19th Symposium IMEKO TC 4 Symposium and 17th IWADC Workshop
Advances in Instrumentation and Sensors Interoperability
July 18-19, 2013, Barcelona, Spain

Synchronization of DLMS/COSEM sensor nodes

Alfonso Attianese, Antonio Del Giudice, Marco Landi, Vincenzo Paciello, Antonio Pietrosanto

DIIn - University of Salerno - Via Giovanni Paolo II, 132 84084 Fisciano (SA), Italy
Email: {aattianese, adelgiudice, malandi, vpaciello, apietrosanto}@unisa.it

Keywords: Synchronization, COSEM, DLMS, WSN, AMI ,Obis code, TPSN.

***Abstract* -** An AMI involves multiple heterogeneous devices, often produced by different manufacturers and which play different roles in the infrastructure, that need to communicate among each other. The DLMS/COSEM seems a good candidate to fulfill the requirement, being targeted to smart meters and supporting different communication channels. However the time resolution provided by the DLMS is in the order of hundredths of seconds: such resolution combined with an appropriate synchronization protocol could contribute to obtain measurements with sufficient resolution in time to make possible an accurate analysis of the power quality in customer's premises. In this paper, the authors will integrate the synchronization capabilities offered by DLMS with some synchronization protocols for WSN making comparisons and reporting on performances and applicability.

I. Introduction

The paradigm of a Smart Grid is widely discussed in literature [1]: the increase in energy prices, the need of reducing CO2 emissions [2] and the ever increasing electricity demand are driving towards the adoption of a network architecture that can effectively integrate heterogeneous players in energy generation, including distributed generation (Fig.1). Moreover it should allow the customers for direct access to electricity market and should be able to react in real time and autonomously to changing conditions on the grid, recurring to user provided services when needed (it is the case of V2G [3] and Demand Response Resources [4]).
Fundamental for this evolution of the grid is the ability to provide accurate and timely effective information on the state of the network, therefore one of the most important developments regards the metering and communication infrastructure, which is usually referred to as AMI (Advanced Metering Infrastructure). Having at disposal information on the actual state of the network allows for more efficient management and more effective planning: load forecasts can be updated with increased frequency, thus being able to respond more promptly to evolution in the demand; management algorithms may be put in place to automatically take actions to meet network changes; true real time electricity market may be realized. The communication needs to be bidirectional, involving both data to be sent to a Management Center and commands for the actuation devices. The communication operation can be divided into two main areas: exchange of data inside the customer's premises, and transmission of data to, and reception of command from, a Management Center.

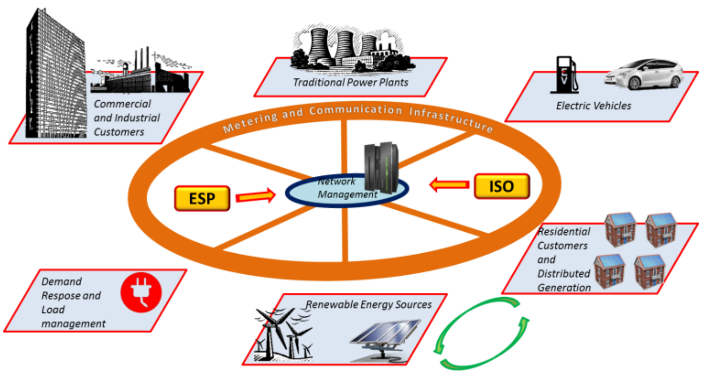

Figure 1. AMI architecture

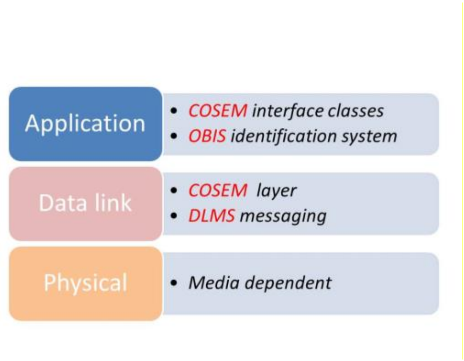

Figure 2. Protocol layout

The paper focuses on the communication inside customer's premises. In such context, different devices, which use different communication means, must be interfaced. Therefore a common communication language is needed: only if the data format is known and shared by all actors in the network the system can be interoperable. The choice fell upon the DLMS/COSEM protocol [5]-[6], which can guarantee the adoption of most of the existing wired and wireless physical channels of communication. In the paper a description of the protocol is given, followed by a characterization of DLMS with particular regard of the synchronization performances that the protocol can obtain.

II. The DLMS/COSEM Standard

Smart meters are fundamental parts of an Advanced Metering Infrastructure. The main actors in a AMI are concentrators and distributed meters (clients and servers respectively). The adoption of a common communication protocol is necessary step on the way of the interoperability. All the subjects in the network have to exchange information, measurements and commands in a full duplex fashion, nevertheless the communication have to be reliable. At the moment DLMS/COSEM [7] can be good choice to satisfy the needs of smart meters. DLMS/COSEM covers all fields of application of AMR/AMI on different physical links. DLMS stands for "Device Language Message Specification" and is a generalized concept for abstract modeling of communication entities. COSEM ("COmpanion Specification for Energy Metering") covers the data exchange in energy meters by laying down rules based on existing standards. The protocol is built up as an OSI stack like HDLC or TCP-UDP/IP. DLMS is responsible for data transport, COSEM defines the application layer data modeling as well as rules for data identification by means of Object Identification System (OBIS). The model provided by COSEM is an abstraction that refers all the functions provided by the meter (Fig. 2). The underlying idea for this work is to realize the time synchronization between smart meters that communicate by means of DLSM/COSEM protocol. To make a node synchronized in a sensor network, a master sensor must transmit its local time information to the slave node. The authors, starting from the analysis [8] of DLSM/COSEM protocol, investigate the possibilities offered from the standard to achieve this goal. In the DLSM/COSEM protocol the date and time information may be represented with data type octet-string, or using the data types date, time and date_time. Data items are uniquely identified by OBIS codes: each code corresponds to an interface class. Two are the objects used in OBIS: with and without the time of capture of the information transmitted.

OBIS

The OBject Identification System [9] makes possible to identify the data items used in common electricity metering equipment. The main advantage of OBIS is the definition of a unique code for data of interest in energy measurement. The general structure of the obis code is composed by six values

A	B	C	D	E	F

Figure 3. OBIS code structure - [9]

Every single value has a range and a standard definition that makes possible to uniquely identify data items.
It is also possible to identify proprietary codes for manufacturer specific purposes.

Code type	Value group						Note
	A	B	C	D	E	F	
Manufacturer specific	0, 1, 4..9	128…199	x	x	x	x	See Note 1
		x	128… 199, 240	x	x	x	
		x	x	128…254	x	x	
		x	x	x	128…254	x	
		x	x	x	x	128…254	
Manufacturer specific abstract	0	0…64	96	50…99	0…255	0…255	See Note 2
Manufacturer specific, media related general purpose	1, 4..9	0…64	96	50…99	0…255	0…255	See Note 2
Utility specific		65…127					See Note 3
Consortia specific	0, 1, 4..9	1…64	93	See Table 7			See Note 4
Country specific		1…64	94	See Table 8			See Note 5

Figure 4. Rules for manufacturer, utility, consortia and country specific codes – [9]

Standard codes cannot be reused from manufacturers with different meaning, on the other hand, an object defined by a manufacturer may be standardized if its use is of common interest.
With the OBIS code even the tariff rates can be exchanged between meters and a remote host.

III. WSN Synchronization

For power quality purposes the meters could have the need of exchanging messages with local time stamps. Also for a collaborative execution of distributed tasks [10], synchronization of meters is indispensable [11]-[14]. The authors want to investigate on the possibility to intervene on the DLMS/COSEM protocol to obtain a synchronization method that guarantees resolution of hundredths of seconds. Two of the features that the synchronization protocol is required to have is scalability and low power consumption policies. Given the wireless capabilities the meters can be seen as sensors interconnected, that is why we can look at common protocols for wireless sensors network to achieve synchronization. Typically in WSN, the synchronization procedure [15] is initiated by the client which makes a sync-request to the server that keeps the reference time. In this case all the nodes represented by the meters are servers, and the gateway is the client that makes queries to obtain the measures. This "upside-down architecture" has to be taken into account to verify the feasibility of the chosen protocol. The most efficient protocols for synchronization in WSN are Timing-sync Protocols for Sensor Networks [16], Reference Broadcast Synchronization [17], Flooding Time Synchronization Protocol [18]. TPSN is a sender-receiver based synchronization protocol and it works in two steps, the first one being the assignation of a hierarchical level to every single node, the second one consisting in the synchronization of nodes of lower level with nodes belonging to higher levels. Advantages of this protocol are good performances and scalability. However, since the synchronization is initiated by the client, in the architecture presented here some modifications have to be done in order to make this protocol work. Also FTSP is based on the concept sender-receiver, in this case a root node sends a packet containing the sender timestamp to all the other nodes in the network that receive the packet and note their local times. Knowing send time and receive time the nodes are able to compute the offset. RBS uses a receiver-receiver synchronization: one of the nodes sends a broadcast beacon with no timing information and the receivers will compare local clock with each other to compute their offsets. The first step for the authors was to implement the synchronization algorithm on microcontroller and then characterize it.

IV. Hardware

The hardware used is composed by two boards [19] with an ARM Cortex M3 processor (STM32F103RE). The two boards are connected by means of a wired serial link; the slave has a second serial connection with the PC for diagnosis purposes. This serial connection is used for testing the system; the next step is to replace cables with a wireless link with devices totally transparent with respect to the serial connection: the goal of this work is to implement a synchronization mechanism inside DLMS protocol, for this reason it is acceptable to initially test the system with a wired connection.
Every sensor node is composed by the minimum hardware that allows communication with other nodes, makes possible interaction with human operator and allow to the microcontroller to keep information about local time.
A block diagram of every sensor node is depicted in figure 5.

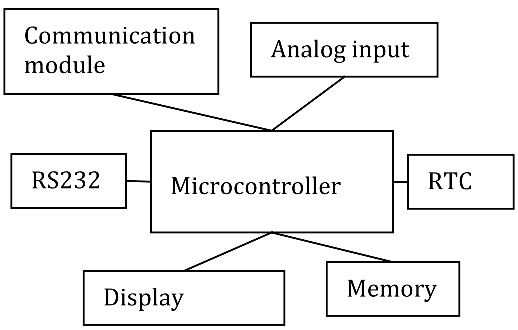

Figure 5. Block diagram of the Sensor Node

The code has been designed to use different physical media as the DLMS protocol do. The goal is find a way to synchronize two or more nodes that use the standard (or manufacturer specific) DLMS protocol. Therefore, if it is possible to realize synchronization between two nodes, the extension to the case of a single master that synchronizes to its clock all the clocks of many other nodes in the network is straightforward.

V. Software

Oscillators drifts, temperature variations, components non linearity can cause significant difference between clocks of different devices even in case they are well synchronized in the beginning. One way to overcome this problem is to establish a reference clock also called master clock and let the other devices periodically synchronize to the clock of the master. There are many algorithms designed for this purpose. The synchronization period is chosen taking into consideration how large is the difference between two or more clocks after a certain amount of time. If the difference is acceptable the period can be larger; in case it isn't the synchronization mechanism have to be repeated with higher frequency to maintain acceptable differences between clocks. In this paper the focus is on the synchronization possibilities offered by the standard DLMS protocol and then once examined its limits the authors tried to make some improvements. Two nodes linked by a serial cable have been taken into consideration for the purposes of this paper. The experiments are referred to a wired connection but the same experiment can be repeated using wireless devices that make transparent the wireless link to RS232 ports. The two nodes are supposed to be two components of an energy measurement system that use DLMS protocol to exchange information about energy data items. The synchronization algorithm chooses is TPSN. The nodes try to synchronize their clocks starting from an undefined initial state. It is assumed that the master node has the reference clock to which the slave node has to synchronize its one. The slave node initiates the synchronization mechanism by asking to the master its local time and then sending to the master the slave_local_time, then local times mutual exchange between master and slave follow.
All the time the master waits for requests from the slave: as soon as an incoming request is arrived, the master answers with its local time, then the slave sends its local time after storing the local time of the master.
Now the master is ready the compute the parameters useful to make the synchronization (offset and delay); once calculated sends them to the slave that immediately synchronizes its clock to the one of the master.
The requests are made using the obis code for the clock object.

```

// Clock
typedef struct Clock{ // class_id:8
    OBIS_CODE logical_name;
    DOUBLE_OCTETSTRING time;  // {yH,yL,m,do
    dlms_LONG time_zone;
    dlms_UCHAR status;
    OCTETSTRING daylight_savings_begin;
    OCTETSTRING daylight_savings_end;
    dlms_INT daylight_savings_deviation;
    dlms_BOOL daylight_savings_enabled;
    dlms_UCHAR clock_base;
}Clock;  // Clock
```

Figure 6. Native Clock structure

The information regarding the clock is encapsulated into a clock structure composed by some fields encoded by the sender and decoded by the receiver.
As first approach the synchronization algorithm based on the precision that the standard DLMS protocol offers with its classes has been implemented. The standard DLMS, thanks to the date_time objects, makes possible to transmit information about the clock with the precision of hundredths of seconds. Given this lower limit to the precision, in case there are into the network some nodes that want to synchronize themselves to the master clock with a precision below the hundredths of seconds it is not possible to reach this performance with the native DLMS messaging system. Another idea in this paper is to improve the precision on the information about local time introducing a manufacturer specific object that permits to exchange information about clock reaching precision of tenth of microseconds. This can be done creating a new clock class with a manufacturer specific logical_name which has an improved precision (one more field) in the codification of time. More precisely the field "time" in the clock structure has to be modified in the field time_1.

```
// Clock
typedef struct Clock{ // class_id:8
    OBIS_CODE logical_name;
    DOUBLE_OCTETSTRING time; // {yH,yL,m,dom,dow,h,m,s,hs,dH,dL,s}
        .
        .
        .
}Clock;  // Clock
```

Figure 7. Standard Clock structure

```
typedef struct Clock_1{
    OBIS_CODE logical_name; // manufacturer specific
    DOUBLE_OCTETSTRING time_1; // {yH,yL,m,dom,dow,h,m,s,hs,us,dH,dL,s}
        .
        .
        .
}Clock_1;  // Clock_1
```

Figure 8. Non-standard Clock structure

In Table 1 are reported first experimental results. The system has been tested using TPSN algorithm to compare the synchronization with and without DLMS packet. The communication link used is RS232 at two baud rates. In the case without DLMS the two nodes exchange only an integer value, that represent the local times in the master board and in the slave board. When the system is tested with the DLMS protocol active, the information is not sent anymore only with integer but the communication happens exchanging OBIS items, in that particular case the nodes acquire times sending and receiving the packet composed by the information of the clock object of each node.

Table 1. Results of test

Baud Rate (bps)	synchronization Interval [s]	**NO OBIS**		**WITH OBIS**	
		Offset [us]	Delay[us]	Offset [us]	Delay[us]
460800	60	60	570	80	1420
	180	220	570	220	1420
	300	380	570	380	1430
57600	60	440	1990	380	7340
	180	1190	2000	1230	7320
	300	1950	2000	2000	7360

From Table1 it is clear that there is not too much difference in the synchronization error using the system with or without DLMS packet; on the other hand looking at the delay when the DLMS packet is used, the time from the coder of the packet by the sender to the decoder of the receiver is increased. The amount of time of witch the delay increase is proportional to number of bytes that are used to exchange the time information.
A future work can be to investigate on the behavior of the system with different synchronization algorithms and different communication links wired or wireless. Also can be interesting to execute several tests at different distances to discover how the error in synchronization changes.

VI. Conclusions

The paper presents how it is possible to synchronize two DLSM nodes and how overcame the limit of the resolution time gives by the DLSM/COSEM scheme. In particular the synchronization algorithm was implemented with and without DLSM/COSEM messages, and then tested on two nodes. The nodes are equipped with an ARM Cortex M3 microcontroller (STM32F103RE).

References

[1] F. Rahimi “Demand Response as a Market Resource Under the Smart Grid Paradigm” Smart Grid, IEEE Transactions on, Vol. 1 no. 1, pp. 82- 88.
[2] International Energy Agency website: http://www.iea.org.
[3] C. Guille, G. Gross, "A conceptual framework for the vehicle-to-grid (V2G) implementation," Energy Policy, Elsevier, vol. 37(11), pp. 4379-4390, 2009.
[4] X.A. Sun, Z. Xiaoxuan “Consumer profiling for demand response programs in smart grids” Innovative Smart Grid Technologies - Asia (ISGT Asia), 2012 IEEE pp. 1-6.
[5] Open meter standard. Website, Available: http://www.openmeter.com/.
[6] (2012) DLMS UA website. Available: http://www.dlms.com/information/whatisdlmscosem/index.html.
[7] G. Kmethy, “IEC 62056 DLMS/COSEM workshop. Part 2: Main concepts”, Metering Europe, Vienna, Sep. 21, 2009.
[8] Flammini, S. Rinaldi, A. Vezzoli, “The Sense of Time in Open Metering Systems”, Proceedings of SMFG 2011 - IEEE International Conference on Smart Measurements for Grids”, no. 6125767, pp.22-27.
[9] International Standard IEC 62056-61. Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: Object identification system (OBIS). http://read.pudn.com/downloads147/doc/comm/637727/IEC%2062056-61-2006.pdf.
[10] D. Capriglione, L. Ferrigno, V. Paciello, A. Pietrosanto, A. Vaccaro, “On the Performance of Consensus Protocols for

Decentralized Smart Grid Metering in Presence of Measurement Uncertainty" Proceedings of 2013 IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2013, 6 - 9 May 2013, MINNEAPOLIS, MN pp.1176-1181.

[11] L. Ferrigno, V. Paciello, A. Pietrosanto, "A Bluetooth-based proposal of instrument wireless interface", IEEE Transactions on Instrumentation and Measurement, Vol.54, no.1 pp. 163- 170, Feb. 2005.

[12] L. Ferrigno, V. Paciello, A. Pietrosanto, "Extending a Short-range Wireless Instrumentation Bus to a Geographic Network" Proceedings of the IMTC 2006 IEEE. Instrumentation and Measurement Technology Conference, 2006. pp. 1268-1273. 24-27 April 2006.

[13] L. Ferrigno, V. Paciello, A. Pietrosanto, "Low-cost visual sensor node for BlueTooth-based measurement networks," IEEE Transactions on Instrumentation and Measurement, Vol.55, no.2 pp. 521-527, April 2006.

[14] L. Ferrigno, V. Paciello, A. Pietrosanto, " Experimental Characterization of Synchronization Protocols for Wireless Networks " Proocedings of Virtual Environments, Human-Computer Interfaces, and Measurement Systems (VECIMS 2008), Istanbul, Turkey, 14-16 July 2008.

[15] D. Capriglione, L. Ferrigno, V. Paciello, A. Pietrosanto, "A step forward the on-line minimization of the synchronization events in TPSN", Proceedings of the I2MTC 2012- International Instrumentation and Measurement Technology Conference, I2MTC 2012, pp. 2780-2784, 13-16 May 2012, Graz, Austria.

[16] S. Ganeriwal, R. Kumar, M.B. Srivastava, "Timing-sync Protocol for Sensor Networks", SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems, pp. 138-149.

[17] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts", Proceedings of the Fifth Symposium on Operating systems Design and Implementation, Boston, MA. December 2002.

[18] Miklos Maroti, Branislav Kusy, Gyula Simon and Akos Ledeczi, "The Flooding Time Synchronization Protocol" Proceeding SenSys 2004 Proceedings of the 2nd international conference on Embedded networked sensor systems, pp. 39-49.

[19] G. Di leo, M. landi, V. PAciello, A. Pietrosanto, "Smart metering for demand side management", Proceedings of the I2MTC 2012 - International Instrumentation and Measurement Technology Conference, I2MTC 2012, pp. 2780-2784, 13-16 May 2012, Graz, Austria.