COMMUNICATION BETWEEN INTELLIGENT DEVICES IN LONWORKS CONTROL NETWORKS

Nicolae MURESAN, Bogdan ORZA, Aurel VLAICU
Technical University of Cluj-Napoca
Email: Nicolae.Muresan@com.utcluj.ro, Bogdan.Orza@com.utcluj.ro, Aurel.Vlaicu@com.utcluj.ro,

Abstract. A modern control network consists of a number of intelligent devices that communicate on one or more channels, using a common protocol. The network's intelligent devices, called nodes, may each include more microprocessors running the application, and the protocol. Also, each device includes a component called transceiver, that provide the electrical interface with the communication channel. In the LonWorks networks, most of the devices (intelligent sensors, controllers, actuators), exploit the advantages of the Neuron Chip, the default LonWorks node, which includes three processors, two of which execute the LonTalk protocol, and the third the device's application-programme. The Neuron Chip supplies the functions of the first six OSI stack layers, the user needing to provide only the configuration and the programming of the application layer. The goal of this paper is to present the comunication mode between intelligent devices, in LonWorks networks, at the MAC and Application levels, based on LonTalk protocol. This analysis is useful in order to know the conditions to be fulfilled for an optimal design of the network, optimal setup and configuration of the devices, and to provide multivendor interoperability. The paper also presents an example of effective design of a lighting control network.

Keywords: *LonWorks, LonTalk, LonMaker Basic Shape, LonMaker design, predictive p-persistent CSMA.*

I. INTRODUCTION

LonTalk is the only protocol supported in the LonWorks systems. It was designed and implemented by means of the ANSI/CEA 709.1 protocols family [6], and it offers a set of communication services allowing the software from device to communicate with other network devices, without need to knowing the network's topology and the addresses, the name or the functions of the other devices.

LonTalk protocol was designed for communications in the control networks, mainly in order to satisfy the needs of certain applications in the fields of domotics and industry. The protocol provides a set of network communication services, with the following characteristics: short messages, very low cost/node, multiple communication media, narrow bandwidth, easy maintenance, multivendor equipment and low cost support. LonTalk is a layered, peer-to-peer packet oriented communication protocol, designed to satisfy the needs and exigencies of a control system. In order for these requirements to correspond to a robust and reliable communication standard, LonTalk protocol design consists of 7 layers, according to ISO/OSI [6], [17] recommendations.

LonTalk is a medium independent communication protocol, supporting several physical layer protocols. For this reason, a sublayer called MAC (Media Access Control) was introduced between the Physical and the Link Layer, in order to facilitate the medium access and to optionally provide priority and collision detection features.

Generally, the MAC protocols for sensor networks have to be able to support a variable and highly correlated traffic. Unlike in data networks, this traffic is generated not by human activity, but is determined by the dynamics of physical processes. Thus, the traditional CSMA (Carrier Sense - Multiple Access) based schemes are, without modifications, inappropriate for sensor networks because they make a fundamental assumption of stochastically distributed traffic [5]. Meanwhile, the traffic in the sensor networking is extremely irregular, especially in case of *event-triggered* applications architecture that LonWorks technology exemplifies. The *event-triggered architecture* is prone to *event showers*, i.e. burst of events, often released by a single physical event that causes congestion of the system [2], [5].

MAC sub layer, responsible with the medium access control, is closely related to LonTalk Link Layer, which is in charge with all the responsibilities of layer 2, including framing and error detection. Frame reception is entirely the responsibility of the Link Layer, which has to notify MAC sublayer of backlog (an estimate of the current channel loading) counter incrementing.

Every LonTalk node communicates to the physical layer in one of two modes: *direct mode* and *special purpose mode*. In direct mode, the Link layer uses differential Manchester encoding. In the *special purpose mode*, data are transferred serially in and out of the node without encoding. In both modes, a 16-bit CRC is generated on transmission and checked on reception. These two modes form an abstraction for all the physical

Manuscript received September 2, 2010; revised December 21, 2010

layers supported by the LonTalk protocol. The interface between the Link and the MAC layers is facilitated by a series of *service interface primitives*, defined using a similar syntax to programming language procedure calls, no implementation technique being implied.

The *predictive p-persistent CSMA* protocol has been designed as the adaptive version of pure *p*-CSMA protocol, and it derives from *p-persistent CSMA* protocol family. Being of *collision avoidance* type, *predictive p-persistent CSMA delay* the *medium access* by means of employing knowledge of *probable channel loading*. Thus, are avoided the collisions that may appear when two or more nodes are waiting for the network to free, in order to transmission start. When they wait the same period between rejection and transmission restarting, repeated collisions occur. The access random delay decreases the collision probability.

If the channel is idle, a node transmits with a probability ***p***. P*redictive p*-persistent CSMA protocol makes the ***p*** value variable and dynamically adjusted to the expected traffic load [6]. Is essential this parameter's adjustment, because larger values of ***p*** may cause excessive collisions rate, and smaller ones may lead to channel bandwidth usage degradation. The evaluation is performed on basis of a *derivative traffic* (for example, acknowledgements and retransmissions).

The prediction built in the CSMA protocol realizes *collision avoidance*, a basic task of media access control in sensor networks. It is based on *backoff schemes* studied at the beginning of the 70s and successfully applied in data networks. Backoff reduce the contention and helps to desynchronize the traffic. *Predictive p*-CSMA uses a backoff without memory.

In *variable-window type* of CSMA protocol class, the p value of *persistency level* maintained by each node is being modified according to the feedback information coming from the network. Usually, the p value is decreased after each collision, respectively increased after each successful transmission, the adjustment being of additive type [6]. In figure 1, the structure of a MAC LonTalk/EIA-709.1 packet sequence is presented. It is composed of two phases, first phase implementing a *reserving protocol*, and second phase implementing the *predictive p-CSMA* protocol. The access to the LonTalk/EIA - 709.1 channel is limited to a specific instance named *slot*. The reservation part from the LonTalk packet cycle is optional and it is dedicated to the priority messages. The priority slots follow immediately after the β_1 inter-packet gap and are dedicated to *improve the response time* of the *critical packets*.

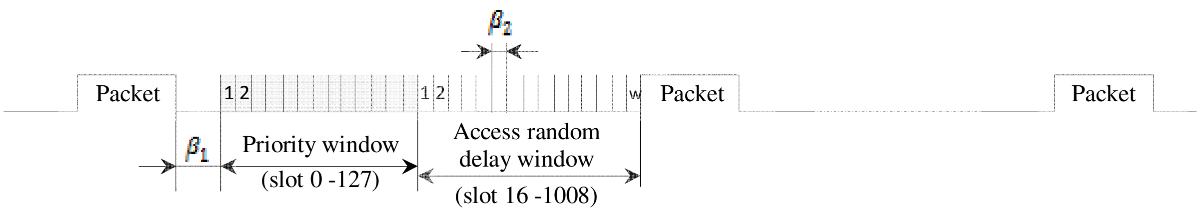


Figure 1. The structure of a LonTalk/EIA -709.1 packet sequence

The number of the priority slots per channel ranges is configurable (between 0 and 127). Lower priority numbers indicate higher priority levels. The p*redictive p-CSMA* code, from the second phase of the MAC packets sequence is divided into slots of β_2 duration.

All nodes tries to force the start of transmission only at the beginning of a slot, and when two packets conflict, they will overlap completely rather than partially, which greatly reduces the probability of collision and provides an increase of channel efficiency. Let's suppose that there are no priority slots. A node attempting to transmit monitors the status of the channel. If the channel is busy, the node continues to sensing, but when it detects no transmission during the β_1 period, the node will delay transmission with a random number of slots of β_2 duration, arbitrarily chosen. If the channel is still idle when the random delay expires, (meaning no other nodes are currently transmitting), the node starts transmission. In this manner, the Link layer of the destination node receives an ingress packet and the proceeding repeats. After transmission, the node receives the incoming packet, and competes again for the channel.

Finally, mentioning that the protocols of the others ISO LonTalk layers perform similar functions with the ones of ISO layer from data transmissions, (see [6]), we will pass to the presentation of the design and implementation manner of a LonWorks *lighting control* application.

II. LIGHTING CONTROL APPLICATION

In order to create a LonWorks *lighting control network*, starting from the functions which must be implemented, firstly we must establish what devices will be used. After a short analysis of these devices we will build the network using the *LonMaker Tool* software.

1. Application requirements, the logic diagram

The application must implement the requirements:

- to measure the illumination level in a room,
- to evaluate the occupancy level of a room,
- to maintain constant the illumination level, in case of room occupancy, regardless the day light.

According to these requirements, the application should contain at least the following components:

- an occupancy sensor,
- a sensor for measuring the room illumination,
- a light controller, which will take over the setpoint by the user and the sensors' feedback,
- an actuator, and a 150 W lamp [1], [2].

We choose, for the aplication, the following devices manufactured by Spega [9]: a multisensor *Lumina MS3-EB*, which includes the sensors and the controller [10], an light actuator/controller *Lumina SC16* [11], a power actuator *Lumina ST4*, and a 150W halogen.

According to *LONMARK Interoperability Guidelines*, the devices software part is described by means of *logical objects*, that are also called *functional blocks*, and which describe the application's behaviour and parametrization [12]. A functional block represents a collection of network variables and configuration properties, for a device.

Lumina MS3 may implement 7 distinct types of such *logical objects:* a *Light sensor*, an *Occupancy sensor*, an *Occupancy controller*, 5 *Switch objects*, a *Constant Light Controller*, a *Spatial Confort Controller* and a *Scene Panel*, the last two types being left unused in the application under discusion [10].

Lumina SC16 may implement two types of *functional blocks: Lamp Actuator* and *Lamp Group Controller* [11]. In this application we will only use the lamp actuator.

In figure 2 is presented a diagram of a possible logical objects connection, able to implement the above required functions. The variable translator is a *Switch object*, which allows the overwriting of the lighting's value with an IR remote control and the feedback loop to be closed.

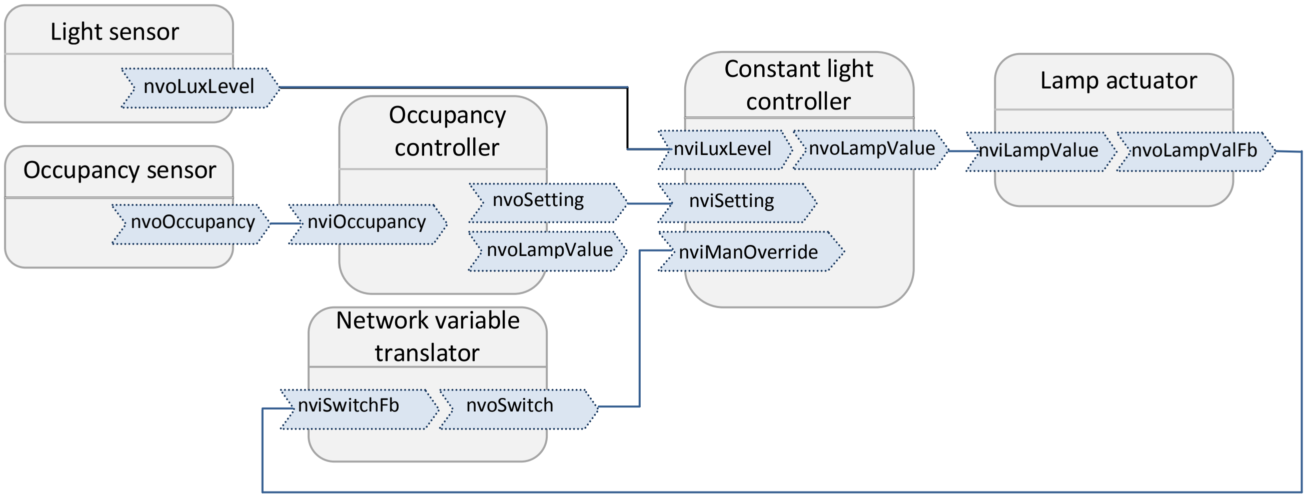


Figure 2. The diagram of logical objects connection

2. Building the network

Before creating the network, must be installed these programs: *LonMaker for Windows Integration Tool*, *LNS DDE Server* and *PCLTA-20 Network Interface Card*, see [4], [7].

Device creating and commissioning

In order to create a new network [3], we open the *LonMaker tool* and in the LonMaker Design Manager window, we select the *New Network* tab, in order to name the network as *Constant light*. In the *Constant light – Echelon LonMaker* window, we will drag in the *LonMaker drawing* (see fig. 3. a, the bottom-right corner) the *Device* form, among the *LonMaker Basic Shapes* (fig. 3. a, left side – at the bottom).

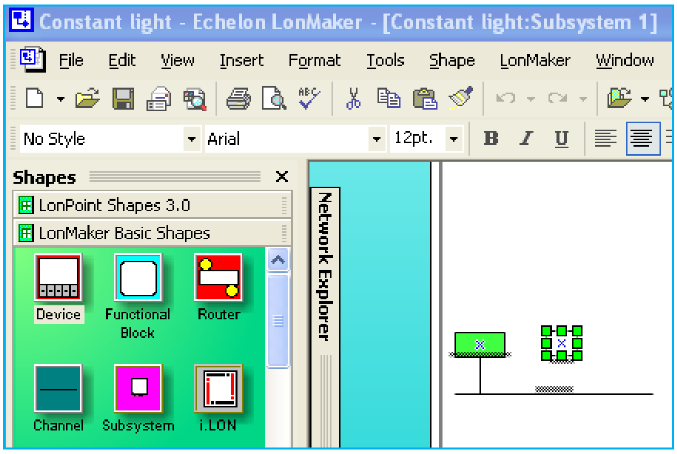
We select the channel type TP/FT 10 (twisted pair), we set-up the name of the device: *Controller ilum*, and load the *application file* SC911103 EC_02.XIF for the *Lumina MS3* device [9]. We commission it (load the *SC011103 EC_02* application file), we set the state of the application on the device after commissioning, and the initial values source of the device's configuration properties [10]. LonMaker initializes the device and loads the selected application files.

Similarly, we create and commission the following device, namely *Actuator ilum* with *Lumina SC16*. Finally, after commissioning, the *LonMaker design* will look as in figure 3. b.

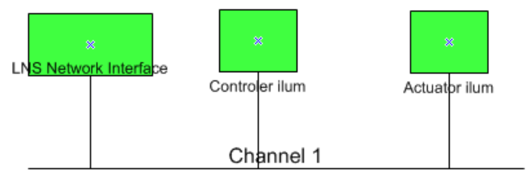
Creating of the functional blocks

The description of the software part of the device's aplications is performed in terms of *logical objects* (functional blocks), according to the recommendations of LONMARK [12], [18], [19]. That objects describe the device applications behaviour and parametrization, and encapsulate *a set of network variables and configuration properties* which define the functions of a device.

In order to create a functional block in a *Lon Maker design*, as in the "*create a device*" section, one must slide the *functional block form* from the LonMaker Basic Shapes (bottom-left corner of fig. 3. a), on the LonMaker design [3].



a. *Constant light* window– *Echelon LonMaker*



b. LonMaker design after commissioning

Figure 3. Creating a new network by means of LonMaker for Windows

Then, the following information has to be introduced:

– *The name of the subsystem* containing the device's functional block.

– *A device* from the list of all the devices, from the selected subsystem, *accepting* the newly created functional block has to be selected.

– *The functional profile* of the created functional block.

– *The name of the functional block*. It may contain up to 85 characters, but may not contain: period, backslash, colon, forward slash, or quotation marks.

– *The number of functional blocks* to be created.

Then, the *Create all Network Variables Shapes* checkbox may be selected, in order to automatically set up forms for all network variables defined in the functional block. Finally, the functional block form is added to the LonMaker design.

3. Creating the software part of the device and running the application

The effective set-up of the application software part is performed according to the steps described in the previous paragraph. Thus, we create the functional blocks: *Senzor lumină, Controler iluminat, Senzor prezență, Controler ocupanță, Actuator iluminat și Flyback* [9], [10], [11].

Then, the functional blocks will be connected by means of the *Connector tool* from *LonMaker Basic Shapes* (in the LonMaker design - bottom-left corner, fig. 3. a). The result is presented in fig. 4 (a *Lon Maker design*), similar to the one in fig. 2, proposed for implementation.

The light controller (*Controler iluminat*), the most significant part of the logical design, is of *Constant Light Controller* type and receives as input:

• The average *illumination level of the room,* by means of the *nviLuxLevel* input variable, from the *Light Sensor.*

• *The occupancy state of the room,* by means of the *nviSetting* input variable, from the functional block: *Senzor de prezență*, through *Controler de ocupanță*, that performs the variable conversion from *nvoOccupancy*, that may take the OC_OCCUPIED/OC_UNOCCUPIED values (occupied or unoccupied room), into *nviSetting*, with the SET_ON/ SET_OFF values.

• *The room's artificial lighting value* (between 0% - 100%) by means of the *Flyback* functional block, of *Switch* type, which reads the *nvoLampValFb* output variable of the *Actuator de iluminat*, and translates it into *nvoSwitch.*

"Controler de iluminat" delivers by means of the *nvoLampValue* output variable*,* the resulting value of the artificial lighting, such that the room is constantly illuminated, regardless of daylight. The adjustment is done only while the room is occupied, otherwise the lighting being disconnected.

The Switch Object may also be controlled by means of an IR remote-control, such that the imposed lighting value may be overwritten. So, the *SNVT_setting* has the following significances and range of values:

0 SET_OFF – device off,
1 SET_ON – device on,
2 SET_DOWN – decrement value,
3 SET_UP – increment value,
4 SET_STOP – stop action,

similarly to a remote-control's keyboard.

Finally, if the physical network is connected and supplied with energy (the devices must be supplied at 24 Volts), and by selecting *Enable Monitoring*, the network will function and the instantaneous values of the network variables will be displayed in the middle of each connection line. These values will change depending on the illumination and occupancy values of the room (see fig. 4, for occupancy mode and a small illumination, of 9 lux).

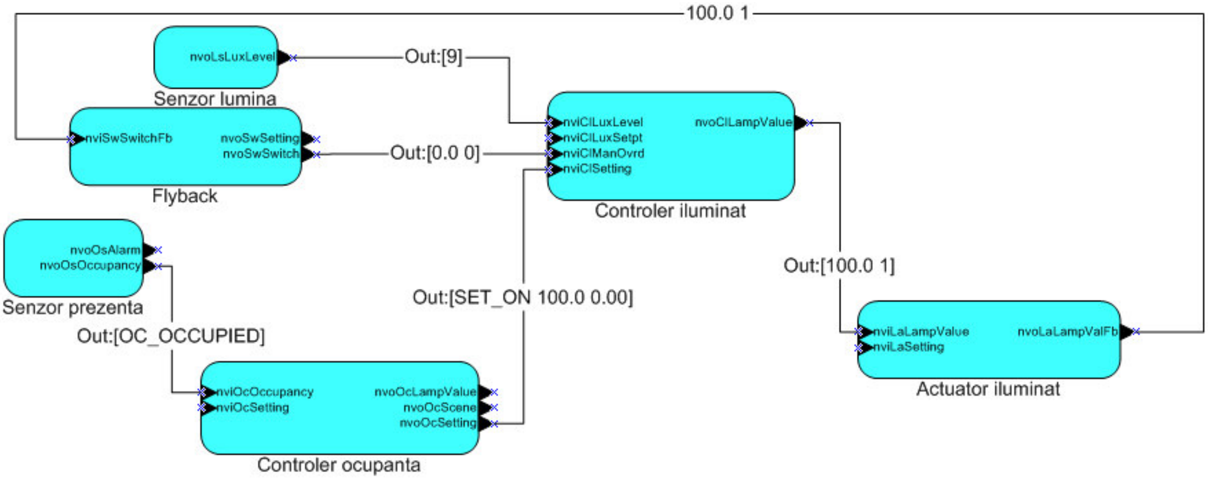
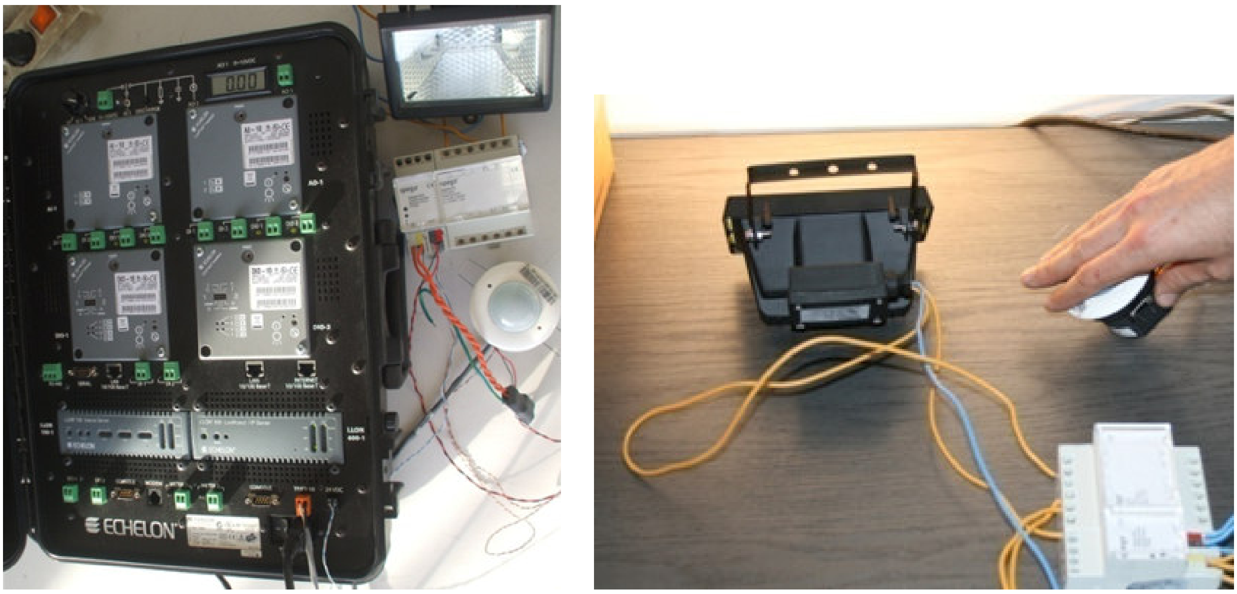


Figure 4. Logical diagram obtained after the functional blocks were connected and the application was run

In figure 5. a, one may see the LonWorks network component devices, together with *IP Connectivity Trainning Center*, an intelligent platform designed for demonstration and testing, from Echelon [8].

In figure 5. b, a sequence of a test performed on the network's logical diagram functioning (occupancy mode and small luminance) is presented.



a. The network and *IP Connectivity Trainning Center* b. A sequence of a test on the network's logical diagram

Figure 5. The physical aspect of the network while running

The hardware requirements for the management PC [4]:

- Microsoft Windows XP (minim Windows 2000)
- Pentium 350 MHz or faster
- 350 megabytes free hard-disk space
- 128 megabytes RAM minimum

- *IP Connectivity Trainning Center*, an intelligent platform for demonstration and testing, from Echelon [8]
- PCLTA-20, a LonWorks Interface designed for connected a personal computer equipped with 32 bit PCI Interface, to a LonWorks control network for installation, monitoring, management, or diagnostic [7].

III. CONCLUSIONS

By means of only two intelligent devices and a solid communication protocol (LonTalk), one may obtain a full lighting control. That network can be extended in order to control, for example, the blinds, by adding another actuator (such controls can be implemented by *Lumina MS3*). In this way, one can obtain a more efficient control and more energy saving. Similarly, other building's or room's parameters may be also controlled.

Such a network, suited to control a room, can be easily extended in order to control a floor, a building, or even a group of buildings. It can be combined with networks based on different types of protocols, or with legacy networks, having classic sensors (by means of IP Connectivity Trainning Center build in *LonPoint integrated modules*). Besides compatibility and extensibility, another major advantage of these networks is the flexibility by means of which one can model the device's functions with the aid of logical diagrams.

Studies carried out by Philips [16] on the theme "*Future House*", showed that people want this house to be like the *house of today*. Thus, based on interviews and research, has discovered that people are afraid of future scenarios in which technology would interfere heavily with their daily lives in the future house. A house is defined in terms of family rituals, such as breakfast together, she must ensure the creation of a feeling of being together, even if some members are far away. The biggest challenge for future technology will be therefore, that to be incorporated into the building not only physically, but also woven into the social context of the future house. Also, to improve people's ability to adapt to new technologies of building parameters control, new techniques of intelligent interaction must developed [1], [18].

LonWorks networks take more advantages in the fact they are: *open, interoperable, multi-vendor and end-to-end* solutions [2], [13], [17]. Elaborated studies, developed according to criteria like: *the pattern of cost comparison*, and performed by Turner & Towsend (a consultancy group in engineering bases) [14], and reports realised by *CABA IIBC* (Continental Automated Buildings Association) [15] and *Strata Resource Inc* [13], about integrated networks for building automatization, clearly proove the benefits of such *open-solutions* and show the fact that *OST* (Open System Technologies) *provide a significant improvement of control and confort*, offering numerous benefits, including increased opportunities to provide:

- easy *installation and integration* of the system,
- device *interoperability and interchange*,
- *data acquisition, diagnosis and maintenance*

facilitating very competitive implementations [17].

There are already many OST technologies in Domotics, three of them have already made history as pioneers of the OST: LonWorks and BACnet (SUA), together with the European standard - KNX/EIB [17]. In Europe are widely used LonWorks and KNX, first technology is preferred to the total control of the building, primarily for the great diversity of products, covering a wide range of functions of building automation. BACnet is focused on traditional HVAC and control systems used more in the SUA. In the lighting systems management, especially in Europe has imposed Digital Addressable Lighting Interface (DALI), a protocol accepted as an industry standard by IEC60929, which facilitates the interchange of the digital ballasts from different manufacturers.

It is hard to predict a global convergence of the technologies, but any such trend is based on extending the idea of interoperability between the OST standards.

The monitoring and control of such a network can be realized locally, or by remote connection, that by means of the internet (attaching an *internet server iLON*), or by means the AC 220 Volts network, using an appropriate transceiver. Traffic problems may occur in larger networks (of tens/hundreds of devices).

As a further improvement, we want to develop *a web interface*, by means of attaching an iLON server to such a network, in order to facilitate the monitoring and control throughout the internet or a mobile phone, of all the network's parameters [19]. Also, we want to extend the network by adding blind actuators, classical sensors, or other types of intelligent sensors and actuators (air, temperature, ventilation, etc.), respectively to ensure wireless facilities of control and monitoring, and from the AC – 220 Volts network.

IV. REFERENCES

[1] Ron Gilster, Helen Heneveld, "Home Technology Integrator", ISBN 0-07-223-132-7, 2004

[2] Aurel Vlaicu (editor) "Clădiri inteligente - Sisteme, Tehnologii şi Soluţii Integrate IT&C", UTPRESS, ISBN 978-973-662-397-4, Cluj-Napoca 2008

[3] Echelon Corporation 2006, "LonMaker User's Guide". [Online]. Available: http://www. echelon.com/support/documentation/manuals/networktools/078-0168-02G.pdf/ [Accessed: Sept. 2008, Oct. 2010].

[4] Echelon Corporation 2006, "Introduction to the LON WORKS Platform". [Online]. Available: http://www.echelon.com/support/documentation/manuals/general/078-0183-01B_Intro_to_LonWorks_Rev_2.pdf/ Accessed: Sept. 2008, Oct. 2010].

[5] Marek Miskowicz and Ryszard Golanski, "LON Technology in Wireless Sensor Networking Applications" ISSN 1424-8220, 2006, by MDPI.

[6] Echelon Corporation 2006, "LonTalk Protocol Specification", version 3.0 , 078-0125-01A

[7] Echelon Corporation, "LON WORKS ® PCLTA-20 PCI Interface User's Guide". [Online]. Available: http://ww.echelon.com/support/documentation/manuals/interfaces/078-0179-01C.pdf/ [Accessed: Sept. 2008, Oct. 2010].

[8] Echelon Corporation 2006, "IP Connectivity Trainning Center". [Online]. Available: http://www.echelon.com/support/documentation/ data shts/72960.pdf/ [Accessed: Sept. 2008, Oct. 2010].

[9] http://www.spega.com

[10] Spega, Technical Datasheets, “E.control - Lumina MS3 – Multisensor”. [Online]. Available: http://www.spega.de/downloads/data sheets/EN/911103_EN_TD.pdf [Accessed: Sept. 2008, Oct. 2010].
[11]Spega, E.control Catalogue 2006 “E.control - Lumina SC16 – Lighting Actuator”. [Online]. Available: http://www.spega.de/downloads/applications/SC121100MC_x5_EN //, http://www.spega.de/downloads/catalogue /EN/e.control_catalogue2010.pdf/ [Accessed: Sept. ’08, Oct. 2010].
[12]http://www.lonmark.org/technical_resources/guidelines/docs/LmPhy34.pdf // 078-0120-01G - LONMARK ® Interoperability Guidelines
[13] Strata Resource Inc, “Investigating Open Systems” [Online]. Available: http://www.strataresource.com/Download/InvestigatingOpenSystems-2006.pdf/ [Accessed: Sept. 2008, Oct. 2010].
[14]http://www.turnerandtownsend.com
[15]http://www.caba.org
[16]http://www.research.philips.com
[17] Nicolae Mureşan, “Stadiul actual în domeniul proiectării sistemelor pentru monitorizarea şi controlul clădirilor inteligente”, Raport de doctorat, Martie 2008,
[18] Nicolae Mureşan, “Standarde şi protocoale utilizate în sistemele de control şi monitorizare”, Raport de doctorat, Octombrie 2008,
[19] Nicolae Mureşan, “Sisteme integrate de control şi monitorizare la distanţă a clădirilor inteligente”, Raport de doctorat, Martie 2009.