IPP-HURRAY! Research Group

Polytechnic Institute of Porto
School of Engineering (ISEP-IPP)

Characterizing the Timing Behaviour of Power-Line Communication by Means of Simulation

Luis MARQUES
Filipe PACHECO
Luis Miguel PINHO

HURRAY-TR-0419
June-2004

relatório técnico

technical report

Characterizing the Timing Behaviour of Power-Line Communication by Means of Simulation

Luis MARQUES, Filipe PACHECO, Luis Miguel PINHO

IPP-HURRAY! Research Group
Polytechnic Institute of Porto (ISEP-IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
Tel.: +351.22.8340502, Fax: +351.22.8340509
E-mail: {npereira, emt, bbatista, lpinho}@dei.isep.ipp.pt
http://www.hurray.isep.ipp.pt

Abstract:
Although power-line communication (PLC) is not a new technology, its use to support communication with timing requirements is still the focus of ongoing research. Recently, a new infrastructure was presented, intended for communication using power lines from a central location to geographically dispersed nodes using inexpensive devices. This new infrastructure uses a two-level hierarchical power-line system, together with an IP-based network. Within this infrastructure, in order to provide end-to-end communication through the two levels of the powerline system, it is necessary to fully understand the behaviour of the underlying network layers. The master/slave behaviour of the PLC MAC, together with the inherent dynamic topology of power-line networks are important issues that must be fully characterised. Therefore, in this paper we present a simulation model which is being used to study and characterise the behaviour of power-line communication.

Characterizing the Timing Behaviour of Power-Line Communication by Means of Simulation[1]

Luís Marques, Filipe Pacheco, Luís Miguel Pinho
IPP-HURRAY! Research Group
Polytechnic Institute of Porto
R. Dr. António Bernardino de Almeida, 431
4200-072 Porto, Portugal
{lmarques,ffp,lpinho}@dei.isep.ipp.pt

Abstract

Although power-line communication (PLC) is not a new technology, its use to support communication with timing requirements is still the focus of ongoing research. Recently, a new infrastructure was presented, intended for communication using power lines from a central location to geographically dispersed nodes using inexpensive devices. This new infrastructure uses a two-level hierarchical power-line system, together with an IP-based network.

Within this infrastructure, in order to provide end-to-end communication through the two levels of the power-line system, it is necessary to fully understand the behaviour of the underlying network layers. The master-slave behaviour of the PLC MAC, together with the inherent dynamic topology of power-line networks are important issues that must be fully characterised. Therefore, in this paper we present a simulation model which is being used to study and characterise the behaviour of power-line communication.

1. Introduction

Using power lines for communication is not a new idea [1]. Using an already deployed infrastructure eases communication costs, not only for energy-related tasks but also for providing new services, such as Internet access and "no-new-wires" home area networks [2]. Nonetheless, the peculiar physical layer of power-line for data communication introduces specific requirements, and the traditional communication technology must be revised to implement an efficient and reliable system [3].

One of the goals of the REMPLI (Real-time Energy Management over Power-Lines and Internet) project [4] is to implement an infrastructure for real-time communication (Figure 1), in order to remotely access monitoring and control equipment [3]. Within the power lines, a two-level hierarchical system is used encompassing the medium-voltage (MV) and low-voltage (LV) electrical power distribution systems. This hierarchical structure implies the development of new protocols for end-to-end communication with timing and reliability requirements.

For this, and due to the inherent characteristics of power-line systems, special consideration must be given to, e.g., redundant path management, data fragmentation and real-time traffic processing. Therefore, within the project, models for traffic patterns and network structure are being set up to simulate the system, allowing developing the communication protocol details and evaluating their behaviour. For the case of end-to-end communication, special concern must be given to the underlying master-slave characteristics of the existent PLC MAC [5]. In order to study this behaviour, a modelling and simulation environment has been implemented, interfacing with the available emulator for the PLC physical layer [6].

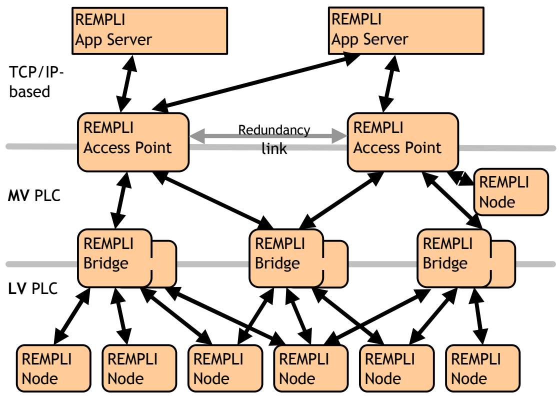

Fig. 1 – REMPLI Architecture

This paper presents this model and some simulation results. The structure of the paper is as follows. Section 2 presents a brief description of the REMPLI system model. Afterwards, in Section 3 we present the

[1] This work is being partly supported by the EC within project REMPLI (NNE5-2001-00825). http://www.rempli.org

implemented simulation model, while in Section 4 we present and discuss some of its results Finally, Section 5 provides a brief summary of the paper.

2. System Architecture

Figure 1 presents the architecture of the REMPLI infrastructure. The Access Points are the entry points on the PLC network. They forward requests from the TCP/IP-based network to the Nodes (via Bridges).

The Bridges provide the connection between MV and LV networks. If needed in a particular installation they can also be used as redundant units providing alternative routes between an Access Point and a Node.

Finally, a complete REMPLI Network can have thousands of REMPLI Nodes covering a large geographical area from a large city to a whole country. REMPLI Nodes have direct connection to/from the installed equipment.

Although a tree-like topology is implied, due to the switching of network segments in the mid-voltage and the low-voltage power-line, nodes can be reached via multiple redundant paths both between Access Points and Bridges and between Bridges and Nodes. Furthermore, this switching is highly dynamic and unpredictable (a simple start of an engine can change dramatically the PLC links, with direct reflection on the communication topology). Therefore, it is also necessary to cope with temporarily not accessible connections.

This two-level, dynamic, communication at the power line (MV and LV) together with the quality of service requirements of the system, are the main reasons for the need for a new protocol, able to provide an end-to-end (Access Point-to-Node and vice-versa) reliable and predictable communication service. This protocol will reside at the Transport Layer of the REMPLI communication stack, but will have also to perform routing at the Bridges level, managing the dynamics of the paths between Access Points and Nodes.

In order to develop this new protocol, one of the requirements is the integrated modelling and simulation of the underlying layers. Note that in this case, it is not important to model in detail the internal functioning of these layers, but to provide their behaviour and results (i.e. a “black-box” model) for the development of the upper layer protocol.

Both the MV and LV PLC networks use a master-slave MAC protocol [5] based in fixed length time slots. The link layer polls the slaves periodically so that data transfers from slaves to masters are possible even without a previous explicit request from the master side. This allows for alarms and asynchronous events to be transmitted and also eases the processing of request/response transactions.

Due to wide distances it may be possible that packets cannot be directly relayed between the master and the slave. In these cases the link layer of the PLC network is responsible for forwarding packets through the slaves until they reach the destination [5]. As a consequence the master must pre-allocate several slots for the transaction with a slave (not only for the request but also for the slave response, if expected).

Furthermore, due to the possible redundant paths, the network must cope with the existence of several masters. This can be done using different frequency bands for each network or through the division of the physical layer in time slots, with pre-allocated time-slots to the foreseen masters. Multiple masters are supported both at the MV and LV levels. The time-division method has several advantages including the possibility of concurrent transactions between a slave and several masters and fast change of parameters.

3. Simulation Model

The simulation environment was developed using the OMNeT++ [7] discrete event simulation platform. OMNeT++ is an object oriented modular discrete event simulator, which provides a reusable component framework, where the system components can be independently built and then characterized and assembled into larger components and models. The basic system components are built using the C++ language and then assembled into larger components and models using a high level language, named NED. An OMNeT++ model consists of hierarchically nested modules. Modules communicate with message passing, where messages can contain arbitrary data structures.

In this implementation, the Simulator takes care of all Master-Slave communication, building up on the existent network management system [5].

The main services to the development of the REMPLI Transport Layer are:

- Master: Send Confirmed, Send Not Confirmed, Receive
- Slave: Send, Receive

These services enable data exchange between Masters and Slaves based on PLC addresses including error recovery and priority queue placement (for QoS support). The Send Confirmed service is intended for request-response master-slave communication since the confirmation frame can also include data.

3.1 REMPLI devices modelling

In the model (see Figure 2 for one example) there are three device modules and the Physical Layer Emulator module. The device modules are the *REMPLI Access Point*, the *REMPLI Bridge* and the *REMPLI Node*. Those modules are composed of sub-modules, which are the *Slave* and the *Node Transport Layer* in the REMPLI Node, the *Master*, the *Slave* and the *Bridge Transport Layer* in the REMPLI Bridge, and, the

Master, the *Access Point Transport Layer* and the *Driver* in the REMPLI Access Point.

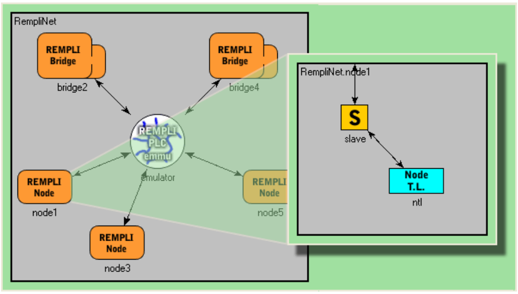

Fig. 2 – Example of a REMPLI Model

The REMPLI Bridge (Figure 3) uses both a *Slave* and a *Master* because it has to forward and (possibly) convert packets from two different network segments. The *Slave* is used in the communication with the Access Points, and the *Master* is used in the communication with the Nodes.

Unlike the *Transport Layer*, which is tailored for each type of device, the *Master* and *Slave* modules are reusable in any device, i.e. the *Master* module in the Access Point is the same in the Bridge, and the *Slave* module in the Bridge is the same in the Node.

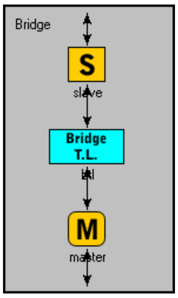

Fig. 3 – Bridge Structure

3.2 Physical Layer Emulator

In order to simulate the physical layer of the system, an interface was made to the Physical Layer Emulator [6], which emulates a PLC time slot, calculating which stations receive which packets, and with which error status.

The emulator itself consists of a set of C++ classes, which allows its simple integration in OMNeT++ projects. The emulator C++ classes receive information about the packets that were sent in a time slot and, when requested, calculate the transmission of those packets for a time slot and return the results. .

To integrate the emulator on the OMNeT++ a module was created (*PLEmulator*) that uses an instance of the emulator. This module interfaces with the emulator, being responsible for:

- Receiving OMNeT++ messages from REMPLI devices, reconstructing the REMPLI packets and schedule them in the emulator.
- Triggering the emulation of a slot time.
- Receiving the output from the emulator, using it to reconstruct the OMNeT++ messages, which are then sent to the respective REMPLI devices.

This approach, where the physical topology of the network is totally emulated and integrated in a module, is very convenient as it permits the transparent use of the module independently of the actual physical topology implemented in the emulator.

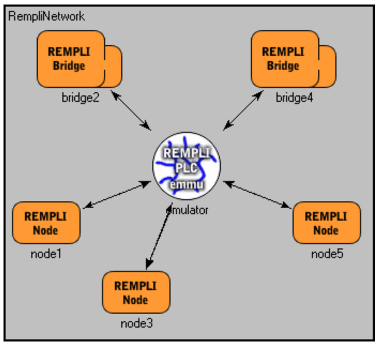

Fig. 4 –Example Scenario

4. Simulation Example

In order to allow discussing some preliminary results, a simulation run was made (for 60 minutes of simulated time, taking appr. 30 seconds in a 1.6 GHz Intel Pentium 4 computer), considering a scenario (Figure 4) with 2 Bridges and 3 Nodes, with the following characteristics:

- 4 time slots in the cycle (each of 100 ms);
- Bridge 2 uses the first time slot, while Bridge 4 uses the remaining 3;
- Each Bridge generates one high priority packet every second that is sent to Node 3 or Node 5;
- Bridge 2 sends 75% of the traffic to Node 3, while Bridge 4 sends 75% of the traffic to Node 5;
- 75% of the packets generated are confirmed requests;
- Each device has two queues (high and normal priority) with space for 4 packets each;
- Node 3 generates one normal priority packet to Bridge 2 every ten seconds;
- Node 1 and Node 5 generate one normal priority packet to Bridge 4 also every ten seconds;
- The necessary link layer slave forwarding levels is a random number between 0 and 2.

Table 1 presents the results obtained in the *Master* of Bridge 2. We verify that Bridge 2 is discarding many packets. This means that the queue is filling quicker than it is able to send packets (remember that after a *Master* sends a packet it has to reserve up to more 5 time slots for the worse case confirmed packets: 2 for the arrival of the packet at the Node, and 3 for the response).

Table 1 – Bridge 2 Packets

SendConfirmed service to Node 3	1473
SendNotConfirmed sent to Node 3	506
SendConfirmed sent to Node 5	493
SendNotConfirmed sent to Node 5	171
Discarded	984
Confirmations Received	1478

Concerning timing behaviour, we measured the time (in seconds) until this *Master* started receiving the confirmatiom packets (Table 2). The minimum and maximum confirmation times are the expected: 0.401 (just 2 hops) and 2.001 (the maximum of 6 hops).

Table 2 – Bridge 2 Time to Receive

Minimum	0.401
Maximum	2.001
Mean	1.165
Standard Deviation	0.649
Variance	0.421

Tables 3 and 4 provide the same results for the Bridge 4 *Master*. Contrarily to Bridge 2, Bridge 4 has no discarded packets, which can be explained by the fact that it can send 3 times more packets than Bridge 2 (3 time slots out of 4). We can also observe that the number of packets sent is only about 2 times the packets sent by Bridge 2. This means that for Bridge 4, two time slots out of 4 would be probably enough, leaving an extra slot for Bridge 2 which is “starving”.

Table 3 – Bridge 4 Packets

SendConfirmed service to Node 3	680
SendNotConfirmed sent to Node 3	226
SendConfirmed sent to Node 5	2037
SendNotConfirmed sent to Node 5	688
Discarded	0
Confirmations Received	2715

Table 4 – Bridge 4 Time to Receive

Minimum	0.401
Maximum	2.001
Mean	1.220
Standard Deviation	0.648
Variance	0.420

Concerning the *Slaves* (Table 5), it is possible to see that in Node 1 all 359 packets generated are discarded, since none of the bridges includes the node in the poll cycle. Node 3 is receiving all packets from Bridge 2, and all but one from Bridge 4.

However, Node 5 has a much lower receiving rate, since it receives only approximately 7% of the packets sent by Bridge 2. This makes the case of the problems in power line communication, where the dynamics of the network are highly unforeseen, thus requiring very adaptive and dynamic higher level protocols.

Table 5 – Slaves' Packets

	Nodes	1	3	5
From Bridge 2	*SendConfirmed*	-	1473	36
	SendNotConfirmed	-	506	12
From Bridge 4	*SendConfirmed*	-	679	2036
	SendNotConfirmed	-	226	688
Discarded		359	0	0
Confirmations Sent		-	2152	2072

5. Summary

This paper outlined the work which is being performed in order to support the development of a new Transport Layer for a particular power-line communication infrastructure. This layer must be capable of providing end-to-end communication with timing and reliability requirements.

For this, we presented a simulation model, which is being used to understand the behaviour of Master-Slave communication in Power-Line Communication Networks. The preliminary results obtained were discussed, providing the required information for the development of the upper Transport Layer protocol.

References

[1] M. Lobashov, G. Pratl, T. Sauter, “Implications of Power-line Communication on Distributed Data Acquisition and Control System”, IEEE Conference on Emerging Technologies and Factory Automation, Lisbon, Portugal, 2003.

[2] Dan Strassberg, “Home/automation buses - Protocols really hit home” - EDN Access - Design Feature - 4/13/1995

[3] A. Treytl, T. Sauter, G. Bumiller, “Real-time Energy Management over Power-lines and Internet”, Symposium of Power-Line Communication and its Applications, Zaragoza, Spain, 2004.

[4] Real-time Energy Management over Power-Lines and Internet, NNE5-2001-00825, http://www.rempli.org

[5] G. Bumiller, M. Sebeck, “Complete Power-Line Narrow Band System for Urban-Wide Communication”, Symposium of Power-Line Communication and its Applications, Malmö, Sweden, 2001.

[6] G. Bumiller, “Power-Line Physical Layer Emulator for Protocol Development”, Symposium of Power-Line Communication and its Applications, Zaragoza, Spain, 2004.

[7] A. Varga, "OMNeT++ Discrete Event Simulation System", v2.3, 2004, http://www.omnetpp.org/