Modelling and simulation of a SFN based PLC network

Raul Brito, Gerd Bumiller, Ye-Qiong Song

HAL Id: inria-00000793
https://hal.inria.fr/inria-00000793

Submitted on 19 Nov 2005

Modelling and simulation of a SFN based PLC network

Raul Brito
LORIA – INRIA - TRIO.
Nancy, France
raul.brito@loria.fr

Gerd Bumiller
iAd GmbH.
Großhabersdorf, Germany
gerd.bumiller@iad-de.com

Yeqiong Song
LORIA – UHP Nancy 1 - TRIO.
Nancy, France
song@loria.fr

Abstract

This work has been carried out as part of the REMPLI project (European program NNE5-2001-00825, www.rempli.org). The main objective of REMPLI is to develop a distributed infrastructure suitable for real-time monitoring and control of energy distribution and consumption. PLC technology has been chosen to form the main communication infrastructure.

The problem addressed in this paper is how to design an efficient network protocol specially suited for SFN based master-slave PLC system [1]. Since the power-line is a complex medium not suited for data transmission, hence classical routing is not efficient. For this purpose, and thanks to the Physical Layer Emulator [2] implemented by iAd, we used the simulation approach to catch the important design parameters such as repeater level variation, packet error rate variation, etc. The understanding of the influence of those parameters allowed us to get an efficient protocol.

INTRODUCTION

This paper shows the integration of the Physical Layer Emulator (described in [2]) implemented in C++, with Opnet, a well known network simulation tool.

Afterwards, we describe the process taken in the development of a flooding protocol for PLC SFN based in a simulation approach complemented by a theoretical study presented in [3]. This process consisted in several steps of simulation, which permitted to develop an efficient protocol. We then compare it against an alternative protocol that allows a faster convergence to the optimal repeater level. The simulations prove that the first protocol behaves better in most cases.

OPNET ENVIRONMENT AND THE POWERLINE EMULATOR

Opnet is a discrete event simulator that possesses vast possibilities and support in terms of simulation and analysis of protocols. In our case, since the power-line is not supported, the main reason of utilisation of Opnet deals with the large array of statistical analysis offered. The C++ Physical Layer Emulator [2] library made available by iAd in the context of the REMPLI EU-project permits to emulate the power-line environment.

Usually, Opnet simulations use predefined Opnet Links for communication between Opnet Nodes. However, since the power-line physical layer isn't readily available in Opnet, it is not possible to use this standard communication mode.

Thus, the adopted solution is the creation of a centralised node that acts as the medium for communication between all network nodes. For the Opnet environment, this means the add of an extra Opnet Node which encapsulates the Physical Layer Emulator, through a shared linked library.

This central Node has the role of receiving packets from the network nodes that wish to transmit in a certain timeslot and return the resulting packets to receiving network nodes, after emulation of the current timeslot. Afterwards, simulation should advance in time.

Another particularity in our system is that the timeslot is the used time unit, since it is normally a parameter that depends of the type of power-line. Typical values for the timeslot size vary between 5 and 20 milliseconds.
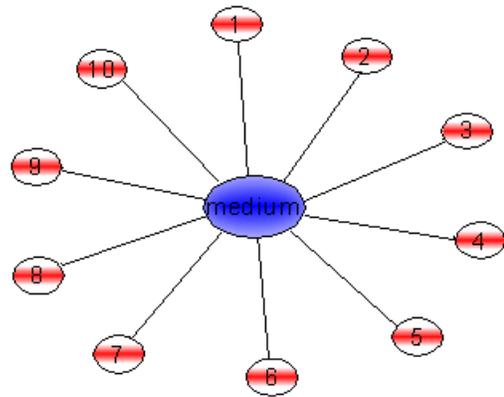
Figure 1 Opnet representation of a 10 node power-line network.

An example scenario with ten nodes is shown in Figure 1. Remark the existence of the central node that acts as power-line medium. The peripheral nodes in the same figure correspond to the actual network nodes.

These peripheral Opnet Nodes provide a communication interface with the central Opnet Node, which permits the sending of a packet for emulation, or receiving a packet from an emulated timeslot.

This communication is assured by Opnet Packets, which are handled by special functions also provided by Opnet. The actual simulated protocols that will be presented next,

are encapsulated in the peripheral nodes, through an Opnet Processor object.

We provide a modular approach that permits an easy integration of different protocols, by associating each protocol to a different Opnet Process. At configuration time we determine which protocol will be used during the simulation.

PROTOCOL DEVELOPMENT PROCESS

The idea behind these simulations consist in the development of an efficient flooding protocol by exploiting the possibility of using the Single Frequency Network (SFN) technique in power-lines. In this way, we expect to derive better results than already existing routing protocols, which require a larger overhead in packet header size and traffic communication for routing information to the master.

We start with some basic simulations, derive a protocol

A. REPEATER LEVEL STABILITY

The first simulations were conducted with the basic goal of collecting information about the repeater level value and its stability in the transmission of a packet, considering different scenarios.

The simulations can be characterised in the following way:

- The Master periodically sends a packet for each node, without imposing a maximum number of repeater levels. Thus, if there is a path to reach the destination, it will be discovered.
- Each Slave acts as a repeater until the packet reaches the destination slave. Each slave only repeats, which prevents the creation of loops in the network.

These simulations were conducted in 4 different scenarios: ring networks of ten and one hundred nodes, and random dispersed networks of one and two hundred nodes. These networks represented typical examples of power-line networks. In all simulated scenarios, the destination slave always received the packet.

Channel Model	Maximum Rep Level	Max Rep Level Diff.	Repeater Level Avg.	Std. Dev. Avg
Ring 10	2	1	0,437	0,139
Ring 100	3	1	1,075	0,032
Random 100	3	2	1,007	0,154
Random 200	4	2	1,493	0,098

Table 1 Repeater level stability for different network topologies.

The results compiled in Table 1 show that there is a certain stability regarding the number of repeaters needed to attain the destination node, and show the low number of necessary repeaters to attain a certain node, even in large network.

B. NETWORK LAYER FLOODING PROTOCOL

The previous set of simulations provided an insightful perspective how the power-line reacts to the SFN technique. A theoretical approach to the same problem [3] also permitted to achieve the same conclusion obtained by simulation. Thus, based in these two elements, a network layer flooding protocol was devised.

The main problem when designing such a protocol regards algorithm which decides about the repeater level from Master to Slave (downlink) and from Slave to Master (uplink). It is important that these values are close to the actual value.

An overestimation of either uplink or downlink repeater level between Master and Slave implies that no more packets can be sent in the network since it might incur into a collision. On the other side, the underestimation of the repeater level between Master and slave (packet didn't reached the destination) forces the use of retries. In this case, timeslots are also lost.

The policy of the protocol relatively to an underestimation of the real repeater level is to retry the packet with an higher repeater level, since besides wasting timeslots, it causes packet losses.

In the case of an overestimation of the repeater level, the protocol acts with more precaution. The intuition is based on the fact that the repeater level stability is high, thus it is better to maintain the actual repeater level for a certain period instead of immediately lower it, since the change could have a sporadic nature. In the case of a more permanent situation, the protocol will eventually update its value, but not immediately. The recent past history of repeater level plays, therefore, an important role in the computation of the next repeater level.

In order for the Master to make a decision of the new repeater levels for uplink and downlink of the slaves, it needs some information from them. Typically, this exchange of information is done through the send/confirm packets of a Master/Slave transmission task.

The complete protocol is under implementation for the REMPLI project.

C. ALTERNATIVE PROTOCOL

In order to evaluate the efficiency of this protocol, an alternative Master-Slave flooding protocol was also proposed and simulated. The main difference of this protocol (Protocol 2) in regards to the protocol presented in the previous section (Protocol 1) is its faster convergence to the optimal repeater level, which permit to take conclusions on the design decisions that support Protocol 1, relatively to its slower adaptation to network disturbances.

In order to obtain a faster reaction to abrupt repeater level changes in the network, Protocol 2 requires the complete information concerning the repeater level of both uplink and downlink of a transmission task at the Master side. In this way, the Master has the possibility to know the exact repeater level of the last transmission task. However, a larger packet header size is needed to carry the necessary information.

This protocol (Protocol 2) also features a repeater level variation degree parameter, useful for small interferences in the repeater levels. For example, in the simulated scenarios this parameter instantiates to 1, which means that with a difference of 1 repeater level no update is made by the Master.

D. PROTOCOL EVALUATION

In this section, we confront the behaviour of the two protocols described above, through extensive simulations on the same topology models given before.

An important measure in our system is the polling cycle. A polling cycle can be defined as the number of timeslots needed by the Master to send one transmission task for all the slaves. This reflects the type of communication existing in the system, and by which slaves must cope to send the data[1].

Figures 2-5 correspond to the polling cycle average in the initial time life of a system, which includes the start-up phase. This is the most interesting period, since afterwards the system tends to stabilise. Posterior interference in the system will also create disturbances in the system and are also interesting to study. However, these disturbances can, be compared to, in a certain extend, a start-up phase. Moreover, these type of interferences is not emulated in the following scenarios. Section E deals with a related problem.

---

[1] Slave initiated communication mechanisms, like Slotted ALOHA, are a viable alternative to Master initiated polling mechanism.
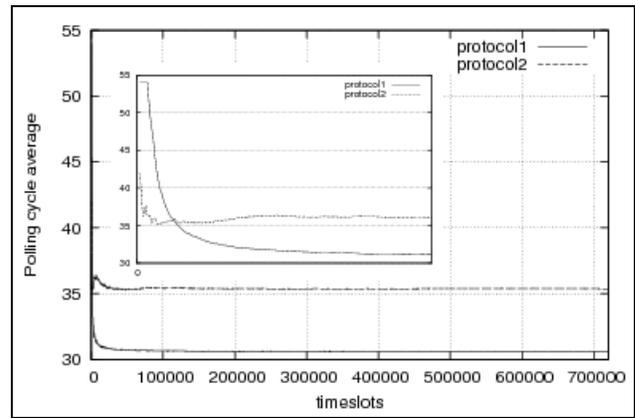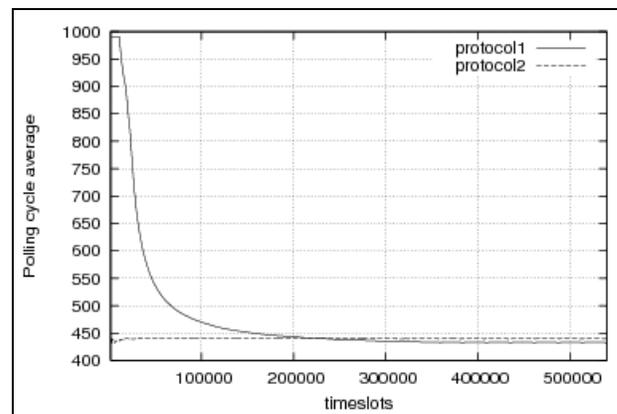

Figure 2 Polling cycle average for ring 10 network with start-up phase zoom.

On all scenarios (figures 2-5), Protocol 1 behaves better. Nevertheless, Protocol 2 provides a faster adaptation to the optimal repeater level at start-up phase, while Protocol 1 needs more time to stabilise.
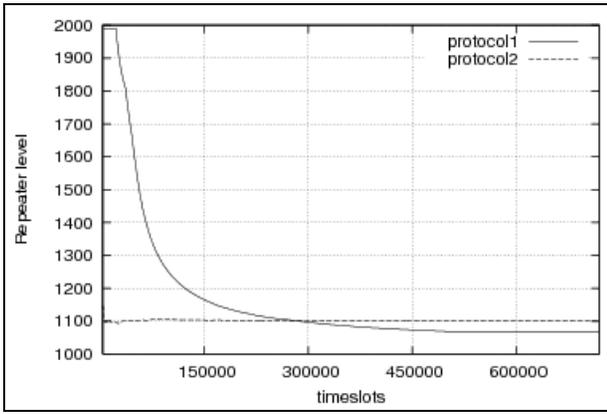
This behaviour is more accentuated in larger networks (100 and 200 nodes), as it would be expected, since it takes longer to descend from the default repeater level to the optimal repeater level.
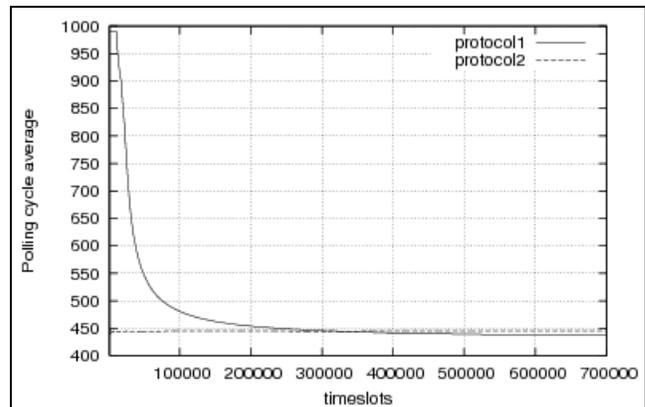
Figure 3 Polling cycle average for ring 100 network.
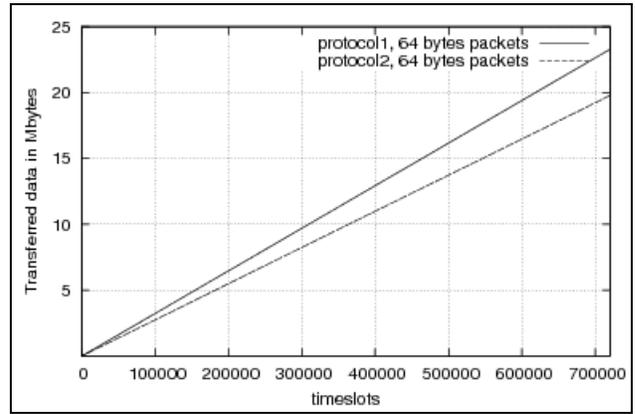

Figure 4 Polling cycle average for random 100 network.

Figure 5 Polling cycle average for random 200 network.

Figure 6 Transferred payload in ring 10 network.

Figure 7 Transferred payload data in ring 100 network.

Figure 8 Transferred payload data in random 100 network.

After the start-up phase, it is also possible to notice a slight advantage of Protocol 1, despite possessing less complete information than protocol 2. This can be explained by the overestimation of repeater level existing in Protocol 2, due to the repeater level variation degree mechanism. In such scenarios, were both repeater level distance and repeater level variation are low, even 1 repeater level is already an important value.
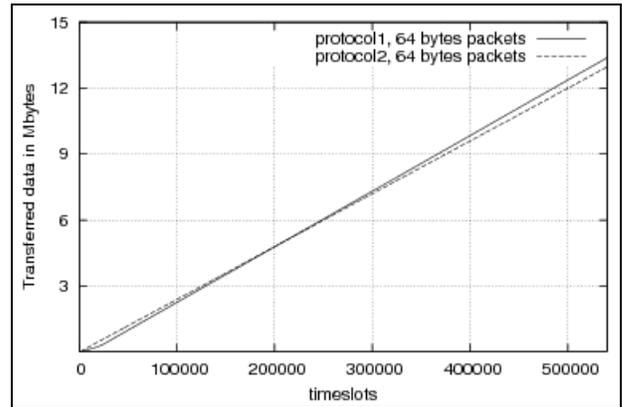
The amount of data transported in each packet is also very important in such a type of limited bandwidth networks. The next set of simulations was conducted with network layer packets of 64 bytes.

In each polling cycle, the transferred data is of (PS-HS)*(N-1)*2 bytes, where PS is the packet size, HS is the packet header size and N is the number of nodes of the network.
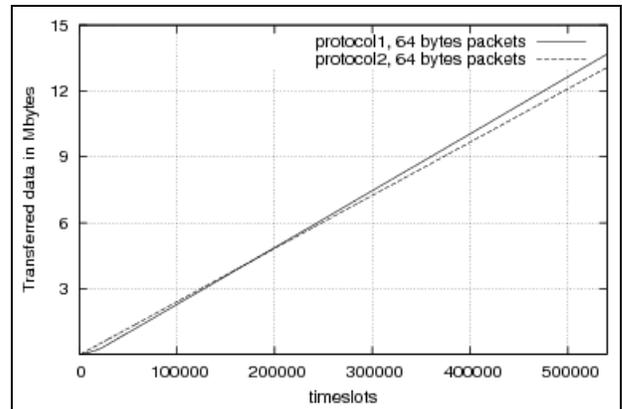
Figures 5-8 show that Protocol 1 systematically provides a higher data throughput. In Figure 6, with a network of 10 nodes, this tendency is recognizable from the start, while in the remaining scenarios, the start-up phase has a greater influence. In smaller network the stabilisation of repeater level by the protocols is faster than in larger networks, which require more time too. In this way, setting up the default repeater level is of great importance, since it controls the start-up phase phenomena observed in the simulations. → We propose that an analysis of the specific power-line characteristics to be made previously before deploying the system.
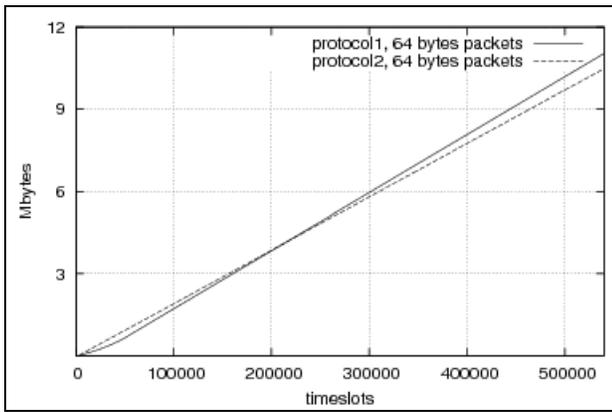
Figure 9 Transferred payload data in random 200 network.

Table 2 summarises the percentage gain in data throughput of Protocol 1 in the simulated scenarios, without the start-phase influence.

Packet size	Ring 10 network	Ring 100 network	Rand 100 network	Rand 200 network
64 bytes	15,0%	3,2%	4,4%	

Table 2 Protocol 1 data throughput gain percentage relatively to Protocol 2.

D. SPECIAL CASE: OPEN RING MODELS

Open rings are a special type of power-line networks that transform its topology by opening or closing a switch. The deployment of this type structure is justified in order to guarantee a higher availability of the network. The Physical Layer Emulator provides such a type of topology with one hundred nodes. After a certain number of timeslots, a switch is alternatively opened at node 20 or node 70. Simulation results are also available for this special model, as seen next.

Simulations were done with alternatively opening and closing switches in periods of 10000 or 50000 timeslots.
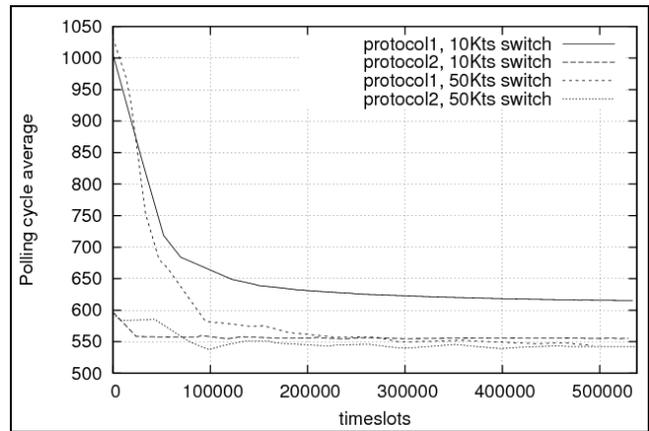
Figure 10 Polling cycle average comparison for open ring 100 network.

This scenario places us in a situation having constantly start-up phases (for analogy with well behaved power-lines), since abrupt topology changes provoke an high variation on the repeater level to attain a certain node.

Figure 10 presents the polling cycle average for the two simulated scenarios in the open ring network with 100 nodes. Both protocols adapt well to the scenario where the period of topology switch is of 50 thousand timeslots. As expected, Protocol 2 adapts faster, but Protocol 1 also behaves well.

The interesting point in this figure regards the scenario where topology changes with periods of 10 thousand slots. Consider the case when, due to a switch closing, both protocols overestimate the repeater levels to attain their slaves. On one hand, Protocol 2 realises this shortly and quickly adapts to this situation. On the other hand, this time period isn't enough for Protocol 2 to adapt, as it is perceptible in Figure 10.

| | Protocol 1 | | Protocol 2 | |
	10K ts switch	50K ts switch	10K ts switch	50K ts switch
% of affected nodes	0%	13%	15%	15%
% of lost packets	0%	0,086	0,339%	0,07%

Table 3 Packet losses in open ring 100 network.

A linked issue that can be raised in this discussion relates with packet losses in the network layer. These sudden topology changes, whether related to open switch or disturbed networks can diminish the efficiency of the protocol. A standard mechanism like packet retry should be well designed .

In all described simulations, each packet sending can be retried two times before considering the packet as lost. In

each retry, the repeater level of both uplink and downlink by one unit.

This is not enough for open ring topologies where the repeater level difference is larger than two repeater levels. Table 3 lists all packet losses occurred in the open ring scenarios. Notice that Protocol 1, in the case of 10 thousand period switches doesn't lose any packet because of the slow swiftness of this protocol to react to the optimal repeater level, maintaining an overestimation of this value, which permits not to lose packets in the next switch period.

Remark that in well-behaved networks two retries are enough since the highest repeater level difference is two (refer to Table 1).

CONCLUSION

In this paper we present an environment for simulation of network layer protocols, based on the Physical Layer Emulator. The main result of this paper is the presentation of an efficient SFN protocol that shows very good results in stable power-line networks. Besides, it also shows a good adaptation to networks with temporary topology disturbances, since its reaction to the optimal repeater level is slower. Its efficiency was analyzed by comparison with another proposed protocol, whose main characteristic is its

faster adaptation to strongly disturbed power-lines, or switched ring networks, at the expense of a larger packet header. We also discussed about the effect of the default repeater level on the start-up phase, and the influence of the retry parameter on network layer packet losses, especially in highly disturbed networks. The next logical step would be to provide a more generalized protocol that behaves well in all types of power-line networks.

REFERENCES

[1] Gerd Bumiller, "Single Frequency Network Technology for Medium Access and Network Management", In *6th International Symposium on Power-Line Communications and its applications*, March 2002.

[2] Gerd Bumiller, "Power-Line Physical Layer Emulator for Protocol Development**",** In *8th International Symposium on Power-Line Communications and its applications*, March 2004.

[3] Gerd Bumiller, Liping Lu and Yeqiong Song, "Analytic performance comparison of routing protocols in master-slave PLC networks", In *9th International Symposium on Power-Line Communications and its applications*, April 6-8, 2005.