IPP-HURRAY! Research Group

http://www.hurray.isep.ipp.pt

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

Communication Protocol for the Rempli Project

Cătălin PORNEALĂ
Mário ALVES

HURRAY-TR-0417
June-2004

relatório técnico

technical report

Communication Protocol for the Rempli Project

Cătălin PORNEALĂ, Mário ALVES

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: {malves}@dee.isep.ipp.pt
http://www.hurray.isep.ipp.pt

Abstract:
The overall goal of the REMPLI project is to design and implement a communication infrastructure for distributed data acquisition and remote control operations using the power grid as the communication medium. The primary target application is remote meter reading with high time resolution, where the meters can be energy, heat, gas, or water meters. The users of the system (e.g. utility companies) will benefit from the REMPLI system by gaining more detailed information about how energy is consumed by the end-users.
In this context, the power-line communication (PLC) is deployed to cover the distance between utility company's Private Network and the end user.
This document specifies a protocol for real-time PLC, in the framework of the REMPLI project. It mainly comprises the Network Layer and Data Link Layer. The protocol was designed having into consideration the specific aspects of the network: different network typologies (star, tree, ring, multiple paths), dynamic changes in network topology (due to network maintenance, hazards, etc.), communication lines strongly affected by noise.

1. Introduction

The focus of the REMPLI project is on the operating of energy distribution networks where an envisaged communication infrastructure is supposed to provide functionality like management of distributed networks, registration of energy consumption, detection of energy loss in the network and tariff management. The related communication infrastructure has to provide connections from the utility companies or other companies to the end user.

I this context, power-line communication will provide connection between the utility company and end user. Medium voltage lines are used for communication between primary transformers, which are already connected to the company's network, and secondary transformers. Low voltage lines are used for communication between secondary transformers and the end user's premises. For this purpose, PLC is used by different types of units within the network, in the following referred to as REMPLI PLC units. These units are:

- the REMPLI Access Points that make up the interface between the company's Private Network and power-line network
- REMPLI Bridges that connect medium-voltage and low-voltage lines
- REMPLI nodes that connect the end customers to the power-line

For communication in this network, iAD already has a well tested protocol based on the NMS concept. However this protocol has a lot of limitations. It is expected that a protocol based on the SFN concept will be more suited for this kind of networks. The purpose of this document is to present a protocol based on the SFN concept and according to the simulation's results to decide if the new protocol is better or not than the existing one.

First, an overview of the goals of the project and the topology of electrical networks will be given. The third section provides an outline of the REMPLI system architecture. Then, the two existing protocol concepts will be presented: NMS and SFN. The NMS concept is based on a system that already exists and is well tested in rough environment. The SFN concept is expected to be better, but potentially the protocol is not defined yet. Section 5 presents a detailed description of the protocol. In the end will be presented the conclusions and the advice whether the newer protocol should be used or not.

2. REMPLI – goals and energy distribution networks characteristics

In order to meet the new needs in the electrical market, such as increasing demand for energy, increasing costs for primary energy, a strong tendency towards liberalization, minimizing the energy loss/leakage, the purpose of the REMPLI project is the definition and implementation of a communication infrastructure for data acquisition and control operations, which is suitable for distributed/remote monitoring and metering, but also for tasks that are not yet planned. The primary objectives of the project relate to the actual operation of energy distribution networks, focusing on the registration of energy consumption, the management of distributed networks, the energy loss detection, tariff management and network usage calculations within a liberalized power market.

The medium- and long-term goals are to provide inclusion of add-on services that may include:

- Metering services that provide data for authorities, industry, marketing and advertisement
- Advanced energy control functions (fine-grained monitoring and control of load profiles and load thresholds, peak limitation, trend analysis)
- Intelligent billing and fraud protection
- Provision of communication infrastructure for third party providers
- Energy and customer consulting
- Providing the customer with add-on services, such as remote control of climatisation, assistance services like gas and water leakage detection, fire detection.

2.1 The REMPLI system architecture

The basis of the REMPLI system is a power-line communication (PLC) infrastructure that allows accessing metering and controlling equipment remotely. The communication platform is open to various kinds of add-on services such as climate control, switching control, burglar alarm.

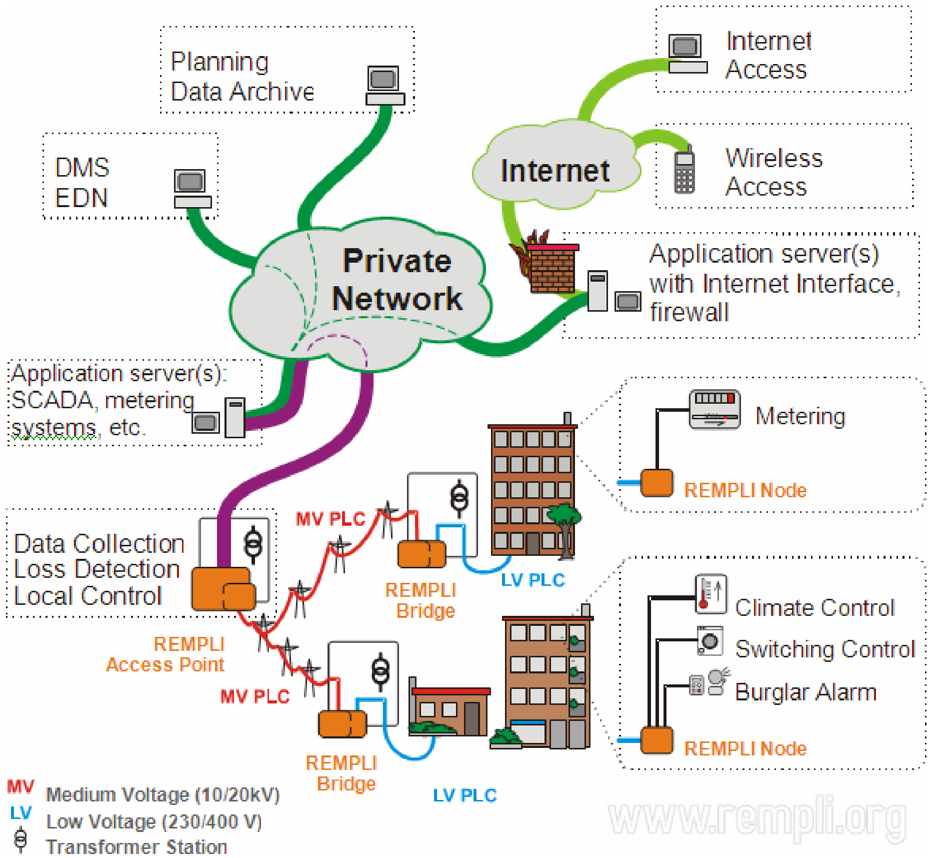


Figure 2.1: The REMPLI system architecture

As shown in the figure, the REMPLI communication infrastructure consists of:

- low-voltage segments (blue lines), which cover groups of energy consumers;
- medium-voltage segments (red lines) between the primary and secondary transformer stations;
- TCP/IP or IEC 60870 based segments (thick purple lines) between the primary transformer stations and the Application server(s).
- TCP/IP communication (green lines) between the Application Servers and their clients. The interfaces provided by the Application Servers could be available only within the Private Network or also by Internet clients.

The bottom-level of the communication infrastructure is comprised of **REMPLI Nodes**, each coupled with a PLC interface (usually a low-voltage PLC slave, in certain cases also with medium-voltage PLC interface). A node is usually installed at the consumer site, e.g. inside an apartment, and has a number of metering inputs. It is also intended to equip each node with digital outputs that would allow switching off and on electrical/heat/gas/water supply for a particular consumer, upon commands from the utility company.

All nodes within a REMPLI installation are connected to a cascaded powerline network. The powerline network consists of one or multiple Low-Voltage and one Mid-

Voltage segment. Communication at both levels is Master/Slave-based. Low- and Mid-Voltage segments are coupled by the **REMPLI Bridge** which is installed at the secondary transformer station, between two parts of the cascade. Physically the Bridge is comprised of an interconnected high-voltage PLC slave and a low-voltage PLC master.

In some cases, when the utility company desires, it is possible to combine the Bridge with an external Node, or replace the whole Bridge with a Node. In this case, the transformer station becomes a communication end-point and no data transmission occurs in the Low-Voltage segment.

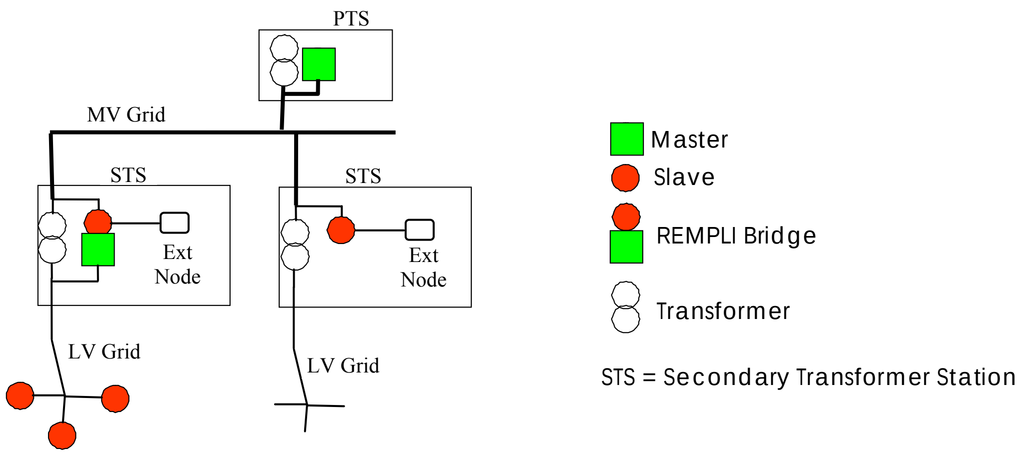


Figure 2.2: Masters and slaves in the electrical power-grid

The transition between PLC and Private Network communication environments is carried out by the **REMPLI Access Point**. Access Points are installed in the primary transformer stations. If Private Network access at the secondary transformer station exists (seldom), the REMPLI Access Point is installed there and only the Low-Voltage PLC segment is used for communication.

An Access Point can be used by several Application Servers simultaneously, although, due to performance constraints, this number is limited.

The Private Network segment can also contain some other hosts, implementing functions, not covered by the Application Server. An example would be a billing server for a utility company, which uses measurement data collected by the REMPLI system. These additional systems, normally do not communicate with Access Points directly, retrieving required data from the Application Servers instead.

3. Relevant characteristics of the MV and LV power-grids

The electrical power grid can be divided into:

- a transport grid and
- a distribution grid.

The transport grid connects the production centers (i.e. power plants) and the distribution network. It comprises lines at voltages between 110 kV and 400 kV. It is possible to have also smaller voltages like those in the French sub-transmission lines of 63 or 90 kV. This part of the network is called the high voltage (HV) segment.

The distribution grid binds the consumers and the transport grid. On a first level the high voltage transport network is connected via substations (REMPLI primary transformer stations) to the medium voltage lines. Transformation posts (REMPLI secondary transformer stations) finally convert the medium voltage level to the end-user voltage level. The distribution grid can be divided in to 2 segments:

- the medium voltage (MV) segment – usually 10 to 60 kV
- the low voltage (LV) segment – 220/400 V.

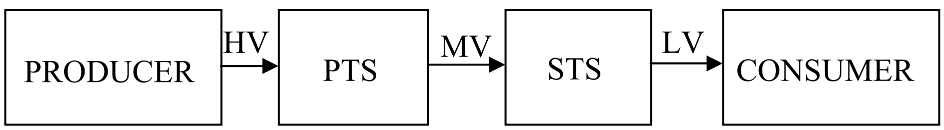


Figure 3.1: The distribution of the electrical energy

The energy from the producer is transported via HV grid to the Primary Transformer Station (PTS) where is transformed to Medium Voltage (MV) energy. Then trough the MV grid is transported to the Secondary Transformer Station (STS) where is converted to Low Voltage (LV) energy, which is finally distributed to the consumers via Low Voltage grid.

Topology of the electrical power grid

The high voltage (HV) segment:

- in Portugal and Bulgaria commonly has a **closed ring topology** with a few cases of o**pen ring topology**
- in France and Netherlands transmission lines are organized in **meshed networks** whereas sub-transmission lines are operated in a **closed loop**.

The medium voltage (MV) segment:

- in Portugal has mainly an **open ring topology** with a few cases of **star topology** both for urban and rural areas
- in Bulgaria has an **open ring topology** where the structure is reconfigurable. MV/LV transformers can be connected(switched) to different HV/MV transformers to install backups in case of failure in transformers and/or power-lines. This fact must be considered in the

design of the protocol, because there is needed to adapt to the disconnected REMPLI bridges and MV Lines.

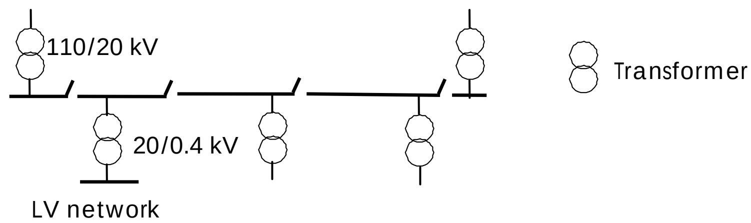


Figure 3.2: Structure of Sofia electrical network

- in France has a radial (tree) structure or an open loop.
- in Netherlands has a meshed network structure build in rings, mostly not closed, which make it possible to switch the supply to a secondary transformer station from different directions.

The low voltage (LV) segment:

- in Portugal has an open ring topology for urban areas and star topology for rural areas
- in Bulgaria is a mixture of star and tree topology
- in France can be radial (tree style), open loop or double shunt
- in Netherlands is typically a tree network.

As we can see, the communication will have to deal with different kinds of network topologies, but this is not a problem because our approach is suited for any type of topology**.**

4. Candidate protocol approaches: NMS and SFN

The power-line networks, we want to work at REMPLI are very large. In this kind of PLC networks repeaters are necessary. Both the NMS and SFN use the master (M) slave (S) concept, with the repetition of the message at the slave level.

The network management system (NMS) is based on a master-slave concept where all modems share the same frequency band and thus a TDMA scheme is applied. The use of different frequency bands enables the coexistence of several logical networks on the same physical network. In order to expand the physical range of the network, two repeater levels are used and every slave can also be a repeater. There is no direct communication between two slaves and hence, the network structure represents a virtual star topology. Slaves are only allowed to transmit on demand of the master. As OFDM performs parallel transmission of bits, decoding can only start after the reception of a complete block. Therefore, responding is not possible within the next TDMA slot but instead an interleaf is specified after which the slave is supposed to respond. Thus, system delays caused by run times of both the physical layer and the layers of the network management are taken into account.

We will consider in the following example a TDMA scheme with 3 logical channels. As iAD suggests, the necessary number of logical channels should be between 3 and 7. In the example are one master (M) and 4 slaves (S1, S2, S3 and S4).

In the first logical channel the master initiates a communication with slave 1 (S1). S1 is supposed to be very close to the master and receives correctly the message transmitted. As soon as the message is received, the decoding begins and then the message is transmitted to the higher level. But the decoding takes some time. Also does the interpretation of the message by the higher level and the response from the higher level. It results that the response is not available when the time reserved for the channel 1 elapses. To solve this problem, time slots are used. The response of the slave 1 is send after an amount of time, when channel 1 is reserved to be used again.

The second logical channel is used for communication with slave 4 (S4). But slave 4 is supposed to be physically far from the master and the signal to it to be degraded. But slave 3 (S3) receives it correctly. S3 retransmits the signal further when the channel 3 occurs again. Now S4 receives it correctly. In the next channel 3 the slave 4 transmits the response to the master. The response does not reach the master with sufficient power and a retransmission by slave 3 is needed.

The third logical channel is used to transmit a non-acknowledge message to slave 4. A repetition by the slave 2 will be required, as the slave 4 is too far from the master and it will not receive the message correctly the first time (when transmitted by the master).

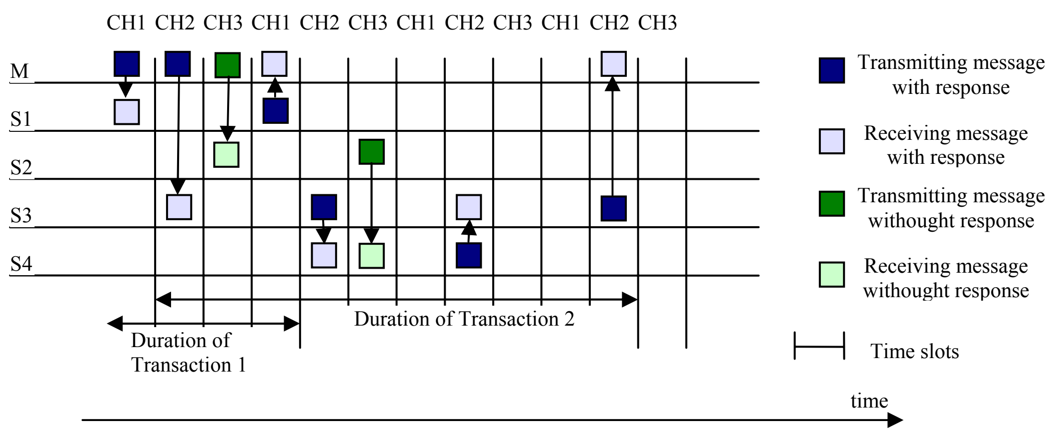


Figure 4.1: TDMA scheme for 3 logical channels - NMS

NMS uses fixed routing for the transmitted message. This means that every slave keeps a table with its favourite repeater slaves and the master periodically asks each slave for the best five entries in its table. Based on this, the master is able to construct a path for the message. The path includes the ID of each slave that the message will have to travel through.

The permanent polling of the participants to measure the channel characteristics involves a high overhead of the communication and complex data processing. It is expected the channel attenuation of power-lines to be strongly time-variant. This way a fixed routing of the communication by the repeaters would not be applicable and a single frequency network (SFN) concept might be a better solution. Also because of the electrical power network characteristics, two levels of the repetition of the message would not be always enough.

The SFN uses a very simple repeater concept: all participants who have correctly received a packet from the master retransmit this packet at the same time, i.e. in the next time slot, which is reserved for the repetition, on the same medium and the same frequency.

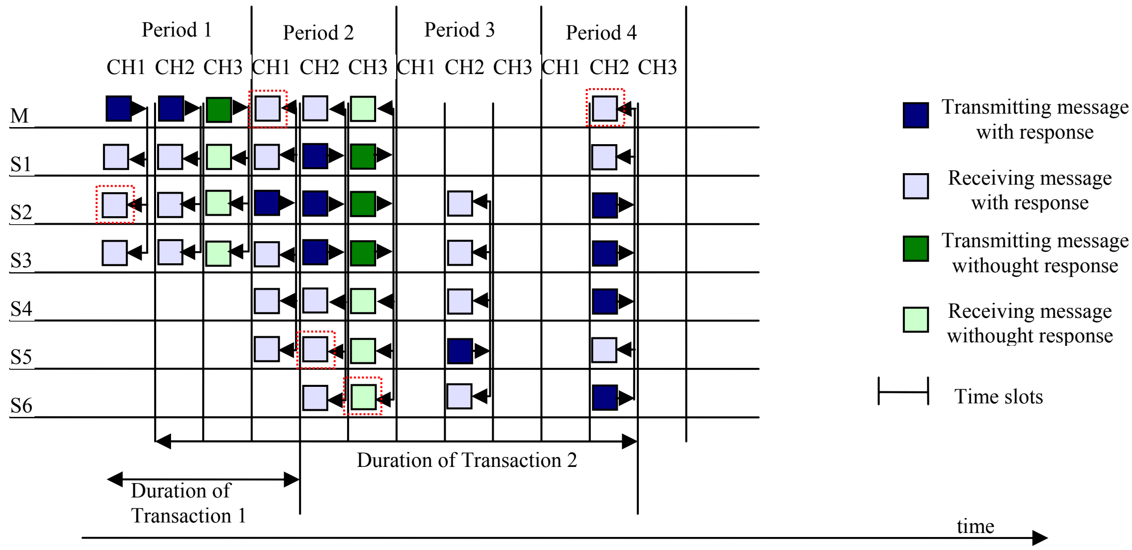


Figure 4.2: TDMA scheme for 3 logical channels - SFN

In the first logical channel the master initiates a communication with slave 2 (S2). The master transmits the message and it is successfully received by the slaves 1, 2 and 3 (S1, S2, S3). The slaves 1 and 3 will ignore the message. As soon as the message is received, it is decoded and transmitted to the higher level. The next time when channel 1 (CH1) occurs, i.e. in the period 2, the slave 2 transmits the response to the master. It will be received by the master and slaves 1, 3, 4 and 5 (M, S1, S3, S4, S5). Only the master will consider the message. The others will ignore it.

In the second channel, master initiates a communication with the slave 5 (S5). But slave 5 is too far from the master. The message will be received correctly only by the slaves 1, 2 and 3. The next time when channel 2 occurs, i.e. in the period 2, the message will be retransmitted by the slaves 1, 2 and 3 (the ones that have received it correctly previously). Now the message will be received correctly by the master and the slaves 4, 5 and 6. Master, slave 4 and slave 6 will ignore it. Slave 5 will transmit it to the higher levels. In the next period, i.e. period 3, the slave 5 retransmits the response to the master. The master is too far from slave 5 and the message will be received correctly only by the slaves 2, 3, 4 and 6. In the next period (period 4), the message will be retransmitted by the slaves 2, 3, 4 and 6. It will be received correctly by the master, slave 1 and slave 5. Slave 1 and slave 5 will simply ignore the message.

The third logical channel is used to transmit a non-acknowledge message to the slave 6. A repetition by the slaves 1, 2 and 3 will be required.

A drawback of the NMS protocol is that it accepts only two retransmissions of the message by the slaves. In many cases because of the characteristics of the electrical power-grids this will not be enough. The new protocol must virtually support an unlimited number of retransmissions.

One major difference between the NMS concept and the SFN concept is that NMS uses a **message routing** mechanism and SFN a **repeating mechanism**. This will allow a master to control a higher number of slaves, due to the fact that routing calculations and the estimation of the quality of connections between slaves will not be needed anymore. In the NMS concept, the number of computations for the routes grows with more than square of the number of stations; the SFN concept eliminates the calculation of the routes.

5. A protocol description for the SFN approach

5.1 Master-Slave (M/S) architecture

Master – Slave management assumes that the slaves only answer to the master's requests. This will solve the problems of resources management and synchronization. Resource management refers to the fact that this way the slaves, which share the same medium, will "know" when to transmit. The synchronization is actually done when at the beginning of each message. Before a message is sent, a synchronization sequence is transmitted. The slaves consider the sequence of bits that follow the synchronization only if the synchronization part was correctly received. The synchronization part is done by the physical layer, so our protocol should not take care about it.

In order to assure reliable communications between masters and slaves, some messages are going to be repeated by the slaves until reaching the destination (master or slave). The number of repetitions for a certain message transaction is defined by the master, which embeds this parameter inside each message. We will assume no upper limit of retransmissions. In practice due to some particular aspects of the initialization of the network (see subsection 5.3), this parameter needs to have a limit. However this limit can have different values for different segments of the network, so it can virtually supposed to be infinite.

We consider two different retransmission approaches:

1. each slave retransmits the message one (and only one) time
2. a received message is retransmitted by each slave that received it correctly and by the master until the remaining number of hops is zero.

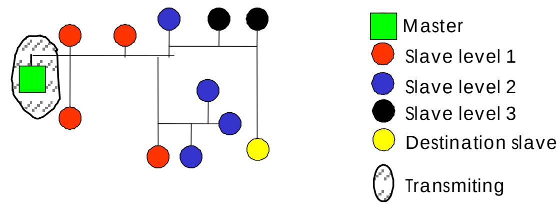
First approach:

- each station retransmits the received packet only one time. This avoids closed loops and other problems caused by the fact that different copies of the message could arrive at the destination with a delay higher than the delay spread that is accepted by the system.

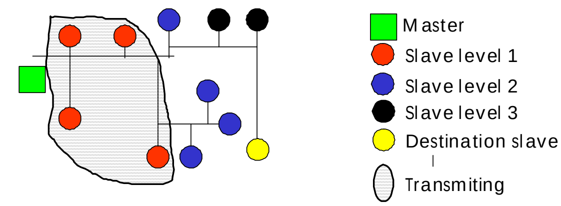
In the following example we suppose that the slaves level 1 are those slaves that receive correctly the message transmitted by the master, the slaves level 2 are those that receive correctly the message retransmitted by the slaves level 1, and so on. The decision that a message was correctly received is done verifying the checksum that is attached at the end of each message.

Example for the first approach: master sends a frame with number of hops 3

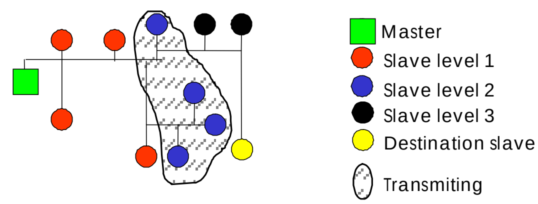
- first period only master is transmitting the frame with number of hops 3



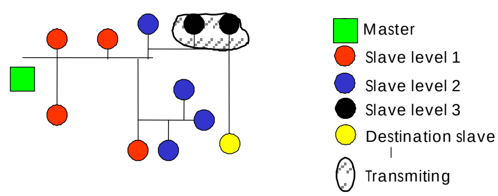
- second period the slaves level 1 are transmitting the frame with number of hops 2



- third period the slaves level 2 are transmitting the frame with number of hops 1



- fourth period the slaves level 3 are transmitting the frame with number of hops 0



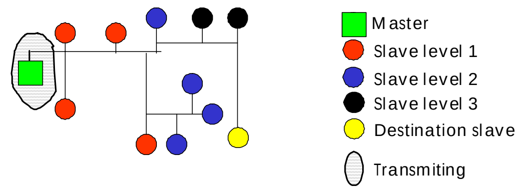
Second approach:

- each station that has received the packet correctly decrements the number of hops and retransmits it until the number of hops is zero.

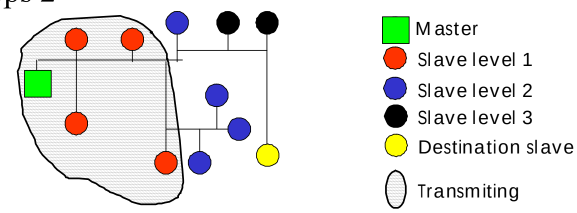
In the following example, we consider that the slaves level 1 are those that have received correctly the message transmitted by the master, the slaves level 2 are those that received correctly the message transmitted by the master and the slaves level 1, the slaves level 3 are those that received correctly the message transmitted correctly by the master, the slaves level 1 and the slaves level 2. The decision that a message was correctly received is also done verifying the checksum that is attached at the end of each message.

Example for the second approach: master sends a frame with number of hops 3

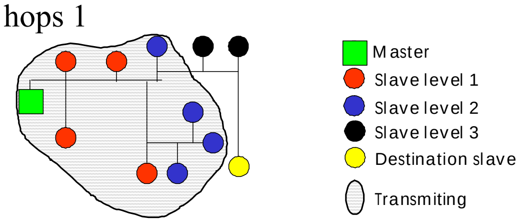
- first period only master is transmitting the frame with number of hops 3



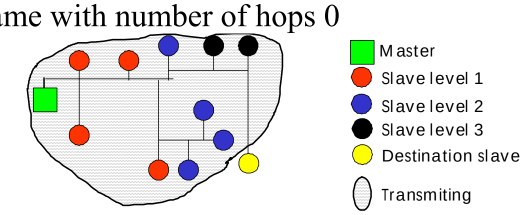
- second period the master and the slaves level 1 are transmitting the frame with number of hops 2



- third period the master, slaves level 1 and slaves level 2 transmit the frame with number of hops 1



- fourth period the master, slaves level 1, slaves level 2 and slaves level 3 transmit the frame with number of hops 0



Using the second method of transmission we could get a better signal in the transmission medium, but due to the superposition of the signals the problem of message jitter may become too complex to handle (the slave will have to decode many packets arriving at different time moments). So we will assume the first method.

5.2 Master Domains

A **Master Domain** consists of one **Master** and the **Slaves** that can be accessed by that **Master**. The **Master** sends requests to the **Slaves** and the slaves execute different tasks and respond to the masters. The management of the moments when slaves should retransmit is done only by the masters.

Usually we will have only one **Master Domain** in a certain segment of the network, i.e. only one master connected to the same physical medium. But there will be many situations when 2 or more masters are connected to the same LV (or MV) segment of the network. The next figure shows a possible case when two masters share the same physical medium. The cases with more than two masters are similar to this one.

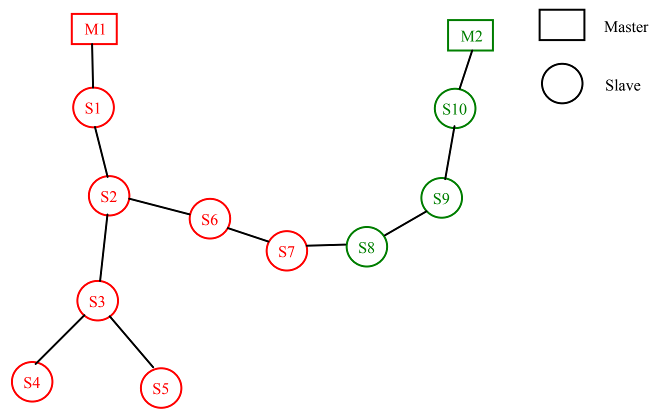


Figure 5.1: Master Domains sharing the same physical medium

In this situation each master will have some of the slaves. The left master will manage only the red slaves, the right one only the green ones. The red master and slaves constitute Master Domain 1 and the green master and slave constitute Master Domain 2. One master and the slaves that respond to its commands represent a **Master Domain**. It is possible to have slaves that belong to more than one **master domain** (Master Domain overlapping):

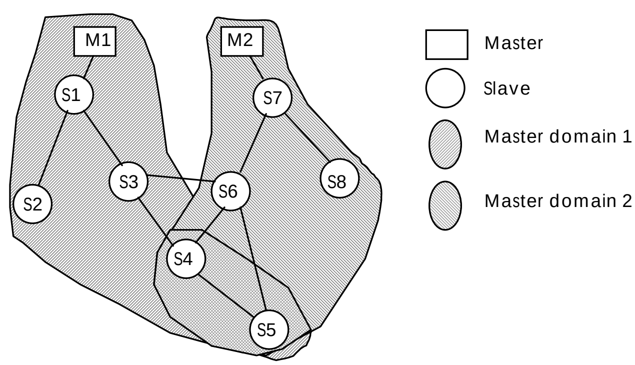


Figure 5.2: Master Domains overlapping

As we can see in the above picture, the slaves 4 and 5 belong to both master domains; they will respond to both Master 1 and Master 2 and will handle different addresses for each of the masters.

If there are two masters on the same medium, there are two options to solve the resource (medium access) management problem:

1. by using two times more time slotted logical channels. As shown in the picture, one master will use half of the channels and the other master the other half of the channels.

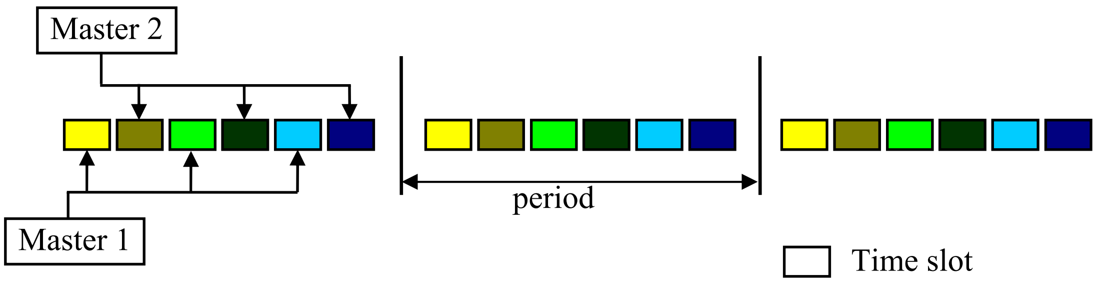


Figure 5.3: Distribution of timeslots between Master Domains

The two masters must previously know which slots from a period of time to who belong, or to have at a higher level a mechanism for determining the slots ownership. Eventually at the network layer might be implemented an algorithm for setting the slots management.

The solution is similar for **n** masters - using **n** times more time slotted logical channels. This method will make the transmission **n** times slower.

2. by using different frequencies for communicating with each master. But this method implies modifications of the decoding scheme that is build into the DSP, and also makes the problem of one slave responding to many masters very complex. The problem of a transmission that would be two times slower will not be solved. This happens because of the particularities of OFMD transmission: one master will take half of the frequencies currently used for multiplexing and the other one the other half. This way the quantity of information that can be transmitted by a master will be divided by two.

So, we will consider the first option.

Having a "big" number (10, 100 or 1000) of masters connected to the same medium would result in a very slow transmission because the dividing the time slots across the masters. This might turn into a serious problem if the situation is met in parts of the network where the transmission is important, such as center of a big city. It also makes the initialization and maintenance phase to take a significant portion of the communication time. The worst case will be when the network will be filled only with maintenance frames.

This aspect leads us to reconsider the suggestion made by Loria in which the maximum number of logical channels can be assumed with 7. Having a maximum number of 7 logical channels will lead to a maximum number of 7 Master Domains sharing the same physical medium: one logical channel for each Master Domain in a period.

5.3 Master Domain initialization phase

Initially, each **Master** must have a list of all the **Slaves** that have permission to belong to its communication domain, called **List of Allowed Slaves** (LAS). Each Slave in the network has a unique identifier **Slave ID** (serial number; physical address), which will be used by the **Master** to build the **List of Live Slaves** (LLS). In the initialization process, the **Master** has to contact each **Slave** using the **Slave ID,** in order to assign it a MAC address (**Slave MAC**); this address will be kept by Master and Slave and used for future communication.

LLS (List of Live Slaves) will contain entries for each Slave that is “aLive” at the moment, each entry made of:

- **Slave ID** (serial number)
- **Slave MAC Address**(the ID that was assigned by the Master)
- **Number of Hops** (NoH) that are necessary to reach the Slave
- **Last Seen Live Timestamp** (LSLT)– the moment when the Slave was accessed the last time

At the moment of initialization, the necessary **Number of Hops** for each Slave is not known, so the process of MAC assignation will also include the determination of **Number of Hops** for each slave, also named “discovery of the network topology”.

If a slave can be accessed by **n** master, it will have **n** MAC addresses and will respond to each master in the assigned timeslot.

The algorithm of Master Domain initialization:

```
for no_of_hops=0 to MAX_NO_HOPS
        for each slave in LAS
            send initialization packet(Slave ID, Slave MAC, no_of_hops, no_of_timeslots)
            if response
                        add slave to LLS with the no_of_hops and time
                        break //breaks the for loop for no_of_hops
            else
                        continue //continues with the next value for no_of_hops
            endif
        endfor
endfor
```

Algorithm 5.1: Master Domain initialization

MAX_NO_HOPS is a constant that represents the maximum distance in hops that is accepted by the domain for a Slave. It may be different from one Master Domain to another Master Domain. If there is more than one master on this medium, the initialization packet that is sent to the Slave will contain information about the period, the number of time slots in a period and which time slots belong to the Master that is sending the initialization packet. Actually the last one does not seem to be important at this moment because the slaves will only retransmits the message or respond to it; so they only need to “know” when to retransmit or when the response should be done. These can be calculated only using the NoH_left and the number of timeslots in a period. This way the Slave will be able to belong to more than one Master Domain and be able to handle different MAC addresses for each Master Domain.

It is possible to have 2 tables for number of hops (one with the number of hops from master to slave and one with the number of hops from slave to master), or only one table (the number of hops from master to slave is equal to the no of hops from slave to master). The first method offers a better usage of the network, but is more complicated to implement and the initialization takes much more time and the maintenance of the tables is much more complicated: the complexity is proportional with the square of the number of hops to the slave ; so the second method (only one table of hops) would be better to implement because it is expected the number of hops M-S (master to slave) and number of hops S-M (slave to master) to be the same or very close.

5.4 Master Domain normal operation phase

The master sends frames and the slaves respond to these frames. It is also possible to send frames that do not require response from the slave.

At this level, when the destination slave is correctly receiving a frame:

a) if the number of hops is 0 transmits the answer the next time period
b) if the number of hops > 0 it will wait a number of time periods represented by the number of hops and then retransmits the answer.

The slave will always embed in the response the number of hops left for M-S transmission (**NoH_left**) that the packet was travelling with when it was successfully received. This way the master will be able to dynamically update the tables with the most adequate number of hops. When the Master receives the response from the Slave, it updates) only if necessary the Number of Hops in the LLS (List of Live Slaves and also the Last Seen aLive Timestamp (LSLT).

During normal operation of the system, supposing that the Master uses 2 tables of hops: one for Master-Slave communication and one for Slave-Master communication are two cases when Master send a frame to the Slave that requires response:

- **case1: Master does not receive answer from the Slave**

A retry is required. The numbers of repeating levels was not high enough. The Master has no information if the transmission failed in the down link or in the up link. To have a high probability, that the retry is successful, the master has to increment the number of hops downlink (**NoH_down**) and the number of hops uplink (**NoH_up**). If the retry was not successful, the Master increments again **NoH_down** and **NoH_up**. This is done until:

- successful transmission
- maximum number of retries
- upper limit for **NoH_down** and **NoH_up** is reached

On the most cases only one retry will be necessary. The Slave can notice, if the first request was successful or not. The Slave will set one special bit in the response, if it has received successful the packet during the last transmission. If the transmission downlink was successful, the failure happened during uplink transmission. So, the increment of the **NoH_down** was not necessary and it will be decremented to reach the old value.

- **case2: The transmission was successful without retry**

In this case if the Slave has received successful the message before the number of hops > 0, it would be better to decrement **NoH_down**. But the gain of a decrement of **NoH_down** is much lower than the losses through a retry. The calculations from iAD show that a decrement of **NoH_down** will be useful if done only after a number of consecutive successful receives before expiration of the number of hops. This number has to be higher than **NoH_down** + **NoH_up**. The case when the Master receives the response before the expiration of the number of hops is similar.

Periodically, the Master must check the Slaves that are in the LLS and have the LSLT < current_time – MAX_ALLOWED_INACTIVE_TIME. The check is done sending a special message that requires confirmation. If no confirmation is received, the failure mechanism described before is to be applied.

5.5 Master Domain maintenance

The system requires that Slave nodes are able to join and leave a Master Domain (turning on and off the equipment). This implies the necessity of periodical querying the Slaves that are in the LAS but are not in the LLS, but the number of Slaves in LAS will be much higher than the number of Slaves in LLS potentially leading to an overload of the network. A solution would be to first check if there is a new Slave in the network. The Master sends a message asking if there is a new Slave. This message will have the number of hops equal to MAX_NO_HOPS (and will contain information about the period of messages, the time slots in one period and which time slots belong to the sending master). Only if the Master receives a response for this message, will initiate the process of searching for new Slaves. Possible failure of the mechanism: a new Slave appears in the network and it is not in the LAS. Solution: the Slave responds to the special message for new slaves only a limited number of times (2 or 3). After this number of receives of the message, the slave will ignore the new slave message and will signal the problem on its display (supposing it has one). This may cause a small loss in efficiency if a new slave appears at the border of two Master Domains.

The algorithm for searching new Slaves is very similar to the algorithm of Master Domain initialization:

```
for no_of_hops=0 to MAX_NO_HOPS
        for each slave in LAS and not in LLS
                send initialization packet(Slave ID, Slave MAC, no_of_hops, no_of_timeslots)
                if response
                        add slave to LLS with the no_of_hops and time
                        break //breaks the for loop for no_of_hops
                else
                        continue //continues with the next value for no_of_hops
                endif
        endfor

endfor
```

Algorithm 5.2: Searching for new slaves

5.6 MV and LV segments interconnection

In this protocol each master is responsible of managing its own slaves. We will have 2 types of masters: the access point (its slaves are the bridges from the MV/LV transformers and other slaves) and the bridge from the MV/LV transformers (its slaves are the nodes). This implies a problem when transmitting a message from access point (AP) to one node (N): the access point (AP) does not know exactly when the response from the node (N) should arrive, only the bridge can know how long it would take the massage to go to the node and come back.

This problem should be solved by the transport layer of the protocol and does not make the subject of this document. In the following we will present a summary solution of the problem, as it can be seen from the point of view of the design of the Data Link Layer and the Network Layer.

When **AP** sends a message to a **Node** in the LV network, it cannot know exactly when the response should arrive. **AP** only knows the number of hops until the **Bridge**. The number of hops from the **Bridge** to the **Node** would be managed by the **Bridge** itself. This problem could be solved using the following communication strategy:

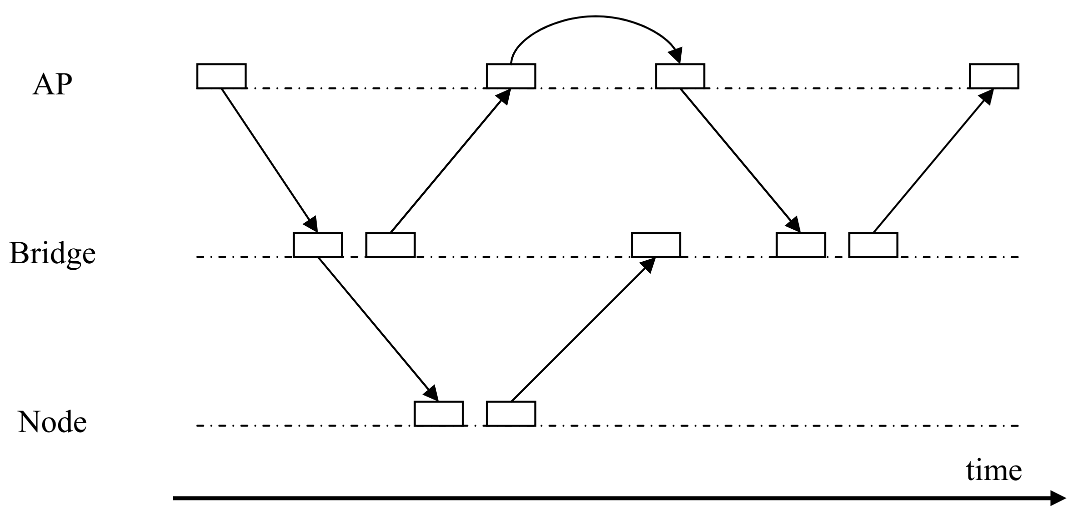
- AP sends a frame which destination **Node** is in the LV network, including in the frame the MAC of the **Bridge** that is responsible to retransmit the message in the LV network
- The Bridge answers telling if the Node is alive or not and if it is alive also sends the **estimated time** when the requested information is supposed to be available. The estimated time is derived from the NoH (number of hops bridge to node) and the load of the bridge.
- After the amount of time specified by the Bridge in the response, the AP asks again the Bridge about the information it has previously requested.
- The response from the Bridge could be: the requested information, information not ready yet – wait until … some timestamp, the Node is not alive.

- If AP receives information not ready yet, after the specified time, asks again the Bridge. The process is repeating until the AP receives either the requested information or Node not alive.

For this mechanism to be possible is needed the AP to have a **List of Requested Jobs** (LRJ) to the Bridges. An entry in LRJ should contain: the id of the job, the id of the slave that should take care of the job and the estimated timestamp when the slave (Bridge) will be able to provide the result.

If any communication failure between Master and Slave (AP/Bridge or Bridge/Node) appears, the failure mechanism described in subsection 5.4 is to be applied.

Sometimes, the interconnection between the MV and LV segments is not needed. It is the case when we have Nodes directly in the MV network or when we have the AP directly at the MV/LV transformer. In this case communication between AP and Nodes will be just a simple Master – Slave communication.



6. Conclusions

The SFN concept shows in many cases to be better than the NMS one. First of all due to the repetition mechanism an estimation of the quality of the links between slaves is not necessary anymore. Also the calculation for the path is not necessary, because the SFN does not use paths; it uses a flooding mechanism. Dynamic changes of topology or channel characteristics do not require an update for the routing tables. The same is true if a repeater goes out of order.

Other participants in the network do not know when the message reaches the destination. The master has to limit the number of repetitions. If the maximum number of repetitions is too low, the message does not reach the destination. If the number is too high, time slots are wasted. The master needs a table of all slaves and the number of repetitions in both directions. Due to the different disturbers at master and slave the quality of the channel is not necessarily identical in both directions and therefore the number of repetitions can differ between up- and downlink. The slave has no table anymore. The slave forwards the information about the necessary number of repetitions with the reply to the cycling polling.

Due to the repeating mechanism, the message is flooded in all directions and if the number of allowed repeater levels is high enough, this situation is equal to a saturation routing. The difference to saturation is the speed. The duration to get a message to all slaves is not higher than a single message on the optimum routing to the far end slave.