

The Nucleolus as a Strategy for Resources Optimization in LANs on Power Line Communications

Juan Carlos Vesga-Ferreira^{1*}, Gerardo Granados-Acuna¹, Javier Emilio Sierra-Carrillo²

¹Escuela de Ciencias Basicas Tecnología e Ingeniería (ECBTI), Universidad Nacional Abierta y a Distancia. Carrera 27 Nro. 40-43. Bucaramanga, Colombia; juan.vesga@unad.edu.co, gerardo.granados@unad.edu.co,

²Facultad de Posgrados, Universidad Pontificia Bolivariana. Circular 1ª No. 70-01. Bloque 9, of. 102. Medellín, Colombia; javier.sierra@upb.edu.co

Abstract

Background/Objectives: At present HPAV lacks an efficient mechanism for the optimization of resources. The objective of this paper is to propose the use of the Nucleolus as a strategy to provide a solution to the problem. **Methods/Statistical Analysis:** The paper raises two separate scenarios, formed by twelve nodes each, in conditions of channel and traffic clearly established. It proceeded to implement a solution to the problem through two methods: Nucleolus and Linear Programming, in order to make a comparison of treatments, through an analysis of means, under the use the test called paired-t and with this evaluate the behavior of the Nucleolus as a strategy for the optimization of resources. **Findings:** The use of Nucleolus is a novel strategy for optimizing resources within the PLC technology, considering the importance that game theory has been gaining as a tool to analyze complex optimization situations in various fields of knowledge and also that no found similar work in the various bibliographic sources consulted. Based on the results obtained it showed that the Nucleolus made a better allocation of BW in comparison with the optimization method of Linear Programming, to minimize the difference between BW requested and the BW allocated for each node, with a 95% confidence. However, although the method is efficient, the computational complexity of the algorithm increases dramatically as increase the number of nodes. **Application/Improvements:** It is recommended to investigate on alternative methods to calculate the Nucleolus that offer a computational complexity and temporary reduced, in order to facilitate its implementation in low-cost embedded systems.

Keywords: Game Theory, LAN Networks, Nucleolus, Power Line Communications, Resource Optimization

1. Introduction

Power Line Communications refers to a group of technologies that allow communication processes to be set by the use of electrical network as physical means of transmission. HomePlug AV (HPAV) is one of the most widely accepted standards on PLC technology that uses both CSMA/CA and TDMA as media access mechanisms, where CSMA

CA is oriented to the transmission of data packets while TDMA is implemented in the transmission of voice and video packets, in order to provide adequate levels of QoS¹. Despite HPAV can achieve transmission rates up to 500 Mbps, it has no any appropriate mechanisms of resource optimization, which greatly affects the performance of the network as the number of users increase, because only one node can transmit at a time². Besides, and consider-

*Author for correspondence

ing each node transmits permanently, a constant dispute arises to get access to the medium and there are cases where the nodes require differential bandwidths depending on the type of traffic to be transmitted, and PLC does not count with an equitable allocation of resources that favors the needs of the nodes. In this order of ideas, the following question comes up: What should be done to distribute equitably the PLC channel capacity among all the nodes that are part of the LAN, in order to optimize the allocation of resources according to the requirements established by each node, providing appropriate QoS levels without affecting the performance of other services and even exceeding the maximum capacity allowed by the channel? To cope with this issue, the use of cooperative game theory as a strategy to solve the established problem appears as a solution.

The game theory is an area of mathematics proposed by John Von Neumann in 1928 and is aimed to evaluate the choices an individual can make within a competitive context of gain or loss, against decisions made by the other competitors. This competitive scenario is called “Game” and the individuals who are part of this scenario are called “Players” (3). Using the game theory provides three ways for modeling a real scenario: *Extensive, strategic and coalition*. The first two are only applicable on *non-cooperative* games where the interest of each player highlights the interest to obtain personal gain, regardless of the outcome of the other players. The third form (coalition) can be applied only on games of cooperative type and corresponds to a game in which two or more

players do not compete with each other but instead, work together to achieve the same objective and therefore they win or lose as a group, increasing the likelihood of obtaining a higher gain versus the one obtained by acting individually⁴. In a cooperative game it is not necessary to analyze the strategies of the players as in non-cooperative game; it is only necessary to know the value each coalition can get and the array of payments associated with the game results^{5,6}.

Based on the above, this paper proposes the use of the *Nucleolus* as a strategy for optimizing resources on a LAN over Power Line Communications, considering that the use of game theory has become a tool of great importance when analyzing situations where decisions are required, with several possible answers, by modeling optimal strategies in order to maximize their utility³. The optimization process is performed on the node that serves as a Central Coordinator (CCo) (Figure 1), which is responsible for setting the media access scheme (Schedule) for each node that is part of the LAN supported in the standard HPAV.

2. Concept Headings

Definition⁵: A cooperative game TU (with Transferable Utility) is a pair (N, v) where $N = \{1, 2, 3 \dots n\}$ is the set of players and $v: 2^N \rightarrow \mathbf{R}$ is called “characteristic function” of game, with $v(\emptyset) = 0$. Any non-empty subset of N is called “Coalition”. For each coalition $S \subset N$ it is associated a number $v(S)$ which represents

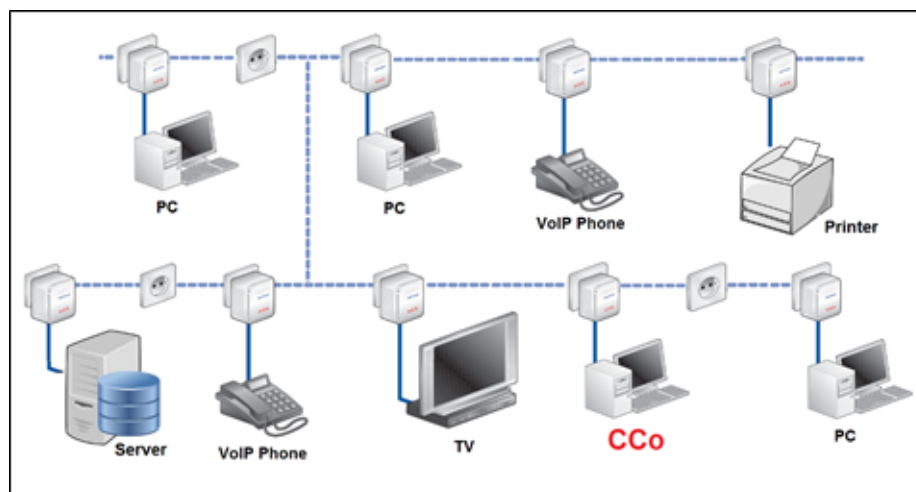


Figure 1. HPAV network.

the payment that can assure the players that are part of S , regardless of what the other players do. The value of a coalition can be considered as the minimum amount that can get a coalition if all the players who are part of it are associated and play in team.

Inside the cooperative game theory, on several occasions it is necessary to divide, in an equitable manner, the net value of an asset or a resource among a set of players, considering that in many cases the amount to divide is insufficient to satisfy the demands of each player. At this point is where the problem known as “Bankruptcy” arises.

Definition⁶: A bankruptcy game is defined as a triplet (N, d, E) where $N = \{1, 2, \dots, n\}$ is the set of creditors, $d = \{d_1, d_2, \dots, d_n\}$ with $d_i \geq 0, \forall i \in N$ is the array of demands of the creditors and E corresponds to the net value that must be distributed among the elements of N .

For each bankruptcy (N, d, E) problem a cooperative game (N, v) can be defined. The set of players N is the same set of creditors or claimants in the bankruptcy problem. The value of the S coalition in the game is defined as the property to be distributed among players and it was not claimed by the claimants that do not

belong to the coalition S . Let $d(S) = \sum_{i \in S} d_i$ the sum of the demands of all creditors who are part of the coalition

$d(N \setminus S) = \sum_{i \in N \setminus S} d_i$ the sum of the demands of all creditors who are not part of the coalition S .

Definition⁵: A cooperative game (N, v) is a bankruptcy game if there is a problem where (1):

$$v(S) = \max\{0, E - d(N \setminus S)\} \quad \forall S \subset N \quad (1)$$

The value of each coalition $v(S)$ responds to a pessimistic assessment of what it can achieve, where the balance is assigned to the coalition S after doing the repartition among the claimants which are not in the coalition.

One of the main problems of cooperative game theory of Transferable Utility (UT) is how to allocate the total gain $v(N)$ among all players in an equitable manner and according to the individual participation of each player. In response, the *Nucleolus* is proposed as a strategy to solve the proposed problem.

The *core* is a concept of solution that has a major difficulty because it can sometimes be a very large set and others an empty set. In view of the above, the concept of

“*Nucleolus*” arises and proposes a solution whenever the set of imputations is not empty. The *Nucleolus* will be able to overcome the weaknesses present in the core, delivering results in an only and non-empty set. Additionally, the *Nucleolus* is part of the core when this is not empty⁸.

Definition⁹: Let us see a game (N, v) with a efficient distribution of payments. $[x = (x_1, x_2, \dots, x_n)] \in \mathbb{R}^n$ among the players, i.e. (2):

$$\sum_{i=1}^n x_i = v(N) \quad (2)$$

Then, the excess of a S coalition with respect to the distribution of payments x is the difference between the value of the S coalition and what the coalition receives by the distribution x , which means (3):

$$e(S, x) = v(S) - x(S) = v(S) - \sum_{i \in S} x_i \quad (3)$$

The expression $e(S, x)$ is because of the measure of the degree of dissatisfaction of the S coalition with the distribution x . The higher $e(S, x)$ is, the greater the degree of dissatisfaction becomes. For each array of distribution of payments x , an array $\theta(x)$ is built so that the excesses are sorted from highest to lowest, in relation to the order of coalitions.

Definition⁹: For each component $x \in I(N, v)$, the array of excess is defined as the array $\theta(x)$, with 2^n components (4):

$$\theta(x) = (e(S, x))_{S \in N} = (\theta_1(x), \theta_2(x), \dots, \theta_{2^n}(x)) \quad (4)$$

Where, $\theta_k(x) \geq \theta_{k+1}(x) \quad \forall k = 1, 2, \dots, 2^n - 1$

Given two arrays of excesses x and y , when compared in lexicographical order, element by element in order to identify which of them presents a lower difference or lesser degree of dissatisfaction. The comparison process begins by evaluating the condition of inequality among the first elements of each array of excess $(\theta_1(x) < \theta_1(y))$. If the condition is met, it can be said that $\theta(x) <_L \theta(y)$

, otherwise the same process is performed again but with the second element of each array and so on to find any difference between their elements. This can be expressed as (5):

$$\max_S \{e(S, x)\} < \max_S \{e(S, y)\} \quad (5)$$

Definition: The *Nucleolus* of a game (N, v) , is defined as the set $N(N, v)$ as follows (6):

$$N(N, v) = \{x \in I(N, v); \theta(x) \leq_L \theta(y), \forall y \in I(N, v)\} \quad (6)$$

Therefore it is said that the *Nucleolus* contains those distributions of payments that are imputations for which the largest degree of dissatisfaction is minimized. For the

Nucleolus can exist and be unique, it is necessary a condition as follows (7):

$$\sum_{i=1}^n v(\{i\}) \leq v(N) \quad (7)$$

To calculate the *Nucleolus* $[x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n]$ for a game (N, v) , it is necessary to solve the following linear programming problem (8):

$$\begin{aligned} v(S) - \sum_{i \in S} x_i &\leq \gamma, S \in N, S \neq \emptyset, S \neq N \\ x &\in I(N, v) \end{aligned} \quad (8)$$

Where γ is the minimum possible value for the problem, which is reached at a x point. Then it can say what x is *Nucleolus*.

Table 1. Value for each coalition $v(S)$ and $e(S, x)$

	Coalition (S)	$v(S)$ [*1E+6]	$(S, x) = v(S) - \sum_{i \in S} x_i$
1	{1}	0.00	$0 - x_1$
2	{2}	0.00	$0 - x_2$
3	{3}	0.00	$0 - x_3$
4	{4}	0.00	$0 - x_4$
5	{1,2}	0.00	$0 - x_1 - x_2$
6	{1,3}	0.00	$0 - x_1 - x_3$
7	{1,4}	0.00	$0 - x_1 - x_4$
8	{2,3}	0.00	$0 - x_2 - x_3$
9	{2,4}	3.59	$3,59 - x_2 - x_4$
10	{3,4}	13.59	$13,59 - x_3 - x_4$
11	{1,2,3}	23.59	$23,59 - x_1 - x_2 - x_3$
12	{1,2,4}	33.59	$33,59 - x_1 - x_2 - x_4$
13	{1,3,4}	43.59	$43,59 - x_1 - x_3 - x_4$
14	{2,3,4}	53.59	$53,59 - x_2 - x_3 - x_4$

For a better understanding of the optimization process through the *Nucleolus* method, the following scenario arises: A PLC network has 83.59 Mbps of total bandwidth to be distributed among four nodes ($i = 1, 2, 3, 4$) in which the demand for bandwidth (d_i) for each i node is 30, 40, 50 and 60 Mbps respectively. To calculate the *Nucleolus* $\llbracket x = (x_1, x_2, x_3, x_4) \rrbracket$, it is proceeded as follows:

Step 1: Setting the values of $v(S)$. Given that the total demand exceeds the total value available, it is necessary to consider the proposed scenario by the PLC network as a bankruptcy game. The value for $v(S)$ can be calculated by (9):

$$v(S) = \max \left\{ 0, E - \sum_{i \in N-S} d_i \right\} \forall S \subset N \quad (9)$$

The values of $v(S)$ are calculated using the following routine, which was developed in Matlab¹⁰. In Table 1 are recorded the estimated values for each coalition:

% Routine to set the number of possible coalitions and the value of each coalition, according to the array of demands (V) for a number of players Nj.

n_coal = 0;

for I = 1:Nj

n_coal = n_coal + nchoosek(Nj, i);

end

M_Coaliciones = zeros(n_coal, Nj); % The array of coalitions initializes

c = 0;

for i = 1:Nj

S = nchoosek(Z, i);

nZ = length(S(:, 1));

for j = 1:nZ

c = c + 1;

Suma_d = 0;

for k = 1:i

Suma_d = Suma_d + V(S(j, k));

M_Coaliciones(c, k) = S(j, k);

end

Suma_dT = BW_Total - (Total_V - Suma_d);

VAux = [0 Suma_dT];

V_Coalicion(c) = max(VAux);

end

end

M_Coaliciones; % Array of coalitions.

V_Coalicion; % Value of transferable utility by coalition.

Step 2: The *Nucleolus* $\llbracket x = (x_1, x_2, x_3, x_4) \rrbracket$ must comply with the following restrictions:

$$\sum_{i=1}^n x_i = v(N) = 83.59$$

$i = 1, 2, 3, 4$

Step 3: Finally, a *minimax* problem can be proposed for calculating the *Nucleolus*:

$$\min_{x_1, x_2, x_3, x_4} \left\{ \max \{ e_k(S, x) \} \right\}$$

Subject to:

$$x_i \geq v(\{i\})$$

$$\sum_{i=1}^n x_i = v(N) = 83.59$$

Step 4: To solve the *minimax* problem it proceeds as follows:

$$\text{Let } \max\{e_k(S, x)\} = \gamma_1$$

Each of the functions that affect the previous maximization must be less than or equal to $\gamma_1 \cdot \gamma_1$. Having seen this, the *minimax* problem can be written as a linear programming problem:

$$\min_{x_1, x_2, x_3, x_4} \gamma_1$$

$$-x_1 \leq \gamma_1$$

$$-x_2 \leq \gamma_1$$

$$-x_3 \leq \gamma_1$$

$$-x_4 \leq \gamma_1$$

$$-x_1 - x_2 \leq \gamma_1$$

$$-x_1 - x_3 \leq \gamma_1$$

$$-x_1 - x_4 \leq \gamma_1$$

$$-x_2 - x_3 \leq \gamma_1$$

$$3,59 - x_2 - x_4 \leq \gamma_1$$

$$13,59 - x_3 - x_4 \leq \gamma_1$$

$$23,59 - x_1 - x_2 - x_3 \leq \gamma_1$$

$$33,59 - x_1 - x_2 - x_4 \leq \gamma_1$$

$$43,59 - x_1 - x_3 - x_4 \leq \gamma_1$$

$$53,59 - x_2 - x_3 - x_4 \leq \gamma_1$$

$$x_1 + x_2 + x_3 + x_4 = 83.59$$

$$x_1, x_2, x_3, x_4 \geq 0$$

Step 5: To solve the optimization problem, it was used the *toolbox* that is included in Matlab R2013A version. In¹⁰ there is more information on using the optimization toolbox. The values for each of the parameters (target function, the restrictions and the starting point of iteration in Array form) for the problem proposed are the following:

F : Array of coefficients of the objective function:

$$F = [0 \ 0 \ 0 \ 0 \ 1];$$

A, b : correspond to inequality constraints, where A is the array of coefficients and b is the array of results for each of the inequalities ($Ax \leq b$) .

A=	-1	0	0	0	0	b=	0
	0	-1	0	0	0		0
	0	0	-1	0	0		0
	0	0	0	-1	0		0
	-1	0	0	0	-1		0
	0	-1	0	0	-1		0
	0	0	-1	0	-1		0
	0	0	0	-1	-1		0
	-1	-1	0	0	-1		0
	-1	0	-1	0	-1		0
	-1	0	0	-1	-1		0
	0	-1	-1	0	-1		0
	0	-1	0	-1	-1		-3.59
	0	0	-1	-1	-1		-13.59
	-1	-1	-1	0	-1		-23.59
	-1	-1	0	-1	-1		-33.59
	-1	0	-1	-1	-1		-43.59
	0	-1	-1	-1	-1		-53.59

A_{eq}, b_{eq} : correspond to inequality constraints, where A_{eq} is array of coefficients and b_{eq} is the array of the results for each of the equations ($A_{eq} x = b_{eq}$) .

$$A_{eq} = [1 \ 1 \ 1 \ 1 \ 0];$$

$$b_{eq} = [83.59];$$

$x0$: Starting point for iteration.

$$x0 = [0 \ 0 \ 0 \ 0 \ 0];$$

Finally, the following expression is used in order to compute the solution to the problem posed:

$$[x, fval] = \text{linprog}(F, A, b, Aeq, beq, x0)$$

Where x and $fval$ correspond to the solution array (*Nucleolus*) and the minimum value the objective function can achieve. The result obtained is the following:

$$\begin{aligned}x_1 &= 17.46 \\x_2 &= 19.58 \\x_3 &= 22.31 \\x_4 &= 24.23 \\y_1 &= 6.52E - 12 \approx 0\end{aligned}$$

Below is the routine developed in Matlab¹² that consolidates every single step mentioned before, in order to facilitate the optimization processes through the use of *Nucleolus* and it can be used in future research:

% Routine for resource optimization processes supported in the *Nucleolus*.

Total_V = sum(V); % Calculates the total demand of the traffic according to the array of demands (V), for a number of players Nj.

Z = 1:1:Nj; % Creates an array with the consecutive numbers from 1 to Nj.

% Routine to set the number of possible coalitions.

n_coal = 0;

for i = 1:Nj

n_coal = n_coal+nchoosek(Nj,i);

end

M_Coaliciones = zeros (n_coal,Nj); % The array of coalitions initializes

c=0;

for i = 1:Nj

S = nchoosek(Z,i);

nZ = length(S(:,1));

for j = 1:nZ

c = c+1;

Suma_d = 0;

for k = 1:i

Suma_d = Suma_d+V(S(j,k));

M_Coaliciones(c,k) = S(j,k);

end

Suma_dT = BW_Total-(Total_V-Suma_d);

VAux = [0 Suma_dT];

V_Coalicion(c) = max(VAux);

end

end

M_Coaliciones; % Array of coalitions.

V_Coalicion; % Value of transferable Utility by coalition.

% Adaptation of parameters to optimize through toolbox.

% Procedure to set Array A of optimization toolbox.

MMZ = zeros(n_coal,Nj);

for i = 1:n_coal

for j = 1:Nj

if M_Coaliciones(i,j)>0

MMZ(i,M_Coaliciones(i,j))=1;

end

end

end

```
MM2 = ones(n_coal,1); % creates a column of
ones.
```

```
MM3 = [MMZ MM2]; % and it is concatenated
as the last column to MM1.
```

```
MM3(n_coal,:) = []; % deletes the last row of the array
MM3.
```

```
MM4 = eye(Nj); % creates an identity array accord-
ing to the number of players Nj.
```

```
MM5 = zeros(Nj,1); % Creates a column of zeros that
is concatenated with the.
```

```
MM6 = [MM4 MM5]; % identity array.
```

```
A = [MM6;MM3]; % concatenates the resulting two
Arrays from Toolbox Array A.
```

```
A = -1*A;
```

```
% Procedure to set Array B of optimization toolbox.
```

```
Mb = V_Coalicion';
```

```
Mb(n_coal) = [];
```

```
MM5 = zeros(Nj,1); % Creates a column of zeros and
then it is concatenated with
```

```
% the identity array.
```

```
b = [MM5;Mb]; % concatenates the resulting two
Arrays from Toolbox Array A.
```

```
b = -1*b;
```

```
% Procedure Aeq, beq, x0, F
```

```
Aeq = ones(1,Nj);
```

```
Aeq = [Aeq 0];
```

```
Beq = V_Coalicion(n_coal);
```

```
x0 = zeros(1,Nj+1);
```

```
F = zeros(1,Nj);
```

```
F = [F 1];
```

```
[x,fval] = linprog(F,A,b,Aeq,beq,x0) % x contains the
BW assigned to each player.
```

3. Results and Discussion

3.1 Proposed scenario

In order to evaluate the performance of the *Nucleolus* to execute processes of resource optimization in a LAN network over *Power Line Communications*, it has been proposed two scenarios consisting of 12 nodes each, in which only data traffic will flow. In each scenario the node 12 acts as Central Coordinator (CCo) and channel requirements for each of the nodes (BW on demand) are set out in Table 2.

Table 2 shows that total bandwidth required is greater than the total bandwidth available, establishing a saturation state of PLC channel in both scenarios. The values that correspond to the Total BW available for the PLC channel were estimated by using the tool “Generador de Canal PLC (GC_PLC)”, developed by PhD Francisco Javier Cañete, belonging to PLC group of the University of Malaga - Spain, which allows to evaluate the behavior of a PLC channel in a frequency range below 30 MHz, according to the parameters associated with the topology of a PLC network, in a typical residential environment^{11,13}.

3.2 Treatment Comparison (Optimal BW-LP vs. BW-*Nucleolus*)

To evaluate the optimization process performed when using the *Nucleolus*, it is necessary to establish an alternative method of optimization that makes calculating the value of BW for each node i possible, and developing a process for comparing treatments. Before this situation it was decided to raise the problem of allocation of resources as a Linear Programming problem (LP). Based on the above, the problem can be stated as follows (10) (11):

Table 2. BW requirements for each proposed scenario

Node	Scenario 1: BW on demand [Mbps]	Scenario 2: BW on demand [Mbps]
1	8.23	4.85
2	14.99	9.37
3	13.41	11.23
4	9.45	8.78
5	4.11	2.09
6	1.8	12.86
7	6.01	1.7
8	12.38	5.27
9	5.36	6.38
10	15.28	14.25
11	10.14	10.14
12 (Cco)	90.2	72.1
Total BW required	191.36	159.02
Total BW available	159.72	120.65

$$\text{Max} \sum_{i=1}^n x_i \quad (10)$$

$$\sum_{i=1}^n x_i \leq BW_T \quad (11)$$

Where n , d_i y x_i correspond to the number of nodes (for the particular case $n = 12$), BW required and BW assigned for i node respectively.

The optimization toolbox included in Matlab is used

one more time to solve the optimization problem which allows the use of various methods of optimization. To use the tool, it was necessary to organize the objective function, constraints and the starting point of iteration in a matrix form. The values for each of the parameters are the following:

F: Array of coefficients of the objective function:

$F = [-1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1 \ -1]$;

A, b : Correspond to the inequality constraints, where A is the Array of coefficients and b the array of results for each of the inequations ($Ax \leq b$).

A=	1	0	0	0	0	0	0	0	0	0	0	0	b=	d_1
	0	1	0	0	0	0	0	0	0	0	0	0		d_2
	0	0	1	0	0	0	0	0	0	0	0	0		d_3
	0	0	0	1	0	0	0	0	0	0	0	0		d_4
	0	0	0	0	1	0	0	0	0	0	0	0		d_5
	0	0	0	0	0	1	0	0	0	0	0	0		d_6
	0	0	0	0	0	0	1	0	0	0	0	0		d_7
	0	0	0	0	0	0	0	1	0	0	0	0		d_8
	0	0	0	0	0	0	0	0	1	0	0	0		d_9
	0	0	0	0	0	0	0	0	0	1	0	0		d_{10}
	0	0	0	0	0	0	0	0	0	0	1	0		d_{11}
	0	0	0	0	0	0	0	0	0	0	0	1		d_{12}
	-1	0	0	0	0	0	0	0	0	0	0	0		0
	0	-1	0	0	0	0	0	0	0	0	0	0		0
	0	0	-1	0	0	0	0	0	0	0	0	0		0
	0	0	0	-1	0	0	0	0	0	0	0	0		0
	0	0	0	0	-1	0	0	0	0	0	0	0		0
	0	0	0	0	0	-1	0	0	0	0	0	0		0
	0	0	0	0	0	0	-1	0	0	0	0	0		0
	0	0	0	0	0	0	0	-1	0	0	0	0		0
	0	0	0	0	0	0	0	0	-1	0	0	0		0
	0	0	0	0	0	0	0	0	0	-1	0	0		0
	0	0	0	0	0	0	0	0	0	0	-1	0		0
	0	0	0	0	0	0	0	0	0	0	0	-1		0
	1	1	1	1	1	1	1	1	1	1	1	1		$\sum d_i$

Table 4. BW assigned to each player for scenario 2: Saturated Channel – Single Class

Nodo	BW request [Mbps] ()	BW assigned Nucleolus [Mbps] (N_i)	BW assigned LP [Mbps] (LP_i)	= -	= -
1	4.85	1.283	4.0258	3.567	0.8242
2	9.37	5.34	7.518	4.03	1.852
3	11.23	6.879	8.8111	4.351	2.4189
4	8.78	4.875	7.0949	3.905	1.6851
5	2.09	0.783	1.541	1.307	0.549
6	12.86	8.268	9.9065	4.592	2.9535
7	1.7	0.258	1.1995	1.442	0.5005
8	5.27	4.814	4.3806	0.456	0.8894
9	6.38	5.499	5.2833	0.881	1.0967
10	14.25	9.461	10.8211	4.789	3.4289
11	10.14	5.967	8.06	4.173	2.08
12 (Cco)	72.1	67.222	52.0083	4.878	20.0917
Total	159.02	120.649	120.6501		

necessary to conduct a hypothesis contrast on average differences with paired samples by using the test called *paired-t*. For this, the following steps are established:

Step 1: A new random variable $Z = X - Y$ is defined and proceeds to calculate the average value and the standard deviation for the variable Z . The result of this process generated values of $4.16E - 5$ and 3.59 for $\bar{Z}$ y S_Z respectively. Furthermore, to define a new variable Z , it is necessary to make an adjustment in the hypotheses posed as follows:

$$H_0: \mu_x \leq \mu_y \rightarrow \mu_x - \mu_y \leq 0 \rightarrow \mu_z \leq 0$$

Step 2: It is proceeded to calculate the statistical value established for the test by using the following expression:

$$d = \frac{Z}{S_Z} \sqrt{n} = \frac{4.16E - 5}{3.59} \sqrt{24} = 5.6864E - 5$$

Where d is the statistical value and n reflects the number of samples for the two proposed scenarios.

Step 3: Set the acceptance range of H_0 for $\{t: t < T_{(\alpha, n-1)}\}$ to 5% of significance ($\alpha = 0.05$) and $n - 1$ degrees of freedom. For the particular case,

the value of $T(0.05; 23) = 1.7139$, defining the range of acceptance of H_0 between $(-\infty, 1.7139)$.

When assessing the statistical value d , it is observed that it is within the acceptance range. That is why H_0 is not rejected. Taking the above into consideration, it can be concluded that the *Nucleolus* may be considered as an appropriate method to perform processes of resource allocation in LANS on PLC, taking into account that the proposed scenarios for the *Nucleolus* conducted a better allocation of BW in comparison with the optimization method LP, to minimize the difference between the BW value requested and the BW value assigned to each node, with a 95% of trust.

4. Conclusions

Facing with the need to make an equitable distribution of resources, according to the demand of the service, among the nodes that are part of a PLC network, it was suggested the use of cooperative game theory of transferable utility as optimization strategy for the allocation of resources based on the needs of each node. The proposal came up considering that the cooperative game theory has become a tool of great importance when analyzing situations in which decision-making is required, with a multiplicity of possible responses, through the modeling of optimal strategies that allow to maximize its utility. Additionally, considering that nodes can work cooperatively increases the probability of obtaining a higher gain against the one obtained by acting individually, where it is just enough to know the utility that each coalition can get and the payments array associated. Considering the above and based on the obtained results, it could be found that the *Nucleolus* can be assumed as an appropriate strategy for resource allocation in LANs on PLC, having in account that for the proposed scenarios the *Nucleolus* made a better allocation of BW compared to the optimization method LP, to minimize the difference between the BW value requested and the BW value assigned to each node, with a 95% trust. The method is efficient. The computational complexity of the algorithm tends to increasing the number of nodes, though. So it is recommended to establish alternative methods for calculating the *Nucleolus*, with a reduced computational and temporary complexity,

for future research, which will facilitate the implementation in embedded systems for low cost.

5. References

1. Anatory J, Theethayi N. Broadband power-line communication systems: Theory and Applications. 1a ed. Southampton, England: WIT Press; 2010. p. 193.
2. Latchman H, Srinivas K, Yonge L, Gavette S. Homeplug AV and IEEE 1901. A Handbook for PLC Designers and Users. 1a ed. New Jersey, USA: Wiley-IEEE Press; 2013. p. 384.
3. Perez J, Jimeno J, Tena E. Teoria de juegos. 1 ed. Madrid, Espana: Pearson-Prentice Hall; 2003.
4. Ramirez RD. Cooperacion en la cadena de suministro de la energia electrica en Colombia. Universidad del Norte; 2008.
5. Peleg B, Sudholter P. Introduction to the theory of cooperative games. Springer; 2007. p. 348.
6. Selvakumari K, Lavanya S. An approach for solving fuzzy game problem. Indian J Sci Technol [Internet] 2015; 8(15). [cited 2016 Jul 2]. Available from: <http://www.indjst.org/index.php/indjst/article/view/56807>
7. Rodriguez M. Contribuciones a la teoría del valor en juegos en forma estrategica y en problemas de bancarrota. Científico S de P e I, editor. Universidad Santiago de Compostela; 2005.
8. Garcia J. Complejidad y algoritmos en juegos cooperativos. [Thesis PhD]. Universidad de Sevilla; 2000.
9. Schmeidler D. The Nucleolus of a characteristic function game. SIAM J Appl Math. Society for Industrial and Applied Mathematics. 1969 Nov; 17(6):1163–70.
10. Puente del Campo MA (Maria A. Aportaciones a la representabilidad de juegos simples y al calculo de soluciones de esta clase de juegos. Universitat Politecnica de Catalunya; 2004.
11. Babu G, Anuhya A, Venkatram N. Digital signal processing scheme for open loop and closed loop IFOG using Matlab/Simulink. Indian J Sci Technol. [Internet] 2016;9(11). [cited 2016 Jul 2]. Available from: <http://www.indjst.org/index.php/indjst/article/view/86935>
12. Berens P. CircStat: A Matlab toolbox for circular statistics. J Stat Softw. 2009; 31(10):1–21.
13. Sudhakar T, Rajan M, Srinivasan K. Modeling and simulation of distribution network with the integra-

- tion of distribution generator using Matlab. Indian J Sci Technol. [Internet] 2016; 9(12). [cited 2016 Jul 2]. Available from: <http://www.indjst.org/index.php/indjst/article/view/89947>
14. Sancha S, Canete FJ, Diez L, Entrambasaguas JT. A channel simulator for indoor power-line communications. 2007 IEEE International Symposium on Power Line Communications and its Applications [Internet] IEEE; 2007 p. 104–9. [cited 2014 Mar 26]. Available from: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4231680>
15. Khemiri R, Sayadi F, Atri M. MatLab-GPU-based 2D-DWT acceleration for JPEG2000 with single and double-precision. Indian J Sci Technol. [Internet] 2016; 9(12) [cited 2016 Jul]. Available from: <http://www.indjst.org/index.php/indjst/article/view/80526>.