

# Allocation of Medium Access Order Over Power Line Communications (PLC) Supported on Weighted Voting Games

Juan Carlos Vesga-Ferreira<sup>1</sup>, Gerardo Granados-Acuna<sup>1</sup> and Jose Antonio Vesga-Barrera<sup>2</sup>

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

<sup>2</sup>Facultad de Ingenieria, Corporacion Universitaria de Ciencia y Desarrollo - UNICIENCIA.Cra. 12 #37-14. Bucaramanga, Colombia; jose.vesga@uniciencia.edu.co

## Abstract

**Background/Objectives:** In PLC networks, the Home Plug for Audio-Video (HPAV) protocol lacks a mechanism that allows optimizing the order of access of the nodes to the channel. The objective of this paper is to propose the use of Power Indexes as a strategy to solve the problem stated before. **Methods/Statistical Analysis:** This paper proposes two independent scenarios, formed by communications equipment (nodes) in clearly established traffic and channel conditions. The solution to that problem is obtained by using three methods applying Power indexes: Banzhaf, Shapley-Shubik and Deegan-Packel. A comparison of the obtained result is carried out and a *Dunnett's test* is applied in order to determine which method has a better performance to solve the problem stated before. **Findings:** The usage of the Game Theory, specially the Power Indexes, is an innovative strategy for the optimization of the order of access to the PLC channel. It is important to state the relevance that the Game theory has acquired as a tool used to analyze complex optimization situations within the communication networks. Furthermore, it is fair to say that no other similar paper has been found in the diverse bibliographic sources consulted. Based on the results obtained by means of the Dunnett test, it was possible to provide evidence that any of the three methods can be considered as a strategy for establishing the order for the nodes to access the channel. Such order will eventually improve the performance of the PLC network (with a 95% confidence). Nonetheless, the optimization method proposed by Shapley-Shubik can be considered as the best method out of the three because it presents the greatest average, and plus, it is the only method in which the difference  $D_i$  is greater than 0 ( $D_i \geq 0$ ). **Application/Improvements:** It is recommended to investigate on alternative methods to calculate Power Indexes that can offer a reduced computational complexity, in order to facilitate its implementation in low-cost embedded systems.

**Keywords:** Banzhaf, Deegan-Packel, Game theory, Home Area Networks, Medium Access, Power Line Communications, Shapley-Shubik

## 1. Introduction

Quality of Service (QoS) is defined as the ability of a network to manage the traffic demand depending on the type of service, to meet the expectations of the user and in accordance with clearly established metrics or levels<sup>1</sup>. The service offered by the majority of the networks (best effort) does not always satisfy QoS requirements for each one of the applications, especially in real-time transmissions such as audio and video streaming, and PLC networks are no exception.

HPAV has established a hybrid model supported in Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and carrier sense multiple access with collision avoidance (TDMA), where CSMA/CA is meant for the transmission of data packages, while TDMA is used for the transmission of video and voice packages, in order to offer accurate levels of QoS<sup>2</sup>. HPAV does not have a method to optimize the order of transmission of the nodes and the access to the PLC channel, which depends on the service demand, the type of packet to send the number of nodes and the conditions of the

\* Author for correspondence

channel. All this is in order to improve the performance of the network for the provision of IP-based services.

The use of the Weighted Voting Games theory is based on the power indexes. This proposal is made in order to supply a solution to the given problem and thus formulate a mechanism to generate the order in which the nodes must access the channel (according to the service demand, the type of package and the conditions of the channel) and to minimize delays and collisions in the network. In addition, processes related to the provision of voice, data and video services are currently a subject of research and interest in the context of PLC, due to increased demand for network services, as well as the need for having adequate QoS levels<sup>4</sup>.

### 1.1 Weighted Voting Games

Definition: A Weighted Voting Game can be defined as a tripleta  $(N, w, q)$  where  $N = \{1, 2, \dots, n\}$  is the set of players,  $w = (w_1, w_2, \dots, w_n)$  with  $w_i \geq 0$  for  $i = 1, 2, \dots, n$ ;  $v$  is a weight distribution and  $q \in \mathbb{R}^+$  is the quota or majority required to win a voting process. Generally  $q > \frac{T}{2}$ , where  $T$  corresponds to the total weight addition  $T = w_1 + w_2 + \dots + w_n$ . A *Weighted Votinggame* is represented as<sup>5</sup>(1):

$$[q; w_1, w_2, \dots, w_n] \quad (1)$$

If in a voting game none of the voters has enough votes to match the established quota, it is necessary to carry out a process of cooperation between several voters, in order to join their votes and establish alliances (so-called "coalitions"), which allow them to exceed the established quota and thus win the voting process.

### 1.2 Power Indexes

In order to create democratic systems, it is necessary to know all possible alliances and the level of power that each group can exercise. Even though the concept of power is considered as an illusion, it is a very useful criterion for decision-making in many systems and the game theory makes a very deep study on the particular subject<sup>6-8</sup>.

Each player can have different power levels over the voting process, which depends on the value or number of votes associated. However, there is no linear dependence with the number of votes due to the fact that there is the possibility that each player can generate coalitions with other players. Hence, it is necessary to establish a

power index that takes into account the various types of strategies that each player can carry out. Among the many types of power indexes, it is possible to mention<sup>5</sup>:

*Banzhaf* power index.

*Shapley-Shubik* power index.

*Deegan-Packel* power index.

The Power Index is a number that is allocated to each player in order to indicate the level of power and it can be interpreted as a percentage of power that each group owns within the system<sup>6</sup>. The power indexes are meant to quantify the chances of cooperation that each group has within the winning coalitions. The greater the weight, the greater the power. However, there are situations in which, although the groups have different weights, they can present equal power indexes<sup>6</sup>.

## 2. Concept Headings

### 2.1 Algorithm for Weighted Voting Games

Calculating the power indexes in Weighted Voting Games is a problem that has several levels of computational complexity when the game contains more than 20 players, due to the use of permutations in classic algorithms<sup>9</sup>. There is a way to do this process without checking all the possible coalitions, by means of the use of generatrix functions. Such functions are based on the combinatory analysis and they offer an indirect method to calculate the number of configurations of diverse types. In<sup>10</sup> diverse aspects related to the usage and properties of generatrix functions are described.

Hereunder, the generatrix functions are used as a strategy in order to estimate the power indexes on a *Weighted Voting game*  $(N, v)$  defined by  $v = [q; w_1, w_2, \dots, w_n]$ .

### 2.2 Estimation of Standardized Banzhaf's Power Index Standardized by Generatrix Functions

The standardized Banzhaf's power index is given by the vector (2):

$$\beta(v) = (\beta_1(v), \beta_2(v), \dots, \beta_n(v)), \text{ con } \beta_i(v) = \frac{\eta_i(v)}{\sum_{k=1}^n \eta_k(v)} \quad (2)$$

Where  $\eta_i(v)$  is the number of pivots of player  $i$  in

the game  $(N, v)$  and  $\sum_{k=1}^n \eta_k w$  is the total number of pivots of all players. It is important to remember that a pivot for the player  $i$  is formed by a couple of coalitions  $(S \cup i, S)$  where  $S \cup i$  is the winner and  $S$  is the loser. Considering the above, the number of pivots of the player  $i$  is (3):

$$\eta_i(w) = |\{S \in \mathcal{W} : S \cup i \in \mathcal{W}\}| \quad (3)$$

Where  $\mathcal{W}$  indicates the set of winning coalitions.

*Definition:* Let  $(N, v)$  a *Weighted Voting Game* defined by  $v = [q; w_1, w_2, \dots, w_n]$ . Then, the number of pivots of player  $i$  is given by (4):

$$\eta_i(w) = \sum_{k=q-w_i}^{q-1} b_k^i \quad (4)$$

Where  $b_k^i$  corresponds to the number of coalitions  $S \subseteq N$  such that  $i \notin S$  and  $w(S) = k$ . The values of  $b_k^i$  can be obtained by means of the use of the expression defined by Brams-Affuso (1976) (6):

$$B_i(x) = \prod_{j=1, j \neq i}^n (1 + x^{w_j}) = \sum_{k=0}^{w(N)} b_k^i x^k \quad (5)$$

Where  $b_k^i$  is the number of coalitions with weight  $k$ . That expression is the one from which the value of  $\{b_k^i\}_{k \geq 0}$  is obtained.

The methodology to estimate the Banzhaf's power index in a *Weighted Voting Game*  $(N, v)$ , where  $v = [q; w_1, w_2, w_3, \dots, w_n]$  is the following:

Step 1: The Value of  $q$  is being calculated as follows (6):

$$q = \left\lceil \frac{\sum_{i=1}^n w_i}{2} \right\rceil \quad (6)$$

Step 2: The estimation of each of the functions  $B_i(x)$  is being carried out. Such functions can be estimated by means of the use of the following expressions (7):

$$B_i(x) = \prod_{j=1, j \neq i}^n (1 + x^{w_j}) \quad (7)$$

Step 3: In order to get the number of *swings*<sup>1</sup> of the player  $i$ , it is necessary to consider that (8):

$$\eta_i(w) = \sum_{k=q-w_i}^{q-1} b_k^i \quad (8)$$

Step 4: Finally, the total number of swings is  $\bar{\eta}(w) = \sum_{i \in N} \eta_i(w)$ . Value with which it is possible to estimate

the standardized Banzhaf's power index in the following way (9):

$$\beta(w) = (\beta_1(w), \beta_2(w), \dots, \beta_n(w)) \quad (9)$$

In order to facilitate future process of research related to the subject of power index, a MATLAB routine has been designed. Such routine allows to estimate the standardized Banzhaf power index for all the players for a *Weighted Voting Game*. The routine is the following:

% Routine to estimate the Standardized Banzhaf power index.

% The variable used within the routine are:

% V: Vector of weight for each player.

% q: Average value to consider the winning coalition.

% Nj: Number of players.

% VBanzhaf: Vector corresponding to the power indexes for each player.

V = [13; 12; 6; 2]; % the vector of weights for each player is defined.

q = ceil(sum(V)/2); % the average value is calculated.

Nj = 4;

VBanzhaf = zeros(Nj,1); % Initialization of vector corresponding to the standardized Banzhaf Value.

Bik = zeros(Nj,1); % Initialization of the vector Bi (x).

% Routine to establish the generatrix functions Bi (x).

syms x z

for I=1:Nj

P = zeros(Nj-1,1); % Initial conditions.

S = 1;

for l=1:Nj % Routine to establish the generatrix functions Bi (x).

if l ~ i

s = s\* (1+x^V(l)); % Establishing the base monomial.

end

end

s1 = expand(s); % make the symbolic multiplication.

s2 = expand(s1\*z);

[cz cx] = coeffs(s2,x); % extract the coefficients of a polynomial.

% cz takes the values of Z abd cx the values of x.

e = log10(cx);

f = subs(e,x,10); %estimate the value of

x=10 in the polinomial e.

ex = simplify (f); % simplifies values by extracting the exponents of x.

n = length (ex);

Linf = q-V(i); % q-Wi

Lsup = q-1; % q-1

for k=1:n

% if the coefficients of the player i is located between the superior or inferior limits.

if (ex(k)>=Linf) and (ex(k)<=Lsup)

Bik(i) = Bik(i)+1; %Bik =number of swings of the player i.

end

end

end

Tswing = sum(Bik); % calculate the total of swings.

VBanzhaf = Bik/Tswing % calculate the value of standardized Banzhaf.

### 2.3 Estimating of the Shapley-Shubikpower Index by Means of Generating Functions

The expression to estimate the Shapley-Shubik power index, for each player  $i \in N$ , in a simple game is the following (10):

$$\Phi_i(N) = \sum_{\{S \in \mathcal{W}: S \cup \{i\} \in \mathcal{W}\}} \frac{S!(n-S-1)!}{n!} = \sum_{j=0}^{n-1} \frac{j!(n-j-1)!}{n!} d_j^i \quad (10)$$

Where  $d_j^i$  represents the number of pivots of the player  $i$  in coalitions of size  $j$ ; that is, the number of coalitions of size  $j$  that go to be winners when the player  $i$  is incorporated in the coalition. On a *Weighted Voting game*  $(N, v)$  defined by  $v = [q; w_1, w_2, \dots, w_n]$  it is possible to estimate  $d_j^i$  under the usage of generating functions described before (11):

$$d_j^i = \sum_{k=q-w_i}^{q-1} a_{kj}^i \quad (11)$$

Where  $a_{kj}^i$  is the number of pivots of the player  $i$  in coalitions formed with  $j$  players, different to  $i$ , and with a coalitional weight  $k$ .

**Definition:** Cantor (1962), established that for a game  $(N, v)$  of weighted voting defined by  $v = [q; w_1, w_2, \dots, w_n]$

, to any  $i \in N$ , the generating function to estimate the  $a_{kj}^i$  coalitions  $S \subseteq N$  formed by  $j$  players, with  $i \notin S$  and  $w(S) = k$ , is given by (12):

$$S_i(\alpha, Z) = \prod_{j=1, j \neq i}^n (1 + x^{w_j} Z^j) = \sum_{k=0}^{w(N)} \sum_{j=0}^n a_{kj}^i x^k Z^j \quad (12)$$

Where  $j$  and  $k$  represent the size of a coalition  $S$  and its weight, respectively. To know the values of the coefficients  $a_{kj}^i$  by means of the use of the generating function  $S_i(\alpha, Z)$ , allows to determine the value of the coefficients  $d_j^i$  that are needed to estimate the Shapley-Shubik power index. In this respect, the value  $d_j^i$  can be identified with the coefficients of a polynomial that encloses the following expression (13):

$$g_i(Z) = \sum_{j=0}^{n-1} d_j^i Z^j \quad (13)$$

And, having into account that (14):

$$d_j^i = \sum_{k=q-w_i}^{q-1} a_{kj}^i \quad (14)$$

When replacing, it is possible to obtain (14):

$$g_i(Z) = \sum_{j=0}^{n-1} d_j^i Z^j = g_i(Z) = \sum_{j=0}^{n-1} \left[ \sum_{k=q-w_i}^{q-1} a_{kj}^i \right] Z^j \quad (15)$$

Leveling  $g_i(Z)$  with the expression  $S_i(\alpha, Z)$  it is possible to deduce that in order to determine the coefficients of  $g_i(Z)$  it is necessary to select the coefficient of the monomial  $x^k Z^j$  in which the exponent  $q - w_i \leq k \leq q - 1$ .

The methodology to calculate the Shapley-Shubik power index on a *Weighted Voting Game*  $(N, v)$ , where  $v = [q; w_1, w_2, w_3, \dots, w_n]$  is the following:

Step 1: The value of  $q$  is calculated according to the Equation (6).

Step 2: Each of the functions  $S_i(\alpha, Z)$  is calculated using the following expression (16):

$$S_i(\alpha, Z) = \prod_{j=1, j \neq i}^n (1 + x^{w_j} Z^j) \quad (16)$$

By using the Generating function  $S_i(\alpha, Z)$  it is possible to determine the numbers  $\{d_j^i\}_{j=0}^n$  that are needed to calculate the Shapley-Shubik power index.

Step 3: The polynomial  $g_i(Z)$  is calculated using the expression (17):

$$g_i(z) = \sum_{j=0}^{n-1} d_j^i z^j = \sum_{j=0}^{n-1} \left[ \sum_{k=q-w_i}^{q-1} a_{kj}^i \right] z^j \quad (17)$$

For each polynomial,  $S_i(x, z)$  those monomials where the exponent of  $x$  is located between  $q - w_i$  and  $q - 1$  are extracted and the resultant polynomial is evaluated with  $x=1$ , whose result is registered in  $g_i(z)$ .

Step 4: The value of Shapley-Shubik  $\Phi_i(v)$  for each player  $i$  is calculated according to the values registered in the polynomials  $g_i(z)$ .

To facilitate future research processes related to the issue of power Index, a MATLAB routine was designed. Such routine allows estimating the power index by means of the use of the value of Shapley-Shubik. The routine is as follows:

% Routine to calculate the Shapley-Shubik Power Index.  
 % The variables used within the routine are:  
 % V: Weights vector for each player.  
 % q: Average value to consider the winning coalition.  
 % Nj: Number of players.  
 % VSh: Vector that corresponds to the Power indexes of each player.

```
V = [14; 11; 6; 2]; % it is defined the
Weights vector for each player.
q = ceil (sum(V)/2); % the value of q is calculated.
Nj=4; % Nj contains the number of
players.
```

```
VSh = zeros(Nj,1); % initial conditions of the
Shapley Matrix.
```

```
syms x z y
for I =1:Nj
P = zeros(Nj-1,1);
S =1;
U = 0;
V1 = zeros(Nj,1);
for l =1:Nj
if l~= i
u = u+1;
V1(u) = V(l);
end
end
s = y*(1+(x^V1(1)*z*y))*(1+(x^V1(2)*z*y^2)); %
```

Base polinomial.

```
PotY = 4;
```

```
for l = 3:Nj-1 % Routine to establish the Generating
functions.
```

```
s = s*(1+(x^V1(l)*z*y^PotY));
```

```
PotY = 2^l;
```

```
end
```

```
s1= expand(s); % Makes the symbolic
multiplication of the polynomial Si (x,z).
```

```
[cxz cy]=coeffs(s1,y); % extracts the coefficients of a
polynomialcxzand takes.
```

```
% the values of xz and the values of cy (auxiliary variable).
```

```
cx = cxz/(z^Nj); % Extracts the X factor of each
monomial.
```

```
[num, den] = numden(cx);
```

```
cx = num;
```

```
PotX = subs(log10(cx),x,10); % evaluates the value
of x = 10 in the polinomial e.
```

```
PotX = simplify(PotX); % extracts the
exponents of x.
```

```
a = x.^PotX;
```

```
cz = cxz./(a); % extracts the factor z of each
monomial.
```

```
[num, den] = numden(cz);
```

```
cz = num;
```

```
PotZ = subs(log10(cz),z,10); % evaluates the value
of t=10 in the polinomial e and
```

```
% registers the total weight of each coalition.
```

```
PotZ = simplify(PotZ); % Extracts the exponents of t.
```

```
n=length(PotX);
```

```
Dij=zeros(n,1);
```

```
Linf=q-V(i); %q-Wi
```

```
Lsup=q-1; %q-1
```

```
for k=1:n-1
```

```
if (PotX(k)>=Linf) & (PotX(k)<=Lsup)
```

```
Dij(PotZ(k))=Dij(PotZ(k))+1; % Calculates the
parameter Dij
```

```
end
```

```
end
```

```
Aux = 0;
```

```
% subroutine to calculate the value of Shapley-Shubik for
each player k
```



```

for j = 1:Nj-1
    Aux = Aux+(factorial(j)*Dij(j)*factorial(Nj-j-1));
end
VSh(i) = (1/(factorial(Nj)))*Aux;

end
VSh % Prints the vector corresponding to the Shapley-
Shubik power index allocated to each player.

```

## 2.4 Estimation of the Deegan-Packel Power Index by Means of Generating Functions

Given a simple game  $(N, v)$  both the Banzhaf power index as the Shapley-Shubik power index of a player  $i \in N$ , obey to the weighted sums corresponding to the number of swings for a player  $i$  in particular. Nonetheless, in the case of the Power index proposed by Deegan-Packel for a player  $i$ , only the minimal winning coalitions are taking into account.

In order to calculate the Deegan-Packel power index, it is necessary to use a generating function that allows to calculate the cardinal of all possible coalitions, of which the winning coalitions must be selected and finally, the minimal. For that,  $m_s^i(v)$  is defined as the number of minimal winning coalitions  $S$  (formed by  $s$  players) to which the player  $i \in N$  belongs. Consequently, Deegan-Packel power index of  $i$  in a *Weighted Voting Game* can be calculated by means of the expression (18):

$$I_{dp} = \frac{1}{M(v)} \sum_{s=1}^n \frac{m_s^i(v)}{s} \quad (18)$$

The methodology to calculate the Deegan-Packel power index in a *Weighted Voting Game*  $(N, v)$ , where  $v = [q; w_1, w_2, w_3, \dots, w_n]$  is the following:

Step 1: The value of  $q$  is calculated with the Equation (6).

Step 2: The generating function  $S(x, Z_1, Z_2, \dots, Z_n, t)$  must be calculated by means of the use of the following expression:

$$S(x, Z_1, Z_2, \dots, Z_n, t) = \prod_{j=1}^n (1 + x^{w_j} Z_j t) \quad (19)$$

Step 3: As of the generating function, it is necessary to eliminate those monomials in which the power of the variable  $x$  is less than  $q$ , in order to determine the winning coalitions.

Step 4: The monomials that can be divided by any other monomial established in the group of winning coalitions are to be eliminated. The number of monomials of the resulting function corresponds to the number of winning coalitions of the game  $M(v)$ .

Step 5: In order to obtain the value of Deegan-Packel of a player  $i \in N$ , those terms of the resulting function, in which variable  $Z_i$  appears, are selected, and the coefficient of this monomial  $m_s^i(v)$  is taken. The exponent of the variable  $t$  indicates that the number of players that are part of such minimal coalition ( $s$ ). Having those elements into account, the corresponding calculation of the Deegan-Packel power index for the player  $i$  is carried out, by means of the following expression (20):

$$I_{dp} = \frac{1}{M(v)} \sum_{s=1}^n \frac{m_s^i(v)}{s} \quad (20)$$

Such routine allows to estimate the standardized Banzhaf power index for all the players for a *Weighted Voting Game*. The routine is the following:

In order to facilitate future process of research related to the subject of power index, a MATLAB routine has been designed. Such routine allows to estimate the Deegan-Packel power index. The routine is as follows:

% Routine to calculate the Deegan-Packel power index.

% The variables used within the routine are:

% V: Weights vector for each player.

% q: Average value to consider the winning coalition.

% Nj: Number of players.

% VDeegan: Vector corresponding to the power indexes for each player.

V=[28; 16; 5; 4; 3; 3]; % the Weights vector for each player is defined

q=ceil(sum(V)/2); % the average value is calculated  
Nj=6;

% the Nj symbolic variables of Zj are created

Z=linspace(1,Nj,Nj); % creates a vector Z with Nj elements

Z=sym('Z',[Nj 1]); % converts the vector Z to a symbolic mode

PZ=1;

fori=1:Nj

```
PZ=PZ*Z(i);      % PZ corresponds to the
product of the elements of Z
end
```

```
% Initialization of the vector corresponding to the value
of Deegan-Packel
```

```
VDeegan=zeros(Nj,1);
[u]=RandSimRepeticao(0,1E4,Nj);      %      10000
random numbers without repeating between 0 and Nj are
generated
syms x t y
```

```
P=zeros(Nj-1,1);
s=y*(1+(x^V(1)*Z(1)*t*y))*(1+(x^V(2)*Z(2)*t*y^2));
PotY=4;
for l=3:Nj      % Routine to establish generating
functions Bi(x)
    s=s*(1+(x^V(l)*Z(l)*t*y^PotY));
    PotY=4^l;
end
s1=expand(s);      % makes the symbolic
multiplication/
[cy]=coeffs(s1,y); % extracts the coefficients of a
polynomial, cy takes the values of y/
% and cy the values of y (Auxiliary variable)
cx=cx/(PZ*t^Nj);      % extracts the factor X of each
monomial/
[num, den]=numden(cx);
cx=num;
```

```
PotX=subs(log10(cx),x,10);      % extracts the value of
x=10 in the polynomial e and the
% total weight of each coalition is registered/
PotX=simplify(PotX);      % extracts the exponents of x/
a=x.^PotX;
ct=cx./(PZ*a);      % extracts the factor t of each
monomial.
[num, den]=numden(ct);
ct=num;
```

```
PotT=subs(log10(ct),t,10);      %evaluates the value
of t=10 in the polynomial e and
% the total weight of each coalition is registered.
PotT=simplify(PotT);      %extracts the exponents of t.
```

```
% the monomials that meet the condition that exponent
of X is >=q are selected.
```

```
m=find(PotX>=q);
PotXWin=PotX(m);      % Powers of X winners.
PotTWin=PotT(m);      % Powers of T winners
```

```
MGanadores=cxz(m);
```

```
nWin=length(PotTWin);
for i=1:nWin-1 % Routine to organize ascendently
PotWin
for j=i+1:nWin % and the array of winners
ifPotTWin(j)>PotTWin(i)
    Aux=PotTWin(j);
    PotTWin(j)=PotTWin(i);
PotTWin(i)=Aux;
```

```
Aux1=MGanadores(j);
MGanadores(j)=MGanadores(i);
MGanadores(i)=Aux1;
```

```
end
end
end
```

```
% Routine to extract the monomials that cannot be
divided by other terms.
```

```
R1=MGanadores; % A copy of the winning monomials
is extracted.
t1=PotTWin;
n1=length(R1);      %The number of winning coalitions
is identified.
Nminimales=1;
while n1>1
% The first minimal monomial is registered.
CMinimales(Nminimales)=R1(n1);
% the power of t of the minimal monomial is registered.
PTMinimales(Nminimales)=t1(n1);
R=R1/R1(n1); % the vector R1 is divided over the
last monomial in order to identify which monomial are
not divisible
[num, den]=numden(R); % The numerator and the
denominator of the last process in each monomial are
extracted.
m=find(den~=1); % Those monomials in which the
denominator is different from 1 are identified.
```

```

    R1=R1(m);    % Those monomials which are not
    divisible during the process are extracted.
    t1=t1(m);
    n1=length(R1);
    Nminimales=Nminimales+1;
end
    n1=length(R1);
    if n1>0
    CMinimales(Nminimales)=R1(1);
    PTMinimales(Nminimales)=t1(n1);
end

    % Calculation of the Deegan-Packel power index.
    Mv=length(CMinimales);    %Mv=total of minimal
    monomials.

    fori=1:Nj
        R=CMinimales/Z(i);    % The vector R1 is
        divided over the last monomial in order to identify which
        are not divisible.
        [num, den]=numden(R);    % The numerator and
        the denominator of the last process of each monomial are
        extracted.
        m=find(den==1);    % Those which
        denominator is different from 1 are identified.
        R1=CMinimales(m); % Those monomial that are not
        divisible during the process are extracted.
        t1=PTMinimales(m);
        n1=length(R1);
        suma=0;
        for j=1:n1    % The Deegan power indexes are
        calculated.
            suma=suma+(1/t1(j));
        end
        VDeegan(i)=suma/Mv;
    end

    VDeegan% Prints the vector corresponding to the power
    index for each player.

```

### 3. Results and Discussion

#### 3.1 Treatment Analysis

In order to compare each of the three proposed methods and calculate the power index of each player (Banzhaf, Shapley-Shubik and Deegan-Packel) to identify those that obtain the best result, according to the posed problema, it

is proposed the use of two scenarios that can be presented on a Home Area Network (HAN). In these scenarios, the Value of Shapely is applied as a strategy for optimizing resources<sup>11,12</sup> and then, by means of the *Dunnett's test*, it will be possible to identify which is or are the methods or treatments that get the best results during the evaluation process.

In order to calculate the total bandwidth ( $BW_T$ ) will be used the tool called "*Channel GeneratorPLC (GC\_PLC)*", written in MATLAB and developed by Francisco Javier Canete, who belongs to Group PLC of Universidad de Malaga (Espana). GC\_PLCallows to estimate the behavior of a channel with the parameters associated to the topology of a PLC network, on a typical household residential environment. Moreover, the tool carries out an evaluation process of the channel below the band of 30 MHz, considering that the network adapters PLC, under the HPAV standard, work in such band of frequency. In<sup>13</sup> all the information for the usage of the GC\_PLC tool is presented.

The tool GC\_PLC generated the values: 159.72 Mbps, 120.65 Mbps and 83.58Mbps. Those values correspond to the capacity that can be present on a PLC channel, under one channel condition: excellent, regular (typical) and deficient, respectively.

The proposed scenarios are the following:

Scenario 1: Channel State non saturated under multiclass traffic conditions.

In this scenario, a HAN network conformed by six (6) PLC nodes is proposed. Each node has a PLC adapter and a traffic source. Each source is able to generate more than one type of traffic  $r$  simultaneously (voice, data, video and telemetry).

The node 6 will be considered as main node or Coordinator (CCo) and will be in charge of establishing the reserve frequency-time (Schedule) for each of the nodes that are part of the PLC network. It will also be in charge of establish and socialize the order in which the nodes can access the channel. By this node will circulate the largest amount of traffic, due to its direct connection with the router for Internet access. Table 1 shows the types of traffic and the established codec (applicable just to Voice and Video), for each node  $i$  and type  $r$  that are part of the proposed scenario.

Table 2 shows the bandwidth required for each type of traffic for the proposed scenario.



According to the recorded values in Table 2, the total bandwidth required by HAN network is 67.68 Mbps, which is less than the total capacity of the channel PLC in any of the three suggested channel conditions, establishing a Non-Saturation state of the channel PLC.

**Table 1.** Type of traffic for each node  $i$  and type  $r$ , scenario 1

Node $i$	Type of Traffic			
	Voice ( $r=1$ )	Video ( $r=2$ )	Control ( $r=3$ )	Data ( $r=4$ )
1			X	X
2	G.723.1			X
3		MPEG-1	X	
4	G.729			X
5		MPEG-2		X
6(Cco)	X	X	X	X

**Table 2.** BW required for each node  $i$  and type  $r$  scenario 1

Node $i$	Type of Traffic			
	Voice [kbps]	Vid-eo [Mbps]	Con-trol [kbps]	Data [Mbps]
1			2,75	9,19
2	23,50			6,14
3		2,32	1,38	
4	44,72			5,45
5		4,07		6,61
6(Cco)	68,22	6,39	4,13	27,38

Scenario 2: Channel State saturated under multiclass traffic conditions.

In this scenario, it is proposed a HAN network conformed by eight (8) PLC nodes. Each node will be made by a PLC adapter and a traffic source, which can generate more than one type of traffic  $r$  simultaneously (voice, data, video and telemetry). The node 8 will be considered as main node or Coordinator (CCo). In Table 3, the types of traffic and the established codec is presented (applicable only to voice or video) for each node  $i$  and type  $r$  that make part of the proposed scenario.

Table 4 presents the BW required for each type of traffic for the proposed scenario.

According to the values registered in the Table 4, the BW total required by the HAN network is 193.86 Mbps, which is superior to the total capacity of the PLC channel in each of the three suggested conditions, establishing a state of saturation of the PLC channel.

**Table 3.** Type of traffic for each node  $i$  and type  $r$  scenario 2

Node $i$	Type of Traffic			
	Voice ( $r=1$ )	Video ( $r=2$ )	Control ( $r=3$ )	Data ( $r=4$ )
1	G.711		X	X
2	G.723.1		X	X
3		MPEG-4		X
4	G.729		X	X
5		MPEG-2	X	
6				X
7	G.711			X
8 (Cco)	X	X	X	X

**Table 4.** BW required for each node  $i$  and type  $r$  scenario 2

Node $i$	Type of Traffic			
	Voice [kbps]	Vid-eo [Mbps]	Con-trol [kbps]	Data [Mbps]
1	81,6		1,5	3,5
2	23,2		2,2	10,2
3		18,4		8,4
4	35,8		2,7	12,7
5		15,38	1,8	
6				16,5
7	102,6			11,6
8 (Cco)	243,2	33,78	8,2	62,9

In the two scenarios it is assumed that the CCo node will be directly connected to the router that allows to connect the HAN network to Internet and in this node will circulate the most traffic. It is also assumed that a great part of the services will depend on internet.

Taking into account that there is no a methodology to calculate the weight of a player on a simple *Weighted Voting game*, the following criterion to establish each of the required weights was adopted:

Since each player can present more of a type of service, it is necessary to calculate a weight for each class  $r$  and player  $i$ , with  $r = \{1, 2, 3, 4\}$  for services of {data, control, video, voice} respectively. The total weight for each player  $i$  can be calculated by means of the use of the following expression (21):

$$W_i = \sum_{r=1}^4 W_{i,r} \quad (21)$$

On the other hand, the weight for each type  $r$  can be calculated by means of the use of the following expression (22):

$$W_{i,r} = FP_{i,r} \left[ \left( \frac{BW_{i,r}}{\min_{j \in N, k=r} BW_{j,k}} \right) \right] \quad (22)$$

Where  $\min_{j \in N, k=r} BW_{j,k}$  corresponds to the minimal value of Bandwidth (BW) allocated between all the players whose type of service be  $k=r$ ,  $FP_{i,r}$  is the priority factor of the type  $r$  player  $i$  and  $\lceil \cdot \rceil$  represents the operator that gets as a result the highest integer nearer to the registered operation.

In Tables 5 and 6 the initial conditions of each proposed scenario are presented. Considering the number of identification of the player  $i$  or node, the Type of Service (ToS), the Priority Factor (PF), the BW allocated to each type by means of the value of Shapley under typical (regular) channel conditions, the weight allocated by type of service and the total node weight, for both proposed scenarios.

In order to analyze if the condition of the PLC channel can in certain way have an effect on the weight allocated to a player  $i$ , the weight for each player was calculated, under excellent, regular and deficient channel conditions. The result of this process of estimation is registered in Table 7, where it is possible to observe that both the total weight and weight for each type of service, under the three channel conditions, are practically the same. Due to this, it is possible to determine that the state of the PLC channel does not influence on a meaningful manner on the estimated weight for a player and over the respective power index. In view of the above, the initial conditions are considered for only one state of PLC channel<sup>7</sup>.

In Table 7, it is possible to observe that  $W_N$  presents

a rather high value compared with  $W_i$ , which can generate as a result  $I_{dp_{i \neq N}} = 0$  e  $I_{dp_{i=N}} = 1$ . The foregoing does not allow estimate adequately the level of power or influence in each node, as the node  $N$  would be considered as a player “dictator”, where such dictator would be the only key player in all the coalitions. The rest of the players (nodes) would be considered “null”, that is that each player would not have any power on the game whatsoever. In view of the above, the use of the following expression is proposed, in order to calculate the adjusted value  $W_N^*$ :

$$W_N^* = \left( \left[ \frac{\max_{i \in N, i \neq N} W_i}{10} \right] + 1 \right) \cdot 10 \quad (23)$$

Where  $W_N^*$  corresponds to the total adjusted weight for the node  $N$  (CCo) and  $\lceil \cdot \rceil$  is the integer part of the registered operation.

In Tables 8 and 9, the value corresponding to the power index estimated in each node is registered for each one of the suggested scenarios. The data are organized on a descending manner in relation to the power index and weight. Observing the result in each proposed scenario, it is possible to see that the position of each node for the three methods proposed is practically the same, which poses the hypothesis that any of the three methods proposed can be considered suitable for calculating the power indexes in each node that is part of a HAN network.

It is important to highlight that the order in which the nodes can access the channel corresponds to the result of

**Table 5.** Initial conditions for the estimation of power indexes - scenario 1

Player (Node) i	ToS (r)	Priority Factor (PF <sub>(i,r)</sub> )	BW Allocated (Shapley) (BW <sub>(i,r)</sub> )	Weight (W <sub>(i,r)</sub> )	Total W <sub>i</sub>
1	Control	2	3,7850	4	6
	Data	1	12,9730	2	
2	Voice	4	3,8060	4	6
	Data	1	9,9230	2	
3	Video	3	6,1030	3	5
	Control	2	3,7840	2	
4	Voice	4	3,8270	8	9
	Data	1	9,2330	1	
5	Video	3	7,8530	6	8
	Data	1	10,3930	2	
6	Voice	4	3,8510	8	20
	Video	3	10,1730	6	
	Control	2	3,7870	4	
	Data	1	31,1630	4	

**Table 6.** Initial conditions for the estimation of power indexes - scenario 2

Player (Node) i	ToS (r)	Priority Factor (PF <sub>(i,r)</sub> )	BW Allocated (Shapley) (BW <sub>(i,r)</sub> )	Weight (W <sub>(i,r)</sub> )	Total W <sub>i</sub>
1	Voice	4	0,0580	16	19
	Control	2	0,0011	2	
	Data	1	2,4870	1	
2	Voice	4	0,0167	4	11
	Control	2	0,0016	4	
	Data	1	7,3497	3	
3	Video	3	13,7348	6	9
	Data	1	6,2702	3	
4	Voice	4	0,0259	8	16
	Control	2	0,0020	4	
	Data	1	9,1822	4	
5	Video	3	11,1597	3	7
	Control	2	0,0013	4	
6	Data	1	11,9820	5	5
7	Voice	4	0,0741	20	24
	Datos	1	8,3769	4	
8 CCo	Voice	4	0,2233	56	30
	Video	3	31,0153	9	
	Control	2	0,0075	16	
	Data	1	57,7519	24	

**Table 7.** Estimation of the total weight for a player I vs. PLC channel condition

Channel Conditions			Excellent			Regular			Deficient		
Player (Node) i	ToS	Priority Factor	BW Allocated (Shapley)	Weight (W)	Total W <sub>i</sub>	BW Allocated (Shapley)	Weight (W)	Total W <sub>i</sub>	BW Allocated (Shapley)	Weight (W)	Total W <sub>i</sub>
1	Voice	4	0,0580	16	19	0,0449	16	19	0,0386	16	19
	Control	2	0,0011	2		0,0008	2		0,0007	2	
	Data	1	2,4870	1		1,9253	1		1,6557	1	
2	Voice	4	0,0167	4	11	0,0128	4	11	0,0110	4	11
	Control	2	0,0016	4		0,0012	4		0,0010	4	
	Data	1	7,3497	3		5,6429	3		4,8459	3	
3	Video	3	13,7348	6	9	9,8756	6	9	8,9741	6	9
	Data	1	6,2702	3		4,5084	3		4,0969	3	
4	Voice	4	0,0259	8	16	0,0197	8	16	0,0170	8	16
	Control	2	0,0020	4		0,0015	4		0,0013	4	
	Data	1	9,1822	4		6,9928	4		6,0367	4	
5	Video	3	11,1597	3	7	8,4300	3	7	7,3611	3	7
	Control	2	0,0013	4		0,0010	4		0,0009	4	
6	Data	1	11,9820	5	5	9,0280	5	5	7,9210	5	5
7	Voice	4	0,0741	20	24	0,0566	20	24	0,0488	20	24
	Datos	1	8,3769	4		6,3954	4		5,5132	4	
8 CCo	Voice	4	0,2233	56	105 (30**)	0,1699	56	102 (30**)	0,0930	36	67 (30**)
	Video	3	31,0153	9		23,5972	9		12,9131	6	
	Control	2	0,0075	16		0,0057	14		0,0031	10	
	Data	1	57,7519	24		43,9392	23		24,0448	15	

**Table 8.** Power indexes for scenario 1

$W_i$	$I_{dp}$ – Banzhaf	Node i	$I_{dp}$ – Shapley-Shubik	Node i	$I_{dp}$ – Deegan-Packel	Node i
20	0,4545	6	0,4667	6	0,25	6
9	0,1212	4	0,1667	4	0,1563	4
8	0,1212	5	0,1667	5	0,1563	5
6	0,1212	1	0,0667	1	0,1458	1
6	0,1212	2	0,0667	2	0,1458	2
5	0,0606	3	0,0667	3	0,1458	3

**Table 9.** Power indexes for the scenario 2

$W_i$	$I_{dp}$ – Banzhaf	Node i	$I_{dp}$ – Shapley-Shubik	Node i	$I_{dp}$ – Deegan-Packel	Node i
30	0,2564	8	0,2786	8	0,1621	8
24	0,1709	7	0,2024	7	0,1437	7
19	0,1624	1	0,1452	1	0,1402	1
16	0,1368	4	0,1167	4	0,1264	4
11	0,094	2	0,0833	2	0,1236	2
9	0,0769	3	0,069	3	0,1132	3
7	0,0598	5	0,069	5	0,1132	5
5	0,0427	6	0,0357	6	0,0776	6

organizing on a descending manner the resulting power indexes. That is, that the node that has the higher power index will be the first in having access to the channel, and the lowest will be the latter.

### 3.2 Evaluation of MCB (Multiple Comparisons with the Best)

In this item, the behavior of the three proposed methods will be evaluated (Banzhaf, Shapley-Shubik and Deegan-Packel), supported in the concept of the power index as a strategy for the allocation of the space frequency-time in the *scheduler*. Such scheduler will regulate the medium access in each node over the PLC channel. In Tables 10 and 11 the samples corresponding to the power index for each case are presented.

Additionally, the values related to the average, and the quadratic error for each player is registered. The quadratic

error is calculated by means of the use of the expression  $[Y_{ij} - \mu_i]^2$ , where  $Y_{ij}$  y  $\mu_i$  corresponds to the power index associated to the optimization method  $i$  player  $j$  and the average value of the optimization method  $i$  respectively. It is important to take into account that the value of the quadratic error is necessary to calculate the value of CME (CuadraticMedia Error) which is fundamental in order to apply the *Dunnett's test*<sup>14</sup>.

Based on the values registered in Table 11, the values corresponding to the Sum of Quadratic Errors (*SCE*), the degrees of liberty (*G.L.*), the Cuadratic Media error (*CME*) and  $M$ ; for a total of 3 treatments ( $N_t$ ) and 14 samples or replicas ( $N_m$ ), are the following:

$$SCE = \sum_{i=1}^{N_t} \sum_{j=1}^{N_m} [Y_{ij} - \mu_i]^2 = 0,1681 + 0,2038 + 0,0202$$

$$SCE = 0,3922 \quad (24)$$

**Table 10.** Estimation of averages in relation to the power index

$W_i$	$I_{dp}$ – Banzhaf	Node i	$I_{dp}$ – Shapley-Shubik	Node i	$I_{dp}$ – Deegan-Packel	Node i
30	0,2564	8	0,2786	8	0,1621	8
24	0,1709	7	0,2024	7	0,1437	7
19	0,1624	1	0,1452	1	0,1402	1
16	0,1368	4	0,1167	4	0,1264	4
11	0,094	2	0,0833	2	0,1236	2
9	0,0769	3	0,069	3	0,1132	3
7	0,0598	5	0,069	5	0,1132	5
5	0,0427	6	0,0357	6	0,0776	6

$$G.L = N_t[(N)_m - 1] = 3(14 - 1) = 39 \quad (25)$$

$$CME = \frac{SCE}{N_t[(N)_m - 1]} = 1.01E - 2 \quad (26)$$

$$M = d_{\alpha,k,v} \sqrt{\frac{2S^2}{r}} = 1,97 \sqrt{\frac{2CME}{N_m}} = 0,075 \quad (27)$$

Table 12 presents the parameters corresponding to the procedure named “multiple comparisons with the Best (MCB)”[100], in order to carry out a process of comparison between the three methods of optimization proposed and to identify which methods can be considered as the best between the three, for ultimately make the process of medium access supported in the use of the concept named “power index”.

Given the results recorded in Table 12, when performing *Dunnett's test*, any of the three methods could be considered as the best, due to the fact that the simultaneous confidence intervals include zero between the superior and inferior limits. However, when evaluating the difference  $D_i$  it can be observed that the method proposed by Shapley-Shubik presents the biggest

average and, additionally, is the only one between the three methods that meets the condition  $\mu_i - \max_{j \neq i} \mu_j \geq 0$  which indicates that the Shapley-Shubik power index Shapley-Shubik can be regarded as the best of the three methods of optimization.

## 4. Conclusions

The evaluation of the three methods proposed to calculate the power index (Banzhaf, Shapley-Shubik and Deegan-Packel) and identifying the best outcome, two scenarios under multiclass traffic conditions were considered: 1. State of non-saturated channel and 2.State of saturated channel. Based on the results obtained by *Dunnett's test*, it was possible to evidence that any of the three methods could be considered as one of the best, as the simultaneous confidence intervals include zero between the superior and inferior limits. When evaluating the difference  $D_i$  it was possible to observe that the optimization method 'proposed by Shapley-Shubik presented the highest average and plus it is the only method that meets the condition  $\mu_i - \max_{j \neq i} \mu_j \geq 0$ , which allows to conclude that Shapley-Shubik power index can be considered as the best

**Table 11.** Estimation of the cuadratic error in relation to the power index

Sample	Power Indexes			Sum of CuadraticErrors (SCE)		
	Banzhaf	Shapley-Shubik	Deegan_Packel	SCE_Banzhaf	SCE_Shapley	SCE_Deegan
1	0,4545	0,4667	0,2500	0,0728	0,0763	9,50E-03
2	0,1212	0,1667	0,1563	0,0040	0,0006	1,42E-05
3	0,1212	0,1667	0,1563	0,0040	0,0006	1,42E-05
4	0,1212	0,0667	0,1458	0,0040	0,0153	4,53E-05
5	0,1212	0,0667	0,1458	0,0040	0,0153	4,53E-05
6	0,0606	0,0667	0,1458	0,0154	0,0153	4,53E-05
7	0,2564	0,2786	0,1621	0,0051	0,0078	9,16E-05
8	0,1709	0,2024	0,1437	0,0002	0,0001	7,79E-05
9	0,1624	0,1452	0,1402	0,0005	0,0021	1,52E-04
10	0,1368	0,1167	0,1264	0,0023	0,0055	6,83E-04
11	0,0940	0,0833	0,1236	0,0082	0,0115	8,37E-04
12	0,0769	0,0690	0,1132	0,0116	0,0148	1,55E-03
13	0,0598	0,0690	0,1132	0,0156	0,0148	1,55E-03
14	0,0427	0,0357	0,0776	0,0202	0,0240	5,61E-03
Total SCE				0,1681	0,2038	0,0202

**Table 12.** MCB on the proposed methods of optimization

Method	Average ( $\mu_i$ )	Max ( $\mu_{(i \neq j)}$ )	M	$D_i$	$D_i - M (L)$	$D_i + M (U)$
Banzhaf	0,185	0,191	0,075	-0,006	-0,080	0,069
Shapley-Shubik	0,191	0,185	0,075	0,006	-0,069	0,080
Deegan-Packel	0,153	0,191	0,075	-0,038	-0,113	0,037



of the three methods of optimization. On the other hand, it was possible to observe a very interesting phenomenon in which, although the values corresponding to the power indexes for each one of the proposed methods are different, the result of organizing the nodes descendent, in function of the power indexes, for the three methods in the two proposed scenarios, got the same result. This means that the three methods do not present significant differences between one another when estimating power indexes for each node that is part of the network.

The process of calculating the power indexes for each node that are part of a PLC network, can be considered as a trouble as it increases the number of nodes because the classical algorithms use permutations, increasing the levels of computational complexity required. Nonetheless, and in order to provide a solution to the problem and a contribution to the study of PLC technology, three MATLAB routines are developed. Such routines allow to estimate the Banzhaf, Shapley-Shubik and Deegan-Packel power indexes by means of the use of the generatriz functions and considering the need of network access as a *Weighted Voting game*<sup>(N, v)</sup>, defined by  $v = [q; w_1, w_2, \dots, w_n]$ , where  $N$  and  $v$  correspond to the node number and the weight vector, respectively. During simulation process and under equal conditions, both in weight and number of nodes, it was possible to observe that the routines that needed, in all scenarios, the highest and lowest execution time were those of Deegan-Packel and Banzhaf, respectively. The development of those three routines in MATLAB can be considered as a great contribution for further researches, due to the fact that those can be used as functions in other simulation scenarios as required.

## 5. References

1. International Telecommunications Union. Recommendation Y.1541. Network performance objectives for IP-based services. IUT-T; Switzerland. 2011. p. 1–66.
2. Lee Y, Latchman H, Ha H, Hwang M. QoS augmented CSMA/CA MAC protocol for high speed multimedia powerline communications. Proceedings of IEEE Int Symp Power Line Commun its Applicationas. IEEE; Mexico. 2012. p. 236–41.
3. Banarwal S, Sharma A, Verma H. Power line communication. Indian Journal of Science and Technology. 2016; 9(21):1–4. Crossref
4. Pi-ero-Escuer PJ, Malgosa-Sanahuja J, Manzanares-Lopez P. Homeplug-AV CSMA/CA evaluation in a real in-building scenario. IEEE Communications Letter. 2011 Jun; 15(6):683–5. Crossref
5. Formacion de coaliciones en los juegos cooperativos y juegos con multiples alternativas. 2017. Crossref
6. Febles ME. Sistema de reparto de poder en las elecciones locales. Numeros. 1999 Jun; 39:13–20.
7. Pounambal M. Survey on channel allocation techniques for wireless mesh network to reduce contention with energy requirement. Indian Journal of Science and Technology. 2016 Aug; 9(32):1–12. Crossref
8. Rafsanjani MK, Aliahmadipour L, Javidi MM. A hybrid intrusion detection by game theory approaches in MANET. Indian Journal of Science and Technology. 2012 Feb; 5(2):2123–31.
9. Complejidad y algoritmos en juegos cooperativos. 2017. Crossref
10. Ribnikov K. Analisis Combinatorio. Editorial Mir Moscu; 1988.
11. Vesga-Ferreira JC, Granados-Acu-a G, Sierra JE. El valor de shapley como estrategia de optimizacion de recursos sobre Power Line Communication (PLC). Ingenieria y Ciencia. 2015, Jun; 11(22):189–211. Crossref
12. Vesga-ferreira JC, Granados-Acu-a G, Sierra JE. The nucleolus as a strategy for resources optimization in LANs on power line communications. Indian Journal of Science and Technology. 2016 Jul; 9(28):1–14. Crossref
13. User guide for PLC channel generator v. 2. 2017. Crossref
14. Holm S. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics. 1979 Jun; 6(2):65–70.

1. International Telecommunications Union. Recommendation