Adaptive Cognitive System Applied to WSN Decisions at Nodes with a Fuzzy Logic Approach

Marcel Stefan Wagner
Department of Electronic Systems Engineering, Polytechnic School, University of São Paulo, São Paulo, Brazil
marcel.wagner@usp.br

Miguel Arjona Ramírez
Department of Electronic Systems Engineering, Polytechnic School, University of São Paulo, São Paulo, Brazil
miguel@lps.usp.br

Wagner Luiz Zucchi
Department of Electronic Systems Engineering, Polytechnic School, University of São Paulo, São Paulo, Brazil
wzucchi@lps.usp.br

Abstract **-**The Adaptive Cognitive System (ACS) presented here is based on the concept of cognition applied to Wireless Sensor Networks (WSN) concerning aspects related to memory, history and decision making over network node tasks. The use of cognitive features in the WSN scope allows the nodes to make better decisions about conflicts or anomalies arising from node or route failures that probably will affect the performance network as a whole. Moreover, with the use of cognitive aspects in feedback processes to make decisions in a multilayer approach, it is possible to obtain an improvement in the data transmitted end-to-end by the nodes. The decision process consists of adjustments in memory, queue, route protocols and energy consumption. A Fuzzy Inference System (FIS) is proposed for decision making from the vector collected from the network, and this logic determines the adjustments to be applied to the network. This inference system is expandable, allowing other rules, metrics and parameters to be added to the analysis for more flexibility and improved performance.

Keywords – Wireless; WSN; Cognitive Network; Fuzzy Logic.

I. Introduction

Wireless Sensor Networks (WSN) represent a major network research area since it fosters different kinds of applications and suits interests that range from a simple exchange of information to the transmission and reception of significant data. Safety aspects are not only the important parameters to be explored in WSN.

Algorithms and protocols focused on the transmission speed, packet delivery increase and node energy economy are well surveyed in this field. But features related to performance and Quality of Service (QoS) are important for evaluating whether the network meets its goals and for rating its acceptance.

Focusing on the improvement of network performance, this work applies a number of cognitive processes within WSN nodes to introduce intelligence based on memory and critical analysis which provide the means for the decision making process, that will be described in the section IV. The combination of cognition with WSN to solve network problems underlines the need for studies encompassing different areas of expertise in search of elaborate alternative solutions.

Cognitive parameters can be inserted in logical mechanisms as the system's intelligent part, allowing it to act on the collected data obtained from network feedback vectors according to history and memory aspects, observing yet the network's policies and purposes. Once the data are gathered, the cognitive process is able to proceed to the decision making stage.

In order to provide the means for node decisions, an analysis based on a decision-threshold is made with system's critical table values collected from the network monitoring vector parameters, as described in next sections. These vectors are

DOI- 10.18486/ijcsnt.2016.5.1.01
ISSN-2053-6283

essential for system analysis because they are the source of the critical tables. Specifically, these vectors are obtained from *Prowler* simulator running over MATLAB platform.

Some researchers in this area treat the theme focusing on prediction routing schemes, such as Jie Li et al [1], who suggest a routing algorithm based on a traffic prediction model known as Efficient Traffic Aware Multi-path Routing (ETAMR). The ETAMR algorithm uses traffic distribution and load to build a multi-path routing using a prediction model. Metrics such as node energy consumption or even network throughput are not considered in this model and are the matter of their future works.

In contrast with [1], instead of relying on a prediction mechanism to change some aspects of network routing, the present work uses the actual readings of network data, collected in predetermined intervals, to execute cognitive processes and provide actuation on network aspects such as routing. This allows for more coherent alterations, not only in routing aspects, but in the nodes processing schemes, buffer operation and energy consumption economy.

The algorithm developed here emphasizes the QoS regarding end-to-end packet delivery ratio among network nodes in scenarios with random node's movement. Therefore, control over the amount of energy consumption in the nodes will be approached as part of the cognitive process.

In [10] another prediction model is presented and in [2] and [11] the importance of WSN integration and application in Internet of Things (IoT) meaning are considered. [3] and [5] gives us examples of WSN simulations using NS-2 software. [6] and [7] treat about cognition applied into WSN environments. [4], [8] and [9] have examples of certain difficulties or anomalies concerning sensor networks.

The rest of paper is organized as follows. Section II describes the node's internal architecture and interface into block's diagrams. Section III represents the environment description and important features related to nodes dimension. Section IV corresponds to the system's development and organization. Section V describes the Fuzzy Logic implementation. Section VI shows the results. Finally, concluding remarks and future survey directions are given in Section VII.

II. Node Description

It is important to know the node's internal architecture, as well as the modules connections of sensor nodes. Therefore, it can be viewed ahead the internal structure of the sensor studied in this paper.

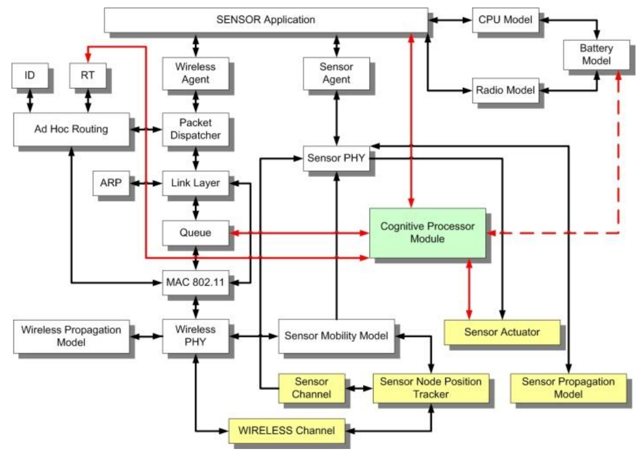

Figure 1.The general node structure with cognitive module.

As can be seen in Fig. 1, the presented structure corresponds to a common sensor node with the insertion of a cognitive module responsible for the execution of cognitive processes.

In the scheme, white blocks represent the traditional internal interface of sensor nodes, yellow blocks are the medium representation and green block represents the cognitive module developed in this work.

The Cognitive Processor Module (CPMod) is the intelligent sensor part with cognitive aspects engaged to execute the Decision Making process (DMp) for acting over sensor layers concerning memory and changing queue modeling according to the necessity. In the following, ID corresponds to the node identification and RT to the route tables. ARP is the Address Resolution Protocol and acts according to 802.11 standard for Wireless Local Area Network (WLAN). The continuous red interfaces were implemented here and dashed line corresponds to the cognitive energy control.

Note that the cognitive module operates in a higher layer concerning the node's internal structure. Moreover, the lower modules will continue to execute their tasks normally, although the decision making running in the cognitive module will affect some internal parameters such as queues rules, buffers allocations, table routes (i.e. ordering to optimize routing protocols operating in the node) and antenna power gain to accomplish network performance adjustments.

III. Environment and Node Details

This work considers the application of dynamic or differentiated media. The study comprises networks with random node movement, so it is not limited to a single scenario. The network environment is variable, thus any link between two nodes should be reestablished if any alteration in the environment occur or due to the changes in nodes positions, so the system can adapt to it and to its logical routes configurations.

To identify the node dimensions for the environment applied, the system-related variable is given by (1):

$$F_V = (\alpha . P_V + \beta . R_V + \gamma . L_V + \sigma . N_V) \quad (1)$$

From (1), F_V represents the node variability number between [0,1] for a single trajectory and it is composed by the following: P_V, accounting for the node position variation; R_V, accounting for the node radius variation; L_V represents the variation in the number of links established by nodes and N_V is the number of nodes variation in the network.

The system variability-related variables, P_V, R_V, L_V and N_V are obtained from the following sequence. To define the position variable P_V, were used the random motion feature of the N_1 node with the displacement from *a* to *b*, as is shown in Fig. 2.

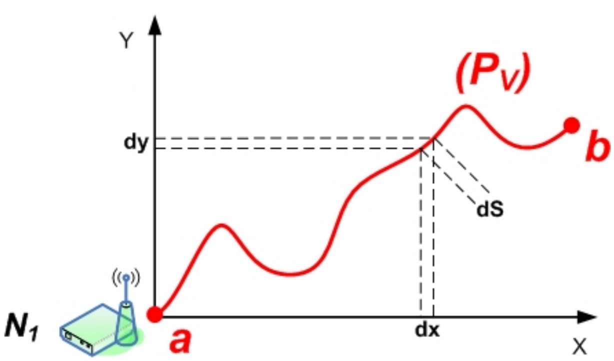

Figure 2. The trajectory from *a* to *b* in a N_1 random movement.

Assuming that *dS* is the differential of the curved line segment representing the trajectory of node N_1 from *a* to *b* and its delimited by *dx* and *dy*, then we have $= \sqrt{dx^2 + dy^2}$. The trajectory from *a* to *b* is referred to as curve *C* in the following. Thus, denoting the continuous trajectory of the node by $f_{(X,Y)}$, the node displacement from *a* to *b* is:

$$P = \int_C f_{(X,Y)} dS \quad (2)$$

In this way, the node displacement is described by (3), where $f_{(x,y)}$ is the function of *a* to *b* segments, which can be a concatenation of several basic functions, such as straight line segments, joined to cover the whole path from *a* to *b*.

$$P_V = \left\{ P_t - P_{t-1} \mid t \geq 1;\ P = \int_C f_{(X,Y)} dS = = \int_a^b f_{(x,y)} . \sqrt{1 + (y')^2} dx \mid y' = \frac{dy}{dx} \right\} \quad (3)$$

From the above, P_V is the position analysis result, comparing the present node location with the previous node location. *P* denotes the node trajectory in the two-dimensional plane (x,y) from a point *a* to *b*, as shown in Fig. 2. The antenna radius is given by:

$$R = r_{t-1} + \int_{r_{t-1}}^{r_t} dr \mid r_t \neq r_{t-1} \quad (4)$$

in the simplified model in Fig. 3, where R_V denotes the signal power in the maximum radius border with connection possibility and *r* is related to the antenna range in *t* or *t-1* intervals.

In Fig. 3, two formats can be observed about the condition radius *r* of node coverage signal. In the arrows, the solid red line represents a decrease in range of the node and the dotted red line shows a gain in the range of the antenna signal. With the variation between the current r_t and the earlier r_{t-1}, and considering yet a difference *dr* between them, then the value of the range for the current time is obtained as:

Therefore, the variation of the R_V range shall be subject to *R*, which may be equal to r_{t-1} if there is no change in the node coverage radius. It is considered in (5) a variation *r* tending to the infinity only to demonstrate that the antenna does not have a predefined limit, which is correct from the mathematical point of view, however, inconsistent from the network point of view due to the real range of the sensors antennas, which are mostly in the tens of meters.

$$R_V = \left\{ R_t - R_{t-1} \mid t \geq 1;\ 0 \leq r \ll \infty;\ r_{t-1} \leq R \leq \left(r_{t-1} + \int_{r_{t-1}}^{r_t} dr \right) \right\} \quad (5)$$

The range of the antenna signal is also directly linked to the signal fading effect in the transmission medium, which in wireless communications is Rayleigh, i.e. the greater the distance between the nodes, the larger is the signal loss according to the Rayleigh curve $f_{(x)}$ of the Probability Density Function (PDF), which is directly related to distance from the source node, and is presented in Fig. 4 and having the notation R ~ Rayleigh(σ), where σ is the scaling factor for the signal fading.

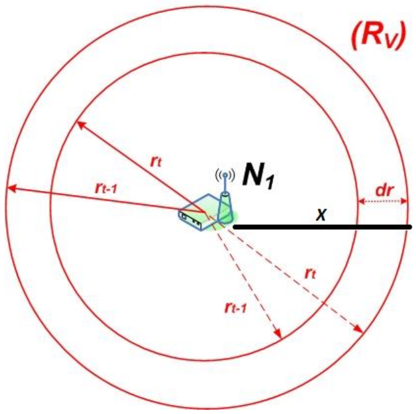

Figure 3. Node signal coverage feature in WSN.

Equation (6) represents the PDF of x. As can be seen in Fig. 4, x is the distance and σ is the fading factor for the curve.

Fig. 4a shows the PDF for a sequence of 5 simulations of σ and Fig. 4b shows the distribution function for the same 5 simulations of σ changes.

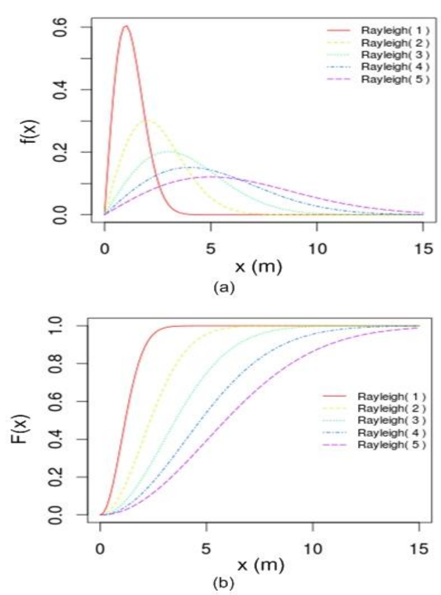

Figure 4. PDF and distribution function of Rayleigh for communication channel.

It is also noted in Fig. 4a that the larger σ gets the straighter the curve. Also, in Fig. 4b, as σ increases, the function spreads wider.

$$f_{(x)} = \frac{x}{\sigma^2} e^{\left\{\frac{-x^2}{2\sigma^2}\right\}}, x \geq 0 \tag{6}$$

Equation (7) shows the distribution function for the presented Fig. 4b.

$$F_{(x)} = 1 - e^{\left\{\frac{-x^2}{2\sigma^2}\right\}} \tag{7}$$

Therefore, considering the fading effect of (6) and applying $F_{(x)}$ from the (7) in the position equation (5), it has:

$$R_V = \left\{ R_t - R_{t-1} \,;\, r_{t-1}.\left(1 - F_{(x)}\right) \leq R \leq \left(r_{t-1} + \int_{r_{t-1}}^{r_t} dr \right).\left(1 - F_{(x)}\right) \right\}, \tag{8}$$
$$1 \leq \sigma \leq 2;\; 0 \leq r \ll \infty; x \geq 0$$

After that, connections will probably will fail with 70% chance and current connections will drop with over 85%.

$$L_v = \left\{ L_t - L_{t-1} \,|\, t \geq 1;\; 0 \leq L \leq (N-1); - N_{neig} \subset R_{neig(S)} \right\} \tag{9}$$

With respect to the L_V parameter, L is associated to the number of links established by nodes. N_{neig} corresponds to the number of neighboring nodes that should be inside the radius area of source node.

$$N_v = \left\{ N_t - N_{t-1} \,|\, t \geq 1 \right\} \tag{10}$$

Equation (10) shows N_V as the current number of nodes in the network. It can be changed according to the insertion or loss of nodes in the network. The ratios among P_V, R_V, L_V and N_V are the factors α, β, γ and σ respectively, and are given by:

$$\sum_{i=1}^{N} (\alpha + \beta + \gamma + \sigma)_i = 1 \Rightarrow$$
$$\Rightarrow \begin{cases} \alpha = \frac{1}{4}.\left|\frac{P_t - P_{t-1}}{\operatorname{argmax}(P_t - P_{t-1})}\right|, \forall\alpha \in \mathbb{R} \mid 0 \leq \alpha \leq 2 \\ \beta = \frac{1}{4}.\left|\frac{R_t - R_{t-1}}{\operatorname{argmax}(R_t - R_{t-1})}\right|, \forall\beta \in \mathbb{R} \mid 0 \leq \beta \leq 2 \\ \gamma = \frac{1}{4}.\left|\frac{L_t - L_{t-1}}{\operatorname{argmax}(L_t - L_{t-1})}\right|, \forall\gamma \in \mathbb{R} \mid 0 \leq \gamma \leq 2 \\ \sigma = \frac{1}{4}.\left|\frac{N_t - N_{t-1}}{\operatorname{argmax}(N_t - N_{t-1})}\right|, \forall\sigma \in \mathbb{R} \mid 0 \leq \sigma \leq 2 \end{cases} \tag{11}$$

DOI- 10.18486/ijcsnt.2016.5.1.01
ISSN-2053-6283

Equation (11) refers to the proportions related to the system's variability factors: α, the position ratio; β, the node radius ratio; γ, the ratio of the node's possible connections and σ, the ratio of number of nodes in the network.

The variables P_V, R_V, L_V and N_V are obtained from (2)-(10) and the relations in the variability detection are presented in pseudo-code part posted ahead, where t represents current time and t-1 represents time in the previous step.

The relationship among system variables P_V, R_V, L_V and N_V is depicted in Fig. 5. The number of nodes in a unique network must be defined (or monitored, if it already exists) based on the monitored/applied environment requirements. In other words, to accomplish the environment necessities is necessary to know the real application of sensor nodes.

Algorithm for sensor variability detection

N = Number of nodes in the network;

$n = 0;$

$t = 0;$

$WHILE\ (n \le N)$

$WHILE\ (t > 0)$

$$LET \quad P_{V(n)} = \phi_0, \quad \alpha_{(n)} = \theta_0$$
$$R_{V(n)} = \phi_1, \quad \beta_{(n)} = \theta_1$$
$$L_{V(n)} = \phi_2, \quad \gamma_{(n)} = \theta_2$$
$$N_{V(n)} = \phi_3, \quad \sigma_{(n)} = \theta_3$$

$\left\{ \mathrm{P_{V(n)}},\ \mathrm{R_{V(n)}},\ \mathrm{L_{V(n)}}\ \text{and}\ \mathrm{N_{V(n)}}\ \text{are readings of network by nodes measurements.} \right\}$

$FOR \quad i = 0 \ \ to\ \ 3$

$$DO \quad \theta_i = \frac{1}{4} \cdot \left| \frac{\phi_{i(t)} - \phi_{i(t-1)}}{\operatorname{argmax}(\phi_{i(t)}, \phi_{i(t-1)})} \right|$$

$$F_{V\,i(n)} = \prod_{i=0}^{3} \theta_i \phi_{i(t)}$$

$$F_{V(n)} = \sum_{i=0}^{3} F_{V\,i(n)}$$

$\left\{ \mathrm{F_{V(n)}}\ \text{represents the percentage relative to the variability in}\ \mathrm{n^{th}}\ \text{node.} \right\}$

$END\ \ FOR$

$t = t + 1$

$n = n + 1$

$END\ \ WHILE$

$END\ \ WHILE$

If the environment being monitored is small and requires few nodes, the number of nodes N is a known parameter instead of a variable one, constraining the variability feature to three variables and the multiplicative factors assume 1/3 ratio unless 1/4. If the random enrolment of nodes to the network is required, then should be used the complete equation with four variables, as given in (1).

According to (2)-(10), if the subtraction shows result 0 the system variable can assume binary value 0 corresponding to no variation in the network parameter, and if exists another value in subtraction result, it indicates that there is occurrence of variation in the parameter affording binary value 1 (it is for P_V, R_V, L_V and N_V values applied in (1)).

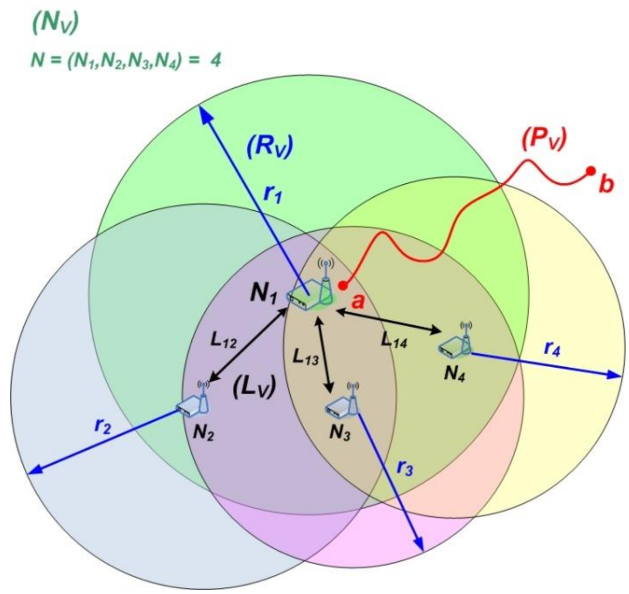

Figure 5. The general features of nodes connection in the network.

The cognitive process proceeds with the monitoring of nodes applying the relationships given in (1) to (11) for each node, and then calculates the variability feature F_N that is the F_V average of an amount of trajectories as shown in (12).

$$F_N = \overline{X}\left(F_{V(n)}\right) = \frac{\sum_{n=1}^{N} F_{V(n)}}{N} \tag{12}$$

In pseudo-code described before, the calculation round find out the parameters and respective ratios to facilitate the F_N achievement. F_N is one of the parameters used by the cognitive process running into

CPMod to provide Decision Making process (DMp) and it will be described in next section.

IV. System Features and Development

The cognitive process uses the memory vector V_M, the variability feature F_N (described in previous section) and the input signal $S_I(n)$ to generate the output signal $S_O(n)$ that controls the system with cognitive process.

In the cognitive process the comparator result is the mean squared error (MSE) $\mathcal{E}$ between the received signal $S_I(n)$ and the output signal $S_O(n)$ and it is used for system adequacy.

For a given input signal $S_I(n)$ composed by network variability feature and memory vector (the memory vector will be described further, but it comprises data obtained from the network's nodes regarding connections successes or failures in addition to route information), the cognitive process uses F_N, V_M and $\mathcal{E}$ to identify possible problems in the network and to act on the output signal $S_O(n)$ with a control vector V_C.

Fig. 6 illustrates the interfaces and connections taking place in the cognitive process regarding internal development of initial CPMod (iCPMod).

The signal $S_I(n)$ derives from the combination of F_N and V_M, while V_C acts on $S_O(n)$. F_N and V_M are indicators of system environment situation. With the analysis of these two parameters, $S_I(n)$ signal can be composed for error calculation. The results applying these indicators are shown in the section VI.

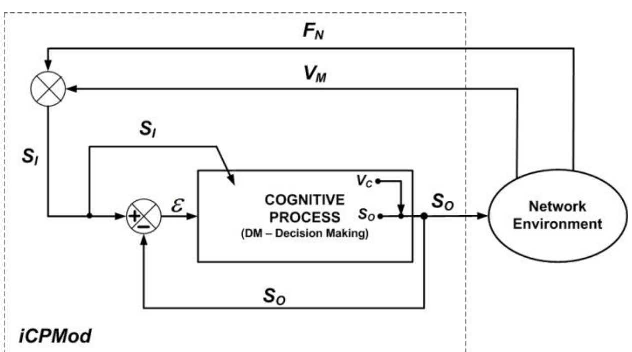

Figure 6. Block diagram of the cognitive process interconnection.

A. Memory Vector

The Memory Vector V_M is in the core of the adaptive cognitive module studied here since it allows the cognitive process to store information regarding network and its nodes, establishing the history of the network situations and memory.

This vector results from the compilation of instant data regarding the monitored environment within time slots after each 5 seconds is recommended. With V_M it is possible to act on the environment using the DMp to generate DM vectors.

$$V_M = (V_S, V_F, V_R) \tag{13}$$

Equation (13) introduces V_M matching to the memory vector composed by: V_S as the vector containing the successes of nodes connections; V_F as the vector containing the failures in the nodes and V_R as a vector of valid routes collected from RT. The components of V_S, V_F and V_R are described briefly in Table I.

Table I. Memory Vector Components

Vector Type	Components and Description		
	Description	***Components***	***Decision***
V_S	Success Vector	Successful end-to-end connection, End-to-end delay, Rate of effectively delivered data.	By the routes with a higher probability of end-to-end packet delivery success.
V_F	Failure Vector	Node disconnected three times in a row, Unresponsive node with reply message (RREP).	Avoiding these nodes for a given period of time.
V_R	Route Vector	Valid routes, Origin node, Destination node, Number of hops to destination.	By the best route according to the valid routes history.

The V_M application for the DM classification will be described ahead and this feature is present on the CPMod as an instance of DM analysis, as in (15).

B. Error Vector

Basically, the error vector $\mathcal{E}$ works with the relative error MSE of the cognitive module measurements input $S_I(n)$ and output $S_O(n)$ signals. $S_I(n)$ is composed by the variability feature and memory vector, and $S_O(n)$ works with control vector (described in D division of this section) and memory vector.

$$\begin{cases} S_I = (F_N, V_M) \rightarrow \zeta S_I = S_I \text{ estimate} \\ S_O = (V_C, V_M) \rightarrow \xi S_O = S_O \text{ real value} \end{cases}$$

$$\mathcal{E}_{(\%)} = \sum_{i=1}^{N} \frac{(\zeta S_I - \xi S_O)_i^2}{S_{O_i}} . 100 \mid 0 \leq \mathcal{E} \leq 100\% \quad (14)$$

The composition of this feature appears in the third parameter for DM analysis, as shown in (15).

C. Decision Making Vector

The procedure for Decision Making process (DMp) is conducted by the cognitive process and it takes into the account of environment variability feature F_N, the memory vector V_M and the error $\mathcal{E}$, seeking for the optimal network configuration and the adjustment of the employed routing protocol.

DM is assigned according to the expected quality of the task and its action is directly related to the Decision Limit (DL). The threshold DL corresponds to a probabilistic value for the adjustment of the system resulting from the combination of F_N, V_M and $\mathcal{E}$ according to the range entrance (DM description) given in Table II.

$$DM = \left(F_N, V_M, \varepsilon\right) \quad (15)$$

Five abstraction levels were adopted based on Quality of Experience (QoE) since they represent the network conditions and allow V_C actuation. DL is calculated using the inverse of the average of DM composition as described in (16).

For the measurement level regarding the network situation, as mentioned before, DL is calculated based on the inverse of DM average, according to (16). The graphical relationship for DL is depicted in Fig. 7.

$$\overline{X}_{(F_N, V_M, \varepsilon)} = \frac{\sum_{i=1}^{N} \left(F_{N_i}, V_{M_i}, \varepsilon_i\right)}{N} \Rightarrow$$

$$\Rightarrow \begin{cases} \overline{X}_{(F_N, V_M, \varepsilon)} \geq 1, \; DL_{(\%)} = \dfrac{1}{\overline{X}_{(F_N, V_M, \varepsilon)}} \times 100 \; (\%) \\ \overline{X}_{(F_N, V_M, \varepsilon)} < 1, \; DL_{(\%)} = 100 \; \% \end{cases} \quad (16)$$

The smaller DL, in percentage, the better is the network condition and, similarly, the greater DL the worst is the network situation, therefore more significant actions should be taken by the CPMod. Furthermore, in order to keep the network with controlled or regular situation, DL level must be under 49.9% range.

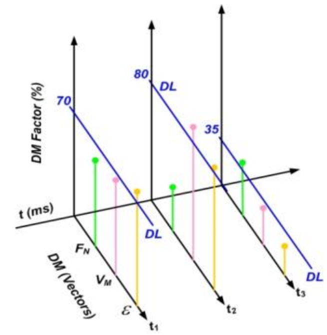

Figure 7. Example of relationship between DL and DM.

Table I. Decision Making According To the Dl Threshold

***DM* Description**	***DL* Configuration**	
	Decision Limit (%) Range [a]	***Action about DM (with DL)***
Urgent	80.0 → 100	Reinitiates the routing table in the nodes. The process evaluates the index (F_N, V_M, ε) originating the urgent situation. V_C is sent to nodes.
Critical	70.0 → 79.9	The process evaluates memory (V_M) and error (ε). Adjusts signal S_O and reads *DL* again. If the situation is maintained a control vector V_C is sent to the node.
Attention	50.0 → 69.9	Adjusts $S_O(n)$ with signal correction according to detected error ε.
Regular	30.0 → 49.9	No need for changes but decisions may be taken to improve QoS. The system is operating normally.
Controlled	0 → 29.9	Network without need for changes. The system is operating in better case.

a. *Measurement level regarding the network situation.*

In other case, the lower is the average in (16) for memory, variability and error vectors, more critical the network condition will be. It can be found in Fig. 7 an example of a

DM data reading composed by F_N, V_M and $\mathcal{E}$ vectors, distributed in times t_1, t_2 and t_3.

D. Control Vector

The iCPMod with DM sends a control data vector in S_O to define the nodes alterations according to the necessity of change by using a control vector V_C. This vector is defined in (17) and it can be adapted to other actions according to the network's needs and to alterations in the settings (such as the insertion of other parameters in the future development).

$$V_C = (Energy, Route, Queue, Memory) \qquad (17)$$

With the settlements of V_C on the output signal $S_O(n)$ it is possible to organize the nodes. The nodes receive and make alterations regarding memory amount sizing through the final control vector to execute tasks. Queues and route protocols (described as RT) can be addressed in this time, correcting the issues related with these features, and energy can be controlled by iCPMod. The cognitive processing allows the system (in a general way) to readjust and, at the same time, execute the network monitoring task. This cognitive system also allows adaptation regarding monitored aspects (e.g., the insertion of a monitoring vector of energy levels at the nodes for a possible control by the cognitive process and a raise in each node's yield according to DL adequacy) as well as DM and V_C routing in $S_O(n)$.

For the optimization of ACS, a development with Fuzzy Logic was realized for CPMod interaction, concerning the DMp with respect to V_C defined aspects in (17).

V. Fuzzy Logic Approach

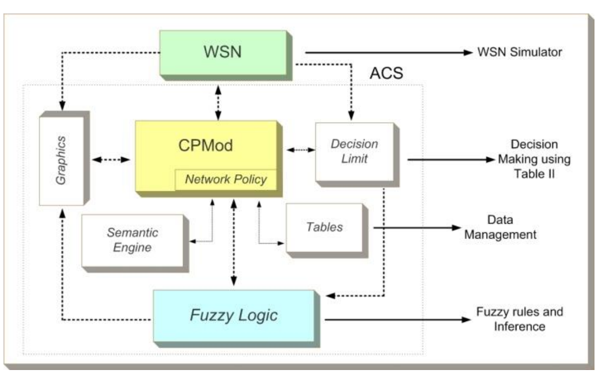

Figure 8. ACS general architecture.

For the Fuzzy Logic implementation, it was defined the ACS architecture as in the Fig. 8. It can be seen that CPMod have interaction with the WSN and Fuzzy Logic blocks. It was used the iCPMod for the algorithm derivation and the CPMod for the block connected into ACS.

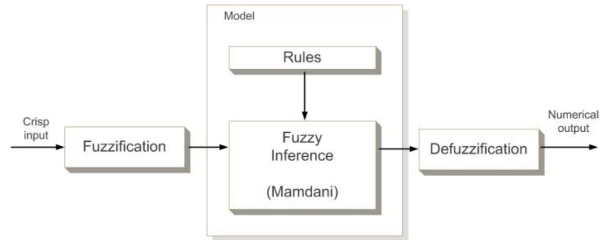

Figure 9. Inference Fuzzy diagram block.

The FIS is intended to be used by the CPMod as an alternative solution for decision making. The inference mechanism aggregates more coherent and fast network analysis using the Fuzzy control and pertinence functions. Considering the provisions of (18), the membership functions mf are shown for the inference and to be applied on the fuzzification and defuzzification steps.

$$A = \{(x, \mu_A(x)) | x \in X\} \text{ and } B = \{(x, \mu_B(x)) | x \in X\}$$

$$\mu_A(x) = mf \Rightarrow \begin{cases} \mu_{A\cap B} = \text{argmin}(\mu_A(x), \mu_B(x)), \forall x \in X \\ \mu_{A\cup B} = \text{argmax}(\mu_A(x), \mu_B(x)), \forall x \in X \end{cases} \qquad (18)$$

On behalf of the membership functions mentioned, the triangular and trapezoidal functions were considered to be part of pertinence functions, according to (19) and (20), respectively.

$$\text{Trim}f_{(x;a,b,c)} = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \qquad (19)$$

The pertinence functions for the fuzzification are presented in Figs. 10-13.

$$\text{Trapm}f_{(x;a,b,c,d)} = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \qquad (20)$$

Equations (22)-(25) assume values from the (16), which relates to five-fold the fuzzy membership degree of *x* in *A*.

$$f_A(x) = (x, T(x), G(x), M(x), X) \qquad (21)$$

From (21), x represents the vector name data, $T_{(x)}$ vector meanings set, $G_{(x)}$ the set of syntactic rules applied to x, $M_{(x)}$ the semantic rule that assigns to each value generated by $G_{(x)}$ Fuzzy set in x and X the universe of discourse that is the X axis.

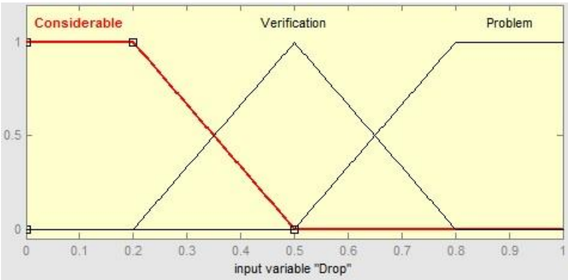

Figure 10. Percentage of input variable drop of packets.

$$f_A(x) = \begin{cases} x\text{: Drop} \\ T(x)\text{: \{Considerable, Verification, Problem\}} \\ X\text{: Number of packets} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \quad (22)$$

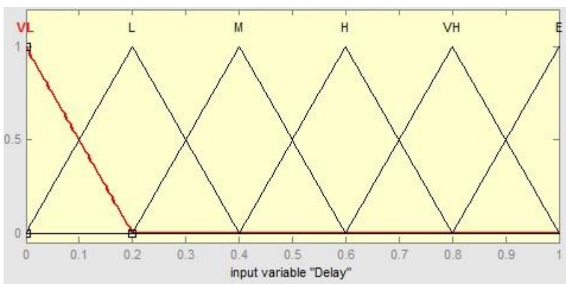

Figure 11. Percentage of input variable delay of packets.

$$f_A(x) = \begin{cases} x\text{: Delay} \\ T(x)\text{: }\{VL, L, M, H, VH, E\} \\ X\text{: Time in miliseconds} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \quad (23)$$

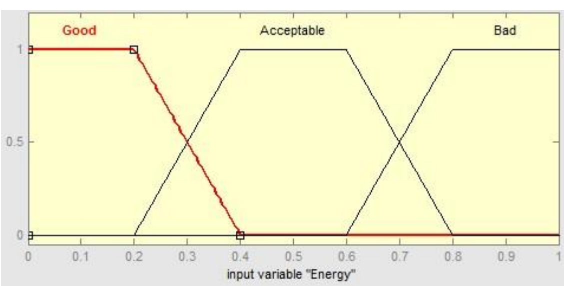

Figure 12. Percentage of input variable energy consumed by nodes.

$$f_A(x) = \begin{cases} x\text{: Energy} \\ T(x)\text{: \{Good, Acceptable, Bad\}} \\ X\text{: Quantity of consumed energy} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \quad (24)$$

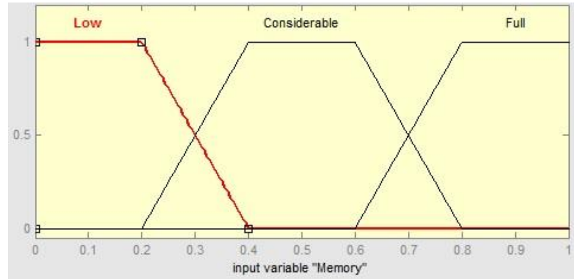

Figure 13. Percentage of input variable memory used by nodes.

$$f_A(x) = \begin{cases} x\text{: Memory} \\ T(x)\text{: \{Low, Considerable, Full\}} \\ X\text{: Quantity of used memory} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \quad (25)$$

For the insertion of Fuzzy Logic into simulation scenario, it was implemented the pertinence functions in MATLAB software. The home screen of Fuzzy Logic is shown in Fig. 14 and represents the initial structure data analysis model in software.

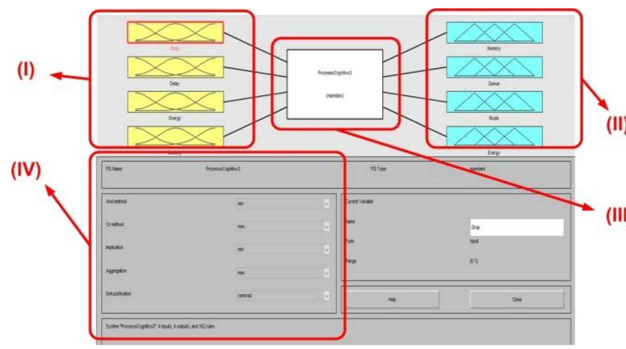

Figure 14. Home screen of Fuzzy Logic in MATLAB.

In screen, were created 4 fuzzification functions to model the crisp data collected from the WSN (I) and 4 corresponding defuzzification pertinence functions to 4 elements present in V_C (II). The item (III) is the use of chosen Mamdani model for inference of the fuzzification membership functions and (IV) is connected to operators and defuzzification method used, which in this case is based on the centroid of defuzzification membership functions.

The pertinence functions for the defuzzification are presented in Figs. 15-18.

Equations (26)-(29) assume values from the (21), which relates to five-fold the fuzzy membership degree of *x* in *A*.

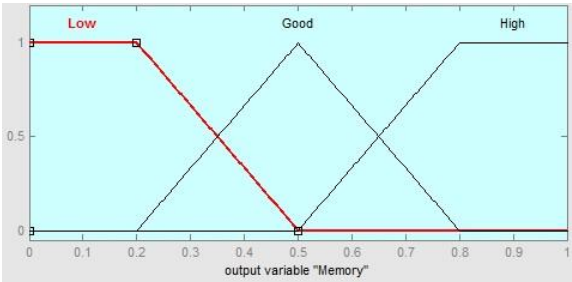

Figure 15. Percentage of output variable Memory.

$$f_A(x) = \begin{cases} x\text{: Memory} \\ T(x)\text{: \{Low, Good, High\}} \\ X\text{: Quantity of available memory} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \tag{26}$$

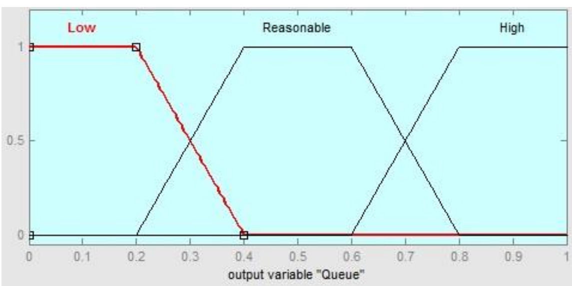

Figure 16. Percentage of output variable Queue.

$$f_A(x) = \begin{cases} x\text{: Queue} \\ T(x)\text{: \{Low, Reasonable, High\}} \\ X\text{: Packet flow} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \tag{27}$$

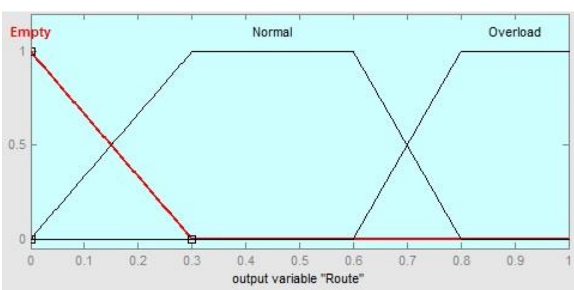

Figure 17. Percentage of output variable Route.

$$f_A(x) = \begin{cases} x\text{: Route} \\ T(x)\text{: \{Empty, Normal, Overload\}} \\ X\text{: Delay in delivering packets} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \tag{28}$$

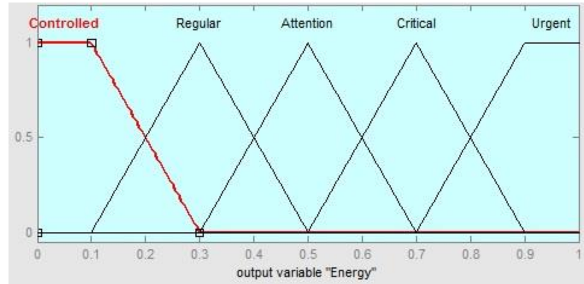

Figure 18. Percentage of output variable Energy.

$$f_A(x) = \begin{cases} x\text{: Energy} \\ T(x)\text{: \{Controlled, Regular, Attention, Critical, Urg} \\ X\text{: Quantity of consumed energy} \\ G(x)\text{: \{rules\}} \\ M(x)\text{: Significate\{rules\}} \end{cases} \tag{29}$$

The Center of Area (CoA) for calculating the centroid is given by (30), which $\mu_{out}(u_i)$ is the area of a membership function and u_i is the centroid position of the individual membership function.

$$u_{defuzzy} = \frac{\sum_{i=1}^{N} u_i \mu_{out}(u_i)}{\sum_{i=1}^{N} \mu_{out}(u_i)} \tag{30}$$

The rules are presented in the MATLAB formatting, representing 162 lines of Fuzzy rules that comprises the overall input-output surfaces present in results.

[System]	[Rules]
Name='ProcessoCognitivo3a'	1 1 1 3, 1 1 2 1 (1) : 1
Type='mamdani'	1 2 1 3, 1 1 2 1 (1) : 1
Version=2.0	1 3 1 3, 1 2 2 2 (1) : 1
NumInputs=4	1 4 1 3, 1 2 2 2 (1) : 1
NumOutputs=4	1 5 1 3, 1 3 2 3 (1) : 1
NumRules=162	1 6 1 3, 1 3 2 4 (1) : 1
AndMethod='min'	1 1 2 3, 2 1 2 1 (1) : 1
OrMethod='max'	1 2 2 3, 2 1 2 1 (1) : 1
ImpMethod='min'	1 3 2 3, 2 2 2 2 (1) : 1
AggMethod='max'	1 4 2 3, 2 2 2 2 (1) : 1
DefuzzMethod='centroid'	1 5 2 3, 2 3 2 3 (1) : 1
	1 6 2 3, 2 3 2 4 (1) : 1
[Input1]	1 1 3 3, 3 1 2 1 (1) : 1
Name='Drop'	1 2 3 3, 3 1 2 1 (1) : 1
Range=[0 1]	1 3 3 3, 3 2 2 2 (1) : 1
NumMFs=3	1 4 3 3, 3 2 2 2 (1) : 1
MF1='Considerable':'trapmf',[0 0	1 5 3 3, 3 3 2 3 (1) : 1
0.2 0.5]	1 6 3 3, 3 3 2 4 (1) : 1
MF2='Verification':'trimf',[0.2	1 1 2 2, 1 1 2 1 (1) : 1
0.5 0.8]	1 2 2 2, 1 1 2 1 (1) : 1

```
MF3='Problem':'trapmf',[0.5 0.8 1 1]

[Input2]
Name='Delay'
Range=[0 1]
NumMFs=6
MF1='VL':'trimf',[0 0 0.2]
MF2='L':'trimf',[0 0.2 0.4]
MF3='M':'trimf',[0.2 0.4 0.6]
MF4='H':'trimf',[0.4 0.6 0.8]
MF5='VH':'trimf',[0.6 0.8 1]
MF6='E':'trimf',[0.8 1 1]

[Input3]
Name='Energy'
Range=[0 1]
NumMFs=3
MF1='Good':'trapmf',[0 0 0.2 0.4]
MF2='Acceptable':'trapmf',[0.2 0.4 0.6 0.8]
MF3='Bad':'trapmf',[0.6 0.8 1 1]

[Input4]
Name='Memory'
Range=[0 1]
NumMFs=3
MF1='Low':'trapmf',[0 0 0.2 0.4]
MF2='Considerable':'trapmf',[0.2 0.4 0.6 0.8]
MF3='Full':'trapmf',[0.6 0.8 1 1]

[Output1]
Name='Memory'
Range=[0 1]
NumMFs=3
MF1='Low':'trapmf',[0 0 0.2 0.5]
MF2='Good':'trimf',[0.2 0.5 0.8]
MF3='High':'trapmf',[0.5 0.8 1 1]

[Output2]
Name='Queue'
Range=[0 1]
NumMFs=3
MF1='Low':'trapmf',[0 0 0.2 0.4]
MF2='Reasonable':'trapmf',[0.2 0.4 0.6 0.8]
MF3='High':'trapmf',[0.6 0.8 1 1]

[Output3]
Name='Route'
Range=[0 1]

1 3 2 2, 1 2 2 2 (1) : 1
1 4 2 2, 1 2 2 2 (1) : 1
1 5 2 2, 1 2 2 3 (1) : 1
1 6 2 2, 1 2 2 4 (1) : 1
1 1 1 2, 1 1 2 1 (1) : 1
1 2 1 2, 1 1 2 1 (1) : 1
1 3 1 2, 1 1 2 2 (1) : 1
1 4 1 2, 1 1 2 2 (1) : 1
1 5 1 2, 1 2 2 3 (1) : 1
1 6 1 2, 1 2 2 4 (1) : 1
1 1 3 2, 3 2 2 1 (1) : 1
1 2 3 2, 3 2 2 1 (1) : 1
1 3 3 2, 3 2 2 2 (1) : 1
1 4 3 2, 3 2 2 2 (1) : 1
1 5 3 2, 3 3 2 3 (1) : 1
1 6 3 2, 3 3 2 4 (1) : 1
1 1 1 1, 1 1 1 1 (1) : 1
1 2 1 1, 1 1 1 1 (1) : 1
1 3 1 1, 1 2 1 2 (1) : 1
[Rules]
1 4 1 1, 1 2 1 2 (1) : 1
1 5 1 1, 1 2 1 3 (1) : 1
1 6 1 1, 1 2 1 4 (1) : 1
1 1 2 1, 2 1 2 1 (1) : 1
1 2 2 1, 2 1 2 1 (1) : 1
1 3 2 1, 2 2 2 2 (1) : 1
1 4 2 1, 2 2 2 2 (1) : 1
1 5 2 1, 2 2 2 3 (1) : 1
1 6 2 1, 2 2 2 4 (1) : 1
1 1 3 1, 3 1 1 1 (1) : 1
1 2 3 1, 3 2 1 1 (1) : 1
1 3 3 1, 3 2 2 2 (1) : 1
1 4 3 1, 3 2 2 2 (1) : 1
1 5 3 1, 3 2 2 3 (1) : 1
3 6 2 1, 2 3 2 5 (1) : 1
3 1 2 2, 2 2 2 3 (1) : 1
3 2 2 2, 2 2 2 3 (1) : 1
3 3 2 2, 2 2 2 4 (1) : 1
3 4 2 2, 2 2 2 4 (1) : 1
3 5 2 2, 3 2 2 5 (1) : 1
3 6 2 2, 3 2 2 5 (1) : 1
3 1 2 3, 2 2 2 3 (1) : 1
3 2 2 3, 2 2 2 3 (1) : 1
3 3 2 3, 3 3 2 4 (1) : 1
3 4 2 3, 3 3 2 4 (1) : 1
3 5 2 3, 3 3 3 5 (1) : 1
3 6 2 3, 3 3 3 5 (1) : 1
3 1 3 1, 2 2 2 4 (1) : 1
3 2 3 1, 2 2 2 4 (1) : 1
3 3 3 1, 2 3 2 4 (1) : 1
3 4 3 1, 2 3 2 4 (1) : 1
3 5 3 1, 3 3 2 5 (1) : 1
3 6 3 1, 3 3 2 5 (1) : 1
3 1 3 2, 3 2 2 4 (1) : 1
3 2 3 2, 3 2 2 4 (1) : 1
```

```
NumMFs=3
MF1='Empty':'trimf',[0 0 0.3]
MF2='Normal':'trapmf',[0 0.3 0.6 0.8]
MF3='Overload':'trapmf',[0.6 0.8 1 1]

[Output4]
Name='Energy'
Range=[0 1]
NumMFs=5
MF1='Controlled':'trapmf',[0 0 0.1 0.3]
MF2='Regular':'trimf',[0.1 0.3 0.5]
MF3='Attention':'trimf',[0.3 0.5 0.7]
MF4='Critical':'trimf',[0.5 0.7 0.9]
MF5='Urgent':'trapmf',[0.7 0.9 1 1]

[Rules]
2 3 1 2, 1 2 2 3 (1) : 1
2 4 1 2, 1 2 2 3 (1) : 1
2 5 1 2, 1 2 2 4 (1) : 1
2 6 1 2, 1 2 2 4 (1) : 1
2 1 1 3, 2 1 1 3 (1) : 1
2 2 1 3, 2 1 2 3 (1) : 1
2 3 1 3, 2 2 2 3 (1) : 1
2 4 1 3, 2 2 2 3 (1) : 1
2 5 1 3, 2 2 2 4 (1) : 1
2 6 1 3, 2 2 2 4 (1) : 1
2 1 2 1, 1 1 1 3 (1) : 1
2 2 2 1, 1 1 1 3 (1) : 1
2 3 2 1, 1 1 2 3 (1) : 1
2 4 2 1, 1 1 2 3 (1) : 1
2 5 2 1, 2 2 2 4 (1) : 1
2 6 2 1, 2 2 2 4 (1) : 1
2 1 3 1, 2 1 2 3 (1) : 1
2 2 3 1, 2 1 2 3 (1) : 1
2 3 3 1, 3 2 2 3 (1) : 1
2 4 3 1, 3 2 2 3 (1) : 1
2 5 3 1, 3 3 2 4 (1) : 1
2 6 3 1, 3 3 2 4 (1) : 1
2 1 2 2, 2 1 2 3 (1) : 1
2 2 2 2, 2 1 2 3 (1) : 1
2 3 2 2, 2 2 2 3 (1) : 1
2 4 2 2, 2 2 2 3 (1) : 1
2 5 2 2, 3 2 2 4 (1) : 1
2 6 2 2, 3 2 2 4 (1) : 1
2 1 2 3, 2 1 1 3 (1) : 1
2 2 2 3, 2 1 2 3 (1) : 1
2 3 2 3, 2 2 2 3 (1) : 1
2 4 2 3, 2 2 2 3 (1) : 1

3 3 3 2, 3 2 2 4 (1) : 1
3 4 3 2, 3 2 2 4 (1) : 1
3 5 3 2, 3 3 3 5 (1) : 1
3 2 1 2, 1 3 2 3 (1) : 1
3 3 1 2, 2 3 2 3 (1) : 1
3 4 1 2, 2 3 2 3 (1) : 1
3 5 1 2, 3 3 2 4 (1) : 1
3 6 1 2, 3 3 2 4 (1) : 1
3 1 1 3, 2 3 2 3 (1) : 1
3 2 1 3, 2 3 2 3 (1) : 1
3 3 1 3, 3 2 2 3 (1) : 1
3 4 1 3, 3 2 2 3 (1) : 1
3 5 1 3, 3 3 2 4 (1) : 1
3 6 1 3, 3 3 2 4 (1) : 1
3 1 2 1, 1 2 2 3 (1) : 1
3 2 2 1, 1 2 2 3 (1) : 1
3 3 2 1, 2 2 2 4 (1) : 1
2 4 3 3, 3 2 2 3 (1) : 1
2 5 3 3, 3 3 2 4 (1) : 1
2 6 3 3, 3 3 2 4 (1) : 1
3 1 1 1, 1 3 1 3 (1) : 1
3 2 1 1, 1 3 1 3 (1) : 1
3 3 1 1, 1 3 2 3 (1) : 1
3 4 2 1, 2 2 2 4 (1) : 1
3 5 2 1, 2 3 2 5 (1) : 1
3 6 3 2, 3 3 3 5 (1) : 1
3 1 3 3, 3 3 2 4 (1) : 1
3 2 3 3, 3 3 2 4 (1) : 1
3 3 3 3, 3 3 2 4 (1) : 1
3 4 3 3, 3 3 3 5 (1) : 1
2 5 2 3, 3 3 2 4 (1) : 1
2 6 2 3, 3 3 2 4 (1) : 1
2 1 3 2, 2 2 2 3 (1) : 1
2 2 3 2, 2 2 2 3 (1) : 1
2 3 3 2, 2 3 2 3 (1) : 1
2 4 3 2, 2 3 2 3 (1) : 1
2 5 3 2, 3 3 2 4 (1) : 1
2 6 3 2, 3 3 2 4 (1) : 1
2 1 3 3, 3 1 2 3 (1) : 1
2 2 3 3, 3 1 2 3 (1) : 1
2 3 3 3, 3 2 2 3 (1) : 1
3 4 1 1, 1 3 2 3 (1) : 1
3 5 1 1, 1 3 2 4 (1) : 1
3 6 1 1, 1 3 3 4 (1) : 1
3 1 1 2, 1 3 2 3 (1) : 1
3 5 3 3, 3 3 3 5 (1) : 1
3 6 3 3, 3 3 3 5 (1) : 1
1 6 3 1, 3 2 2 4 (1) : 1
2 1 1 1, 1 2 1 3 (1) : 1
2 2 1 1, 1 2 1 3 (1) : 1
2 3 1 1, 1 2 1 3 (1) : 1
2 4 1 1, 1 2 1 3 (1) : 1
2 5 1 1, 1 3 2 4 (1) : 1
2 6 1 1, 1 3 2 4 (1) : 1
2 1 1 2, 1 1 2 3 (1) : 1
```

DOI- 10.18486/ijcsnt.2016.5.1.01
ISSN-2053-6283

	2 2 1 2, 1 1 2 3 (1) : 1

The weight of all rules is the same, so the system can be in the balanced state when inference runs over data collected from WSN, in support of providing the results to CPMod realize the DM.

VI. Results and Comments

Added either as vectors or only as another variable to the network data reading, the mechanism gains robustness but tends to lose processing response time since the cognitive process needs to conduct a larger number of analysis with the insertion of new monitoring elements, however this case occur when the system is running in the initial stage.

In the study scenario were used 0-100 sensor nodes with possibility to each node establishing 0-9 connections, and each node has antenna radius variability of 3-5 meters and displacement of 1-10 meters in random trajectory. For simplicity, was adopted a basic antenna model. The data collected from nodes were generated by *Prowler* simulation over MATLAB.

As mentioned before, the CPMod operates according with network/nodes situation. Nevertheless, the procedure to execute data analysis (described in previous section) is based on Table II and it is explained as follows.

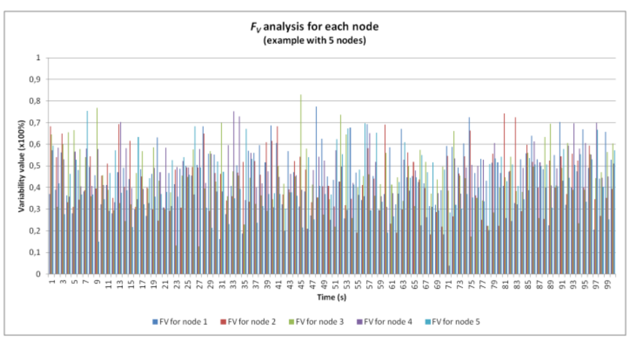

Figure 19. Graphical analysis concerning F_V variability for 5 nodes.

The graphs were obtained from the random nodes statistics samples, applying the expressions mentioned in previous sections.

From the Fig. 19, it is possible to identify the relation between Table II and variability values in the graph. As can be seen, the most of data collected about variability is in the range of 20% to 50%, so according to the Table II, these 5 nodes are operating normally and the situation is controlled and regular. DL is present here according to each peak signal.

There are yet a big amount of circumstances demanding attention and few cases where nodes require action with V_C intervention, because they reach the critical level. In this sampling, at approximately 45 seconds an urgent condition occurred. Therefore, in this case, a V_C was sent to the respective node (node 3) to perform the adjustment task (queue scheme modification).

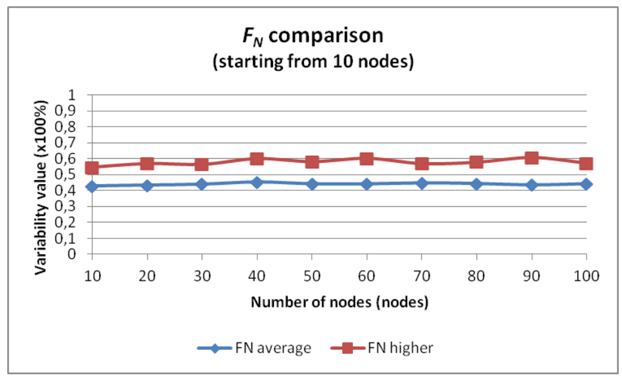

Figure 20. Graphic representing the F_N comparison for variability.

This graph shows us the occurrence of stable condition in the system, this means that the nodes are operating in approximately 45% of variability, according to the position, antenna radius, possible links and number of nodes. The higher situations are appearing over 50% and less than 70%, so in the attention range according to Table II.

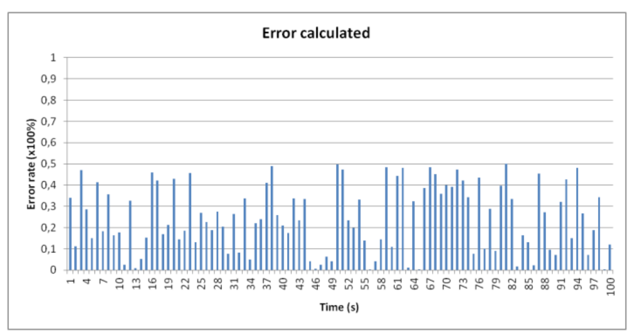

Figure 21. Error rate concerning S_I and S_O.

In the Fig. 21, it can be seen that the error calculated over network data resulted in the Controlled and Regular levels of Table II.

DOI- 10.18486/ijcsnt.2016.5.1.01
ISSN-2053-6283

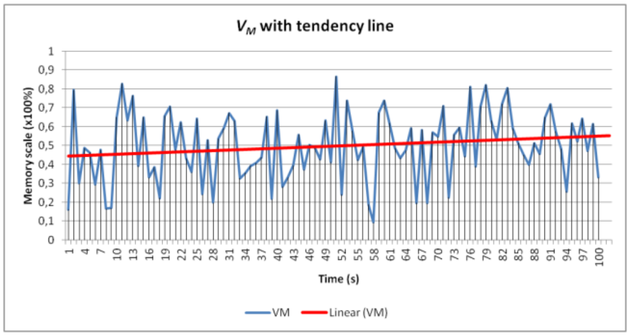

Figure 22. V_M with tendency line according to the memory status.

From the Fig. 22, is understood that with the network time running, the memory usage tends to be higher according to the tendency line. It can be seen that exists 6 peaks surrounding 80% memory usage, and in that cases V_C is sent to the respective nodes to try corrections in memory, queue or route tables.

The idea that this mechanism will overload the network with unnecessary data packets seems reasonable but, when the sensors/actuators are synchronized within the nodes and respective features after the initial setting stage, the system pre-stabilizes and acts according to the alterations done by the CPMod.

Using F_N, V_M and $\mathcal{E}$ parameters, the DMp can be executed according to the Table II to regularize the nodes situations regarding memory, queue, energy and route adjustments. DL is obtained too from these parameters and tends to be near to their higher values, concerning the average relation described in (16).

The real strictly direct correlation between CPMod process and QoS assurance is not depicted from the above graphs, but it will be treated in a sequence work.

With respect to the Fuzzy Logic applied to DM in ACS architecture, the surfaces of Figs. 23-32 present the specific analysis surrounding each input-output element, according to (X, Y, Z) coordinates. In the graphs, X and Y represent input elements and Z represents the output of Fig. 9.

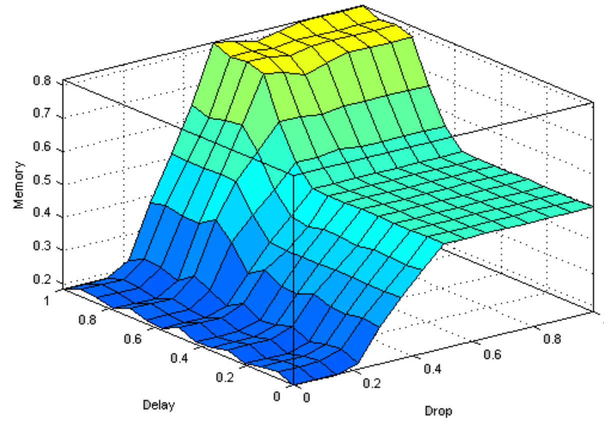

Figure 23. Surface of Drop x Delay x Memory.

From the Fig. 23, it can be seen that with growing of packets dropped in network in conjunction with delay to transmit data end-to-end indicates an elevated necessity for memory available in nodes, or in other words, the memory will be a bottleneck for the network soon.

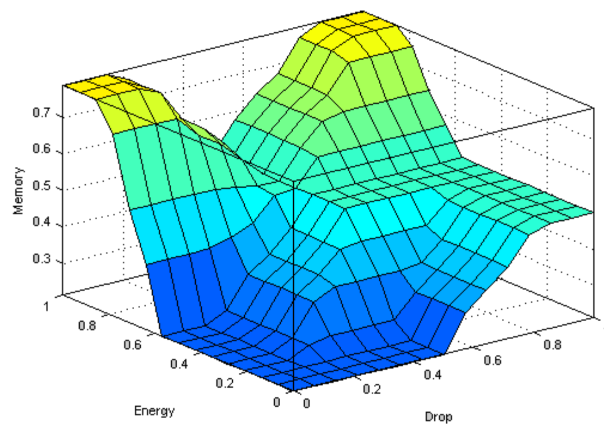

Figure 24. Surface of Drop x Energy x Memory.

The Fig. 24 shows that even with low drop rate of packets, if energy consumed by nodes raises over 60%, the memory is also committed.

DOI- 10.18486/ijcsnt.2016.5.1.01
ISSN-2053-6283

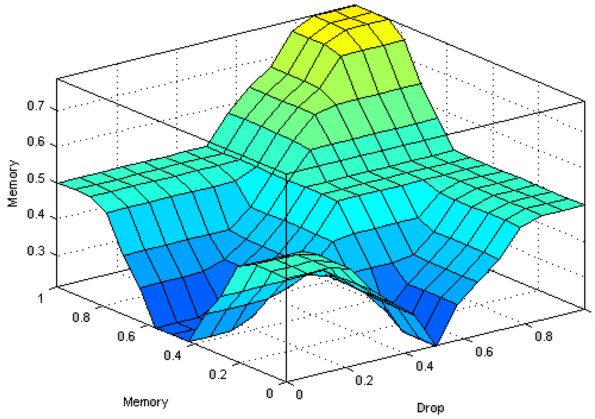

Figure 25. Surface of Drop x Memory x Memory.

In the Fig. 25, when the memory usage by nodes according to the packets dropped raises, the memory available turns a bottleneck too.

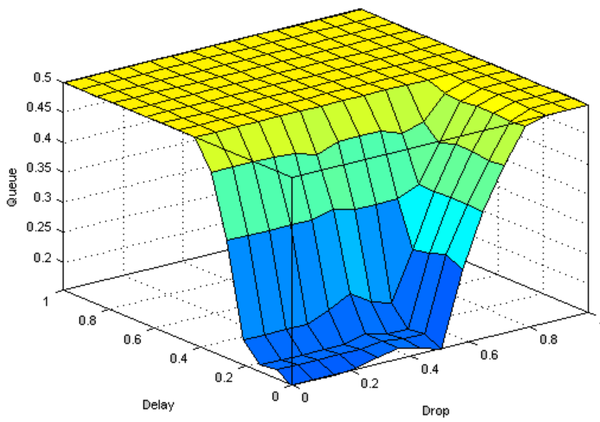

Figure 26. Surface of Drop x Delay x Queue.

It can be seen in the above graph, that the relation between drop and delay does not affect seriously the network in the queue issue.

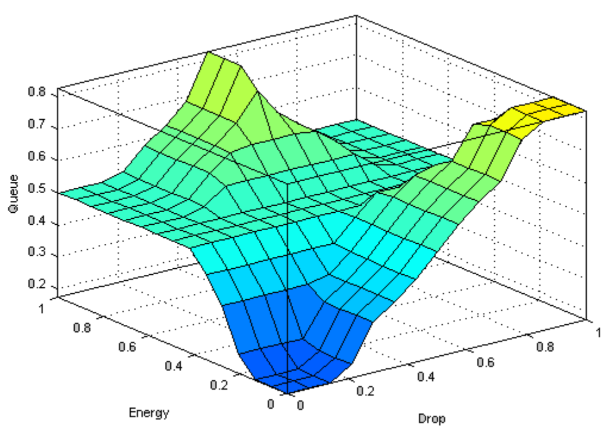

Figure 27. Surface of Drop x Energy x Queue.

Even with low energy usage by nodes, if drop raises the queue is affected directly.

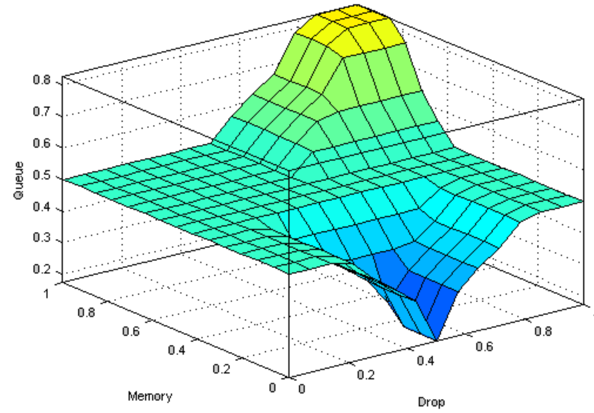

Figure 28. Surface of Drop x Memory x Queue.

The Fig. 27 indicates that with a high rate of packets dropped in network, the Critical and Urgent levels of Table II are meeting, forcing the CPMod to act on Link layer.

From the Fig. 28, with high usage of memory and packets dropped excessively, the queue tends to raise fast over 50% of both.

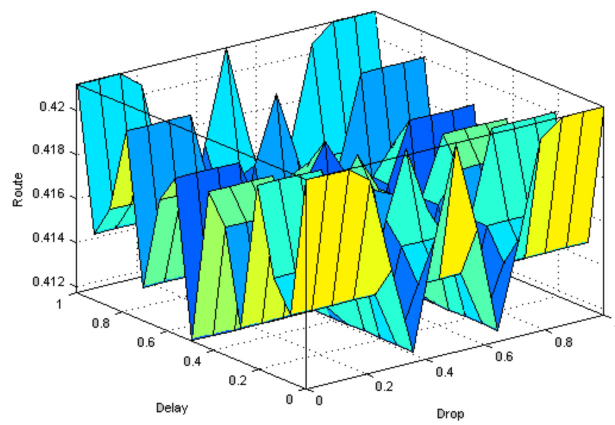

Figure 29. Surface of Drop x Delay x Route.

From the Fig. 29, can be verified that the route or routing scheme is oscillating in the range of 41 to 43% for the relation between rates of packets dropped and delay end-to-end. This means the route protocol is affected by drop in three instances and delay can occur due to the links between nodes reestablishments, nodes position modifying, and other variable features. The results for (Drop x Energy) and (Drop x Memory) related to Route do not presented significant implications to changes in routing protocols on Network layer.

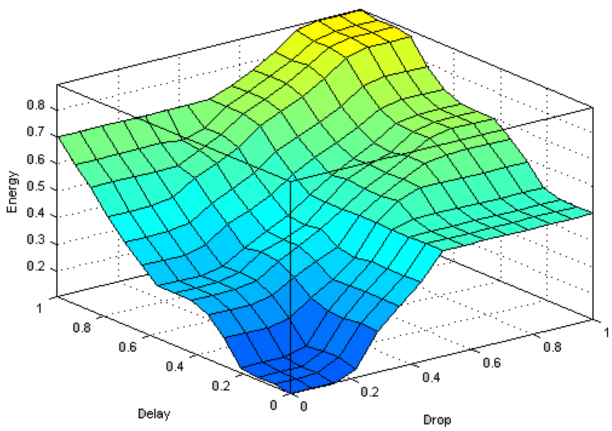

Figure 30. Surface of Drop x Delay x Energy.

Finally, according to Energy output, the drop and delay ongoing affect gently the energy as they raises. With high drop and delay, the energy consumed by nodes is too high, because more processing and routing of packets are needed.

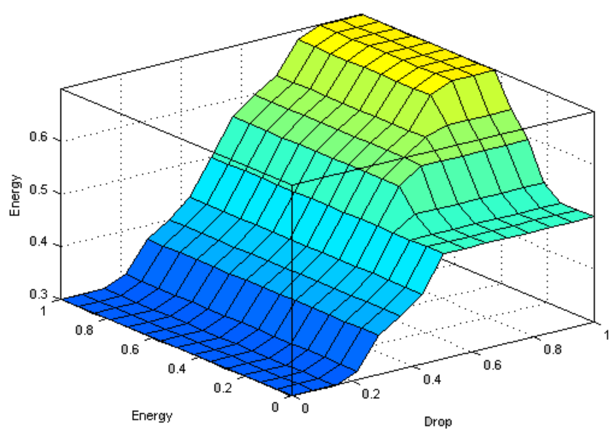

Figure 31. Surface of Drop x Energy x Energy.

The same occurs when Energy > 20% and the drop of packets raises too, until the queue does not support more packets and all of new received packets falls into dropped packets by nodes.

In Fig. 32 is shown that the evolution of memory used by nodes does not affect directly the Energy output; however the combination of memory usage with drop of packets results in an energy spent by nodes. The energy spent is directly associated with the growing of rate of data packets dropped.

As can be seen, the elements that comprises the network features do not work alone and their interactions allows the control of WSN, trying to keep the network balanced into rules defined in Fuzzy Logic and in the DL of Table II.

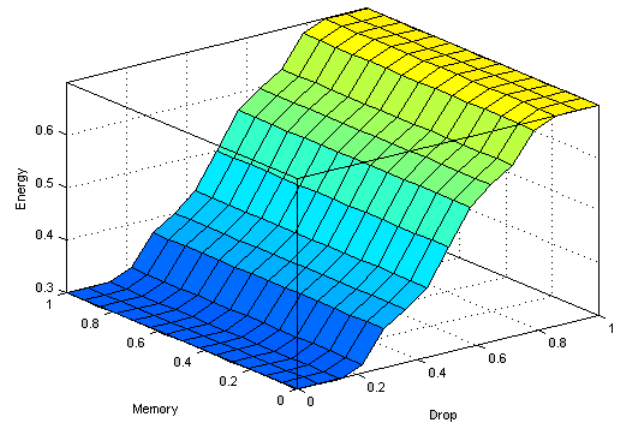

Figure 32. Surface of Drop x Memory x Energy.

The implemented FIS block permits the increase of elements, which can be added for future observation and control of network. As well as the rules and pertinence functions applied for fuzzification and defuzzification, that can be modified and increased.

VII. Conclusion

The cognitive process adds intelligence aspects to the Wireless Sensor Networks control mechanism, attaching memory, history and decision making. These features reflect some inherent aspects of human beings that represent a strong tool to the improvement of QoS in WSN. With the use of measurement, monitoring and analysis mechanisms by nodes where the cognitive processes occur, it is possible to improve the network metrics with the actuation of Decision Making process. The DL verification represents a relation of the vectors that integrates the DM to provide the system with the action rules sent by V_C. This procedure guarantees the establishment of actions directly in the nodes and, therefore, an improvement in the performance regarding to the data packet delivery. In fact, with CPMod insertion into sensor nodes, more internal tasks in processing are needed, but this control results in a great gain concerning error corrections in the nodes mechanisms, covering queue, routing protocols adjustments, energy consumed by nodes and in the node memory usage for sensor applications. The use of Fuzzy Logic to identify the network situation turns the DMp more robust and dynamic. The V_C with Fuzzy Logic is a strong tool to help the CPMod in the decision making, setting up the CPMod to apply changes in multiple layers of the network.

For future researches some issues should be treated as follows: more nodes can be inserted into DM analysis, the feature concerning energy usage can be explored expanding F_N, V_M and V_C, another control scheme can be defined as complementary solution for DMp, add more elements in Fuzzy block for observation and control of network, the rules and pertinence functions can be modified and increased, and comparisons with other WSN protocols concerning control and metrics improvement in the network point of view.

Acknowledgment

M.S.W. thanks all his coworkers and friends and especially his wife for participating in the accomplishment of this work that is a part of his PhD thesis.

This work was supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) under Grant 307633/2011-0 and by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under Grant 2012/24789-0.

References

[1] LI, J. et al., "Efficient Traffic Aware Multipath Routing Algorithm in Cognitive Networks", Fifth International Conference on Genetic and Evolutionary Computing (ICGEC), Xiamen, **(2011)** September.

[2] ALCARAZ, C. et al., "Wireless Sensor Networks and the Internet of Things: Do We Need a Complete Integration?", Computer Science Department of University of Malaga, 1st International Workshop on the Security of the Internet of Things (SecIoT'10), Tokyo, Japan, **(2010)** December.

[3] EZREIK, A. and GHERYANI, A., "Design and Simulation of Wireless Network using NS-2", 2nd International Conference on Computer Science and Information Technology (ICCSIT'2012), Singapore, **(2012)** April.

[4] INTANAGONWIWAT, S.; GOVINDAN, R. and ESTRIN, D., "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", Proceedings of the 6th annual International Conference on Mobile Computing and Networking (MobiCom'2000), Boston, USA, **(2000)** August 6-11.

[5] SINGH, N.; LAL DUA, R. and MATHUR, V., "Network Simulator NS2-2.35", International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE), vol. 2, issue 5, **(2012)** May, pp. 224-228.

[6] SOMOV, A.; DUPONT, C. and GIAFFREDA, R., "Supporting Smart-city Mobility with Cognitive Internet of Things", Conference of Future Network & MobileSummit'2013, International Information Management Corporation (IIMC), Portugal, Lisboa, **(2013)** July.

[7] VIJAY, G.; BDIRA, E. B. A. and IBNKAHLA, M., "Cognition in Wireless Sensor Networks: A Perspective", IEEE Sensors Journal, vol. 11, n. 3, **(2011)** March, pp. 582-592.

[8] WANG, Q., "Traffic Analysis & Modeling in Wireless Sensor Networks and Their Applications on Network Optimization and Anomaly Detection", Macrothink Institute, Network Protocols and Algorithms, vol. 2, n. 1, Sundsvall, Sweden, **(2010)**, pp. 74-92.

[9] WEI YE et al., "Evaluating Control Strategies for Wireless-Networked Robots Using an Integrated Robot and Network Simulation", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001), Seoul, Korea, **(2001)** May.

[10] ZHANG, N.; GUAN, J. and XU, C., "Traffic Prediction Model for Cognitive Networks", Proceedings of International Conference on Advanced Intelligence and Awareness Internet (AIAI'2011), Shenzhen, China, **(2011)** October.

[11] ZHANG, M. et al., "Cognitive Internet of Things: Concepts and Application Example", International Journal of Computer Science Issues (IJCSI), vol. 9, issue 6, n. 3, November, **(2012)**, pp. 151-158.

[12] GS Tomar, Tripti Sharma, Brijesh Kumar, "Fuzzy based Ant colony Optimization Approach for Wireless Sensor Network", Wireless Personal Communication, Vol.84, No.1, pp.361-375, May 2015.