# An Approach for Network Selection Based on Artificial Neural Networks in Heterogeneous Wireless Environments

Pablo Rocha Moreira, Claudio de Castro Monteiro, Mauro Henrique Lima de Boni and Fabiana Ferreira Cardoso
Department of Informatics
Federal Institute for Education, Science and Technology of Tocantins (IFTO)
Palmas, Brazil
prm.gredes@gmail.com, {ccm, mauro, fabiana}@ifto.edu.br

***Abstract*—The concept of Always Best Connected (ABC) is in great demand today, and algorithms that come on mobile devices cannot provide that. This paper describes a proposal for a Neural Network (NN) based network selection mechanism, which is intended to be a piece to be integrated into a handover system environment. Our neural network had a good behavior after trained; it has a good generalization capability for new data presented to it.**

***Keywords-Handover; Network Selection; Neural Networks; Quality of Service.***

## I. INTRODUCTION

Internet has grown in ways without precedent, and we can realize that there are many kinds of services available to the public, such as: video/audio streaming, videoconferencing, social networks and much more. As of April 2014, the number of end systems connected to the Internet was predicted to be almost 3 billion by the end of 2014, and the number of mobile-broadband will reach 2.3 billion [1]. Therefore, in today's world everyone desires a ubiquitous connection, they want to be best connected anywhere and anytime [2], and consume various types of services.

Many research groups around the world have studied network selection, and yet there is no final solution to solve this problem. Network selection is the mechanism that works as a trigger to start a migration from a network A to a network B. WLAN-first scheme [3] is the default network selection mechanism in today's mobile devices. It prefers Wireless Local Area Network (WLAN) networks over other technologies, for instance, Universal Mobile Telecommunications System (UMTS), in which, when the migration is between WLAN networks, the decision is based only on signal level. Therefore, a better feasible solution is needed to solve this problem.

This research aims to suggest a solution for network selection. The goal is for this solution to have low computing cost and guarantee an efficient mobility management. To accomplish that, we have chosen a client-server architecture (see Figure 1), which gathers most of processing on the server side, and the classification process uses an approach based on neural networks.

This paper is organized as follows. Section 2 presents the background. Section 3 gives an overview on our proposed mechanism. In section 4, we present the materials and methods utilized to perform this research. Section 5 presents the results. Section 6 concludes this paper and presents future work.

## II. BACKGROUND

Network Selection is the object of study in several research groups, and it still lacks a good solution in today's mobile devices. To solve the network selection problem, support techniques and strategies for decision are found in the literature, such as fuzzy logic, genetic algorithms and Multiple Attribute Decision Making (MADM) methods [4].

In [5], the Enhanced Power-Friendly Access Network Selection (E-PoFANS) mechanism is presented, which is proposed to be integrated in user devices to perform an energy-efficient network selection for multimedia services. In [6], the use of MADM methods in network selection is discussed, and they conclude that Analytic Network Process (ANP) combined with Mahalanobis is the best match for weighting algorithm in order to select the best access network.

Most of related works are client-side solutions, and it is known that mobile devices lack good computational resources, e.g., memory, processing power, battery autonomy. Users want to benefit from ABC [2], but they also want the best performance of their devices. Thus, a client-side based solution is not the best choice due to the impact it will cause in Quality of Experience (QoE). Our proposal is a hybrid solution; the majority of processing is done on the server side.

Neural Networks were the chosen technique in our approach because they are utilized to solve many difficult tasks, such as: speech recognition, image processing, autonomous systems, etc. According to [7], the biggest neural networks virtue is to be capable of learning from input data with or without a supervisor. This capability has turned the use of this kind of algorithm more frequent. Neural Networks is the machine learning technique that has grown and evolved much more than others techniques since its reappearing in the early 1980's.

## III. PROPOSAL

The core of our proposal is a NN-based classifier; the topology of the proposed NN is displayed in Figure 2. Input neurons handle 6 inputs: jitter, delay, packet loss, signal

 ISBN: 978-1-61208-433-6 

level, throughput, and cost (monetary), and they give different weights (defined during training) to each one of them. Output neurons compete against each other, and the winner neuron defines which class the given network fits the best.

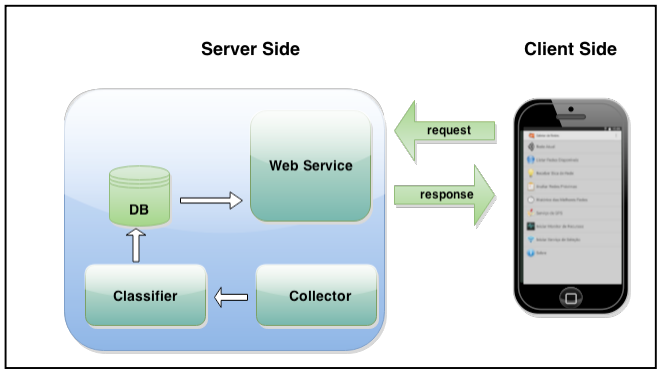


Figure 1. System Architecture.

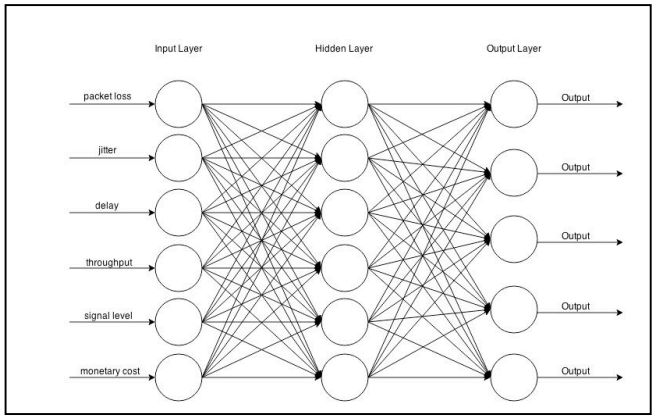


Figure 2. Topology of the proposed NN.

Table I shows the possible classes in which a network can be classified. Depending on which output neuron wins the competition, the network is classified as bad, poor, fair, good or excellent. It is important to highlight that a single input (attribute) cannot define the class that a network will be classified in, but rather it is the sum of all attributes and their weights that will do so.

TABLE I. MOS/CLASSSIFICATION

| Stars/Score | Classification |
|---|---|
| 5 | Excellent |
| 4 | Good |
| 3 | Fair |
| 2 | Poor |
| 1 | Bad |

The Mean opinion Score (MOS) [8] has been used for decades in telephony networks to measure the human user's view of the quality of the network. Therefore, we found it appropriated to use this notation to classify the quality of the networks. In [4], [9], [10] studies were made about Quality of Service (QoS) and quality of networks, and based on that information, we present Table 2. This table relates interval values of each QoS parameter with MOS, and it was the base to generate the dataset utilized to train the NN. Also, the validation dataset was obtained from traces collected in [9].

TABLE II. TABLE TYPE STYLES

| Class | Jitter | Delay | Packet Loss | Signal Level | Throughput | Cost |
|---|---|---|---|---|---|---|
| Excellent | 0 - 1 | 0 - 0.5 | 0 - 0 | 60 - 100 | 5000 - 15000 | 0 - 5 |
| Good | 1 - 3 | 0.5 - 1 | 0 - 1 | 40 - 60 | 1500 - 5000 | 1 - 6 |
| Fair | 3 - 5 | 1 - 3 | 1 - 3 | 30 - 40 | 800 - 1500 | 2 - 7 |
| Poor | 5 - 10 | 3 - 10 | 3 - 10 | 20 - 30 | 500- 800 | 3 - 8 |
| Bad | 10 - 15 | 10 - 15 | 10 - 15 | 0 - 20 | 0 – 500 | 4 - 9 |

We can split up the project of the NN in four stages: getting a dataset, choosing the NN architecture, training the network, and last validating the generalization capability of it. We will be covering each stage in the next paragraphs.

Getting a dataset: In order to get a dataset to train our NN with we needed a model to define what a excellent, good, fair, poor or bad network is. The dataset used to train the neural network has 10,000 samples and was created based on Table 2. This table was constructed based on previous works performed in our research group by other colleagues.

Choosing NN architecture: this NN is for a classification task; it was implemented making use of the Python library PyBrain [11]; it is a fully connected Feedfoward NN with 6 neurons in the input layer, 6 neurons in the hidden layer, and 5 neurons in the output layer, one for each class, as can be seen in Figure 2. This was the topology that best worked for us. We tried other topologies and the NN was not converging. Each synapse has a different weigh, which was defined during the training sessions.

Training the NN: the NN was trained over the dataset for up to 1,000 epochs and the best version of the NN was saved for future use in a Extensible Markup Language (XML) file. The NN converged with about 200 epochs, with about 3% of error. In Algorithm 1, you can see the pseudo code for the training algorithm.

**Algorithm 1:** NN Training Algorithm

```
1. variables:
2. dset = readFromFile("dset.csv")
3. nn = buildNetwork()
4. trainer = BackPropTrainer(nn, dset)
5. p_error = 100.0
6. Begin
7.   for i = 0 to 1000 do;
8.      error = trainer.train();
```

```
9.       if error < p_error then
10.         nn.save("nn.xml");
11.      end if
12.    end for
13. End
```

Validating NN: in order to make sure our NN was behaving as expected, we tested it on a dataset from a real production environment, and analyzed the results.

**Algorithm 2:** Collector Algorithm

```
1. variables:
2. nn = readNetworkFromFile("nn.xml")
3. Begin
4.    while true do;
5.       networks = getNetworks()
6.       for net in networks do
7.          connectTo(net)
8.          qos = calculateQoS(net)
9.          stars = nn.activate(qos)
10.         persitToDB(net, qos, stars)
10.      end for
12.   end while
13.End
```

In Algorithm 2, the collector algorithm is presented. This algorithm scans all networks in collector server's range, after which it connects to each network, collects QoS parameters, and then it uses the NN to give a classification for each network; results are stored in the database and then made available to queries.

Thus, our proposal is divided into three modules: Collector, Classifier and Web Service. Each one of them has a well-defined function and they work together to accomplish the selection process and to serve the information of the best available networks.

The Collector is the piece of software responsible for collecting signal level and Quality of Service (QoS) variables for every network in range in order to perform the classification process in an ulterior step. This module calculates QoS variables packet loss, jitter, delay and throughput based on Internet Control Message Protocol (ICMP) packets using the Linux PING command, and it obtains the signal level using the Linux IWLIST command.

In the Classifier module, the collected networks are classified among the 5 possible classifications by the neural network, and then the result is persisted into the database for future consultations. The trained neural network is loaded from an XML file.

The Web Service part of our proposal is responsible for serving information to any application that would consume it. This module has a simple socket script, which returns a JavaScript Object Notation (JSON) with networks that have the biggest stars number amongst the collected networks.

## IV. METHODOLOGY

The materials utilized to perform our research are the following: a netbook Asus Eee PC with Ubuntu 14.04.1 LTS 32-bit as our server, Python version 2.7.6 as programming language, to implement the neural networks we have chosen a library called PyBrain in its version 0.31, and to persist data we utilized the SGBD MySQL version 5.5.41.

To train our neural network we made use of a dataset containing 10,000 samples generated by a script respecting interval of values for every attribute for each class. It was trained for about 200 epochs until it converged. The neural network converged with about 3% of error for new data presented to it; on the other hand it had 97% of right classifications, which is a pretty good result. We utilized the PyBrain library to implement, train and test our neural network and then we integrated it to our classifier module.

We tested our neural network classification capability on three Wi-Fi networks with ESSIDs GREDES_TESTE, IFTO_RDS, and IFTO_LABINS up to 6776 cycles, each cycle correspond to collecting QoS variables of the networks, classifying them and after that showing the best networks. It must be said that all of these networks are in a real production environment.

Our solution is based in a NN with an oriented training. Due our work is based (or within) of a big project; we used a subset of values of networks variables used by this project. So, when test were run we had not access to other technologies collected data.

## V. RESULTS

Tested networks presented different levels of quality due to several factors as: distance from the Access Point (AP), obstacles, number of connected users, network saturation level, etc. GREDES_TESTE, IFTO_RDS and IFTO_LABINS were considered a 5 stars network in 93%, 76% and 49% of the tests, respectively.

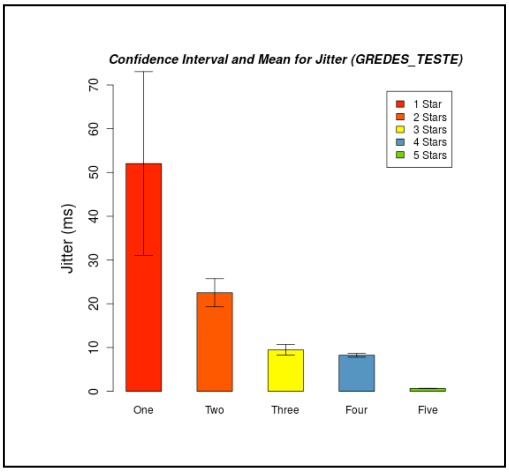


Figure 3. Jitter values for GREDES_TESTE.

In Figure 3, we present a chart that shows mean and confidence interval values of jitter for GREDES_TESTE for each star. It is easy to perceive that the value of jitter and the number of stars are inversely proportional, in other words the lesser jitter the more stars a network will get, and it shows that our neural networks has a good behavior.

 ISBN: 978-1-61208-433-6 

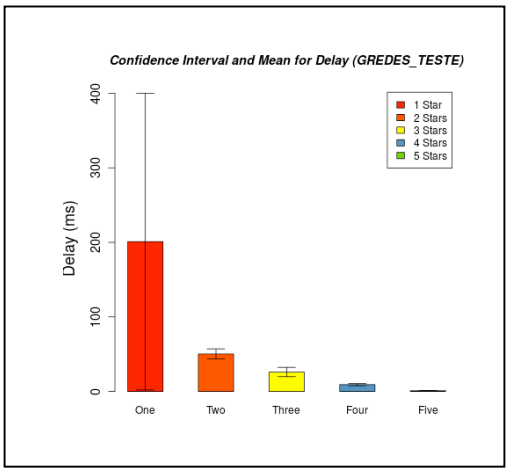


Figure 4. Delay values for GREDES_TESTE.

Our neural network had a similar behavior for delay compared with jitter as can be seen in Figure 4. The value decreases when the stars increase.

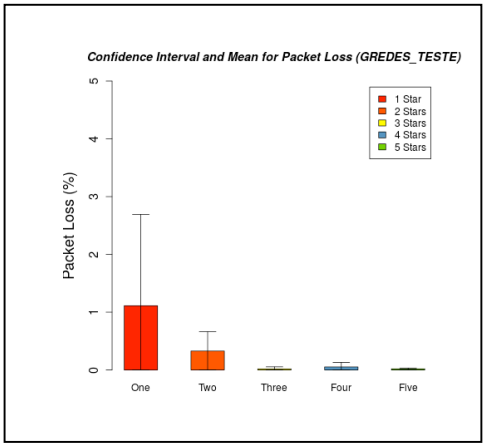


Figure 5. Packet loss values for GREDES_TESTE.

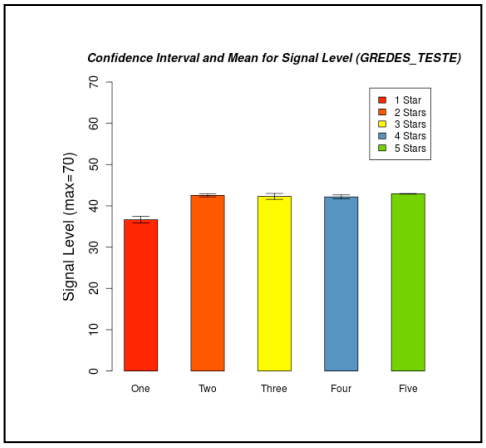


Figure 6. Signal level values for GREDES_TESTE.

In Figure 5, we can see that the values for packet loss are really low, but they present decay just like the other QoS parameters.

In Figure 6, we can see values of signal level for GREDES_TESTE and for each star. The values are almost the same for each star. We can infer from this information that there is no relation between signal level and other QoS parameters, because QoS parameters are varying in each stars while signal level remain almost the same value.

GREDES_TESTE is the closest AP (Access Point) from our Collector server, 2 meters of distance more precisely, and has less obstacles than the other APs, only one wall, as well it has less users allowed to use it, only a few users from our research group are authorized to use this network.

The results above show that GREDES_TESTE is a 5-stars network. This network was classified as excellent in about 93% of the tests, and its values for jitter, delay, packet loss and signal level presents the best levels when compared with the results from the other networks.

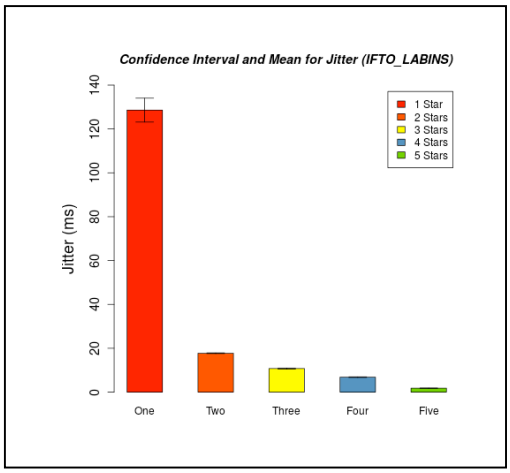


Figure 7. Jitter values for IFTO_LABINS.

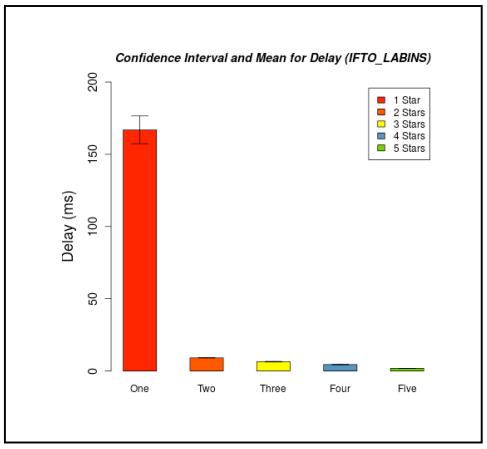


Figure 8. Delay values for IFTO_LABINS.

 ISBN: 978-1-61208-433-6 

In Figure 7, we present a chart that shows mean and confidence interval values of jitter for IFTO_LABINS for each star. In this scenario, our neural network also performed as expected, the value falls as the number of stars raises.

In Figure 8, we present a chart that shows the mean and confidence interval values of delay for IFTO_LABINS for each star. The change of values are more subtle but we still can perceive that it has a descending pattern as the stars increase.

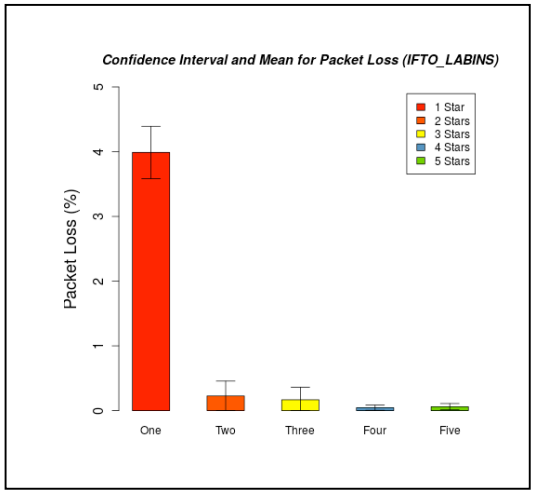


Figure 9. Packet loss values for IFTO_LABINS.

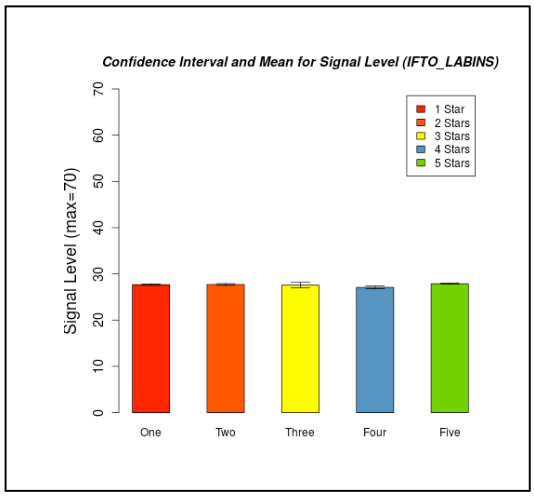


Figure 10. Signal level values for IFTO_LABINS.

In Figure 9, we present a chart that shows mean and confidence interval values of packet loss for IFTO_LABINS for each star. Packet loss for this network was bigger than from the previous network due to several reasons, but the pattern is still the same.

In Figure 10, we present a chart that shows mean and confidence interval values of signal level for IFTO_LABINS for each star. We can see that the values are pretty much the same for all stars.

IFTO_LABINS is about 52 meters of distance from our collector server, and it has multiple obstacles, including 4 walls. This network is open to all students on this campus, and it has many users using it during the day.

The results above show that IFTO_LABINS is a 5-stars network in 49% of tests, and its values for jitter, delay, packet loss and signal level are the worst results from all the three networks.

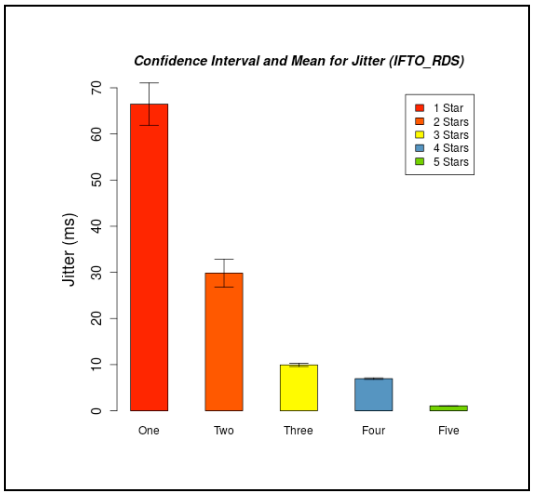


Figure 11. Jitter values for IFTO_RDS.

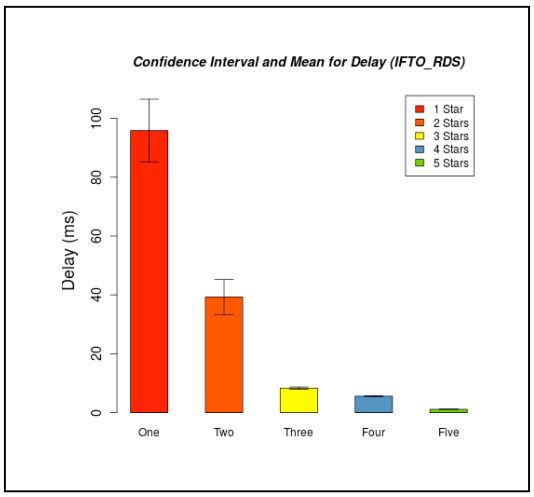


Figure 12. Delay values for IFTO_RDS.

In Figure 11, we present the mean and confidence interval values of jitter for IFTO_RDS and for each star in a chart. In this test the result is like the other results obtained from the other networks.

In Figure 12, values of delay for IFTO_RDS for each star are presented; the pattern is similar to the one from the other networks.

 ISBN: 978-1-61208-433-6 

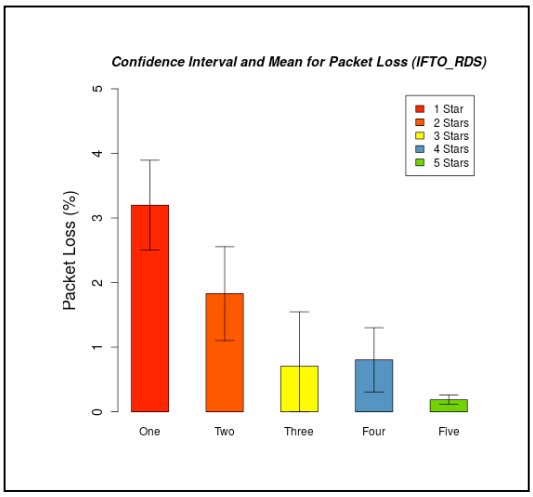


Figure 13. Packet loss values for IFTO_RDS.

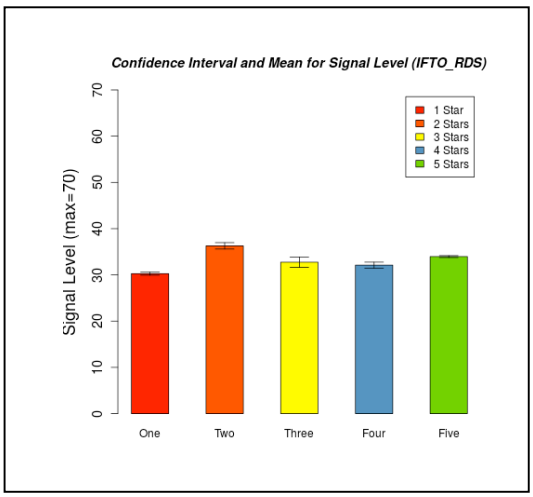


Figure 14. Signal level values for IFTO_RDS.

In Figure 13, we present mean and confidence interval values of packet loss for IFTO_RDS and for each star. The values get down as the number of stars gets up.

In Figure 14, we present a chart that shows mean and confidence interval values of signal level for IFTO_RDS and for each star. Values presented a variation but it was most likely following a pattern of increasing as the number of star raises, because they are directly proportional. It is important to highlight that due to the kind of problem we are dealing with, we do not always observe the expected pattern.

IFTO_RDS is about 63 meters of distance from our Collector server, and it has multiple obstacles, 2 walls and some trees. This network is open to all students on this campus, and it has many users using it during the day.

The results above show that IFTO_RDS is a 5-stars network in 76% of tests, and its values for jitter, delay, packet loss and signal level are the second best from all the three tested networks.

In most cases jitter, delay and loss decrease as the number of stars increases because it is known that the lesser of them the better. Signal level is different from the other parameters, when it increases, the number of stars also tend to increase. It is interesting that the values of all variables but signal level follows a pattern of decay in all tested networks when the number of stars raises.

## VI. Conclusion and Future Work

The foregoing discussion has attempted to highlight the importance of a good mechanism for making decision regarding the handover process in heterogeneous networks, and the lack of a good solution in today's mobile devices.

Our approach based on neural networks seems really promising; our neural network is able to classify networks of any technology type. Classification is given based on QoS parameters, and those parameters can be obtained from networks of any technology. We conclude that there is no causal correlation between QoS variables and signal level. Our neural network had a good behavior so that we could validate that based on the values of QoS/Signal Level and the number of stars that were assigned by it to the tested networks.

As future work, we can compare the efficiency of this approach against other approaches that use other methods, such as: MADM, genetic algorithms, fuzzy logic, etc., this will be easy to do once our architecture allows us to replace the selection method by any method we like just by replacing the algorithm in the classification module. We would also like to add user preferences as an attribute taken into account on the selection process.

## Acknowledgment

We would like to thank the National Counsel of Technological and Scientific Development (CNPq) for supporting our research. We would also like to thank our host institution (IFTO) for providing resources in order to conduct our work.

## References

[1] International Telecommunication Union. *ICT Statistics*. [Online]. Available from: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf 2014.03.25

[2] E. Gustafsson and A. Jonsson, “Always Best Connected”. Wireless Communications, IEEE, vol. 10, no 1, 2003, pp. 49-55.

[3] S. P. Thiagarajah, A. Ting, D. Chieng, M. Y. Alias and T. S. Wei, “User Data Rate Enhancement Using Heterogeneous LTE-802.11n Offloading in Urban Area”. Wireless Technology and Applications, IEEE, 2013, pp. 11-16.

[4] V. d. M. Rios, “Wireless Network Selection based on Decision Support Techniques”. Masters' dissertation, Electrical Engineering Department - University of Brasilia (UnB). Brasilia - DF, Brazil, 2012.

[5] R. Trestian, O. Ormond. and G. M. Muntean, “Enhanced Power-Friendly Access Network Selection Strategy for

 ISBN: 978-1-61208-433-6 

Multimedia Delivery Over Heterogeneous Wireless Networks". TBC, vol. 60, no 1, 2014, pp. 85-101.

[6] M. Lahby, L. Cherkaoui and A. Adib, "Novel Validation Approach for Network Selection Algorithm by Applying the Group MADM". Computer Systems and Applications, IEEE, 2013, pp. 1-4.

[7] O. Ludwig and E. Costa, Neural networks - fundamentals and applications with C programs. Publisher Ciencia Moderna, 2007.

[8] M. Fiedler, T. Hossfeld and P. Tran-Gia, "A Generic Quantitative Relationship Between Quality of Experience and Quality of Service". IEEE, vol 24, no 2, 2010, pp. 36-41.

[9] D. C. d. Silva, "A Mobile Network Selection Architecture for Heterogeneous Environments". Masters' dissertation, Electrical Engineering Department – Federal University of Minas Gerais (UFMG). Belo Horizonte - MG, Brazil, 2015.

[10] F. L. d. Silva, "Low Resources Consumption Network Selection Architecture for Mobile Devices". CONNEPI, 2014.

[11] Schaul, T., Bayer, J., Wierstra, D., Sun, Y., Felder, M., Sehnke, F., ... and Schmidhuber, J., "PyBrain". The Journal of Machine Learning, vol. 11, 2010, pp. 743-746.

 ISBN: 978-1-61208-433-6 