Analysis of the Logical Proximity between 802.11 Access Points

Ricardo Sousa
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
rmps@inescporto.pt

Ricardo Morla
INESC TEC
Campus da FEUP
4200 - 465 Porto
Portugal
ricardo.morla@inescporto.pt

Abel Maio
FEUP
Campus da FEUP
4200 - 465 Porto Porto
Portugal
ei07102@fe.up.pt

João Coelho
FEUP
Campus da FEUP
42 - 465 Porto
Portugal
ei07118@fe.up.pt

***Abstract*—802.11 campus networks have their access points deployed across the area the network has to cover. The physical proximity between these access points is often well understood. Network managers have maps of the campus and of the location of the access points. Due to reflections indoor and between buildings, a mobile station may have connectivity to only one of two physically nearby access points. This may mislead the network manager into thinking that an area is densely covered when in fact mobile stations cannot connect to all of the access points in that area. This problem can have a dynamic nature if we consider people and objects that move around and change the properties of the propagation medium. In this paper we explore an alternative measure to physical proximity based on mobile station connectivity to the access points, which we call logical proximity. We take the ping-pong effect as a proxy to logical proximity. We use this proximity to characterize 802.11 campus network with over 200 access points and 14k users over a 2 year period. We report on the magnitude of the ping pong effect, the clustering of access points, and the degree distribution of the resulting access point proximity network.**

***Index Terms*—Local area networks.**

I. Introduction

The deployment of wireless networks technology has attracted attention to the issues that network managers face. One of such issues is to adequately characterize the logical proximity of access points. This enables the network managers to view the network from the perspective of the user and of possible access points that mobile stations can connect to. In this work we aim to identify the logical proximity of the access points using the ping pong effect as a proxy for such proximity. The ping pong effect can be characterized by a series of consecutive connections to different access points and is the result of the aggressive nature of 802.11 interfaces that try to connect to an access point with a better signal once the signal from the current access point drops below a threshold.

In the past, research has been developed to understand network topology. The authors in [1] propose a mobility-aware clustering algorithm that uses roaming events as the metric to evaluate the proximity to access-points (APs) without using any geographical information. In this work, they make reference to three main goals for studying the mobility of the users on wireless networks: (1) to understand what are the implications and how mobility can have an impact on the network services, (2) to create realistic mobility models to evaluate the performance of protocols and algorithms, and (3) to propose new communication solutions that can adapt to the specificities of each user.

Another investigation that approach this subject is project SPOTS [2]. In this project, the aim is to create a better understanding of the daily working and living patterns of the MIT academic community, which changes due to the emergence of WiFi itself. Tang and Baker in the paper [3] analyze the network for overall user behavior (when and how intensively people use the network and how much they move around), overall network traffic and load characteristics (observed throughput and symmetry of incoming and outgoing traffic), and traffic characteristics from a user point of view (observing a mix of applications and number of hosts connected to by users).

[4] analyzes the Wi-Fi network as a proxy to space usage aiming to use it as a mean for the characterization of physical spaces and, consequently, as a source of information for a dynamic symbolic model representing those spaces. In [5] a general methodology is presented for extracting mobility information from wireless network traces, and for classifying mobile users and APs. Kim and Kotz [6] propose a model of user movements between APs. In their paper they define three goals in developing a mobility model. First, the model should reflect real user movements, second, the model should be general enough to describe the movements of every device and third, the model should consider the hourly variations over a day. Another area with many interesting works is network tomography, originating from a research by Vardi [7]. One of the main applications of network tomography is to detect heavily loaded links and subnets [8]–[10]. Another important work is presented in [11] describing the use of a novel and efficient data structure called neighbor graphs, which dynamically capture the mobility topology of a wireless network as a mean for pre-positioning the stations context ensuring that the stations context always remains one hop ahead. [12] proposes a new way of measuring and extracting proximity in networks called cycle free effective conductance (CFEC). Their proximity measure can handle more than two

end points, directed edges, is statistically well-behaved, and produces an effectiveness score for the computed subgraphs.

The investigation presented on project NearMe [13] is a way to find people and things that are in your physical proximity using Wi-Fi. The project consists in a server, algorithms, and application programming interfaces (APIs) for clients equipped with 802.11 wireless networking (Wi-Fi) to compute lists of people and things that are physically nearby.

The research described in this paper is related to several other projects and technologies in ubiquitous computing, including location sensing, proximity measurement, and device discovery. The main difference is that it looks at the ping pong effect as an indicator of logical proximity. We intend to identify access points that are near each other, without any prior knowledge about the location or proximity between them. The main contribution of this paper is to develop an algorithm that will generate a graph to describe the closeness between logical access points using only historical data from the wifi network. In Section II we describe the dataset we use, followed by the algorithm to find the proximity of the access points in section III. Section IV presents results of applying the algorithm to the dataset. In Section V we present our conclusions.

II. Dataset

A. Attributes

Each record with index $r = 1..R$ in the dataset represents a session of a user at any given access point as recorded by the RADIUS authentication server [14], [15] and has the following attributes:

- User identifier, index $u = 1..U$
- Access point identifier, index $a = 1..A$
- Location description, index L^a
- Session start time (resolution in seconds), as $v\_start_i^{(u,a)}$.
- Session duration (resolution in seconds), as $s\_time_i^{(u,a)}$

Index i ranges from 1 to the number of sessions user u has in access point a.

B. Description

The data set that we use in this paper was collected from November 2006 to March 2009. $U = 14,167$ users were observed to connect to $A = 217$ access points in $R = 6,249,992$ sessions. Figure 1 shows the time series of the ratio of active users, active access points, and observed sessions per week. Active users and access points are those for which at least one session has been observed during the week.

The ratios are against the maximum of 2707 users in a week, 206 access points in a week, and over 115 thousand sessions in a week. We can observe a crest of the number of sessions and of active users during the summer terms, which drop from approximately 50% to less than 10%. The first semester (Fall) in both years shows a sustained increase of these numbers, whereas after January these are still high but much more irregular until the summer crest. The number of active access points also has a crest in the summer terms but with a much smaller variation (approximately 75% to 65% in the first crest and 95% to 80% in the second crest).

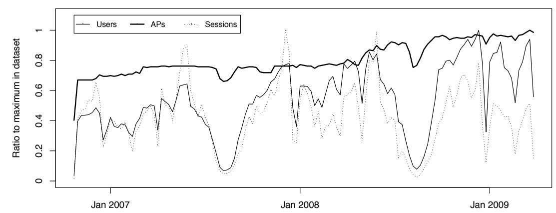

Fig. 1. Weekly time series of the dataset.

This means that the users that remain on campus during the summer use almost all of the access points that are used during the first and second semesters.

Figure 2 shows the cumulative distribution function (CDF) of the number of users and sessions per access point. A few access points seem to have much more sessions than the rest: more than 90% of the access points have less than 65k sessions while fewer than 10% of the access points have between 65k and 200k sessions. Users distribution per access point seems to be less skewed: more than 90% of the access points have sessions from less than 2800 users while fewer than 10% access points have between 2800 and 4500 users.

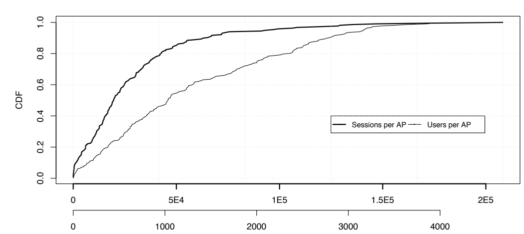

Fig. 2. CDF of the number of users and sessions per access point.

Figure 3 shows the CDF of session durations. More than 70% of the sessions have less than 5 minutes and more than 20% have sessions smaller than 20 seconds. This points to a large majority of small sessions. We also notice a log-linear distribution for sessions between 5 minutes and 4 hours. More than 99% of the sessions are smaller than 4 hours.

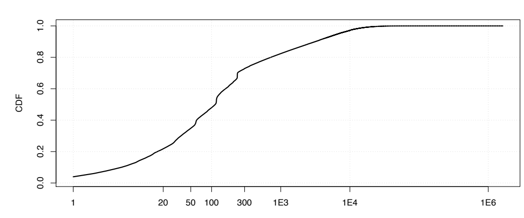

Fig. 3. CDF of session duration.

Figure 4 shows the CDF of the time between consecutive sessions by the same user. In more than 30% of the cases the time between sessions is reported as 0s (notice that the granularity of the record is 1s) and in more than 24% of the cases it is reported as 1s (the log scale prevents plotting these values on the figure). In 71% of the cases, the time between sessions is smaller than 15s. This means that most of the session changes are due to handover and only a small portion due to users connecting and disconnecting their mobile devices.

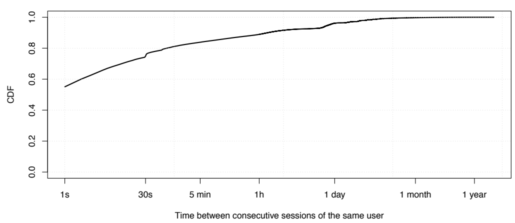

Fig. 4. Intersession CDF.

III. Algorithm

The goal of the algorithm presented in this paper is to identify the logical proximity of the access points of a wireless network. We take the ping-pong effect as a proxy to logical proximity. Ping-pong effect refers to the succession of associations-dissociations between two ore more Access points. So it is necessary to analyze the processes of commutation between access points that are made by each user on the network usage.

Before we start reading the data presented in Section II, we must introduce the threshold value that defines the segments of the session. The first step of the algorithm is to read the data, to organize and order the events by the user. This action also creates a list of users in the dataset. The central idea of this algorithm is to create different segments of sessions for each user. The segment of the session consists of grouping the set of access points, when the commutation time is lower than the threshold value. When creating segments of session of the user, the algorithm can identify the access points used in this segment. This way it is possible to identify the access points that are near each other.

In certain cases, when the connection is established, the user may use different access points. The purpose of the algorithm is to interpret the fast commutations made between access points to obtain a logical topology proximity between the access points using only the raw data network.

For consecutive access points, the algorithm checks if the commutation time between access points is lower than the threshold. If that's the case, the algorithm groups the access points in the same segment session. In case the commutation is higher than the threshold, the algorithm creates a new segment.

For each segment, the algorithm identifies which is the dominant access point. The dominant access point is the more frequent in the segment of the session. After the dominant access point is identified, the algorithm updates a commutation matrix. This matrix allows to draw a network graph and creates a view of the topology of the logical proximity of the access points. To update the commutation matrix (Matrix_Commutation), firstly the algorithm needs to identify the line position of the dominant access point. Then it needs to identify the position of the other access points in the columns of the matrix. When the position of the corresponding cell is identified for the two access points, the value 1 is added.

Algorithm 1: proximity of ap based on segmentation of user sessions

```
Data: Dataset → Collection of network events
Result: Matrix_Mean_Commutation
input : time threshold
Dataset ⇒ Group and order the collection of events by user
/* create a list of all the users */
List_users→ all(username in Dataset)
for val_users ∈ List_users do
    /* foundRows is the structure collects the links of each user */
    foundRows=subset(Dataset.username == val_user)
    for (i = 1; i <= foundRows.count; i++) do
        v_source=foundRows[i-1][Apcode]
        v_destine = foundRows[i][Apcode]
        v_start = foundRows[i-1][Time_Stamp]
        v_end = foundRows[i][Time_Stamp]
        commutation = Convert.ToDouble(End-Start);
        ResCom → function.save(val_user,v_start,commutation,v_end
        ,v_source,v_destine);
    end
end
for val_users ∈ ResCom do
    result_Session_user → grouping the sessions when the
    commutation time is less threshold defined
    foreach Session ∈ result_Session_suser do
        Ap_Dominat → search the access point with the largest
        presence in the session
        List_near_ap → list of other access points in the session
        Matrix_Commutation → In Matrix update the line of the
        dominant access point
    end
end
```

A. Example

To better understand the algorithm we analyze one simple example. After reading the data, the first step of the algorithm is to group and sort the events for each user, which means the events are ordered by Time_stamp. The second step is to identify the users present in the dataset and store them in a structure called List_users. For each user is created a structure called foundrows. This structure temporarily stores each user's events.

Username	Time_stamp	Sessiontime	Apcode
10	t_stamp_0	s_time_0	6
10	t_stamp_1	s_time_1	3
10	t_stamp_2	s_time_2	3
10	t_stamp_3	s_time_3	4
10	t_stamp_4	s_time_4	2
10	t_stamp_5	s_time_5	5
10	t_stamp_6	s_time_6	1
10	t_stamp_7	s_time_7	2
10	t_stamp_8	s_time_8	1

TABLE I
EXAMPLE THE STRUCTURE FOUNDROWS FOR ONE USER

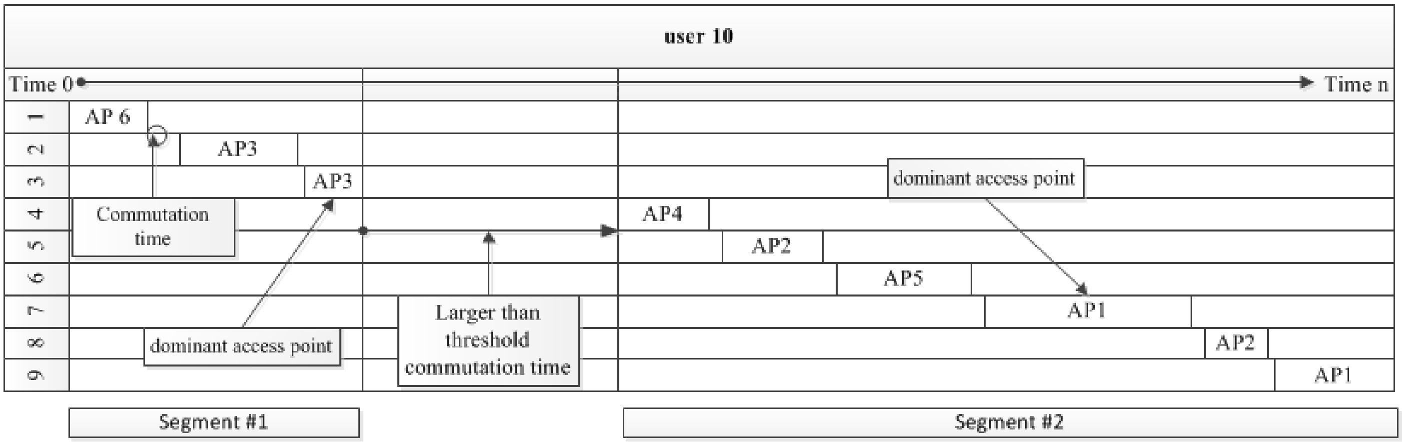

Fig. 5. Figure to represent the activity the user in network

The Third step of this algorithm is important since it will define the structure of the commutation times we call ResCom, which calculates the comutation time between access points. Therefore, we used the equation commut, which aims to determine the time required for commutation between access points. We calculate the time end (v_end) by removing the Sessiontime of the Time_stamp.

$$commut[i-1] = t\_stamp\_[i] - s\_time\_[i] \tag{1}$$

From the subset of data relating to the example presented it is possible to obtain the following results in table II:

user	v_star	v_end	commut.	v_source	v_destine
10	t_stamp_0	t_stamp_1 - s_time_1	1	6	3
10	t_stamp_1	t_stamp_2 - s_time_2	6	3	3
10	t_stamp_2	t_stamp_3 - s_time_3	21785483	3	4
10	t_stamp_3	t_stamp_4 - s_time_4	1	4	2
10	t_stamp_4	t_stamp_5 - s_time_5	2	2	5
10	t_stamp_5	t_stamp_6 - s_time_6	8	5	1
10	t_stamp_6	t_stamp_7 - s_time_7	170	1	2
10	t_stamp_7	t_stamp_8 - s_time_8	1	2	1

TABLE II
STRUCTURE RESCOM FOR COMMUTATION BY USER

Observing graphically the results of the commutation time described in with the previous table, we get to the graph of 6:

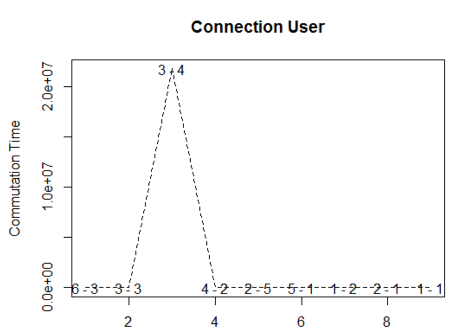

Fig. 6. Graph the commutation of user

For example, observing figures 5 and 6, it's can be seen that at a given moment the time of commutation between access points is much bigger, which means that for some reason the user has left the premises, e.g., he returned after a few hours or finished the day's work. The algorithm will identify for this case two different moments of use. This way it will create two distinct segments of user session.

To create the matrix commutation, the algorithm needs to examine the segments of session. First it identifies the dominant access point. When the dominant access point is identified the algorithm searches the remaining access points present in the same session.

Returning to the example, in the first block that defines the user session, the algorithm extracts the access point 3 as dominant. This happens because it is the access point with the largest presence in the session. Then, it finds the other access points in the same session. In this case we only have the access point 6.

To update the Matrix_Commutation, the first step consists of finding the position of the dominant access point in the first row. In the following step, it finds the position in the columns for the other access points present in the segment of session, adding the value 1 to the corresponding cell.

Updating the matrix, with the results of the first session of this user, the algorithm in cell $\{6,3\}$ adds the value 1.

For the second session of this user, the algorithm makes the same process, but now we obtain the access point 1 as the dominant. In this segment of session the following access points $\{4,5,2\}$ are present. To update the line of the dominant access point, we add 1 to the corresponding access points that are present in the session.

Next we assume that the user 11 presents the following events in the structure ResCom:

user	v_star	v_end	commut.	v_source	v_destine
11	t_stamp_0	t_stamp_1 - s_time_1	1	6	3

TABLE III
COMMUTATION BY USER

In this case, as there is no dominant access point, the algorithm assumes that the first access point is the dominant. In the previous example the dominant access point is 6. Then it increments the value 1 to the cell {6,3} in the Matrix_Commutation.

For this example, we obtain the Matrix_Commutation of the following table IV:

AP	1	2	3	4	5	6
1		1		1	1	
2						
3						
4						
5						
6			2			

TABLE IV
MATRIX WITH NUMBER COMMUTATION BETWEEN ACCESS POINT

The algorithm draws the network graph, as shown in the figure 7. The connection between the access points 6 and 3 is represented by one larger and darker line, Which means these two access points are probably closer. This happens because the number of commutations between these two access points is more evident.

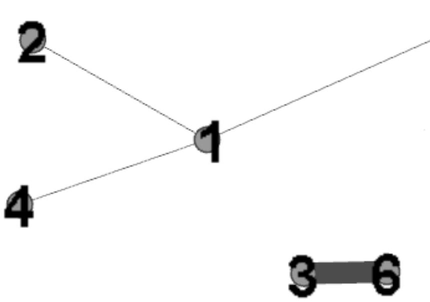

Fig. 7. Graph of frequency of AP

This algorithm has some advantages, the first of which is that it enables a idea of a ping-pong effect between access points. Other advantage is that identifies the dominant access point which has more impact in the network.

IV. RESULTS

To implement the algorithm we used the C# language and developed the tool ProxAP as can be seen in Figure 8. With this tool, it is possible to visualize the matrix commutation generated by the algorithm.

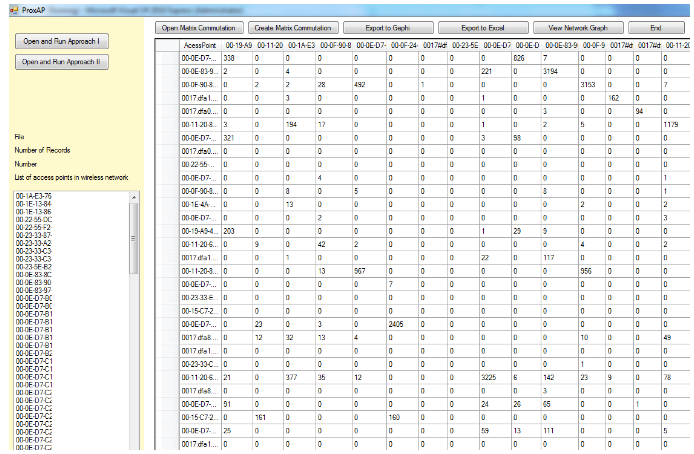

Fig. 8. Tool ProxAP

This tool allows to export the results to the Gephi [16]. It is an open-source software to visualize and analyze network graphs. This way, it is possible to observe the results generated by the algorithm presented in this article.

When we apply the data presented in section II to the algorithm, with a a threshold of 100 seconds, we obtain the commutation matrix. When these results are used in the Gephi tool, applying the algorithm Yifan Hu, with a filter higher than 200 connections, we obtain the figure 9.

We decided to use the algorithm Yifan Hu and filters it allows a better visualization of network graphs.

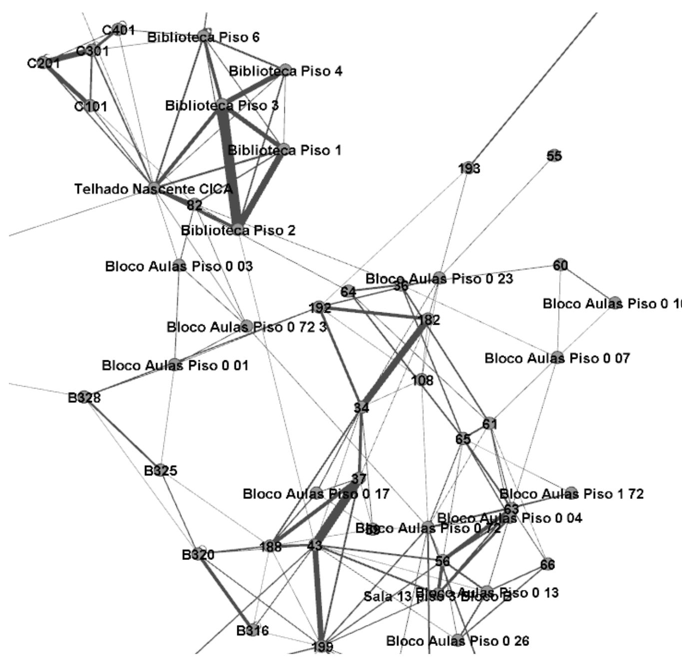

Fig. 9. Graph of frequency of AP

Using the application Gephi, the administrator can obtain many views of network topology. Making a closer view in the graph to the access point "Biblioteca Piso 3" we obtain:

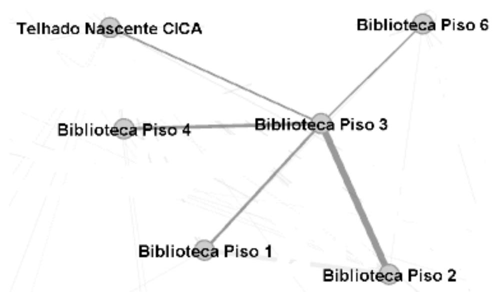

Fig. 10. Graph of frequency of AP

Using another location with a strong impact on the graph created, we obtain:

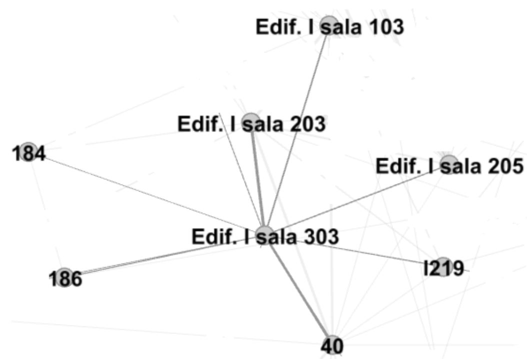

Fig. 11. Graph of frequency of AP

This way, the network administrator has the opportunity to see the access points with bigger ping-pong effect on the network.

A. Analysis of number of commutations

In this section we analyze the number of commutations of one user from one access point to the same access point and to other access points.

Figure 12 shows the CDF of the number of commutations to the same access point in absolute value and relative to the number of sessions in the access point. The number of commutations to itself is relatively small (up to tens of thousands) compared to the number of sessions which can go up to hundreds of thousands of sessions. This is because each commutation is a segment of a potentially large number of sessions. More than 90% of the access points have less than 10% ratio of the number of commutation to their number of sessions. There are 4 access points with this ratio above 20% have a small number of sessions (less than 300), which puts them in the very beginning of the session per AP distribution and says these are scarcely used access points.

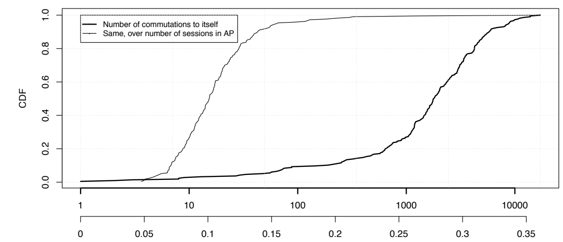

Fig. 12. CDF of commutation to self.

Figure 13 shows the CDF of the percentage of commutations to other access points relative to the total commutations (both to itself and to others). No access point has only commutations to itself (the minimum is 42% of the total commutations). Most access points have more commutations to other access points than to themselves (70% have between 50% and 80% commutation ratio). A significant percentage of access points (5%) has more than 85% of their commutations to other access points.

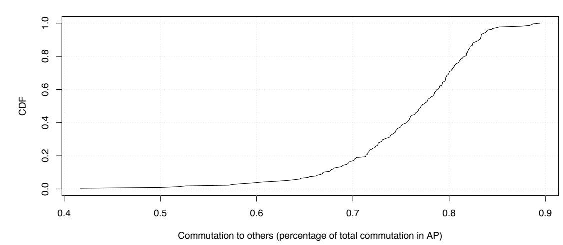

Fig. 13. CDF of commutation to others.

B. Chinese Whispers Clustering

The goal of applying this clustering technique is to easily identify access points considered near by the algorithm. Using the algorithm Chinese Whispers Clustering in Gephi tool we obtain 24 clusters. The Clustering is the process of grouping together objects based on their similarity to each other [17]. This means that for our example we have 24 clusters with access points considered near each other.

Selecting randomly one cluster we obtain the following figure 14.

Fig. 14. Selected the first cluster

For example, if we choose the cluster that has the largest number of access points, we get figure 15:

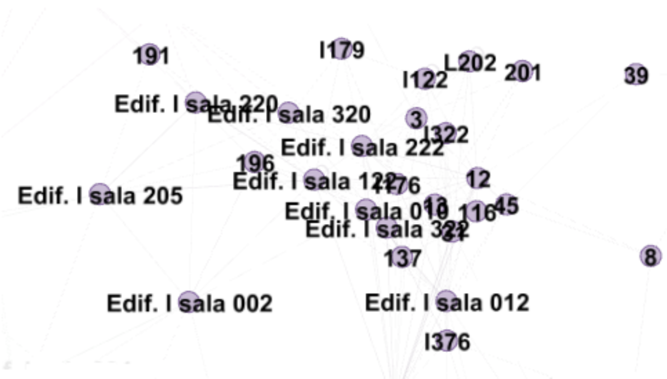

Fig. 15. Cluster with the largest number of elements

This way, the network administrator has information that many access points are concentrated in building ”I”. This important so that he a precise knowledge about the use and proximity between access points.

C. Average Degree Distribution

In this article we explore the Average Degree Distribution metric that is available in the Gephi [16] to evaluate the results generated by the algorithm.

In the study of networks, the degree of a node in a network is the number of connections it has to other nodes and the degree distribution is the probability distribution of these degrees over the whole network [18].

From the example discussed in this paper we obtain 6,958 as the Average Degree Distribution.

For Average Degree Distribution 6,958:

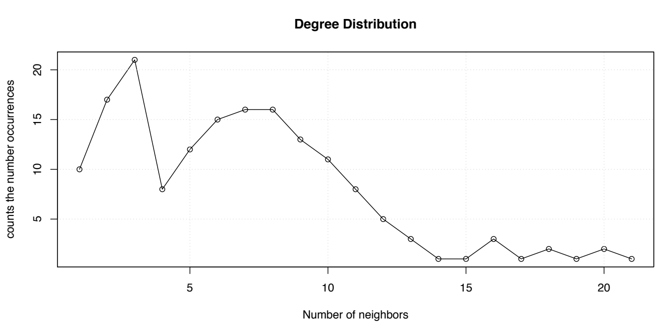

Fig. 16. Average Degree Distribution 6,958

One of the advantages in using this metric to enable the administrator is to understand how the access points are distributed. The figure 16 shows that there is a small set of access points where the number of occurrences is bigger. This information can be useful to identify the access points that have this behaviour and to understand the reason why this happens.

Figure 17 shows the CDF of the node degree. As we can see most access points have a value smaller than 10. This means most of the access points have less than 10 access points nearby.

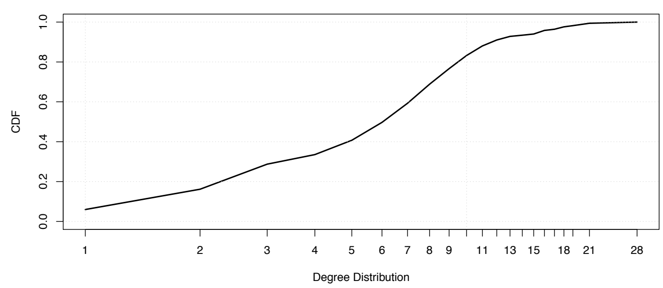

Fig. 17. CDF Degree Distribution

V. Conclusion

In this article we developed and explored an algorithm to identify the proximity of access points in 802.11 campus network. The principal objective of the algorithm developed is to provide network administrators with a logical view of the proximity between access points. For the logical topology of proximity between the access points, the algorithm makes the analysis of raw data collected for a period of about 2 years.

The main task of the algorithm is to characterize the ping-pong effect between access points. Thus, it builds the matrix which defines the logical proximity between access points. This way the network administrator can overlap a graph of nearby access points with metrics and indicators of network usage and performance and visually detect geographic correlation of lower quality indicators.

Acknowledgment

This work is financed by the ERDF- European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT- Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within projectS PTDC/EIA-EIA/113933/2009 and (FCOMP-01-0124-FEDER-015064).

References

[1] M. Boc, A. Fladenmuller, and M. de Amorim, “Towards self-characterization of user mobility patterns,” in *Mobile and Wireless Communications Summit, 2007. 16th IST*, july 2007, pp. 1 –5.

[2] A. Sevtsuk, S. Huang, F. Calabrese, and C. Ratti, *Mapping the MIT campus in real time using WiFi*. Hershey, PA: IGI Global, 2009.

[3] D. Tang and M. Baker, “Analysis of a local-area wireless network,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom ’00. New York, NY, USA: ACM, 2000, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/345910.345912

[4] K. Baras and A. Moreira, “Symbolic space modeling based on wifi network data analysis,” in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, june 2010, pp. 273 –276.

[5] W. jen Hsu and A. Helmy, “On modeling user associations in wireless lan traces on university campuses,” in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, april 2006, pp. 1 – 9.

[6] M. Kim and D. Kotz, "Modeling users' mobility among wifi access points," in *Papers presented at the 2005 workshop on Wireless traffic measurements and modeling*, ser. WiTMeMo '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 19–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1072430.1072434

[7] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, pp. 365–377, Mar. 1996. [Online]. Available: http://dx.doi.org/10.2307/2291416

[8] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statistical Science*, vol. 19, pp. 499–517, 2004.

[9] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, "Toward the practical use of network tomography for internet topology discovery," in *INFOCOM, 2010 Proceedings IEEE*, march 2010, pp. 1 –9.

[10] D. Ghita, K. Argyraki, and P. Thiran, "Network tomography on correlated links," in *Proceedings of the 10th annual conference on Internet measurement*, ser. IMC '10. New York, NY, USA: ACM, 2010, pp. 225–238. [Online]. Available: http://doi.acm.org/10.1145/1879141.1879170

[11] A. Mishra, M. Shin, and W. Arbaush, "Context caching using neighbor graphs for fast handoffs in a wireless network," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol.(xxxv+2866).

[12] Y. Koren, S. C. North, and C. Volinsky, "Measuring and extracting proximity graphs in networks," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 3, Dec. 2007. [Online]. Available: http://doi.acm.org/10.1145/1297332.1297336

[13] J. Krumm and K. Hinckley, "The nearme wireless proximity server," in *UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings*, ser. Lecture Notes in Computer Science, N. Davies, E. D. Mynatt, and I. Siio, Eds., vol. 3205. Springer, 2004, pp. 283–300.

[14] C. Rigney, A. C. Rubens, W. A. Simpson, and S. Willens, "Remote authentication dial in user service (radius)," Internet RFC 2865, June 2000.

[15] C. Rigney, "RADIUS Accounting," RFC 2866 (Informational), Internet Engineering Task Force, June 2000, updated by RFCs 2867, 5080. [Online]. Available: http://www.ietf.org/rfc/rfc2866.txt

[16] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[17] C. Biemann, "Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems," in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, ser. TextGraphs-1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 73–80.

[18] Wikipedia, "Degree distribution," 2012, [Online; accessed 10-Set-2012]. [Online]. Available: http://en.wikipedia.org/wiki/Degree_distribution