Community-based approach to handle wireless connectivity context data

Roberto Rigolin Ferreira Lopes
University of São Paulo
São Carlos, Brazil
rigolin@icmc.usp.br

Bert-Jan van Beijnum
University of Twente
Enschede, The Netherlands
beijnum@ewi.utwente.nl

Edson dos Santos Moreira
University of São Paulo
São Carlos, Brazil
edson@icmc.usp.br

***Abstract*—Connectivity management in heterogeneous wireless networks can be done combining context-aware techniques, aiming at perform optimal handovers. For instance, mobility and QoS predictors use, as input, georeferenced historical network context data. Moreover, the predictor's accuracy is strongly related to the quality of this required data set. Wireless connectivity context data is composed by date, time, geographical localization, and QoS metrics, to cite the most common. Usually, these data are available in public databases with considerable size and hardly updated. Nevertheless, few efforts are directed to improve the quality, freshness and handiness of this fundamental source of information. In this work we propose a community-based solution to allow mobile users collaborate to discover wireless connectivity islands. Also, we introduce a smart way to summarize and delivery connectivity context data. It is result of a prototyping effort and we focus the discussion on its feasibility in terms of storage size, power consumption and QoS metrics.**

***Index Terms*—Connectivity management, IP connectivity, virtual community.**

I. Introduction

Advanced connectivity engineering is leading to improved bandwidth, coverage, and mobile access, but it is, also, getting expensive, in terms of both capabilities and cost [21]. Gather, store, synchronize and process network context data are fundamental functionalities of several context-aware wireless connectivity management solutions, e.g. [2], [9], [12], [15], [23]. While the mobile user is roaming, the current and historical *connectivity context data* (also referred as mobility information or profile) is used, as input, by handover mechanisms, mobility predictors, QoS predictors, and mobility management standards and protocols.

The most common source of this kind of data are the searchable *wardriving* [13] databases. It, usually, has a considerable size, become outdated quickly and just provide basic information, such as the ones in 802.11 beacon packages [10]. Another problem is that these mobility techniques have their costs higher than their gains. More complex techniques may be preferred for more accurate estimations but they usually require more history information, more computation time, and more sophisticated parameter configuration [9]. However, while the user is moving there may not be enough time to the mobile device perform such complex processing.

In our previous work [11], we proposed a community-centric way to handle the context data used to manage IP connectivity in heterogeneous wireless environments. We introduced a graph based model to represent mobile user's IP connectivity experiences and use it as social media. Each mobile user is member of a virtual community, and all their time-spaced set of IP connectivity experience is represented as a digraph called *connectivity path*. These experiences are uploaded to the community and an algorithm is applied to combine them in a *connectivity graph* that is the media socialized.

In the current work, we started with the hypothesis that people will have their opportunities of communication and collaboration enhanced inside smart environments, such as houses, shoppings, universities and hospitals. Virtual communities can be a key tool to achieve it, and it is an interesting place to store, combine and share wireless network context data. We verify the feasibility of put people to collaborate in order to improve their wireless connectivity experiences inside a smart and virtual environment.

Most of the context-aware solutions uses raw QoS historical data or preprocess it using statistical tools like averages and probabilities. In this work, we studied how this kind of database can grows in function of time and in function of the probability of the mobile device take a connectivity opportunity during a roaming. We also propose a smart delivery system for those data, also called IP connectivity experiences. The ultimate goal is improve the quality of this set of data in terms of updatability, handiness and size.

The valuable contributions of the current paper is the summarized connectivity graph that leads to less storage size and power consumption. It is applied inside a feedback loop that, also, converges to better QoS metrics. This is the result of a prototyping effort using PDAs and its specifics APIs to gather connectivity context data at the client side, and common web technologies at the server side. We faced some implementation and execution issues during the prototyping and experimentation phases. This experience motivated us to discuss the feasibility of our solution, and we believe that it can be useful for similar approaches.

The rest of this paper is organized as follows. In Section II, the main related works are examined. The fundamental concepts related to connectivity management are discussed in Section III. Section IV describes our community-based solution. In section V, we discuss the feasibility of the proposed approach. Finally, conclusions are given in Section VII.

II. Related work

Several works in the literature proposes manage wireless connectivity combining QoS historical data, mobility prediction and handover mechanisms in a context-aware way. Most of them agree that an ideal connectivity management would be possible if, before the current access point (AP) becomes unusable, the device knew which AP it will use next, had already completed association, and had already received a IP address [12]. One of the most studied problem is define which AP will be the next one during an arbitrary roaming.

Some works, [14], [15], [23], exploited the knowledge of mobility patterns in order to properly allocate network resources and enhance the performance and QoS experienced by applications in a mobile device [15]. For Cheng et al., in [4], the mobility prediction algorithms usually consist of two steps: to assign conditional probabilities to the elements of a given the users movement history, and to use these values to predict the next term in the sequence.

Yami et al., in [23], proposed a scheme based on user mobility profile to allow call admission resource reservation in celular networks. These profiles are used for path prediction and contain frequent paths and their associated probabilities and the information related to mobile user's behavior. The mobile device will have the responsibility to collect the mobility data of a user that uses this mobile device. The method guarantees that the bandwidth allocated to a real-time call never drops below the minimum bandwidth requirement of the call ensuring that the corresponding application can still function at an acceptable level [23].

Prasad et al, in [15], have proposed a generic framework for modeling and prediction of user movement in wireless networks. We have used a hidden Markov model to perform mobility prediction in a wireless network in conjunction with a closed loop feedback control system. The proposed scheme is computationally simple and can be incorporated into existing wireless network architectures. The main advantage of this technique is that it is a general framework and can be used to effectively manage mobile users in a network. It make use of a predictor that can accurately estimate the next location visited by a mobile user, given current and historical movement information. In sequence, a feedback control system provides information regarding network resource and utilization for better overall resource management.

Nicholson et al., in [12], explore mobile devices with multiple network interfaces that perform the MAC layer in software. Juggler is a link-layer implementation of an 802.11 virtual networking service. The ultimate goal is enhance data throughput when wireless bandwidth is superior to that of an APs wired, backend connection, by striping data across many networks. To do it, Juggler switches between wireless networks in just over 3 ms, or less than 400 ms if networks share the same wireless channel. They show how mobile clients can enjoy nearly instantaneous 802.11 handoff by reserving 10 percent of the radio duty cycle for background AP discovery, while minimally impacting foreground transfers.

III. Background

During the design and implementation of our solution we reused and adapted several ideas, standards and mechanisms, previous published. The majority of these works are related to connectivity management in heterogeneous wireless environments. Connectivity management focuses primarily on mobility and network QoS mechanisms needed to access ubiquitous services offered through pervasive smart spaces [20]. For this work, the fundamental components of connectivity management are: mobility prediction, handover mechanisms and mobility management as presented in Fig. 1 (1). In addition, the Fig. 1 (2) shows the historical context database in the center of a control loop, where all entities store or search data there.

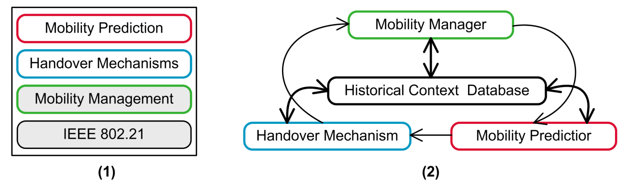

Fig. 1. Fundamental components of the connectivity management (1), and the execution loopl and data flow among them (2).

Mobility prediction models are applied to predict the user's geographical localization, generally, aiming at pre-provision network resources in an eventual handover. When the user reaches a location where the current network does not fulfill the application's QoS requirements, it is necessary to activate a **handover mechanism**. In order to keep the transport connections alive it is necessary to make use of a **mobility management** mechanism, like Mobile IP at layer 3 or TCP Migrate [18] at application layer, to cite two. Moreover, an evolving standard, called IEEE 802.21 [8], aims seamless roaming between heterogeneous networks enabling information exchange cross network layers.

In the following subsections, we present these essential components. They provide the basic knowledge and infrastructure used to design and implement our solution at application layer. The subsections are organized in 3 main topics: mobility management, handover mechanisms and mobility prediction.

A. Mobility Management

While on the move, a user cannot rely on a single wireless network, but must exploit a wide variety of connectivity options [14]. In addition, it is necessary maintain all the user's connections alive when the mobile device reconnects from one network to another. The mobility manager focuses on enabling different mobile networks to interoperate with each other at lower layers to ensure terminal and personal mobility [3].

During the design phase, we had in mind that the action of change the point of attachment, also called handover or handoff, triggers a set of mechanisms to keep the user connected. First, the physical connection must be handled (at layer 1 and 2) and, in sequence, the IP connections (layer 3) need to be managed to not disrupt any transport connection. To do it, there is a developing standard called 802.21 at layer 1 and 2, and mechanisms such as Mobile IP at layer 3, both are briefly described in the following.

The IEEE 802.21 draft standard describes a media independent handover architecture that provides link-layer services to enable handovers among different radio air interfaces [8]. In addition, the standard is capable of exchange networks' information to upper layers, aiming at optimizing handovers. An interesting scenario is the inter-radio mobility between 802.16m (100 Mbps, 250km/h) and 802.11 VHT (1 Gbps, low velocity) that will satisfy both the low-mobility and fully mobile user velocity versus data rate requirements [6]. This type of efforts justifies the several researches related to context-aware handovers management being conducted, because in a near future will be possible deploy and test them effectively.

One layer up, mobile IP provides layer 3 mobility to allow mobile devices to roam around wireless LANs maintaining intact all transport sessions [24]. After a handover, the mobile device react sending bind updates to its home agent and its correspondent nodes independent of the locality and magnitude of its movement. Consequently, the same level of package signaling load is introduced in the network regardless the user's mobility pattern [1]. This problem motivates the many efforts addressing scalability and performance transparency of mobile IP, always considering the growing of the Internet and the increasing number of mobile nodes. This is a key functionality to allow on-line seamless full mobility in all-IP networks environments.

We are not investigating solutions to the problems related to mobility management. The aforementioned mechanisms were used as previous infra-structure to design and implement a consistent solution at application layer. Thus, when a handover occurs, we assume that 802.21 will handle connectivity in the lower layers, and solutions like mobile IP will be able to keep all the IP-connections alive. In this way, our community-centric solution is a complementary approach to gather pertinent context data placed on top of these techniques.

B. Mobility Prediction

Mobility prediction is the process of forecast a roaming user's next location given their current one and prior movement history [19]. It is an elementary mechanism for systems that aims service provisioning [14], [16], [17], [22], like bandwidth reservation when the user is handing over between access points. Our implementation uses the *connectivity graph* of a particular person to provide data the movement predictor, and data from all community members' to estimate network conditions, the details are examined in Section IV. To accomplish it, we adapted a Markov predictor similar with the one used in the BreadCrums system [14].

This type of predictor takes as input a sequence of symbols (a_1, a_2, $\ldots$, a_n) as a history string, and tries to predict the next symbol from the current context, that is, the sequence of the k (order) most recent symbols in the history (a_{n-k+1}, $\ldots$, a_n) [19]. The second-order Markov model applied works with states composed by the position where the device was during the last state, and its current position. In the end, the predictor returns a list of locations and their respectively probability of movement. Which, also, can be represented as a graph, where the points are the locations and the edges' weight are probabilities of the user take that direction.

The predictor's accuracy is limited by the density of the available historical information and by the precision of the estimated localization, of a particular mobile user in question. In this case, the availability depends on the community members' behaviors inside the building. Our indirect contribution consist in make available a set of rich and summarized QoS data result of collaboration among community members', at a particular place, aiming at better predictions and, consequently, better handovers decision making.

C. Handover Mechanisms

Handover mechanism controls the change of a mobile device's point of attachment, in order to maintain connection with the moving node during active one or many data transmissions [20]. This work focus on improve indirectly the handover decisions applying a feedback loop where the collaboration of mobile users can converge to a better IP-connectivity experience of all participants. Focusing on real usage and validation, we implemented a mechanism similar to VIRGIL, proposed in [13]. It scans all open access points available, test IP-connectivity of each one associating and trying to get a DHCP address, if the tests succeed meaningful properties are stored, e.g. network name, MAC address, signal strength and opened transport ports. One key difference is that we don't perform a set of tests in the access point, instead just log real QoS experiences of the users.

In the current work, we discuss a comparative evaluation confronting the QoS performance of 3 connectivity management techniques namely and described here:

- **Strongest signal strength** (SSS): is the plan of action currently used by the majority of common operating systems for automatically selecting an access point simply scanning for access points and then choosing the unencrypted one or one stored in the profile with the highest signal strength [6].
- **Mobility prediction**: uses an order 2 Markov mobility predictor [19] using as input the particular user movement history and non-structured data collected by all users, similar to the approach used in [14].
- **Community-based**: use the same predictor cited above accessing the socialized information as historical data [11].

The results and discussion are present in Section V. The next section, IV, describe in details our community-base solution for wireless connectivity management.

IV. Community-based approach

In our previous work [11] we designed a specialized virtual community to assist the process of gather, combine and share connectivity context data. In the current work we complement these functionalities adding a function to summarize quantitative QoS metrics and verify the feasibility of the proposed solution. In the following subsections we present the methodology and the model applied, and how the quantitative QoS data is summarized and delivered. We examine them in subsections IV-A, B, C and D, respectively.

A. Methodology

The solution is composed of three key elements: mobile device, mobile user and a virtual community as shown in Fig. 2. In this figure, at the mobile device, client side, we assume that are implemented a mechanism responsible to manage the connectivity. Moreover, there are mobile users carrying on these devices and experiencing particular quality of service in a shared location, e.g. a building, a neighborhood or a downtown. All the mobile users are members of a community, server side, where they can upload their IP connectivity experiences and download the combination of all members' feedbacks [11].

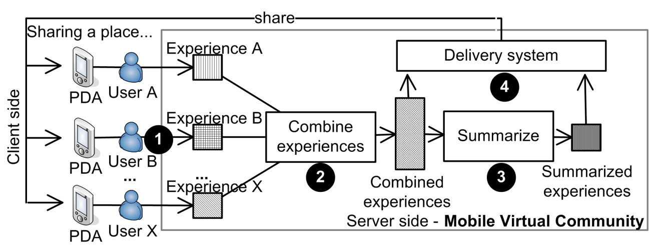

Fig. 2. Methodology applied: (1) Gather user's IP-connectivity experiences; (2) Combine the experiences; (3) Summarize the numerical QoS related data; (4) Share the combination in a virtual community.

The methodology applied has four fundamental methods disposed in a feedback loop also spotted in Fig. 2. A mobile user's IP connectivity experience at a particular place can be acquired, stored and upload to a mobile virtual community, it's is our first method as show in Fig. 2 (1). The client side is in charge of fetch context data, e.g. date, time, throughput and signal strength, using the operational system's programming interfaces and supported protocols. A virtual community can be created to put together people interested in sharing their unique feedbacks. The second method, Fig. 2 (2), is in charge of combining all users' experiences in an interesting format. To achieve it, we apply a graph based model that combines all the community members' connectivity experiences in a connectivity graph [11]. This model is briefly described in subsection IV-B.

At a given place, several mobile users would had a connectivity experience. In this case a search for feedbacks will return many context instances. To solve it, we summarize, Fig. 2 (3), the quantitative data using common statistical tools described later in subsection IV-C. Lastly, this information needs to be accessible to all community's members, Fig. 2 (4). We explore the RSS (Really Simple Syndication) feeds functionalities to make available the combined and summarized experiences data in channels, as described later in subsection IV-D.

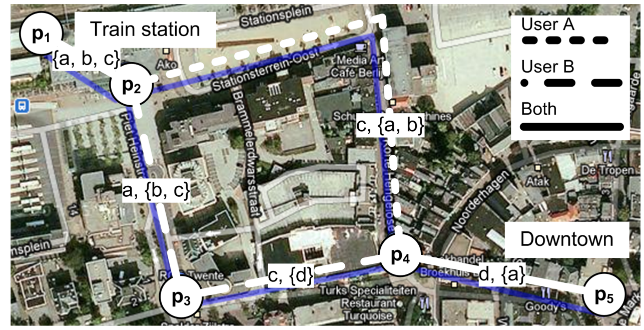

Fig. 3. Connectivity paths of two mobile users walking from the train station to the city's downtown.

B. Model

Considering a hypothetical scenario, two persons carrying a mobile device can have IP connectivity on the move between the train station and a restaurant in downtown, as shown in Fig. 3. During the roaming, may it be necessary to perform eventual handovers at particular points to keep the device always connected, let's call it *handover point*. In the end of the trip, the route connecting the *handover points* of each user is named *connectivity path*. In others words, each user's time-spaced set of IP connectivity experience is called *connectivity path*.

In the Fig. 3, we overlap the *connectivity graph* of two users, A and B in a hypothetical path between the train station and downtown. The two users had a different set of handover points in the path, however, there are few common points among the five ones (p_1 to p_5) exposed. The combination of these paths represents all the information discovered by the community members about this particular wireless environment.

The *connectivity path* shows the APs used and the other ones available at the handovers points. The socialized media, called *connectivity graph*, combine all the community members' experiences showing the APs used and the people that had experiences there. To illustrate, the Fig. 4 shows the digraph, G_{conn}, that summarizes the experience of the two users aforementioned. The labeling of the edges is $< currentNet >, < \{allUsers\} >$. All the five points are connected by edges representing the wireless network used between five handover points and how many users took the same choice.

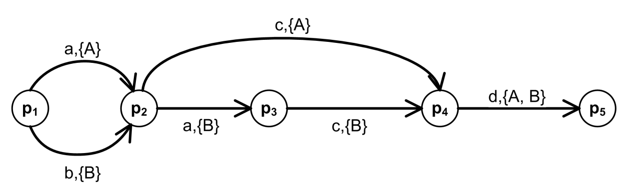

Fig. 4. Two connectivity paths, overlaped earlier in Fig. 3, combined in one connectivity graph G_{conn}.

C. Summarizing

For us, IP connectivity experience is a set of context data, including QoS parameters, related to an access point used, by a mobile user with a portable device, at a certain location in a specific time. The set of context data to comply with this definition was determined and structured considering the vision of Sun et al., in [20], who stated that mobility is a logical concept rather than a physical one, in which mobility means the change of the logical location of network's access points instead of user's geographic position.

We used in our prototype set of context data from the mobile user, mobile device, time, location, access point and transport connections, as shown in Fig. 5. Handling this set of information properly we can answer these questions: (1) Who is the mobile user? (2) Where and when he or she was using IP-connectivity? (3-4) Attached to which access point? (5) Doing what? (i.e. transport connections opened). Answering these questions properly we are able to perform context-aware handovers comparing the current context and using historical QoS data condensed in the *connectivity graph*, often referred as IP connectivity experiences.

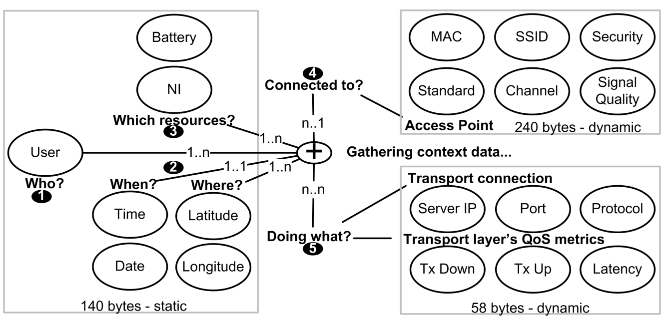

Fig. 5. The set of context data gathered and stored.

At each handover point the users can fetch a set of quantitative QoS metrics, e.g. throughput, latency and signal quality as shown in Fig. 6. It is known that this databases can have a consider size that is not manageable by a mobile device. In this way, we propose to condense the quantitative data using common statistical tools, such as quartiles. Quartiles divide the historical data into 4 quarters each containing 25% of the data. It is simple to compute and the decision making algorithms can have a more details about the particular AP at the specific location.

D. Delivery system

The fundamental idea of our methodology is create a feedback loop to allow all community members to share their IP connectivity experiences aiming at better experiences for everyone. To achieve it we propose make use the RSS feed's capabilities to share the connectivity graph inside the virtual community.

The feeds are a collection of standardized XML-based formats used to publish frequently updated applications, e. g. blog posts and news headlines. It provides advantage to

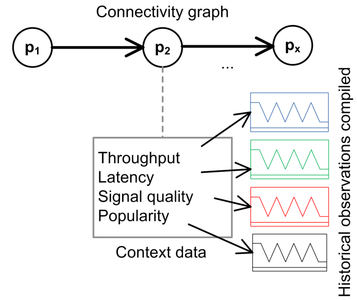

Fig. 6. The quantitative set of context data in a particular handover point.

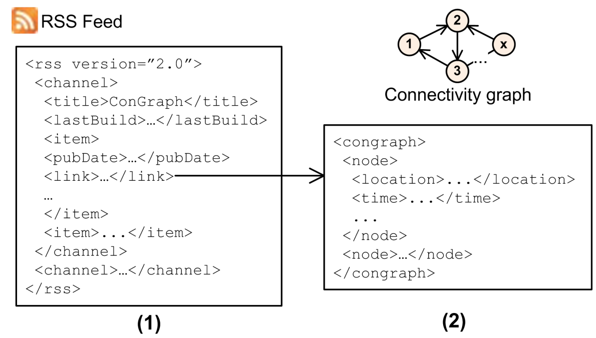

Fig. 7. A fragment of a RSS feed file (1), and a textual representation of the connectivity graph (2).

publishers, in this case a community, allowing them combine digital content automatically. Likewise, benefit readers who want to subscribe to timely updates from web applications or to aggregate feeds from many sources into one place.

The document can include full or summarized text, and metadata such as publishing dates and authorship, as shown in Fig. 7 (1). The tag $< channel >$ define a channel which the user can subscribe to. Each channel has a tag $< lastBuildDate >$ that is used by the clients to verify if there updates in the channel. The information is published in a channel as a item, tag $< item >$. When a connectivity graph is created or updated a new item is added to the channel. With that, the graph's evolution is recorded and is available for all community members'. In case of a malicious attack the respective item can be deleted to isolate the problem.

Every time that a graph is up-to-date one new item is created to make a reference to the new information with the respective timestamp. In this way, the community members can get updated just reading the newest item available. Inside a item the connectivity graph is referenced in the tag $< link >$ that point to a textual representation of the graph as show in Fig. 7 (2).

A community member can subscribe to several feeds as shown in Fig. 8. This figure shows a web user interface, in (1) we have personal user data, in (2) the feeds, and in (3) the friends from Google opensocial. In addition, each smart environment can have an specific channel. The storage overhead depends on how many channels the user is subscribed to. Thus, the scalability of our model is enhanced by the creation

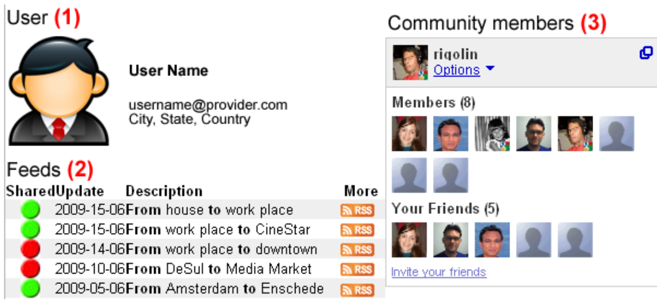

Fig. 8. Web user interface, (1) personal data, (2) subscribed feeds and (3) community members.

of channel for each specific smart environment. For instance, inside a university campus each building can have a channel updated by the community of people that work or study there. Hence, the users can subscribe to the channels of the buildings that they visit frequently, which can be done automatically using a localization system and knowing the feed's address.

V. Feasibility

During the prototype effort we faced some issues that put in doubt the feasibility of our solution. In this section we describe these experiences with focus on storage size, power consumption and QoS metrics. To start, lets take a look on a snapshoot of the process in execution at the mobile device and the virtual community in Fig. 9. This figure provide an abstract view of the RAM and primary memory of the client and server side. Starting at the client side, where the mobile device is having an experience and, consequently, needs to gather and store the connectivity path, and, also make use of the historical data to perform handovers. When its finished, the mobile device upload the experience to the server side. The server store the experience and combine them in a connectivity graph as examined earlier in section IV.

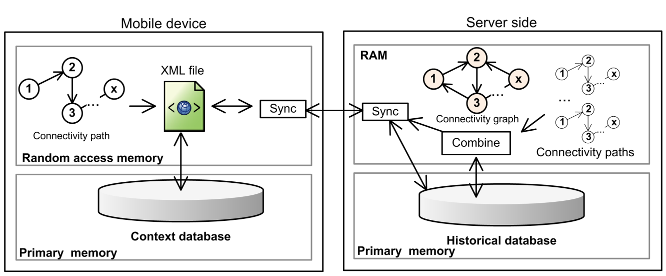

Fig. 9. The effective usage of the context data.

The following subsections examine in details the overhead related to storage size, power consumption and experimental in the subsections V-A, B and C, respectively.

A. Storage size

At each scan the context data size can vary in function of the number of networks available and transport connections opened. Taking as example the set of data in Fig. 5 we can identify the static and dynamic subset of data in terms of

Place	APs/scan [13]	Per scan (bytes)
Downtown	2.4	576
Residential	2.00	480
Suburban	1.80	432

TABLE I
Estimative of context data collected in 3 different urban areas: downtown, residential and suburban.

size. At a particular scan, the data related to date, time and mobile user will appear just once in the set, also referenced as static subset. In other hand, the set of data related to visible wireless networks will have an instance for each individual AP, and the same will occur to the opened transport connections. Considering these parts we can determine the context data storage size, C, using the Equation 1:

$$\mathbf{C}ontext = c + \sum_{i=1}^{n} x_i * s_i. \tag{1}$$

Where c is the size of the static part of the context data, x_i is the number of the dynamic's subset instances, and s_i is the size of this particular data subset. As an example, the data related to the mobile user, date and time sums approximately 140 bytes, $c = 140$. The AP's data has about 240 bytes, $s_i = 240$ and let's assume that this arbitrary scan found 3 APs, $x_i = 3$. In addition, the set of data related to the opened transport connections is 58 bytes, $s_2 = 58$, and lets say that the mobile device has 2 active connections, $x_2 = 2$. Applying the Equation 1 we have: $C = 140 + (3*240) + (2*58) = 976$ bytes.

The work reported in [13] quantified the average number of available access points in different urban areas, namely: downtown, residential and suburban. The Table I shows these numbers and the average size of the context data per scan disregarding the number of active transport connections.

While roaming, the probability of the user take a connectivity opportunity is strong related to how often the device perform network scans looking for APs available, as shown in Table II. In this table, we have the scan interval, in seconds [7], for each probability of take a connectivity chance available. Also, shows the storage size for each probability after 1 day of continued scan of one mobile user.

There is a linear grow till 80%, and a disproportional grow for more than 80%, referenced as $> 80\%$, about 16 times bigger than less than 80% of connectivity chance. These results leads us to think about a smart mechanism to perform handover using our *connectivity graph*, it is going to be discussed in the next subsection V-B.

At this point we can generalize a function, Equation 2, to estimate the storage size of the context database in function of time and numbers of users.

$$\mathbf{D}atabase = \sum_{i=1}^{u} \frac{t_i}{o} * C_i. \tag{2}$$

Where u is the number of users, t_i the user's total usage time, o is the scan interval that varies in function of the

Opportunity (%)	Scan interval (s) [7]	Size (Kbytes)
20	1500	32.40
40	1000	48.60
60	500	97.20
80	250	194.40
>80	<15	3,240.40

TABLE II
DATABASE SIZE GROWING IN FUNCTION OF THE PROBABILITY OF TAKE A CONNECTIVITY OPPORTUNITY.

probability of take a connectivity opportunity, earlier presented in Table II, and C_i is the context data's size in a particular scan. Using this relation we can estimated the database grow in function of time and number of users for each probability of take a connectivity opportunity.

B. Power consumption

We can, also, verify how much power is necessary to perform scans aiming at discover connectivity opportunities during a roaming. We assume that the scanning operation is equivalent in terms of power consumption to spending $5ms$ in reception mode at $300mW$ [5]. The Fig. 10 (1) shows the estimated power consumption in function of time, for each probability of take a connectivity opportunity. We can see that that the power consumption grows faster for higher probabilities of take a connectivity opportunity.

It is more evident in Fig. 10 (2), where we can see the power consumption in function of the probability of take a connectivity opportunity. With these two charts we can see how expensive is take more than 80% of the opportunities. It motivates the design of a smart technique to save resources. In this work, it is done identifying the areas with the highest handover density in the *connectivity graph*.

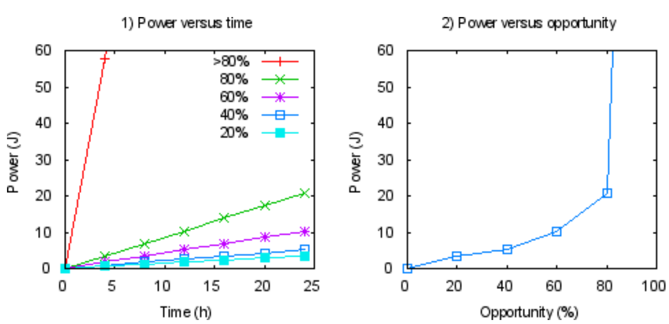

Fig. 10. Power consumption in function of time (1) and in function of the probability of take a connectivity opportunity.

C. Experimental results

We executed our prototype for approximately 4 months with 3 users in an indoor testbed with approximately $1,600m^2$. Using the *connectivity graph* we discovered the areas with highest handovers density as shown in Fig. 11. In this figure, there are five points, in the center of each area, namely here as: $Alpha$, $Bravo$, $Charlie$, $Delta$ and $Echo$.

In first experiment, the mobile user took a path that goes through the points the five way points, $Alpha$, $Bravo$,

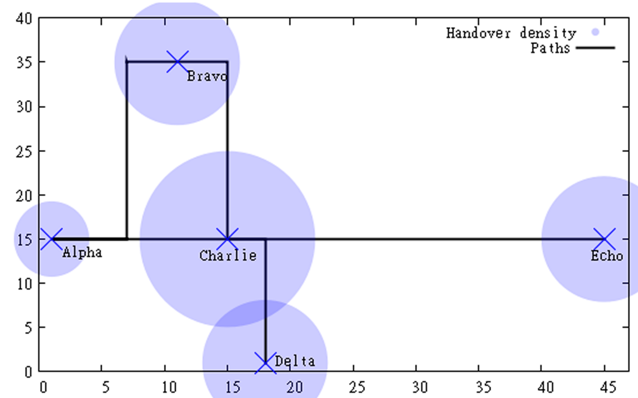

Fig. 11. Areas with highest handover density in the building.

$Charlie$, $Delta$ and $Echo$, in this sequence. The data size accumulated in function of time and the power consumption until the end of the path are shown in Fig. 12 (1) and (2). With a consistent connectivity graph we can apply a smart way to perform scans in the wireless environment. The Fig. 12 (1) shows how the databases size grows using a constant loop to take more than 80% of the connectivity opportunities and confronts with 2 preemptive approaches to take more than 80% and 80% percent of the opportunities, in this sequence. The 2 preemptive approachs has saved significant storage space.

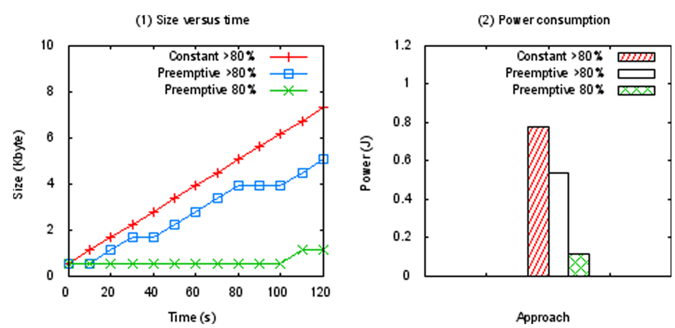

Fig. 12. Contex data size (1) and power consumption (2) while walking through a path.

Another benefit of our solution is the positive impact on transport QoS metrics. The Fig. 13 (1) shows the throughput (rx) in function of time during a roaming between $Bravo$ and $Delta$ earlier illustrated in Fig. 11. In this figure, The mechanisms are referenced to as A, B and C, SSS, mobility predictor and comunity-based, respectively. The community-based, C solution has a better performance than the other two with an average of 3.8 Mbps in comparison to 3.3 and 2.9 Mbps, B and C. The Fig. 13 (2) compare the average signal quality of the three methods, 55, 64 and 78%, SSS, mobility predictor and community-based, in this ordem.

The first one, A, did 3 unnecessary handovers, and avoids one necessary to keep the highest throughput walking from $Alpha$ to $Charlie$. Even with an excellent coverage the reactive decision making system performed poorly. One problem observed is the random connection to access points in others

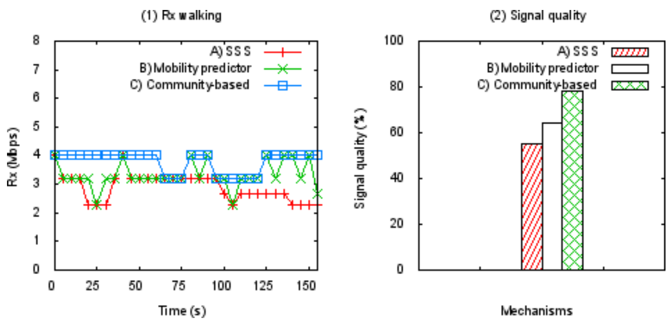

Fig. 13. Throughput in function of time (1), and average signal quality during the roaming (2).

floors or far way, even with a better one available. The mobility prediction, B, did one handover unnecessary. Probably, just one user needs more time to discover the environment. Our solution choose a close optimal combination of access points in the path, also with lower latency 1.5 ms, against 2.6 and 3.4 ms in average, A and B.

VI. Acknowledgment

This work is supported by Capes (grant 3993-08-6) and FAPESP (grant 2008/06862-6).

VII. Conclusions

This paper discussed the benefits of handling network context data inside a virtual community by measuring the overhead of storage size, power consumption and how it impacts on QoS metrics, such as throughput, latency and signal quality. This kind of data plays a fundamental role in the context-aware connectivity management solutions, because its quality has direct impact on mobility and QoS predictors.

We started discussing the main related works and the fundamental background of connectivity management. In sequence, we examine the community-based approach to handle network context data, and finished discussing its feasibility. We provided an analytical approach to describe the storage overhead, and used experiments results to discuss power consumption and QoS metrics.

The experimental results show that the socialized connectivity graph can be used to improve QoS experiences, reduce the storage overhead and power consumption. Another valuable contribution is the delivery system that allow the mobile users get updated about the wireless environment periodically. A topic for further research is determine optimal QoS conditions and localization to perform updates using the delivery system.

References

[1] C. Castelluccia. Hmipv6: A hierarchical mobile ipv6 proposal. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(1):48–59, 2000.

[2] J. Chan, S. Zhou, and A. Seneviratne. A qos adaptive mobility prediction scheme for wireless networks. In *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, volume 3, pages 1414 –1419 vol.3, 1998.

[3] P. Chan, R. Sheriff, Y. Hu, P. Conforto, and C. Tocci. Mobility management incorporating fuzzy logic for heterogeneous a ip environment. *Communications Magazine, IEEE*, 39(12):42–51, dec 2001.

[4] C. Cheng, R. Jain, and E. van den Berg. Location prediction algorithms for mobile wireless systems. pages 245–263, 2003.

[5] C. Desset, N. Ahmed, and A. Dejonghe. Energy savings for wireless terminals through smart vertical handover. In *Communications, 2009. ICC '09. IEEE International Conference on*, pages 1 –5, 14-18 2009.

[6] L. Eastwood, S. Migaldi, Q. Xie, and V. Gupta. Mobility using ieee 802.21 in a heterogeneous ieee 802.16/802.11-based, imt-advanced (4g) network. *Wireless Communications, IEEE*, 15(2):26–34, april 2008.

[7] M. H. Falaki. Wlan interface management on mobile devices. Master's thesis, University of Waterloo, Waterloo, Ontario, Canada, 2008.

[8] IEEE802.21. *IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services*. IEEE, New York, December 2009. Draft standard D9.0.

[9] T.-H. Kim, Q. Yang, J.-H. Lee, S.-G. Park, and Y.-S. Shin. A mobility management technique with simple handover prediction for 3g lte systems. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 259 –263, sept. 2007.

[10] A. LaMarca, J. Hightower, I. Smith, and S. Consolvo. Self-mapping in 802.11 location systems. In *UbiComp 2005: Ubiquitous Computing*, volume 3660 of *Lecture Notes in Computer Science*, pages 87–104. Springer Berlin / Heidelberg, 2005.

[11] R. R. F. Lopes, B.-J. van Beijnum, and E. S. Moreira. Managing wireless IP-Connectivity experiences as mobile social media. In *International Conference on Ultra Modern Telecommunications (ICUMT 2010 (Telecommunications))*, pages 1–8, Moscow, Russia, 2010.

[12] A. Nicholson, S. Wolchok, and B. Noble. Juggler: Virtual networks for fun and profit. *Mobile Computing, IEEE Transactions on*, 9(1):31 –43, jan. 2010.

[13] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall. Improved access point selection. In *MobiSys '06: Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 233–245, New York, NY, USA, 2006. ACM.

[14] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 46–57, New York, NY, USA, 2008. ACM.

[15] P. Prasad and P. Agrawal. A generic framework for mobility prediction and resource utilization in wireless networks. In *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, pages 1 –10, 5-9 2010.

[16] A. Rahmati and L. Zhong. Context-for-wireless: context-sensitive energy-efficient wireless data transfer. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 165–178, New York, NY, USA, 2007. ACM.

[17] A. Sleem and A. Kumar. Handoff management in wireless data networks using topography-aware mobility prediction. *Journal of Parallel and Distributed Computing*, 65(8):963–982, 2005.

[18] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 155–166, New York, NY, USA, 2000. ACM.

[19] L. Song, U. Deshpande, U. C. Kozat, D. Kotz, and R. Jain. Predictability of wlan mobility and its effects on bandwidth provisioning. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–13, april 2006.

[20] J.-Z. Sun and J. Sauvola. From mobility management to connectivity management. *Computers and Communications, IEEE Symposium on*, 0:307–312, 2005.

[21] M. Tlais and F. Weis. Centralized mobility prediction support in a hierarchical architecture. In *Wireless and Mobile Communications, 2006. ICWMC '06. International Conference on*, pages 81 –81, 29-31 2006.

[22] E. Z. Tragos, A. Mihovska, E. Mino-Diaz, P. Karamolegkos, P. Vlacheas, and J. Luo. Access selection and mobility management in a beyond 3g ran: The winner approach. *Telecommunication Systems*, 42(3-4):165–177, December 2009.

[23] E. Yami and V. Vakili. Adaptive bandwidth allocation and mobility prediction in mobile cellular networks. In *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. Third International Conference on*, pages 343 –348, 15-18 2009.

[24] H. Yokota, A. Idoue, T. Hasegawa, and T. Kato. Link layer assisted mobile ip fast handoff method over wireless lan networks. In *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 131–139, New York, NY, USA, 2002.