Managing wireless IP-connectivity experiences as mobile social media

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*—Social media is the meaningful digital content that results from interaction or collaboration between users through web-based applications, e.g. blogs and virtual communities. While on the move, people can create or consume mobile social media using portable devices with IP-connectivity. Connectivity is a fundamental resource for these users that are going from one location to another using heterogeneous multi-access systems. Our hypothesis is that mobile user's IP-connectivity experiences can be represented, implemented and handled as mobile social media. Thus, these set of particular experiences can be combined in a mobile virtual community to be shared among the members. The ultimate goal is improve the quality of member's experiences in a common environment. We proposed a model to represent and combine user's connectivity experiences. A prototype has been implemented, and experiments were performed in an indoor testbed during few months. The results show that our solution converges to better experiences, even with a small community.**

***Index Terms*—IP-connectivity, social media, user mobility.**

I. Introduction

The availability of ubiquitous wireless IP-connectivity, portable multi-homed devices and web applications, e.g. virtual communities, enhanced people's communication adding freedom of time and space. One interesting form of communication and collaboration is social media. It joins people, technologies and rich digital content that enables users to share their experiences with other ones, thus building a shared meaning among virtual communities [6].

In other words, social media can be a set of experiences in real life mediated by technologies as a form of digital content [11]. The feasibility of this relevant digital content as mobile social media is dependent on applications such as *mobile virtual community*, just referred as *community*. This kind of community connect people with common interests and who are already together, sharing the same locality, giving them the opportunity to take advantage of their close proximity to form more tightly-coupled virtual communities [5]. In this paper we investigate the hypothesis that mobile user's IP-connectivity experiences can be modeled and implemented as mobile social media, and if it can be used to effectively to improve IP-connectivity experiences.

The fundamental idea is to use a community-centric way to handle the context data used to manage IP-connectivity in environments with overlapping access networks of different technologies and providers. We propose a graph based model to represent mobile user's IP-connectivity experiences and use it as social media. This media enables people sharing a place to collaborate inside a community. These users, carrying on mobile devices, are able to discover wireless connectivity islands, in a cheap and accurate way, converging to an optimal IP-connectivity management of all community members.

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. Each user's time-spaced set of IP-connectivity experience is called *connectivity path*. We focused in how to gather, share, combine, and update these set of data, arguing that mobile virtual communities are an interesting place to reach people that shared a wireless environment in daily life. The combination of all members' connectivity paths is called *connectivity graph*, and it is the media socialized.

A prototype has been implemented using standard web technologies. We validated this idea showing that handover decision making mechanisms can have better results with our data model. Our prototype was used in an indoor environment for few months, by a small community of mobile users. Even with a few members, the results show that our solution yields to better QoS metrics than just use the typical operational system's reactive management or a mobility prediction model using non-shared data. It is a complementary approach for context-aware solutions that intends to allow full mobility across heterogeneous wireless networks.

The contributions of this work are: (i) the methodology used to represent IP-connectivity experiences as mobile social media; (ii) an algorithm to combine distinct connectivity paths in a connectivity graph; and, (iii) the discussion related to the implementation and experimentation of a prototype. Envisioning further, mobility and QoS prediction models, like in [1], [7], [8], [10], [14], [15], can access and sample the connectivity graph aiming at optimal results.

This paper is organized as follows. First, the previous works in section II. The methodology applied to represent and share connectivity experiences is presented in section III. Then, the key implementation details are discussed in section IV. In section V, the experiments and their results are described and discussed. Finally, section VII concludes the paper.

978-1-4244-7286-4/10/$26.00 ©2010 IEEE

II. Related Work

Nicholson et al. [8] explored the fact that people are creatures of habit that takes similar paths every day. Complementing this vision, people are social creatures that interact and collaborate with other ones in daily life. Here, this characteristic is investigated within a virtual community and handling IP-connectivity experiences as mobile social media, in a context-aware way. There are several context-aware solutions to manage IP-connectivity in the literature [8], [9], [13], [14], [15], [17], [18]. Most of them combine mobility prediction and handover mechanisms in a particular way aiming at ubiquitous connectivity.

In this section, we examine the 3 works used as basis and motivation for the present one, which are: BreadCrums [8], HOPES (Handoff Prediction and Enhancement Scheme) [10] and QoSIS.net [17]. All of them use somehow a set of historical data in order to improve the handover decision making. We argue that our community-centric approach is complementary to these ones, and it is how this subsection unfolds.

BreadCrumbs explores the derivative of connectivity of a mobile user to perform context-aware handovers. The system maintains a history of observed networking conditions and a personalized mobility model on the user's mobile device. It works tracking the device's owner, and use together the predictions of the mobility model and the access point's quality database to generate connectivity forecasts. The authors showed that these forecasts can be used to allow applications take domain-specific action in response to upcoming network conditions. The ultimate goal is to improve communications performance, or to reduce power consumption, or a combination of both [8].

Sleem at al. [10] argue that the network can predict the user mobility. Then, it is possible to take action, preemptively, before the movement happens, e.g. network services in the new location can be used without conventional registration. To achieve this, the authors designed HOPES that aims at exploring the use of prediction techniques in mobility managements in order to improve the end-to-end traffic quality. It uses a topography-aware predictive approach that combines the mobile host's movement history, current state, and the topography of the network's cells. This combined knowledge is used in predicting intra-cell mobility, inter-cell, and hence the need for a handover. The proposed architecture provides the network with timely information necessary to proactively respond to user movements instead of passively handling it after it happens [10].

The QoSIS.net system offers anywhere, anytime, anyhow, accurate location-based QoS predictions service via user-collaborative information sharing [17]. A map of user's historical mobility and predictions is kept on his or her device. This map is automatically updated, either at regular time-intervals or at the moment a prediction is requested. The prediction system applies data mining algorithms to retrieve information patterns from a historical database of QoS measurements performed by mobile users and providers. A request for a predictions service, results in instantaneous processing of large quantities of QoS information by prediction engine [16]. There is no technical evidence of how the users' mobile device gets updated automatically, and the complexity of time of the QoS predictions requests.

There are several works investigating handovers optimization using mobility prediction and relying on public or private QoS databases. However, few efforts are direct to provide better quality data in terms of facility to access, handle and update. We believe that BreadCrums can have better forecasts using our community-based solution if compared to just use common wardriving databases. HOPES can use the connectivity graph socialized to parameterize and choose algorithms. Finally, QoSIS.net would use our solution to delivery data to the users.

III. Methodology

Our community-based solution consists of three key elements: mobile device, mobile user and a virtual community as shown in Figure 1. In this figure, at the mobile device, 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 where they can upload their IP-connectivity experiences and download the combination of all members' feedbacks, i.e. all the knowledge about the particular environment collected by the community members'.

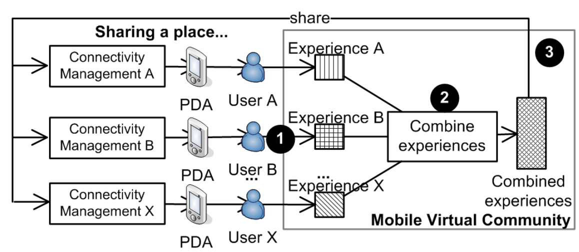

Fig. 1. Methodology applied: (1) Gather user's IP-connectivity experiences; (2) Combine the experiences; (3) Share the combination in a virtual community.

The methodology applied has 3 fundamental methods disposed in a feedback loop also spotted in Figure 1. 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 Figure 1 (1). A virtual community can be created to put together people interested in sharing their unique feedbacks. The second method is in charge of combining all users' experiences in an interesting format, Figure 1 (2). Lastly, this information needs to be accessible to all community's members, Figure 1 (3).

In the following subsections, we describe each one of these methods, focusing on how they were designed to help to verify our hypothesis. They are responsible for gathering, combining and sharing IP-connectivity experiences. We examine them in subsections III-A, III-B and III-C, respectively.

A. Gathering

Illustrating a hypothetical scenario, a person possessing a mobile device can have IP-connectivity on the move between his or her work place and a restaurant in downtown, as shown in Figure 2. During the roaming, it is necessary to perform eventual handovers at particular points to keep the device always connected, let's call it *handover point*. At these points, the set of context data [9], [15], often referenced as *IP-connectivity experience*, is gathered and the handover decision is taken based on the current network QoS information and application requirements. In the end of the trip, a route connecting the handover points is referred to as *connectivity path*. In others words, each user's time-spaced set of IP-connectivity experience is called *connectivity path*.

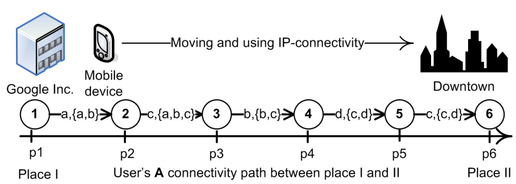

Fig. 2. A hypothetical connectivity path, and its formal representation.

The connectivity path of a mobile user can be modeled as a directed graph, G_{path}=(V_{path}, E_{path}), as shown in Figure 2. Where V_{path} is a set of handover points and E_{path} is a set of edges. Each edge indicates the network being used and the set of available networks with labeling like $< currentNet >, < \{availableNets\} >$. For instance, from vertex p_1 to p_2 the user was connected to network a and were available at that place a and b. The connectivity path summarizes all the context data gathered during the user's trip.

B. Combining

Along a shared path, not necessary with the same starting point and end point, mobile users may experience different QoS performance using the access networks available. As a result, the handover points in every user's connectivity path can be different. However, there is a probability of the user take handovers almost in the same point. It varies in function of the number of mobile users, surroundings characteristics and the size of overlapping areas of two or more wireless networks.

In the Figure 3, we overlap the connectivity graph of 3 users, A, B and C in a hypothetical path between place I and II. The 3 users had a different set of handover points in the path, however, there are a few common points among these 9 points (p_1 to p_9) exposed. The combination of these paths represents all the information discovered by the community members about this particular wireless environment.

The challenge now is represent it in an interesting way to be socialized within a community. We have chosen build a digraph, $G_{conn} = (V_{conn}, E_{conn})$, that joins these experiences

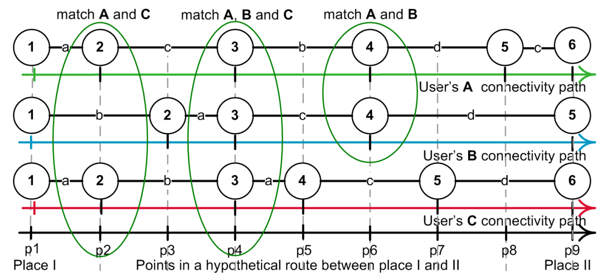

Fig. 3. Overlaping of 3 user's connectivity path.

in an useful way. The final result is the connectivity graph that summarizes the experience of the 3 users, as shown in Figure 4. The labeling of the edges is $< currentNet >, < \{allUsers\} >$. All the 9 points are connected by edges representing the wireless network used between handover points and how many users took the same choice.

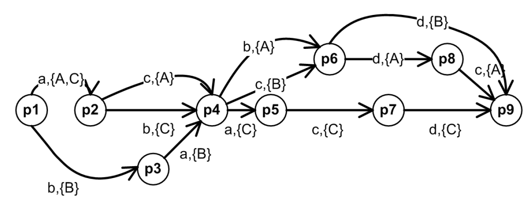

Fig. 4. The connectivity graph of the hypothetical example.

To build this digraph we need to identify the handovers points of each user and then look for matches searching in areas with pre-defined size. A simple way is choose an area with the diameter bigger than the error of the localization technique applied. In this case, we used a localization system with approximately 9 meters of error and implemented an area search with complexity of $O(nlogn)$. At the first time, the algorithm goes through all the users and they respective handovers points, as shown in Algorithm 1. Then the graph is constructed step by step adding nodes and edges. Before include the node is necessary look for location matches in the set of nodes already discovered.

Algorithm 1 COMBINE(G_{path})

Require: An digraph $G_{path} \neq null$.
Ensure: The connectivity graph G_{conn}.

```
count ← 1
for i = 0 to ALL_USERS do
    for j = 0 to G_path[i].V.length do
        if match(V_conn, G_path[i].V[j]) then
            G_conn.E[count] ← G_path[i].E[j]
        else
            G_conn.V ← G_path.V[j]
            add(edge[count].user, user[i].name)
            G_conn.E ← edge[count]
            count ← count + 1
        end if
    end for
end for
return G_conn
```

Once built, the complexity to add or remove a handover point in a connectivity graph is $O(logn)$ times the search complexity. A pertinent remark is that the graph reduction simplify the updates, because the historical data is not used by the algorithm all the time. However, scalability is a problem to this approach, since the complexity of time and size to build a *connectivity graph* restrict the coverage of large areas with high density of users, e.g. an university campus or a big city. That's why we choose mobile virtual communities to generate, update and share these digraphs. The next subsection, III-C, describes a way to increase the scalability just using a smart way to delivery information.

C. Sharing

In the end of our methodology we have the connectivity graph and need to share it among the members somehow. Also, taking in consideration that heterogeneous wireless IP networks are in constant change. We come up the hypothesis the context-aware connectivity manager should be updated about these environments as people get updated about news or weather. To achieve it we applied a publish and subscribe approach using RSS (Really Simple Syndication) feeds.

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 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 Figure 5. 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.

```
<rss version="2.0">
<channel>
  <title>IP-connectivity experiences</title>
   <link>http://community.org</link>
   <description>Connectivity graph</description>
   <pubDate>Tue, 10 Jun 2009 04:00:00 GMT</pubDate>
   <lastBuildDate>11 Nov 2009 09:41 GMT</lastBuildDate>
   <docs>http://community.org/rss.jsp</docs>
  <item>
    <title>Connectivity Graph</title>
    <link>http://community.org/rss.jsp</link>
    <description></description>
    <pubDate>Tue, 11 Nov 2009 09:41:01 GMT</pubDate>
    <guid>http://community.org/rss.jsp?item=573</guid>
  </item>
  <item>Next item…</item>
</channel>
</rss>
```

Fig. 5. Fragment of a RSS document.

We create a channel for the connectivity graph and for each user, anonymzing the data. 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.

With this approach, each smart environments can have an specific channel. The storage overhead depends on how many channels the user is subscribed to. Thus, scalability is added creating a channel for a 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.

IV. Implementation Details

A prototype has been developed in order to validate the proposed approach in a real-world scenario. Experiments were performed inside a university building with 2 floors covered by 3 known WiFi networks. To achieve this, was also necessary implement a beacon-based indoor localization system, since localization is a key context data for the proposed solution. All the information that compose the IP-connectivity experience were retrieved by an application running on the mobile device. In addition, at the server side we developed a specialized virtual community with enough features to put members to collaborate and proof our hypothesis.

In this section we present and discuss the key details of our prototyping effort. We start with the environment and hardware applied, in subsection IV-A. In subsection IV-B, we present the indoor localization system deployed. And, finally, we discuss how the mobile virtual community was developed, in subsection IV-C.

A. The Environment for Experimentation

The building has offices and laboratories distributed in approximately $3,200m^2$ divided in 2 floors. Both floors have an excellent WiFi coverage with 6 known 802.11a/b/g access points (CISCO Aironet 1200). The physical location of these access points in the building was used as reference to the indoor localization system described later in Subsection IV-B. The computational environment is composed by 2 machines at the server side and PDAs at the client side as shown in Figure 6. The workload, during the experiments, was divided between the 2 servers; one executing the community inside a container web, and the other one with the relational database system managing the entire set of context data collected.

All the servers and access points are connected to a gigabit Ethernet network. The mobile users were using PDAs (HP iPAQ 214) with a network wireless interface 802.11b/g and, using the operational system Windows Mobile 6. In this way, the set of context information that includes time and network QoS data was gathered using specifics programming interfaces of the operational system. Also, the interaction between the device and the virtual community were done by standard HTTP requests through the air interface.

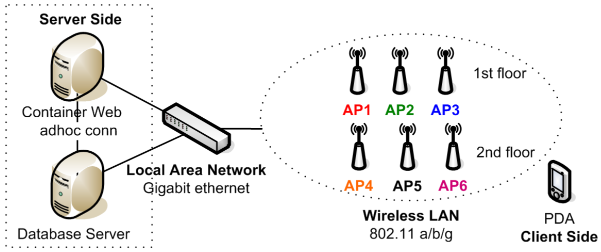

Fig. 6. Set up of the servers and access points in the building.

The access points were managing 3 different networks. We setup our client side application to change the network every time that a handover is necessary. Therefore, we make sure that all handover were done inter-cell, which means that the user is always jumping from a network to another. In other terms, all the handovers in the testbed is done between different IP domains that requires a new IP address and network gateway. At the PDA, we implemented 3 different connectivity management mechanisms to evaluate the solution by comparison. Details of the mechanisms and the experimentation results are presented and discussed in Section V-B.

B. Indoor Localization System

The mobile device's localization is estimated extracting the signal strength from beacons frames broadcasted by the known 802.11 access points in the building. It is also necessary have previous knowledge the physical location of each access point, apply a model to estimate the distance among the mobile device and at least 3 access points, and finally use triangulation to calculate the estimated coordinates.

First, was chosen a point of reference in the building to derive all the 3D coordinates. Then, was defined 28 different points distributed in the testbed with their respectively 3D coordinates. Each point was visited to collect the signal strength of the known access points using a PDA. With these data in hands, we calculated the real distance and the estimated distance among the device and the access points in each point, using Seidel's model simplified in [4]: $\mathbf{d} = 10^{-(32+\mathbf{ss})/(10\mathbf{n})}$. Where d is distance, ss is signal strength and n a variable, between 2 and 5, defined as function of general surroundings.

Using n equals 2.5, as recommended in [4], we got an error of about 12 meters to one access point, which sums, approximately, 58 meters of error, as shown in Table I. So, we tried to workaround using 2 different techniques of function fitting, namely here as: *fminsearch* and *polyfit*. The *fminsearch* function uses an unconstrained nonlinear optimization to find the minimum of a scalar function of several variables. Using it we found a better coefficient to the Seidel's model, n equals 2.92, with an error about 5 meters, in the worst case almost 18 meters of error. This error motivates trying the function *polyfit* to find the coefficients of polynomial of any required degree that fits the data in a least squares sense. We observed an error about 2.5 meters of error, with total error around 9 meters. The Figure 7 (1) shows the accuracies aforementioned for one access point, plotting the distance in function of the signal strength.

Technique	Error (m)	Std. Dev.	Norm	Total Error
Estimated	12.82	12.00	86.96	58.53
fminsearch	5.30	3.66	32.01	17.90
polyfit	2.44	1.68	14.69	9.34

TABLE I
ERROR OF THE 3 MODELS APPLIED TO CALCULATE THE DISTANCE IN FUNCTION OF THE SIGNAL STRENGTH.

An important decision was just use beacons from access points located on the same floor of the mobile device. The attenuation introduced by the concrete floor mischaracterized the distance as function of signal strength in Seidel's model, as shown in Figure 7 (2). This figure compares the distance in function of the signal quality of an access point in the same floor of the device, and another one in a different floor.

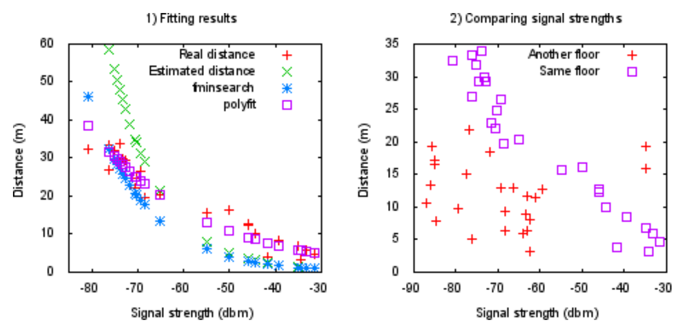

Fig. 7. 1) Distance in function of the signal strength observed in 28 points in the building. 2) Comparing the estimated distance of 2 access points in different floors.

There are several similar localization systems previous published. The main difference among them is the accuracy of the estimation that is strongly related to how much knowledge the system has about the environment. For instance, PlaceLab [3] has precision around 20-30 meters, and the system reported in [2] has 10 meters of error or less. The first one just use the localization of access points and cellular towers and can provide coverage in a whole city. In other hand, the last one uses an intrusive calibration technique fully aware of the topography of the environment and can just provide coverage in a building. Our system is in between of these two, and we believe that localization will be a common service in smart environments, because as LaMarca et al. [3] said these systems are most useful where we live, socialize and shop.

C. Mobile Virtual Community

The community is composed by a web application executing at servers, often cited as *server side*, and a mobile application at the users' mobile device, in short *client side*. They interact through an IP network using standard HTTP requests, e.g. GET and POST. The server side is responsible to provide a graphical user interface and a set of basics services to the client side. These services allow the user to upload the IP-connectivity experiences, combine the experiences in the connectivity graph and share it inside the community and also provide a user interface and manage the community members' using Google Friend Connect, as shown in Figure 8.

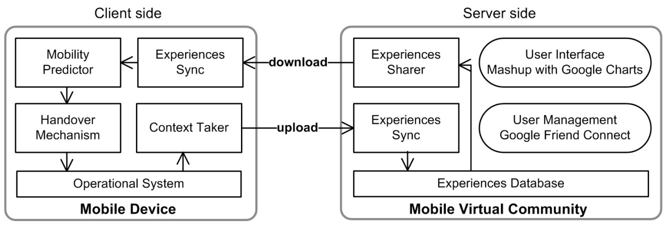

Fig. 8. Main elements of the mobile virtual community and the data flow among them.

As an OpenSocial[1] application, Friend Connect facilitates the interaction between social and non-social web applications and standardizes the handling and presentation of social applications and content. It uses several open standards, e.g. as OpenID for signin, and OpenSocial for applications. The last one is a set of common application programming interfaces for web-based virtual social network applications, developed by many social networks. Applications implementing the OpenSocial programming interfaces will be interoperable with any social network system that supports them.

The *client side* is in charge of gather context data, upload it to the community, and access the socialized RSS feed to get updated about the network conditions. The localization is retrieved by the localization system, describe earlier in Subsection IV-B, and a set of context data like date, time, signal strength are fetched by the operational system's programming interfaces. Regarding the connectivity management, the client executes the mobility predictor and a variation of it that uses our community-centric solution.

V. Experimentation

The effectiveness of this work is demonstrated through a comparative study among 3 connectivity management techniques, including one that implements the proposed solution. The comparison involves monitoring QoS metrics to assess the end-to-end traffic improvement achieved from the proposed model. We executed our solution for approximately 4 months with 3 users in the testbed described earlier in Section IV. The metrics measured are: download throughput, upload throughput, latency, and signal quality. In this section we discuss quantitative results obtained from our experiments.

A. Experiments Setup

The testbed configured allows inter-cell mobility creating an area with high probability of handovers. Inter-cell mobility includes all user movements that involve crossing the boundaries of an area covered by one access point to area that is covered by another access point [10]. With this setup each handover is done between different 802.11 networks. In others terms, each reconnection demands retrieve a new IP address, name server, and gateway, for instance.

We selected three points in the building, Alpha, Bravo and Charlie, and two connectivity paths. The paths are: from Alpha to Bravo, in short Alpha-Bravo, and from Alpha to Charlie, Alpha-Charlie. After 4 month we had in the database 8 different networks, 55 access points, and about 250.000 scans in distinct timestamps. The proportion of data collected among the 3 points picked in the building is plotted in Figure 9.

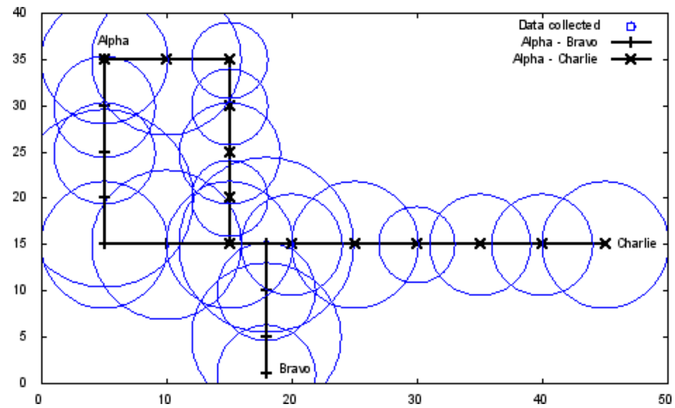

Fig. 9. Minimum (1) and average (2) values over the respective maximum ones observed.

Focusing on real usage and validation, we implemented a handover mechanism similar to VIRGIL, proposed in [7]. 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.

The comparative evaluation confronts the QoS performance of 3 connectivity management techniques namely and described here:

- Strongest signal strength: 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 [12] using as input the particular user movement history and non-structured data collected by all users, similar to the approach used in [8].
- Community-based: use the same predictor cited above accessing the socialized information as historical data.

The following subsection examines the performance of these 3 mechanisms in the testbed earlier detailed in Section IV.

B. Results

This subsection describe and discuss the quantitative results observed in the 2 paths arbitrary chosen, namely earlier:

[1]http://code.google.com/apis/opensocial/

Alpha-Bravo and Alpha-Charlie. The results observed in both paths are presented here in different ways aiming at explore the results from different angles. Thus, in subsection V-B1, we analyze the QoS metrics observed in the Alpha-Bravo path. In sequence, we examine the connectivity graph and the throughput in function of signal quality and time from Alpha-Charlie, in subsection V-B2.

1) From Alpha to Bravo: The main metrics observed in the Alpha-Bravo path are summarized in the Table II. This set of results revels that the testbed has a good coverage because the worst case in average is regular, and all the connectivity management mechanisms examined achieved the same maximal value. The Figure 10 presents the minimum and average values, of the 3 main QoS metrics, over the maximum values observed, both are in Table II. At a first look, we can see that in the worst case, Figure 10 (1), our solution has a predominantly performance if compared to the other 2 mechanisms. We avoid bad network conditions with a superior performance about 25, 9 and 33 %, tx down, tx up and signal quality, in this sequence. In the average, Figure 10 (2), our solution had a marginally superior performance than the others techniques.

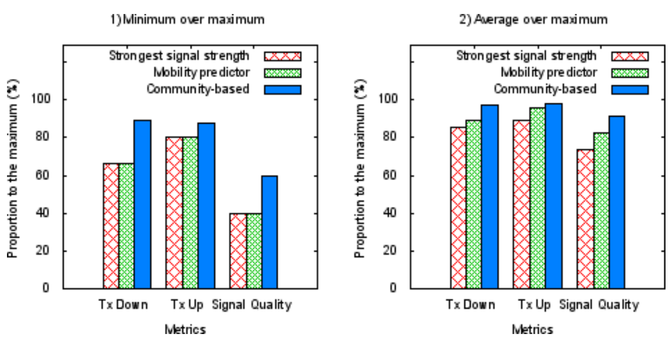

Fig. 10. Proportion of the main metrics to the maximal values observed.

2) From Alpha to Charlie: The *connectivity graph* generated by the 3 handover mechanisms is presented in Figure 11, with edge labeling $< accessPoint >$, $< \{handoverMechanisms\} >$. The mechanisms are referred to as A, B and C, strongest signal strength, mobility predictor and comunity-based, respectively. The first one, A, did 3 unnecessary handovers, and avoids one necessary to keep the highest throughput walking from Charlie to Alpha, as presented in Figure 12. Even with an excellent coverage the reactive decision making system performed poorly. One problem observed is the random connection to access points in others 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.

We need to have in mind that the Figure 12 shows the results between Alpha and Charlie and vice-versa. At time $0s$ all the users were at Alpha and walking to Charlie. Few seconds later, at $120s$ they arrived in Charlie. Then, they

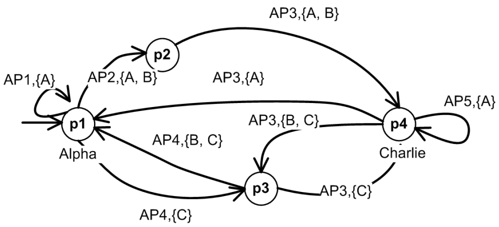

Fig. 11. Connectivity graph between Alpha and Charlie. A) Strongest signal strength, B) Mobility predictor and C) Community-based.

started to come back to Alpha, arriving around $227s$. The figure shows the upload throughput and the signal strength in function of time (seconds). In this case, time also represents space, since the users were walking in the building during the experiments. Concerning the results, our solution has better performance than the others 2, converging to a homogeneous experience in both metrics plotted; throughput and signal quality. Between 50 and 75 seconds, our solution had throughput about 25 % lower than the other 2. However, seconds later the throughput was 20 and 45 % higher, on the best and worst case, respectively. Returning to Alpha, the mechanism A come connect to the same access point and experienced the worst QoS conditions observed.

VI. Acknowledgment

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

VII. Conclusions

This research investigated the feasibility of modeling and implementing mobile user's IP-connectivity experiences as mobile social media. These experiences are a set of common context data, represented as a digraph and shared in a virtual community in a collaborative fashion. We started discussing the fundamentals components of connectivity management and related works. In sequence, we discussed the details related to the methodology applied and about the implementation issues. To finish, we described a testbed and discussed the main results of the experiments performed.

The main idea was address the challenge of design tools to allow mobile users, sharing a common place, to cooperate for the discovery of connectivity islands in a cheap and accurate way. Our community-based approach adds handiness to IP-connectivity QoS data and makes it easier to access, update and sample than the common QoS databases.

Our results showed that IP-connectivity experiences can be improved even in environments with excellent coverage. In addition, the virtual community is a interesting place to share experiences and can be a good source of feedbacks for access and service providers. By means of the research reported herewith, we are closer to achieve the challenge of developing tools to share connectivity experiences as mobile social media.

Metrics	Strongest signal strength				Mobility predictor				Community-based			
	Min	Avg	Max	S. Dev.	Min	Avg	Max	S. Dev.	Min	Avg	Max	S. Dev.
Tx Down (Mbps)	4.00	5.14	6.00	0.81	4.00	5.37	6.00	0.32	5.33	5.85	6.00	0.29
Tx Up (Mbps)	3.20	3.56	4.00	0.39	3.20	3.82	4.00	0.22	3.50	3.92	4.00	0.14
Latency (ms)	2.00	3.63	5.00	0.81	2.00	3.49	7.00	1.20	2.00	3.03	6.00	0.89
Signal Quality (%)	40.00	73.71	100.00	27.34	40.00	82.29	100.00	22.11	60.00	91.43	100.00	13.09

TABLE II
MAIN QOS METRICS OBSERVED IN THE ALPHA-BRAVO PATH.

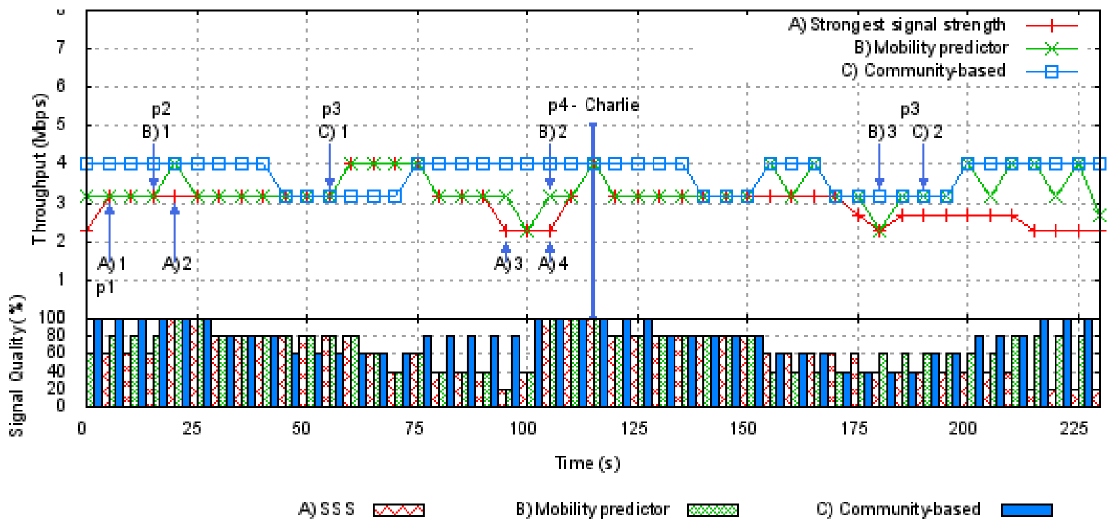

Fig. 12. Throughput and signal quality over time between Alpha and Charlie (and vice-versa).

REFERENCES

[1] J. Ghosh, M. J. Beal, H. Q. Ngo, and C. Qiao. On profiling mobility and predicting locations of wireless users. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 55–62, New York, NY, USA, 2006. ACM.

[2] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84, New York, NY, USA, 2004. ACM.

[3] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *Pervasive Computing*, volume 3468 of *Lecture Notes in Computer Science*, pages 116–133. Springer Berlin / Heidelberg, 2005.

[4] 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.

[5] C. E. Morr and J. Kawash. Mobile virtual communities research: a synthesis of current trends and a look at future perspectives. *Int. J. Web Based Communities*, 3(4):386–403, 2007.

[6] J. Multisilta and M. Milrad. Sharing experiences with social mobile media. In *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–3, New York, NY, USA, 2009. ACM.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] M. Smith, V. Barash, L. Getoor, and H. W. Lauw. Leveraging social context for searching social media. In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 91–94, New York, NY, USA, 2008. ACM.

[12] 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.

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

[14] 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.

[15] C.-L. Tsao and R. Sivakumar. On effectively exploiting multiple wireless interfaces in mobile hosts. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 337–348, New York, NY, USA, 2009. ACM.

[16] K. E. Wac, R. G. A. Bults, B. J. F. van Beijnum, H. Chen, and D. Konstantas. Toward mobile web 2.0-based business methods: Collaborative qos-information sharing for mobile service users. In M. Head and E. Li, editors, *Mobile and Ubiquitous Commerce*, volume 4 of *Advanced E-Business Methods*, pages 60–81. Information Science Reference, UK, May 2009.

[17] K. Wack. *Collaborative Sharing of Quality of Service-Information for Mobile Service Users*. PhD thesis, University of Geneva, Geneva, Switzerland, 2009.

[18] 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.