Achieving All the Time, Everywhere Access in Next-Generation Mobile Networks *

Marcello Cinque[1] *macinque@unina.it*

Domenico Cotroneo[1] *cotroneo@unina.it*

Stefano Russo[1,2] *sterusso@unina.it*

[1]Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli, Federico II
Via Claudio 21 - 80125 Naples, Italy

[2]Laboratorio ITEM - Consorzio Interuniversitario Nazionale per l'Informatica
Via Diocleziano 328 - 80124 Naples, Italy

The anytime, anywhere access view of nomadic computing is evolving towards all the time, everywhere views of pervasive computing. The all the time access requires mobile devices to be always connected, even if connectivity may be compromised due to Access Point overload and to transient signal degradations. The everywhere access requires mobile devices to use heterogeneous Access Points (ranging from Bluetooth and 802.11 to 2.5G and 3G cellulars), leading to high variability of the connection status. A mobility management solution that leverages connection availability, while enabling applications to be aware of the connection status, is thus needed. This paper proposes a Last Second Soft Handoff scheme that leverages the availability of connection. The proposed scheme has been integrated in a mobility management architecture, which provides connection awareness via an API, named NCSOCKS. Implementation issues are discussed and experimental results are provided.

I. Introduction

As mobile computing increases in popularity, more and more are the mobile devices (MDs) with multiple wireless interfaces, both for local area access (e.g. the IEEE 802.11 family or Bluetooth) and wide area access (2.5G and 3G cellular technologies). The nomadic computing view of mobility, in which MDs are nomads in wireless domains served by means of Access Points (APs), is nowadays supported by several proposal of architectures and frameworks, which aim to provide users with the “anytime, anywhere access”, according to Kleinrock's view [1].

Current research trends are foreseeing the use of MDs in a more pervasive fashion from user's perspective. Users will expect to use services that are spread in the environment, and to access to the Internet as well, in any moment, wherever they are located. This trend is leading to the realization of the so-called Next-Generation Mobile Network (NGMN), which arises new issues and challenges.

As nomadic computing aims to become pervasive, new mobility management solutions are needed in order to leverage the availability level of wireless connections, thus providing the “all the time, everywhere access” view of pervasive computing [2]. This means to take into account heterogeneity constraints and to define general and extensible vertical handoff procedures, since MDs cannot rely on an unique ubiquitous wireless network infrastructure. Addressing heterogeneity issues represents a fundamental milestone for the pursuit of the Next-Generation Wireless Internet (NGWI), as discussed in [3], which aims to provide Internet access through a diverse set of wireless architectures or “meshed” networks, e.g., wireless local area network (WLAN), 3G cellular, and satellite network. Besides, the wireless channel heterogeneity arises the problem of high bandwidth variations. For instance, a vertical handoff from a 802.11 Wi-Fi AP to a 2.5G GPRS AP may involve several order of magnitude of bandwidth decrease. For this reason, applications should be aware of the connection status in order to adapt their behavior accordingly, e.g., applications which adapt their flow rate to meet new channel conditions. Furthermore, mobility management solutions (i.e. handoff triggering and new AP selection) should be tailored with respect to application requirements, such as the expected bandwidth and link costs (a 3G link may be more expensive than 802.11 one). Adaptation of mobility management subsystem and of applications can be achieved by using a cross-layer ap-

* This work has been partially supported by the Italian Ministry for Education, University and Research (MIUR) in the framework of the FIRB Project “Middleware for advanced services over large-scale, wired-wireless distributed systems (WEB-MINDS), and by Regione Campania in the framework of “Centro di Competenza Regionale ICT”.

proach, relaxing the classical layered model. Finally, new architectures for the NGMN have to take into account the great deal of existing applications around, which were not designed with the objective of operating in such environments. Nevertheless, they often are valuable assets of enterprises, and have to be transparently ported to the mobile environment.

This paper presents a novel mobility management architecture for the NGMN, dealing with the mentioned issues. In particular, the proposed solution is designed in order to satisfy the following two fundamental requirements:

- *Leveraging communication availability* in spite of transient signal degradations, due to: i) shadowing caused by both physical obstacles and electromagnetic interference, and ii) APs overload. This results in the design of a mobility management solution using a novel handoff scheme, named Last Second Soft Handoff (LSSH), which is able to reduce the number of unnecessary handoff operations and the APs overload;

- *Providing applications with a connection awareness support*, dealing with the highly variable network conditions, and allowing them to describe their requirements. This results in i) the provisioning of a connection aware transport level API, the Nomadic Computing SOCKetS (NCSOCKS), and ii) letting the mobility management solution to use application level information to perform its tasks.

The above mentioned requirements have to be satisfied by taking into account the following constraints: i) wireless networks heterogeneity (APs can use different wireless data-link technologies); ii) Interoperability with existing client applications (the communication infrastructure has to be compliant with a plenty of existing Internet applications); and iii) energy efficiency (MDs rely on batteries).

The remainder of the paper is organized as follows. Next section reviews mobility management solutions. The proposed LSSH scheme, is presented in section III, whereas section IV describes the proposed architecture. Section V gives some implementation details and, in order to evaluate our proposal, section VI discusses the experimental results we obtain via an approach based on combined use of simulation and prototype-based measurements. Finally, section VII concludes the paper.

II. Related work

In recent years, a plenty of mobility management solutions have been proposed, including handoff schemes in a variety of flavors. These can be classified along several dimensions.

First, handoff schemes can be classified as reactive based and proactive based approaches. Reactive approaches, based on broken link recognition [4, 5], initiate the handoff as soon as the connected AP becomes available. Conversely, proactive approaches allow the handoff initiating when the AP is still available. For this reason, these approaches should be preferred. Proactive handoffs can trigger a migration procedure with respect to several criteria, such as monitoring the Receiver Signal Strength Indicator (RSSI) [6, 7, 8, 9], using a set of defined quality parameters [10, 11], and/or using a user-driven triggering [12]. In the direction of RSSI schemes, some of the above mentioned proactive solutions, such as [6, 9] are based on a fixed threshold mechanism, that is, the handoff is initiated when the RSSI falls below a certain threshold. Other solutions use a more complicated RSSI processing, for example using fuzzy controllers [7], or mobility prediction [8].

Second, handoff schemes can be classified as hard and soft handoff. If MDs are allowed to have two or more simultaneous connections to different APs, then the handoff is said to be soft; otherwise it is said to be hard [13]. Reactive handoff schemes are an example of hard handoffs, whereas proactive approaches can be either hard or soft. Indeed, among the above mentioned proactive solutions, some of them are hard [6, 9], and the rest are soft. Soft handoffs have the advantage of providing continuity of service during AP switching. On the other hand, they result in more resource consumption and in a low availability due to overload conditions. As discussed later, our LSSH scheme is a compromise of hard and soft handoffs that takes advantages of both the schemes.

Finally, one more distinction is between vertical and horizontal handoffs. A vertical handoff involves two technologically different network interfaces, whereas a horizontal handoff occurs between two APs that use the same technology and interface. Among the above mentioned solutions, some of them are vertical [10, 11], whereas the others are horizontal. Following the NGMN view, we are concerned with vertical handoff solutions.

It is worth noting that none of the above mentioned solutions i)take into account availability issues, and ii)provide application developers with a connection aware APIs. We rather believe that the time is ripe to provide both availability and connection awareness in terms of an unique solution for the NGMN.

III. Achieving availability: the proposed handoff scheme

A handoff procedure consists of a sequence of actions which is performed when a device migrates from an area served by an AP to another one. This procedure is composed of the following three phases [10]:

- Initiation: the network status is monitored to decide when to start a migration;
- Decision: once the need for handoff is triggered, a new AP has to be selected;
- Execution: the connection to the selected AP is established.

Our focus is on minimizing unavailability periods. Starting from the assumption that the device is in a zone covered by APs, otherwise, no connections can be established, unavailability periods can be caused by two kind of events: i) handoff occurrence, and ii) cell overload during an handoff, that can occur whenever a device tries to connect to an AP which cannot manage more connections. We explicitly note that the unavailability period for the event i) is negligible as compared to the time spent for the event ii). Furthermore, event i) does not occur if a soft handoff scheme is adopted. For this reason, we assume a soft handoff strategy. This means that we are only concerned with the event ii). Under the above mentioned assumptions, the availability, i.e. the probability that the connection is available during system operations, can be expressed as:

$$aval = 1 - unav = 1 - P_r(O \cdot H) = 1 - P_r(O) \cdot P_r(H) \quad (1)$$

where $P_r(H)$ is the probability that a handoff occurs and $P_r(O)$ is the APs overload probability. Thus, our goal is to minimizing both $P_r(H)$ and $P_r(O)$. The overload probability can be evaluated as follows:

$$P_r(O) = \sum_{i=C+1}^{N} \left(\begin{array}{c} N \\ i \end{array} \right) \left(\frac{1}{N_{AP}}\right)^i \left(1 - \frac{1}{N_{AP}}\right)^{N-i} \quad (2)$$

where N_{AP} is the number of APs distributed on the environment, C is the maximum average number of connections that can be handled by the APs, i.e., their average capacity, and N is the total number of admitted connections. In order to take into account soft handoff schemes, this model can be extended by assuming N be a random variable. Indeed, N can be evaluated by multiplying the number U of MDs in the environment by the number of simultaneous connections N_C required for each MD:

$$N = N_C \cdot U \quad (3)$$

The N_C term can be modeled as an uniformly distributed random variable, whose expected value depends on the adopted handoff scheme.

Although we assumed to use a soft handoff strategy, it is should be noted that, in order to minimize the $P_r(O)$ term, soft handoff schemes are not the best choice. In fact, as previous studies stated [14], the overload probability often increases with respect to hard handoff as the number of channels used by mobile terminals grows. However, we are concerned with soft handoff schemes as they help in minimizing the unavailability period due to the handoff per se. Thus, a trade-off between soft and hard handoff should be adopted.

Our proposal consists of using a Last Second Soft Handoff (LSSH) scheme, in which the initiation phase takes place using only the information about the AP currently in use, as in hard handoff, and only in the decision phase multiple connections are established, as in soft handoff. Hence, the LSSH scheme presents the characteristic of using a wireless interface at time during the initiation phase. This also results in i) better energy efficiency due to low power consumption, and ii) interference reduction, indeed using for instance Bluetooth and Wi-Fi simultaneously may produce significant interference [15].

III.A. Initiation: reducing $P_r(H)$

As observed in section II, the initiation phase can be performed using diverse sets of information and techniques, such as broken link recognition and AP monitoring through RSSI or other measures and metrics. Our solution is RSSI based, for several reasons: i) it allows the handoff to be proactive, ii) the RSSI parameter is already provided by the wireless interface, without performing intrusive measures needed to obtain other parameters, such as throughput or delay; this also reduces the power consumption, and iii) RSSI is an indication of the device position with respect to APs; this helps to achieve load balancing on APs depending on device distribution in the environment. Furthermore, in this way, locationing techniques can be implemented, as shown in our previous work [16]. It should also be noted that a RSSI based solution does not exclude the possibility to let applications to trigger handoff operations, if their requirements are not meet. According to the LSSH scheme, the initiation has to be performed using only the RSSI of the AP in use. It is thus crucial to carefully discriminate transient signal degradations, from permanent ones. Indeed, transient signal degradations can trigger unnecessary handoff procedures, increasing the probability $P_r(H)$.

As outlined in section II, most of handoff proposals are based on a simple threshold mechanism, that is, the handoff is initiated when the RSSI falls below a certain threshold. In this case, the value of $P_r(H)$ can be calculated as:

$$P_r(H) = P_r(RSSI < S_{RSSI}) \tag{4}$$

where S_{RSSI}, is a fixed threshold on RSSI. We can argue, as experimental results confirm, that this kind of initiation leads to a poor availability due to transient RSSI degradations, which do not strictly require any handoff.

Among the heuristics based on the concept of threshold, the α-count mechanism appears to be particularly interesting for our purposes due to the clear and simple mathematical characterization, the thorough analysis already conducted, and the minimal computational complexity [17]. The α-count function $\alpha^{(L)}$ is a count and threshold mechanism. It takes the L-th measured RSSI as an input, then $\alpha^{(L)}$ is incremented by 1 as the current RSSI falls below the threshold S_{RSSI}. Similarly, $\alpha^{(L)}$ is decremented by a positive quantity dec if the L-th measured RSSI is greater than the S_{RSSI}. A handoff is triggered as soon as $\alpha^{(L)}$ becomes greater than a certain threshold α_T. The function $\alpha^{(L)}$ is thus defined as follows:

$$\alpha^{(L)} = \begin{cases} \alpha^{(L-1)} + 1 & \textit{if} \quad RSSI^{(L)} < S_{RSSI} \\ \alpha^{(L-1)} - dec & \textit{if} \quad RSSI^{(L)} \geq S_{RSSI} \\ & \quad\quad \textit{and}\ \alpha^{(L-1)} - dec > 0 \\ 0 & \textit{if} \quad RSSI^{(L)} \geq S_{RSSI} \\ & \quad\quad \textit{and}\ \alpha^{(L-1)} - dec \leq 0 \end{cases}$$

Figure 1 shows how the α-count mechanism avoids to trigger handoffs procedures due to transient RSSI degradations. Indeed, a handoff is triggered if the degradation becomes permanent, i.e. $\alpha^{(L)}$ reaches α_T. Obviously, values for α_T, dec and S_{RSSI} have to be accurately tuned in order to achieve a tradeoff between early and late handoffs. These parameters are also tuned on the basis of application requirements. For instance, a location-sensitive application may require to use the closest AP. On the other hand, a bandwidth-sensitive application can also avoid to use the closest AP if its bandwidth requirements are satisfied. Section VI.B shows an experimental tuning procedure based on the reaction time one expects from the handoff procedure in the context of a case study prototype.

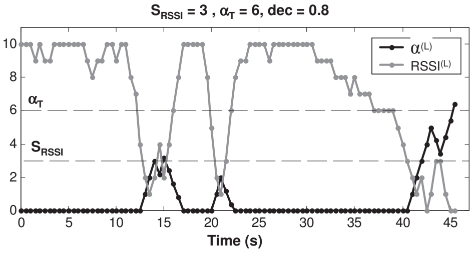

Figure 1: α-count behavior related to RSSI

A further improvement can be obtained by increasing the tolerance of the α-count, using a variable threshold, and in particular a step threshold $\alpha_T^{(L)}$, which assumes two distinct values: $big\alpha_T > \alpha_T$ after an handoff and for a time $duration$, and α_T until the next handoff. In other words:

$$\alpha_T^{(L)} = \begin{cases} big\alpha_T & \textit{if} \quad L < \textit{duration} \\ \alpha_T & \textit{if} \quad L \geq \textit{duration} \end{cases}$$

with L initialized to 0 at the beginning of each initiation phase. The step-threshold strategy is coherent with the normal user motions: once a user has entered in a cell, he/she spends a certain time to cover it. This is particularly useful in the worst case of an user standing at the boundary between two cells. In this case, lower RSSI values lead to a higher value of AP switching frequency, thus affecting the availability level. This frequency is reduced by using the step-threshold scheme as illustrated in section VI.C.

By using the α-count, the probability $P_r(H)$ can be calculated as:

$$P_r(H) = P_r(\alpha^{(L)} \geq \alpha_T^{(L)}) \tag{5}$$

As we expect, experimental results, which are discussed in section VI.C, show that $P_r(H)$ as evaluated in formula 5 is lower than the one as evaluated in formula 4.

III.B. Decision

During the initiation phase, only one connection is established and one RSSI signal is monitored to trigger the handoff. On the other hand, during the decision phase, multiple connections have to be established, in order to implement the soft handoff scheme. To simplify the selection of the new AP among the active ones, a topology-based scheme is adopted. In fact, the decision algorithm can elect the new AP only among the neighboring APs. We assume two APs to be neighbors if their distance d is less then a certain value d_{max}. The value of d_{max} can be evaluated for each couple of APs as one half the sum of their propagation radius. This means to consider two APs as

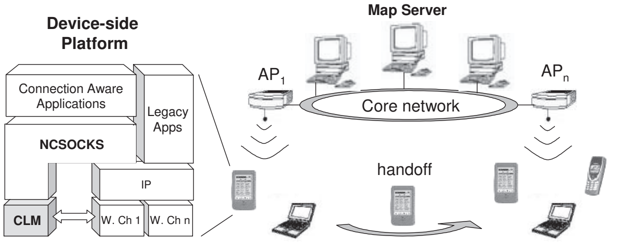

Figure 2: The proposed architecture

neighbors when they cover a common zone, assuring soft handoffs to be better performed.

The decision is taken by using a score criteria: let $N = \{ng_1, ..., ng_n\}$ to be the set of neighboring APs. For each $ng_i \in N$ a score $s(ng_i)$ is evaluated on the basis of several parameters: the RSSI R_i, the bandwidth B_i, the delay D_i, and the link cost C_i (in terms of how much the user pays to use that link). These parameters are adimensioned and scaled in order to be of the same order of magnitude. Once the scores are evaluated, the decision algorithm selects the AP ng^* with the score $s^* = max_{ng_i \in N}\ s(ng_i)$. The score function we propose, is a weighted sum of such parameters; in other words:

$$s(ng_i) = W_R \cdot R_i + W_B \cdot B_i + W_D \cdot D_i + W_C \cdot C_i \qquad (6)$$

The weights W_R, W_B, W_D, and W_C vary in the same range and assume discrete values, let say $[0, W_1, ...W_n]$, and they are chosen according to application requirements. Applications can easily specify their requirements via the NCSOCKS API, using several constants defined in the API. For example, the bandwidth requirement can be expressed with the following constants: NO_BAND, LOW_BAND, MED_BAND, and $HIGH\_BAND$. Similar constants are defined for other parameters. A constant contains the actual value of the associated weight. Applications' choices influence the score, as can be argued from the formula 6. In fact, applications with high bandwidth requirements result in a higher W_B, letting the decision to choose the AP which best fits bandwidth requirements, whereas location sensitive applications will have higher values of W_R. Similar considerations can be done for the link cost and the delay. As discussed later in section IV.A, NCSOCKS are able to notify applications when different applications, which run on the same MD, specify conflicting requirements, that is, for instance, high and low bandwidth. The proposed decision strategy, exploits the cross layer approach, since application level information modifies the behavior of low level mobility management components. The strategy, here proposed for four fundamental parameters, can be easily extended as new parameters need to be taken into account.

It is worth noting that, since is likely that each AP has only a few neighbors, the decision time is minimized as compared to the time needed for scanning and comparing all the APs in the environment, as some solutions suggest [7, 4]. In the context of our architecture, the AP topology is provided by a specialized component, the Map Server, that resides on the fixed infrastructure. Further details about the Map Server and other components that build our communication infrastructure can be found in the next section.

IV. Overall architecture

The proposed LSSH scheme is integrated in the architecture depicted in figure 2. Three are the major components that build such an architecture: the Connection and Location Manager (CLM), the Nomadic Computing Sockets (NCSOCKS), and the Map Sever. The first two components run on the device-side platform, whereas the third is deployed on the core network (the fixed infrastructure).

The CLM is in charge of handling connections with APs. In particular, it handles horizontal and vertical handoff operations among different APs, using the LSSH scheme, which parameters (α-count parameters and score function weights) are set by mobile-enabled applications via the NCSOCKS API. Furthermore, as the name suggests, the CLM manages information about the current location of the mobile device. As the CLM aims to perform vertical handoffs, it has to use different wireless channels (`W.Ch 1`, `W.Ch n` in figure 2). For this reason, it is designed according to the interface-based approach (see figure 3), so as to handle all the wireless channels through the same interface, which provides several common operations, such as searching APs, creating/destroying connections, building an IP interface upon the wireless media, and getting the current RSSI, delay, bandwidth and cost values. These operations are then implemented for each technology which characterizes the pervasive environment. By this way, the CLM can perform its operations without worrying about technological details. Furthermore, once defined an interface, it is always possible to extend the architecture with other wireless infrastructures.

As already mentioned, the decision phase is performed by the CLM by keeping a map of the neighboring APs. Such a map is managed by the Map Server. This component is in charge of accepting map requests from MDs' CLM. Furthermore, since the topology may change (an AP may fail, or the system administrator may decide to add or remove a covered zone), the Map Server has to recognize topology changes and provide the mobile terminals with the last updated map. As a centralized component, the Map Server may represent a bottleneck of the architecture, as the number of APs increases. To avoid the problem, we grouped APs in clusters and used a Map Server for each cluster. Clusters should be set up taking into account the physical topology. For instance, if the system has to cover a building, clustering should associate a Map Server for all the APs covering the same floor.

IV.A. Achieving connection awareness: the NCSOCKS

NCSOCKS were designed according to the cross-layer approach. As figure 2 shows, the NCSOCKS use both IP communication facilities and information gathered from the data link layer, through the CLM. Thanks to the cross-layer approach, information flows from/to the application to/from the CLM, via the NCSOCKS interface. Indeed an application can set its QoS requirements in terms of bandwidth, delay, link cost, and locationing precision, and can request, or be notified, about the current connection status. The status consists of several information: i) the availability (coded in *CONNECTED*, *DISCONNECTED* and *HANDOFF*), ii) the wireless technology being used and its cost, iii) the bandwidth level, iv) the delay level, and v) the MD's location in terms of its coarse grained symbolic location (i.e. the room name). The current symbolic location, the wireless technology and the cost are provided by the Map Server, knowing which AP is currently used. Both status information and requirements are defined and accessible by applications in terms of constants defined in the API (e.g. $HIGH\_BAND$ and LOW_BAND for the bandwidth), except from the location, that is expressed in strings which values depend on the physical environment and are configured via the Map Server.

The NCSOCKS are designed as a set of classes (see figure 3). In particular, two classes (`TCPClient` and `UDPSocket`) provide transport level services, both connection oriented and connectionless. Both the classes raise exceptions when the communication channel is unavailable for an unacceptable time, compared to delay requirements. On the other hand, packet lost during acceptable disconnections, are buffered and retransmitted, once the channel becomes available. In order to set its requirements and to read or to register their interest in connection status information, applications are provided by the NCSOCKS with a class, the `Connection Monitor`. The monitor provides a set of `setRequirement()`-like methods to set application level requirements. Since different mobile-enabled applications on the same device may ask for contrasting requirements, exception are raised in order to let the application to relax such requirements, if any, or abort the execution. Unspecified values for some connection attributes are automatically set to non-conflicting default values. In this way, all applications running on the same MD are forced to agree with the same non-conflicting requirement set. On the other hand, the monitor allows applications to read connection status information, and to register a callback with it to be executed when specified changes occur. By this way, applications can adapt their behavior accordingly. Finally, applications can also force a handoff triggering, if the expected requirements are not satisfied. In order to avoid frequent handoff requests, the CLM ignores further requests coming within a certain temporal window. To exemplify, the following C++ code segment shows how to exploit NCSOCKS facilities. The example consists of a UDP client that adapts, through asynchronous notification, the transmitted packet size with respect to the current delay. It requires to use low cost links with low delay.

```
// classes declaration
ConnectionMonitor CM;
UDPSocket sock;
DatagramPacket bigPkt, smallPkt, pkt;
// packets initialization ...
...
// delay callback declaration
class myCallBack: public DelayCallBack {
  public:
   void handle (string newDel) {
    if (newDel>LOW_DELAY) pkt = smallPkt;
    else pkt=bigPkt;
    if (newDel=HIGH_DELAY) this->CM.handoff();
  } };
...
// requirements settings
try { CM.setCost(LOW_COST); CM.setDel(LOW_DELAY);
catch (const ConflictingRequirement& e) {
  // asking the user to relax the requirements...
}
// callback registration
myCallBack cb(&CM);
CM.notifyDelayChanges(&cb);
while (!finished) {
   try { sock.send(pkt);
   } catch (const ChannelDisconnected& e) {
         sock.waitConnection(CONNECTED, 10)
   } }
```

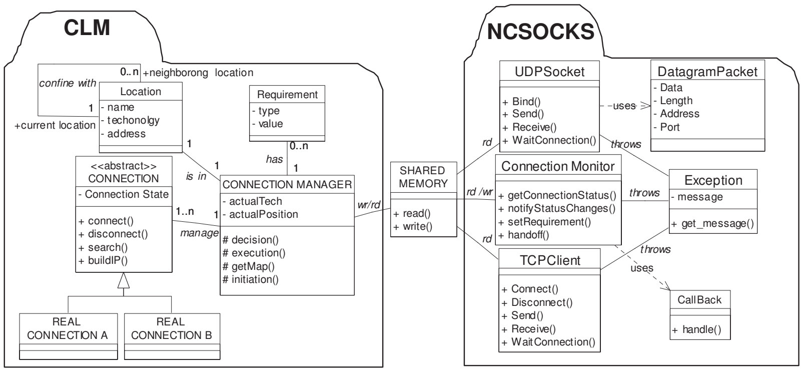

Figure 3: CLM and NCSOCKS class diagrams

IV.B. Interoperability issues

Handoff procedures can be classified according to the level in which they operate [5, 18]. We refer to an Ln handoff for an handoff procedure that works at level n of the ISO-OSI stack. In this paper, the focus has been on L2 handoff, i.e. vertical and horizontal handoff performed at data link layer. In order to pursue the portability of legacy IP-based applications, handoff at network (L3) layer should be supported as well. To this aim, since CLM provides IP communication facilities, L2 handoff can be completed with an IP-based mobility management solution, such as Mobile IP [19], which is a general technique to perform handoff of IP traffic between IP subnets using straightforward routing techniques and IP-IP encapsulation. However, L2 solution can support also UDP and some TCP applications with a minimal configuration effort at IP level (i.e. using DHCP or configuring the IP address by means of Zero Conf IP - `http://files.zeroconf.org`). Indeed, UDP does not require any connection and, as discussed in [18], some TCP applications, such as SMTP mail upload, POP or IMAP mail retrieval, and web browsing, are characterized by TCP connections being short enough to make the cost of having to re-attempt an operation relatively small.

V. Implementaiton issues

CLM and NCSOCKS have been implemented according to the class diagram depicted in figure 3. In particular, the `Connection` class represents the interface that has to be implemented in order to integrate different wireless technologies in the architecture. NCSOCKS and CLM communicate by means of a `Shared Memory` class, that encapsulates shared memory system calls. In the case study prototype, we have been concerned with wireless short-range access networks, and, in particular, we used Bluetooth and 802.11 wireless technologies. Thus, the `Connection` class has been implemented for these two wireless technologies. Beyond simplicity purposes, the utilization of Bluetooth as one of the “last meter” access network, as Gerla et. al. also proposed [20], represents an opportunistic and cost-effective way to improve the connection availability of the wide-spread existing 802.11 networks. Indeed, many portable devices are now equipped with both Bluetooth and 802.11 wireless interfaces. Furthermore, as the number of APs increases, Bluetooth has a better behavior than 802.11 in terms of bandwidth, delay, fairness and energy efficiency, as previous results have already outlined [21]. In this context, current implemented solution represents a mobility management solution for Bluetooth networks (it is noteworthy that Bluetooth specifications do not consider mobility management issues [22]), that natively integrates also 802.11 and then can be used to extend via Bluetooth existing 802.11 networks. Thus, this approach can be applied to integrate other wireless short-range technologies (e.g. Infra red Data Association) and wide area wireless solutions as well (e.g. 2.5G and 3G cellular), towards the NGMN view.

As far as the implementation is concerned, we have started using Linux operating systems. The CLM has been conceived as a daemon process running on mobile devices. As for 802.11, the implementation of its `Connection` class was straightforward, in that the IP abstraction is already provided by Wi-Fi adapters. This is not the case of Bluetooth technology, where a more deep study of BlueZ (`http://bluez.sourceforge.net`), the official Linux Bluetooth stack, has been needed. In particular, since Bluetooth does not support IP natively, the Personal Area Network (PAN) profile and the Bluetooth Network Encapsulation Protocol (BNEP) [22] have been used. On the AP side, we have experienced the implementation of Bluetooth APs by using simple Bluetooth USB dongles attached to PCs. Since APs have to continuously accept incoming connections, we have implemented a specific daemon process, called NAPdaemon, which is in charge of accepting incoming PAN connections and bridge them upon a unique virtual data link interface, as suggested by BlueZ implementors. To this aim, we used a Linux kernel (2.4.19) compiled with Bridge Control (brctl 802.1d) option. Behind the AP we enable a Network Address Translation Server (NAT), allowing bluetooth-enabled devices to use private IP network addresses.

VI. Experimental results

In order to evaluate our proposal, we adopted an approach based on combined use of simulation and prototype-based measurements.

VI.A. Testbed Description

A testbed was established on the laboratories of department building. The layout of these labs is shown in figure 4. It has dimensions of 30m by 40m with about 4 different rooms, including computer labs, offices, and the storeroom. As shown in the figure, APs are placed in the middle of four rooms. As for MDs, we used both a Compaq iPAQ 3970 equipped with Bluetooth and 802.11 modules, and with the Linux Familiar 0.7.0 operating system and a ASUS laptop with centrino technology and Linux Mandrake 10 distribution. The wireless APs are stationary and are configured to accept connections from mobile devices. In our scenario the APs used were three ANYCOM Bluetooth dongles programmed with our NAPDaemon and an Orinoco Access Point 802.11b. The mentioned scenario can be viewed as a case in which a 802.11 AP is enriched with some Bluetooth APs in order to leverage availability. Indeed rooms C and D are better covered by Bluetooth APs, whereas, B area, in which 802.11 signal is shadowed by a column, is covered by another Bluetooth AP.

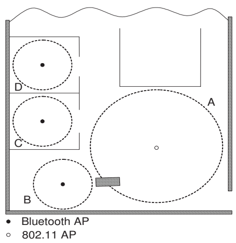

Figure 4: Test bed layout and AP placing

VI.B. α-count tuning

The proposed α-count scheme affects the *reaction time* of the initiation strategy. We define the reaction time T_r as the time within which the handoff is triggered, once the mobile device reaches the cell boundary. Obviously, T_r is a function of α-count parameters, that is $T_r = f(S_{RSSI}, \alpha_T, dec)$; hence, once the expected dimension of a cell is fixed, it is necessary to tune the α-count parameters in order to achieve a certain T_r when the boundaries are reached. We define an experimental means to tune the α-count parameters, given the cell dimensions and the expected T_r. For each cell, the tuning process encompasses two steps: **i)** experimental evaluation of RSSI frequency distribution at the cell boundary, and **ii)** simulation of the α-count algorithm with the previous evaluated RSSI distribution in order to estimate the T_r as function of the triple $(S_{RSSI}, \alpha_T, dec)$. Simulation results allow to choose the correct parameter values which guarantee the expected T_r. For this reason, each AP has its own α-count parameters stored in the topology map. As an extension, this kind of tuning strategy can be adopted also for the step-threshold scheme. To exemplify, let consider the tuning of the Bluetooth cell B (see figure 4). According to the step i), we capture RSSI values in several parts of the cell boundary region. As for the step ii), α-count simulation results are depicted in figure 5, in order to evaluate T_r as function of α_T, dec, and S_{RSSI} parameters (other figures with different S_{RSSI} are not shown due to the lack of space). Hence, once fixed the desired T_r (as an example $T_r = 4$ in the figure, which is emphasized by a dashed line), it is possible to determine different $t = (S_{RSSI}, \alpha_T, dec)$ triples that produce the expected T_r (for example both

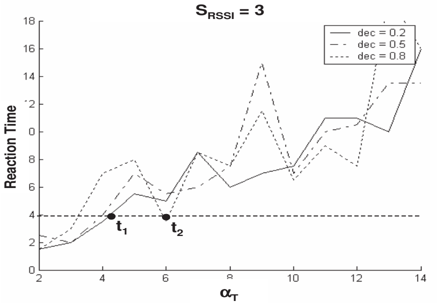

Figure 5: α-count parameters tuning

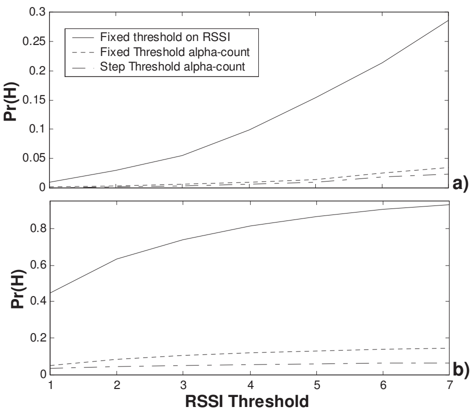

Figure 6: Unavailability estimation for the $P_r(H)$ term; a) average case, b) worst case

$t_1 = (S_{RSSI} = 3, \alpha_T = 4.3, dec = 0.2)$ and $t_2 = (S_{RSSI} = 3, \alpha_T = 6, dec = 0.8)$ triples can be used). By evaluating the RSSI frequency distribution in several cell boundaries, it is possible to achieve different parameters values for different cell sizes. Then the correct set of values can be chosen by the CLM with respect to application requirements (e.g. for location sensitive applications, it is better to use small cells, whereas for high bandwidth applications is better to use big cells in order to rarely change the current good AP).

VI.C. Availability measurements

The two proposed α-count schemes, simple and step-threshold, are compared with the fixed threshold mechanism (that is, an handoff is triggered when the RSSI value is lower than a fixed threshold), in order to evaluate the availability improvement. In particular we show unavailability reduction by evaluating the $P_r(H)$ (i.e. the probability that an handoff occurs) and the $P_r(O)$ (i.e. the overload probability), as defined in equation 1. As a proof of concept, the measurements are performed with respect to RSSI-based triggering, ignoring application-level handoff triggering.

As far as the $P_r(H)$ is concerned, we compare the three initiation strategies as a function of S_{RSSI}, since this parameter is the only one that affects the fixed threshold scheme. A simulation model is developed using the MATLAB environment (`http://www.mathworks.com`). Such a model performs the following two tasks:

1. simulation of the three initiation strategies with an increasing S_{RSSI} in order to evaluate the following two numbers: $N_{RSSI<S_{RSSI}}$, that is how many times the RSSI is lower than S_{RSSI}, and $N_{\alpha^{(L)}>\alpha_T^{(L)}}$, that is how many times $\alpha^{(L)}$ is greater than $\alpha_T^{(L)}$;
2. $P_r(H)$ estimation for each strategy.

The probability is estimated with respect to the frequency approach. It can be thus evaluated as $P_r(H) = N_{RSSI<S_{RSSI}}/N_{tot}$ according to equation 4, whereas, for α-count strategies, it is $P_r(H) = N_{\alpha^{(L)}>\alpha_T^{(L)}}/N_{tot}$, according to equation 5, where N_{tot} is the total number of measurements.

The simulation is performed by populating the models with realistic values. Hence, RSSI values were obtained on the experimental field. In particular, we measured RSSI values with respect to the average case and the worst case. For the average case, we measured the RSSI of a typical user moving with his iPAQ around the lab, from area A to area B, from room C to room D, and so on, for several days. As for the worst case, we place a MD between two rooms to force continuous handoffs due to lower RSSI values. The worst case scenario allows us to estimate the minimum guaranteed level of availability, with respect to the $P_r(H)$ term. Once these measurements have been performed, the initiation strategies have been simulated using RSSI measured values, and values of S_{RSSI} from 1 to 7. Simulation results are shown in figure 6. As we expected, the $P_r(H)$ is an increasing function of the S_{RSSI} threshold. The figure points out the benefits of our strategies, both in the average and in the worst case. In the average case, the α-count schemes produce respectively a 15% and 16% improvement on average, compared with the fixed threshold on RSSI scheme. These percentages increase respectively to 45% and 51% on average in the worst case. In the average case, the two α-counts exhibit quite the same behaviour. In the worst case, instead, the step-threshold scheme improves of about 6% on average with respect to the simple threshold α-count scheme.

As far as the $P_r(O)$ term is concerned, we have compared our LSSH scheme with two handoff schemes: an ideal hard handoff scheme, which always requires one connection per device, and an ideal soft handoff scheme, which always requires two simultaneous connections per device. Once the number of APs N_{AP} (including their average capacity C) is fixed, the probability $P_r(O)$ can be easily evaluated. In the case of our testbed, we have $N_{AP} = 4$; as for AP capacities, the three Bluetooth APs have a capacity of 7 connections, since Bluetooth devices can

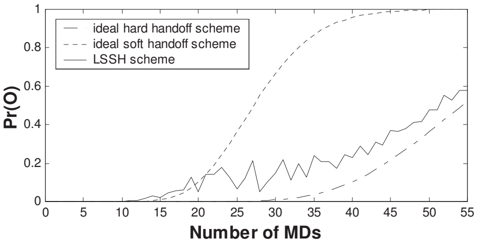

Figure 7: Unavailability estimation for the $P_r(O)$ term

manage at most 7 simultaneous connections. As for the 802.11 AP capacity, we can assume an average number of 30 connections. This value is an average estimation of a 802.11 AP capacity, since its values range from 4 up to 60 simultaneous connections, depending on the required bandwidth by each device. In fact, 802.11 MAC protocol is a random access protocol, conversely from Bluetooth, that is based on time division multiplexing. By averaging these capacity, we obtain $C \cong 13$.

However, in order to achieve the parameter N of the equation 2, we had to simulate the N_C random behavior, according to the equation 3. In the above mentioned ideal soft and hard handoff schemes, N_C can be viewed as a deterministic number, and in particular it is equal to 1 in the case of the hard handoff scheme, and 2 in the case of the soft handoff scheme. As for our LSSH scheme, N_C is a random variable. Its expected value has been evaluated by keeping track of users movements in our labs and by knowing that in the initiation phase, the scheme requires 1 connection. For the decision phase we assumed the worst case of 3 required connections, for simplicity purposes. Based on users movement, and with α-count parameters tuned for our testbed (with respect to defined default application requirements), we experienced that for the 15% of the time a MD is in the decision phase. Hence, we derive that only 1.3 connections on average are required for each MD using the LSSH scheme.

Once performed the above mentioned experiments, we have evaluated $P_r(O)$ for hard, soft and LSSH handoff schemes as a function of the number of MDs U (which varies from 0 to 55). The results we obtained are depicted in figure 7. The result is coherent with the assumption that the LSSH scheme is a trade-off between hard and soft handoffs. The interesting result is that the overload probability obtained through the LSSH scheme is closed to the one obtained for the hard handoff scheme. Hence, without loosing soft handoff advantages, we obtain availability results similar to the ones achievable by an hard handoff scheme, with an improvement of about 33% on average with respect to soft handoff schemes.

VII. Conclusions

This work presented a solution designed with the objective of pursuing the all the time everywhere access view of the NGMN. The wireless connection availability is leveraged by using a novel handoff scheme, the LSSH, which has been integrated in a communication infrastructure for the NGMN that provides application with connection awareness. Such an infrastructure has been implemented using Bluetooth and 802.11 wireless technologies. Exploiting the cross layer approach, applications can tailor the CLM behavior by specifying their requirements via the NC-SOCKS. The CLM has been designed according to interface-based approach, in order to easily integrate different wireless technologies, such as 2.5G and 3G cellulars. We claim that integrating different wireless technologies can help the NGMN view to become a reality. Experimental results have demonstrated that the proposed hand off scheme reduces unavailability periods, due to transient signal degradations and AP overloads, significantly. Future work will concern security issues, which have to be addressed, and the context awareness support, which has to be extended, providing more context resources, such as light and noise level.

References

[1] L. Kleinrock. Nomadicity: Anytime, Anywhere in a disconnected world. *Mobile Networks and Applications*, 1(1):351 – 357, December 1996.

[2] D. Saha and A. Mukherjee. Pervasive computing: A paradigm for the 21st century. *IEEE Computer, IEEE Computer Society Press*, pages 25 – 31, March 2003.

[3] O.B Akan and I.F. Akyildiz. ATL: An Adaptive Transport Layer Suite for Next-Generation Wireless Internet. *to appear in IEEE Journal on Selected Areas in Communications*, 2nd Quarter 2004.

[4] S. Baatz, M. Frank, R. Gopffarth, D. Kassatkine, P. Martini, M. Scheteilg, and A. Vilavaara. Handoff support for mobility with IP over Bluetooth. *Proc. of the 25th Annual IEEE Conf. on Local Computer Networks (LCN 2000)*, November 2000.

[5] J. Tourrilhes and C. Carter. P-handoff: A protocol for fine grained peer-to-peer vertical handoff. *Proc. on the 13th IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communcations (PIMRC '02)*, 2002.

[6] M. L. George, L. J. Kallidukil, and J. M. Chung. Bluetooth handover control for roaming system applications. *Proc. of the 45th Midwest Symposium on Circuits and Systems. MWSCAS-2002.*, August 2002.

[7] G. Bianchi, N. Blefari-Melazzi, M. Holzbock, Y. Fun Hu, A. Jahn, and Ray E. Sheriff. Design and validation of QoS aware mobile inernet access procedures for heterogeneous networks. *Mobile Networks and Applications, Special Issues on Mobility of Systems, Users, Data and Computing*, 8(1):11–25, February 2003.

[8] P. Bellavista, A. Corradi, and C. Giannelli. Mobility Prediction for Mobile Agent-based Service Continuity in the Wireless Internet. *to apper in Proc. of the 1st Int. Workshop on Mobility Aware Technologies and Applications (MATA'04)*, October 2004.

[9] S-H Chung, H. Yoon, and J-W Cho. A Fast Handoff Scheme For IP over Bluetooth. *Proc. of 2002 Int. Conf. on Parallel Processing Workshops (ICPPW'02)*, 2002.

[10] P. Reynolds. Mobility management for the support of handover within a heterogeneous mobile environments. *Proc. of First Int.Conf. on 3G Mobile Communication Technologies*, 27-29 March 2000.

[11] L-J Chen, T. Sun, B. Chen, V. Rajendran, and M. Gerla. A Smart Decision Model for Vertical Handoff. *Proc. of The 4th ANWIRE Int. Workshop on Wireless Internet and Reconfigurability*, 2004.

[12] R. Beyah, C. Turner, C. Corbett, and J. Copeland. A Mobility Enhancement for Switched Wireless Ethernet with Soft Handoff. *Proc. of 2002 IEEE Int. Midwest Symposium on Circuits and Systems (MWSCAS 02)*, 2002.

[13] B. Hamdaoui and P. Ramanathan. A network-layer soft handoff approach for mobile wireless IP-based systems. *IEEE Journal on Selected Areas in Communications*, 22(4):630 – 642, May 2004.

[14] Y. B. Lin and A. C. Pang. Comparing soft and hard handoffs. *IEEE Trans. on Vehicular Technology*, 49(3):792–798, May 2000.

[15] J. Lansford, A. Stephens, and R. Nevo. Wi-Fi (802.11b) and Bluetooth: Enabling coexistence. *IEEE Network*, pages 20 – 27, September/October 2001.

[16] D. Cotroneo, F. Cornevilli, M Ficco, S. Russo, and V. Vecchio. Implementing positioning services over an ubiquitous infrastructure. *Proc. of 2nd IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS 2004)*, May 2004.

[17] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. *IEEE Trans. on Computers*, 49(3):230 – 245, March 2000.

[18] E. Wedlund and H. Schulzrinne. Mobility support using SIP. *Proc. of the 2nd ACM Int. Workshop on Wireless Mobile Multimedia (WoWMoM'99)*, 1999.

[19] Network Working Group, IETF. *IP mobility support, RFC 2002*, 1996.

[20] M. Gerla, P. Johanssona, R. Kapoor, and F. Vatalaro. Bluetooth: "last meter" technology for nomadic wireless internetting. *Proc. of 12 th Tyrhennian Int. Workshop on Digital Communications*, 2000.

[21] P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla. Personal Area Networks: Bluetooth or IEEE 802.11? *International Journal of Wireless Information Networks Special Issue on Mobile Ad Hoc Networks*, April 2002.

[22] Bluetooth SIG. *Specification of the Bluetooth System - core and profiles v. 1.1*, 2001.