ALMA MATER STUDIORUM - UNIVERSITÀ DEGLI STUDI DI BOLOGNA

Dipartimento di Elettronica, Informatica e Sistemistica
Dottorato in Ingegneria Elettronica, Informatica e delle Telecomunicazioni
Settore Disciplinare: ING-INF/05

Middleware for Communication Interface and Context Source Management in Heterogeneous Wireless Environments

Dissertazione presentata
da
Carlo Giannelli

Coordinatore del Dottorato
Chiar.mo Prof. Ing. Paolo Bassi

Relatore
Chiar.mo Prof. Ing. Maurelio Boari

Ciclo XX

Esame Finale 2008

to the Patience, alias Frà

Abstract

The full exploitation of multi-hop multi-path connectivity opportunities offered by heterogeneous wireless interfaces could enable innovative Always Best Served (ABS) deployment scenarios where mobile clients dynamically self-organize to offer/exploit Internet connectivity at best. Only novel middleware solutions based on heterogeneous context information can seamlessly enable this scenario: middleware solutions should i) provide a translucent access to low-level components, to achieve both fully aware and simplified pre-configured interactions, ii) permit to fully exploit communication interface capabilities, i.e., not only getting but also providing connectivity in a peer-to-peer fashion, thus relieving final users and application developers from the burden of directly managing wireless interface heterogeneity, and iii) consider user mobility as crucial context information evaluating at provision time the suitability of available Internet points of access differently when the mobile client is still or in motion.

The novelty of this research work resides in three primary points. First of all, it proposes a novel model and taxonomy providing a common vocabulary to easily describe and position solutions in the area of context-aware autonomic management of preferred network opportunities.

Secondly, it presents PoSIM, a context-aware middleware for the synergic exploitation and control of heterogeneous positioning systems that facilitates the development and portability of location-based services. PoSIM is translucent, i.e., it can provide application developers with differentiated visibility of data characteristics and control possibilities of available positioning solutions, thus dynamically adapting to application-specific deployment requirements and enabling cross-layer management decisions.

Finally, it provides the MMHC solution for the self-organization of multi-hop multi-path heterogeneous connectivity. MMHC considers a limited set of practical indicators on node mobility and wireless network characteristics for a coarse-grained estimation of expected reliability/quality of multi-hop paths available at runtime. In particular, MMHC manages the durability/throughput-aware formation and selection of different multi-hop paths simultaneously. Furthermore, MMHC provides a novel solution based on adaptive buffers, proactively managed based on handover prediction, to support continuous services, especially by prefetching multimedia contents to avoid streaming interruptions.

Contents

Chapter 1 - Introduction

The last decade has been characterized by the evolution of the traditional fixed distributed scenario into a more powerful but articulated and complex mobile one, with increasing user expectations to access services continuously and seamlessly, with high heterogeneity of client devices, and with high probability of intermittent wireless connectivity.

In particular, the spread of portable devices with relatively great computational capabilities permits to advance from the traditional static computational model to a newer one where the user is not confined to predefined locations: the user can access resources and invoke services despite her current location, even while moving. In addition, the user has available a plethora of highly heterogeneous mobile clients, e.g., powerful laptops, whose capabilities are comparable to traditional fixed desktop computers, Personal Digital Assistants (PDAs), with lower computational capabilities but sometimes longer battery life, and even cell phones, greatly limited in relation to available memory and display size but almost always connected to the telecommunications network infrastructure. The traditional wired connectivity has been replaced by wireless communication technologies, IEEE 802.11, GPRS/UMTS, and Bluetooth among the others, providing the capability of accessing remote resources everywhere. Note that the currently available wireless technologies are characterized by highly heterogeneous capabilities; for instance, IEEE 802.11 connectivity provides great bandwidth at the cost of relatively high power consumption, UMTS great coverage range usually at not negligible economic costs for final users, and Bluetooth low power consumption but coupled with rather limited throughput. These differences are sufficient to point out that there is not a wireless technology overcoming the others in every environment and under every condition; on the contrary, available wireless technologies should be considered potentially complementary from the point of view of provided functions and properties.

Moreover, in current deployment scenarios there is an increasing and increasing availability of heterogeneous context sources characterized by different abstraction levels. These context sources can provide useful and meaningful information for service provisioning toward mobile clients/users, ranging from user geographical location provided by a GPS receiver to the internal clock, from the user agenda to her current mobility degree. Context information can greatly improve the full and effective exploitation of available capabilities, e.g., by driving management operations of low-layer components. For example, while applications may tend to select IEEE 802.11 interfaces due to their large bandwidth to improve user perceived quality of service, the operating system could impose, when the battery level is below a fixed threshold, to use a Bluetooth interface due to its lower power consumption to maximize the mobile client battery life. This simple example not only exhibits how context information could fruitfully bring to more suitable management operations, e.g., by denying power consuming interfaces exploitation not to compromise the whole mobile client, but also how complex is the integration of the many context information and requirements residing at different abstraction levels, for instance in order to harmonize potentially conflicting requirements such as large bandwidth and limited power consumption.

The availability of many heterogeneous context sources and communication interfaces motivates the work toward a novel and more powerful **Always Best Served (ABS)** scenario further improving the mobile one. The envisioned ABS scenario is characterized by applications desiring to get full advantage of the many available context sources and connectivity opportunities, namely ABS applications. In particular, each ABS application is potentially able to **select both the wireless interface and the remote node providing access to the Internet**, namely *connector* as thoroughly detailed in the following chapter, **based on its own requirements and context information** available on the mobile client itself. In other words, the availability of many networking opportunities permits the best selection of the networking opportunity to use in a context-aware way, i.e., based on current system state, user requirements, involved operating systems, and running applications.

The new ABS scenario not only improves application capabilities but also greatly increases the complexity of mobile client management operations: applications have to access and monitor the many (locally and remotely available) context sources and control mobile client behavior accordingly. In addition, local applications and remote services can no longer assume the availability of rather standardized capabilities on the mobile client but should adapt the content in relation to actual capabilities, e.g., not providing detailed Web pages to a cell phone with a display of few inches. Finally, the lack of reliable connectivity should be considered the usual operation condition strictly related to the dynamic nature of the wireless connectivity: low performance due to wireless channel interferences or connection disruption due to mobile client moving away from the connector, is not an exception due to congestion or rare hardware failures as in more traditional and fixed execution environments.

1.1 The Need of Middleware for Context-Aware ABS Applications

The lack of a set of well-standardized features to access and control context sources and wireless interfaces have limited the easy exploitation of information and communication opportunities available in ABS scenarios, pushing application developers to exploit only a limited subset of available capabilities.

The thesis proposes and thoroughly evaluates a middleware solution to support applications aiming to exploit the whole spectrum of capabilities potentially available in the ABS scenario. In particular, the proposed middleware has the ultimate goal of supporting the **context-aware synergic management of several networking opportunities**, considering both heterogeneous interfaces locally available on the mobile client and heterogeneous connectors available remotely, that is supporting both infrastructure single-hop and ad hoc multi-hop connectivity.

ABS scenario heterogeneity pushes for the **exploitation of context information** to take full advantage of the many available capabilities. In fact, it is obvious that it is unsuitable to exploit the available low-level networking components in a pre-defined manner, e.g., by classifying wireless interfaces on the mobile client in a fixed priority order. Instead, it is required to dynamically adapt mobile client and application behavior by carefully evaluating the capabilities in the current context of the mobile client, the applications running on top of the mobile client, and the preferences of the user accessing the applications. For example, the availability of context information related to the user geographical location could suggest exploiting a pre-configured free-of-charge Wi-Fi hotspot when approaching user's office or home, while switching to UMTS connectivity while walking in a park. In addition, based on context information it is possible to maximize the tradeoff among exploited resources and provided capabilities; for instance, based on application requirement knowledge it could be possible to automatically turn of energy consuming IEEE 802.11 and UMTS whenever the less performing but less consuming Bluetooth interface is present and considered enough.

We claim the need of **adopting middleware-based solutions** to properly manage ABS service provisioning scenarios. Middleware adoption is justified by the inherently complexity of gathering and monitoring several context information and requirements at different abstraction level and consequently changing the mobile client behavior. In the last years, middleware-based solutions have demonstrated their effectiveness in the control of low-level networking components while providing high-level Application Programming Interface (API) to other management facilities/systems and to the application layer. Considering the specific case of the ABS scenario, the necessity of a middleware solution is starting to be widely recognized: it is clearly unsuitable to delegate to the final user or the application developer the burden of monitoring and controlling context sources to have complete visibility of the current state. For example, users have neither the skill to access and configure the many heterogeneous components nor the capability to periodically monitor their dynamically varying state. Instead, a middleware solution can support application developers by providing a homogeneous and transparent access to low-level components, e.g., disclosing at the application layer many low-level details and features useful to easily provide richer applications. In addition, it is possible to reuse/extend available middleware capabilities improving the basic set of features with new ones potentially useful for many different applications in many different scenarios. In this manner it is possible to greatly simplify the development and deployment of new ABS applications while taking advantage of the many opportunities provided by the ABS scenario.

A middleware solution supporting context-aware access to and control of low-level components can greatly make easier the development and deployment of ABS applications. In fact, due to the inherently complexity derived by their heterogeneity only few context sources are widely recognized as useful and currently exploited in industrial mobile services, e.g., GPS data to enable car navigation systems, while the dynamic selection and management of communication interfaces is not currently supported at all. In addition, the great unreliability characterizing wireless connectivity if compared with traditional fixed networks greatly limits its adoption: for example, wireless connectors providing access to the Internet may become available and disappear just simply because a user moves from location to location. The intrinsic minor reliability of wireless links suggests to exploit only simple network topologies that facilitate and enhance connectivity durability instead of considering the wide possible set of networking opportunities. In particular, wireless interfaces can be exploited to give direct access to the Internet via infrastructure-based connectors, i.e., IEEE 802.11 Access Points (APs) and GPRS/UMTS Base Stations (BSs), while other available networking opportunities are completely neglected, e.g., not considering the possibility to access the Internet via a multi-hop ad hoc network based on Bluetooth scatternets. However, in some circumstances it could be more suitable to adopt ad hoc multi-hop topologies, even if at the cost of reduced performance due to the longer path to the Internet and higher power consumption since intermediate mobile clients have to forward flows to the next hop. For instance, by adopting a multi-hop heterogeneous ad hoc network, a mobile client equipped with only Bluetooth could be able to access a UMTS BS via an intermediate node getting UMTS connectivity and providing it via Bluetooth in a peer-to-peer fashion. A middleware solution supporting the easy exploitation of the many context sources and communication interfaces could certainly spread the adoption of ABS applications.

Let us stress that the goals of middleware-based solutions for ABS scenarios should be not limited to context monitoring and gathering but should also include the context-aware **active management** of the mobile client and running applications on top of it. It is worth noting that such a middleware-based solution permits to autonomously monitor context sources and consequently change mobile client behavior to best fit the different requirements provided by users, operating systems, and applications. In addition, there is the need for proper solutions for the fusion of heterogeneous context information, e.g., merging and harmonizing location information provided by GPS and other positioning systems based on wireless interfaces (additional details are in the following chapter). In other words, middlewares could permit to autonomously monitor mobile clients and the execution environment in order to control the local behavior accordingly, by considering and eventually harmonizing multiple different requirements provided by many entities. In addition middleware-based solutions can provide users and applications with a mediated access and control to underlying low-level components. For example, an application interested in downloading a huge update file could simply notify to the middleware the necessity of a large bandwidth connection while a user expresses her willingness to maximize battery life. The middleware is in charge of autonomously gathering context information related to power consumption and bandwidth of available interfaces and of properly managing the priority among requirements; a possible simple solution could be to exploit a Wi-Fi connection if the battery level is low, thus accomplishing only the application requirement, while denying any remote connection if the battery charge is almost finished, thus postponing the application update. Finally note that the previous solution could be suitable only in specific target scenarios, e.g., considering applications which can postpone the access to the Internet. However, a middleware solution permits to easily change the adopted evaluation metric, thus adapting the mobile client behavior not only in relation to the context, i.e., the battery level in the previous case, but also to specific objectives, e.g., taking into consideration deployment scenarios where it is unsuitable to deny remote access to applications.

1.2 Primary Guidelines: Translucent Access/Control and Heterogeneous Mode/Technology Connectivity

The ultimate goal of this thesis is to leverage the ABS scenario by considering and taking full advantage of its specific characteristics, instead of completely abstracting from them. This implies not only the identification and modeling of new context information related to user behavior and wireless technologies performance, but even the effective monitoring of context information to execute informed control procedure on mobile clients and, eventually, even on the infrastructure side. In particular, the thesis aims to support a context-aware simultaneous exploitation and management of the many heterogeneous networking opportunities, by considering mobile client current state, peculiarities of locally available wireless interfaces, and characteristics of dynamically discovered connectors.

To actually support ABS applications we claim the need for a novel middleware solution able to interact with and integrate multiple low-level components, autonomously managing these components in order to exploit mobile client capabilities at best. At the same time, such a middleware solution should provide users and the application layer with the capability to interact with underlying components in a homogeneous and properly aware manner, to gather low-level details useful to influence management decisions and to support the control of low-level components themselves. In particular, we have identified the three guidelines below any middleware solution supporting ABS applications should follow:

1) **translucent access** to underlying heterogeneous context information, i.e., providing the application level with the capability to gather context information and control context sources in both a transparent and visible manner to simultaneously provide a simple and fully aware interaction;
2) **both infrastructure and peer-to-peer connectivity support**, i.e., accessing the Internet not only via direct connection to APs and BSs but even via mobile clients that self-organize themselves in ad hoc multi-hop networks providing connectivity to the infrastructure network;
3) **mobility-aware connectivity management**, i.e., explicitly considering user behavior characteristics when managing active connections, to provide durable and reliable communication channels and, whenever required, even proactively configure local and remote resources in order to minimize perceived connectivity interruptions while changing connectors.

In relation to the first guideline, we claim that, in order to take informed context-aware management decisions, a middleware should provide easy access to low-level characteristics and control features of context sources and communication interfaces. That visibility should sometimes be disclosed also to advanced applications, in a highly portable and extensible way, which could take application-level service management choices depending on the awareness of low-level positioning details. We call **translucent** the original approach of middlewares that can support applications with both transparent and visible access to dynamically available context sources and communication interfaces in an integrated way. The ultimate goal is a highly dynamic, flexible, and reconfigurable application support capable of mediating visibility of low-level component characteristics/data and of managing heterogeneity in a context-dependent way. In fact, on the one hand, low-level components should propagate via the integrating middleware any capability they are able to offer, dynamically retrieved by middleware components and made accessible to the application level in a properly simplified way. On the other hand, applications should be able to command the reconfiguration of context sources and communication interface behaviors in relation to their current requirements, e.g., by keeping switched-on only the wireless interface with minimum energy consumption and satisfying bandwidth when the client battery lifetime is under a specified threshold. However, it is crucial that the visibility of low-level details and the synergic control of context sources and communication interfaces do not increase too much the complexity of application design and implementation.

In relation to the second guideline, we claim that the progress in wireless technologies is pushing towards more complex, flexible, and collaborative deployment scenarios than the traditional mobile one, where i) clients are equipped with and able to **simultaneously exploit several heterogeneous interfaces**, and ii) connectivity opportunities include both infrastructure-based equipment, e.g., IEEE 802.11 APs or UMTS BSs, and **wireless peers offering connectivity in ad-hoc mode**. These connectivity opportunities could be profitably combined together at runtime, to establish dynamic chains of multi-hop peers forwarding traffic to/from the wired Internet. In other words, given the multiplicity of wireless interfaces and increasing client-side computing resources, there is the need of middleware supports that exploit at best the potential of any available connector in a mixed infrastructure/ad hoc manner, even without affecting final users and service developers. Furthermore, mobile clients can easily have **simultaneous availability of multiple accesses to the Internet**, permitting local applications to exploit a different path for each required connection, e.g., to best fit each connection requirement or simply to get a larger throughput jointly by using several different paths simultaneously.

The third guideline refers to the exploitation of mobile clients as connectors, namely peer connectors, that could be far more complex than infrastructure-based usage and requires novel support approaches to effectively tackle newly introduced issues. For instance, peer-based connectivity tends to be less reliable, also in the case of a non-moving client, since peer connectors can move out of client radio range or abruptly revoke their connectivity offer. In such a complex and dynamic scenario, we claim that the evaluation process of monitoring and quantitatively evaluating networking opportunities cannot be based only on traditional raw monitoring data from the physical layer, such as Received Signal Strength Indication (RSSI) and Signal to Noise Ratio (SNR). In fact, we claim that, among the whole information characterizing provisioning context, **mobility degree** of both clients and peer connectors is of primary relevance. The main idea is to exploit mobility degree information to reduce the set (and management complexity) of available connectors. For instance, while a Bluetooth connector may be suitable to reduce power consumption, it should be discarded due to its limited coverage range in the case of a rapidly moving client with strict requirements about channel durability.

In conclusion let us note that the above guidelines are able to simplify the development and deployment of ABS applications and potentially spread the adoption of the ABS scenario. In fact, as better detailed in the following parts of the thesis, a middleware solution based on the above guidelines:

- enables the easy development and deployment of applications interested in exploiting the many available context sources and wireless interfaces, while not leaving to application developers the burden of dealing with the complexity derived from context/interface heterogeneity and their simultaneous exploitation;
- gathers and exploits context information specifically related to the management of ABS environments. The aim is to provide more durable connections despite the unreliability of wireless technologies: considered connection opportunities are not limited to infrastructure based APs and BSs but enlarged to even ad hoc based networking;
- explicitly considers connectivity disruption due to wireless connectivity unreliability and user behavior. For instance user location and mobility are not hidden; instead, they are considered as crucial data exploited to improve the informed management of connectivity opportunities.

1.3 Structure of the Thesis

The starting point of the work has been an in-depth study of the literature not only to precisely identify the current state of the research in the mobile scenario, in particular related to context and interface integration and management, but also to propose novel models and taxonomies to simplify the description and comparison of proposed solutions. After a long process of analysis of the state-of-the-art and of their constraints and limitations, the final step has been the development, deployment, and experimental validation of a middleware solution able to simplify ABS application development and mobile client control. In particular the thesis presents an actual middleware implementation that fully takes into account all above guidelines and able to take full advantage of already available capabilities on mobile nodes.

Chapter 2 provides some additional details about ABS, by delineating and comparing the different context sources and networking opportunities. In addition, it presents the novel Context-aware Autonomic Management of Preferred network Opportunity (CAMPO) model that identifies the main components an ABS scenario is made of. In particular, the CAMPO model proposes a common terminology to describe all those systems characterized by the control of networking opportunities in a context-aware fashion.

Chapter 3 thoroughly analyzes the state-of-the-art in the literature related to context source integration, remote resource discovery, and synergic management of different communication interfaces. Positioning system integrating middlewares are analyzed in relation to their capability to provide the application layer with information about and control of low-level components; resource discovery solutions are presented in relation to the mobility degree of the target environment they are intended for; communication interface integration solutions are analyzed and compared in relation to the characteristics of the deployment scenario, the dynamicity of the adopted evaluation procedure, and the capability of supporting active connection continuity when changing interface or Internet point of access.

Chapter 4 presents how it is possible to propagate differentiated levels of visibility up to the application level and synergically manage heterogeneous positioning systems depending on application requirements, user preferences, device characteristics, and overall system state.

Chapter 5 illustrates an innovative middleware solution supporting the exploitation of multiple heterogeneous communication interfaces simultaneously, not only to dynamically change the adopted point of access to the Internet, e.g., from an IEEE 802.11 AP to an UMTS BS, but also to actually exploit novel networking opportunities creating and managing in a self-organizing fashion multi-hop multipath infrastructure/ad hoc hybrid topologies. In addition, the chapter presents how it is possible to gather information related to user mobility and to exploit it to improve user-perceived quality of service in a mobile scenario. In particular, we target the specific case study of the continuous provisioning of on-demand audio/video streams, even during client handovers between different and heterogeneous Internet points of access.

Finally, Chapter 6 presents some conclusive considerations related to lessons learned through the design and prototyping of ABS middleware solutions. Directions of on-going research work end the thesis.

Chapter 2 - A Unifying Model for Autonomic Management of Preferred network Opportunity

As briefly depicted in the introduction, in our opinion the growing presence of powerful mobile clients with relatively high wireless bandwidth, e.g., via UMTS, IEEE 802.11, and Bluetooth 2.0 connectivity, is going to leverage the novel ABS scenario. Based on this novel scenario we envision the spread of context-aware applications, in particular Location Based Services (LBSs) depending on geographical location information, and full exploitation of multiple connectivity opportunities, based on infrastructure and ad hoc connectivity. The ultimate goal of the ABS scenario is to get full advantage of all available networking opportunities to perform mobile client and wireless interface management operations depending on context, i.e., to control their behavior in relation to the full contents of current state and requirements.

The chapter first presents our envisioned ABS scenario, focusing on heterogeneity of positioning systems and communication technologies; such heterogeneity is not to be considered a limitation but instead an opportunity to provide final users with richer applications. Then, it details the novel Context-aware Autonomic Management of Preferred network Opportunity (CAMPO) model, which provides a common vocabulary to describe not only the ABS scenario, but also all those systems characterized by the control of networking opportunities in a context-aware fashion. The CAMPO model has the additional objective of making easier the presentation and comparison of currently available and newly proposed CAMPO systems. Finally, some example applications based on the proposed model will show how it is possible to improve user experience by exploiting an ABS scenario with multiple context sources and wireless interfaces. In particular, the described use cases will point out the issues that ABS scenario heterogeneity and complexity rise and explain how it is possible to exploit middleware solutions to support the easy implementation of new ABS applications. Five design rules are identified and presented with the purpose of facilitating the management of different context sources and connectivity opportunities. We claim that the proposed five design rules greatly simplify the development of middleware-based solutions. Their adoption permits to support ABS applications following the already delineated three crucial guidelines: translucent access to low-level components, simultaneous exploitation of infrastructure and ad hoc connectivity, and the gathering and exploitation of context aware information strictly related to the user mobility behavior.

2.1 ABS Context and Connectivity Heterogeneity

The ABS scenario is characterized by a great number of low-level components providing a wide set of context information and supporting communication in several different manners. It is possible to identify heterogeneity related to i) context sources, and particularly positioning systems, ii) networking technologies and protocols, and iii) communication interface operating modes, i.e., infrastructure or ad hoc, single- or multi-hop.

Before analyzing the issues of heterogeneity of context sources, wireless technologies and interface operating modes, let us sketch a brief use case to clarify how an ABS scenario could enhance user perceived quality of service (more detailed use cases of the ABS scenario are in Section 2.3). Everyday Alice goes to work while carrying her PDA equipped with multiple wireless interfaces (IEEE 802.11, Bluetooth and UMTS) and a GPS receiver. At home the PDA connects to the domestic and free-of-charge Wi-Fi hotspot downloading her e-mail. When Alice moves out the GPS receiver starts receiving data from the satellites; the GPS receiver notifies the location change and the PDA reacts connecting to the more expensive UMTS BS that had greater coverage range; in other words triggered by context information provided by the GPS receiver, the PDA autonomously performs a vertical handover. Since the PDA connects to the UMTS BS before the Wi-Fi hotspot becomes unavailable, e-mail downloading is accomplished correctly: Alice does not perceive any service interruption. While driving to her office, the PDA starts downloading via UMTS brief traffic information related to the whole route. At the same time it performs IEEE 802.11 ad hoc multi-hop channels with nearby vehicles; in this manner it is able to collect the average speed of vehicles in the following kilometer, e.g., in order to have detailed and up-to-date traffic information related to her current district. Note that in this case the

PDA simultaneously exploits different and heterogeneous wireless interfaces in both an infrastructure and an ad hoc fashion, even exploiting multi-hop paths. In this manner it is possible to get differentiated services; a global one via UMTS, a local one via IEEE 802.11. Once in her office, the PDA connects to the local network via IEEE 802.11 to get a large bandwidth access to the Web and at the same time to her desktop via Bluetooth to synchronize the agenda and documents. Furthermore, it turns off the UMTS interface, since the Wi-Fi network is more suitable, and the GPS receiver, which does not properly work in indoor environment, to save battery power. Hence, the PDA interacts with underlying interfaces not only to access the most suitable networking opportunities, even sinergically, but also to autonomously change the mobile client behavior by controlling the activation of its equipment.

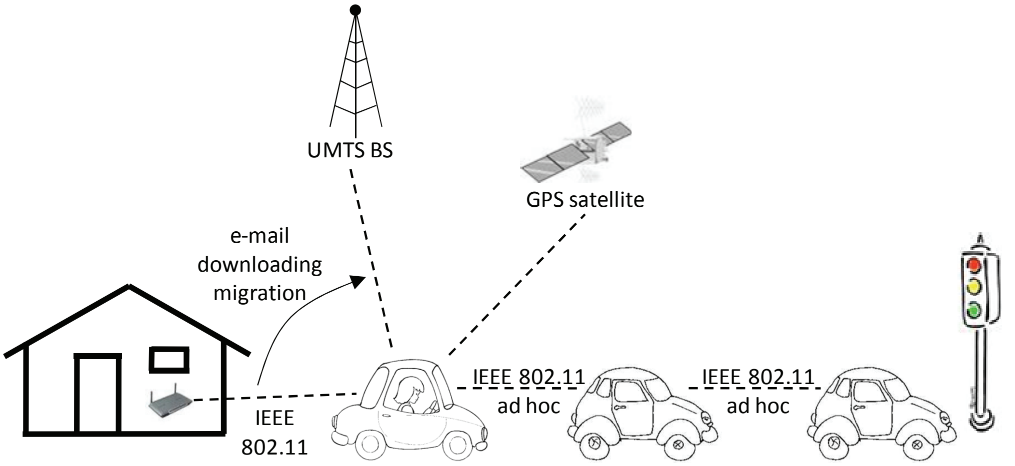

Figure 2.1 Alice moving to her office.

There are **several context sources available**, providing information on the mobile client, e.g., its battery level, the user, e.g., its agenda, end the environment, e.g., the number and type of remote nodes providing connectivity. However, the currently more investigated and exploited context information is the user geographical location provided by positioning systems. For this reason we focus on positioning systems and location information heterogeneity; however, it is possible to easily generalize our considerations to a wider set of context sources and information.

Based on location information provided by positioning systems, LBSs can provide service contents depending on the current position of served users, on the mutual location of clients and accessed server resources, and on the mutual position of users in a group [Chen and Kotz 2000]. To enable LBSs, the availability of low-cost and effective positioning systems is crucial. Several research activities have deeply worked on evaluating positioning mechanisms, techniques and systems: some solutions have been specifically designed for determining location, e.g., the well-known Global Positioning System (GPS) [McNeff2002]; other proposals try to estimate localization by monitoring characteristics of general-purpose communication channels, such as the IEEE 802.11-based Ekahau [Ekahau] and our originally proposed Bluetooth-based BTProximity (additional details in Chapter 4). Detailed surveys about positioning solutions and systems can be found in [Hightower and Borriello 2001a, Hightower and Borriello 2001b].

The point motivating our research activity is that the relevant work recently accomplished on positioning techniques has produced a wide set of currently available solutions that greatly differ on capabilities and provided facilities. For instance, they exhibit deep differences on:

- model used to represent location information. The representation model could be either physical (location information is provided as a longitude, latitude, and altitude triple), or symbolic (e.g., room X in building Y), or both;
- deployment environment. For instance, GPS can properly work outdoor, while another positioning system, such as Ekahau, may be more suitable for indoor environments;
- accuracy and precision of the positioning information. Accuracy is defined as the location data error range (10 meters for GPS), while precision is the error range confidence (95% for GPS);
- power consumption. The energy required for positioning typically depends on location update frequency;
- user privacy. Some systems provide high quality security features, other low: high for GPS because it determines the localization information in a completely client-side way with no explicit server-side visibility [McNeff2002], low and deployment-dependent for Ekahau because a cen-

tralized Ekahau server positions clients by monitoring the received signal strength of their wireless interfaces [Ekahau];

- additional system-specific attributes, such as the possibility to provide positioning data as a probability distribution function.

That heterogeneity of current positioning solutions, while being evidence of the relevant academic/industrial interest in the field, significantly complicates the development and deployment of LBSs. LBS developers currently have to know the details of the positioning system that will be available when deploying their services; LBS implementation is typically not portable and depends on the characteristics of the target positioning system (sometimes on the specific implementation of that positioning solution). Therefore, also due to the fact that current wireless clients tend to simultaneously host several wireless technologies useful for positioning (e.g., terminals with Wi-Fi and/or Bluetooth connectivity and/or equipped with GPS), there is a recent and emerging research trend in support infrastructures for uniformly integrating heterogeneous positioning techniques. The ultimate goal is easy LBS portability over different positioning solutions dynamically retrieved at LBS provisioning time.

Let us stress again that the same heterogeneity can be experienced while considering other types of context sources residing at a different abstraction levels. For example, it could be possible to exploit as context source the battery level of the mobile client (physical layer), the current throughput of exploited wireless interfaces (network layer), and the user agenda depicting her planned meetings (application layer). While in the following we specifically consider positioning systems, the same considerations can be easily generalized and applied to different kinds of context sources.

At the same time, to achieve network connectivity while preserving mobility, current mobile devices can exploit **several heterogeneous wireless technologies** with different characteristics, such as available bandwidth, transmission range, allowed client mobility, power consumption, ... This heterogeneity is also justified by the specific suitability of different connectivity solutions for different deployment environments: for instance, GPRS/UMTS for Wireless Wide Area Networks (WWAN), IEEE 802.11 for Wireless Local Area Networks (WLAN), Bluetooth for Wireless Personal Area Networks (WPAN or simply PAN).

Today, it is very common that one portable device hosts client equipment of multiple wireless technologies and, when needed, can exploit more than one of them simultaneously. For example, most laptops and PDAs are already equipped with both IEEE 802.11 and Bluetooth cards; UMTS/GPRS smart phones can usually exploit also infrared and Bluetooth connectivity, while the additional Wi-Fi option is emerging as a crucial market element due to the envisioned diffusion of Wi-Fi Voice over IP telephony.

That widespread availability of mobile devices with many wireless connectivity options calls for novel support solutions to exploit the dynamically available networks in the most appropriate manner, e.g., by selecting an IEEE 802.11 network if a running application requires large bandwidth and the Wi-Fi cell is not congested, or by choosing Bluetooth if the client is willing to preserve its battery and the running applications have compatible bandwidth requirements.

As pointed out by these cases, the choice of the most suitable connectivity solution at a given time depends on a large variety of elements, from user preferences to application requirements, from runtime environment conditions to expected stability in the availability of peers offering connectivity. In other terms, the choice of which connectivity to exploit should depend on context, intended as any information that describes the user (preferences, needs, location, …) and the environment where she is operating (date, time, resource state, ongoing service sessions, …) [Bellavista et al. 2006].

In addition to the aforementioned technology heterogeneity, the ABS scenario is characterized even by **different operational modes** each wireless interface is able to provide. In fact, the spread of powerful mobile clients equipped with multiple and heterogeneous wireless interfaces pushes for innovative envisioned scenarios where clients can both require and provide connectivity in a peer-to-peer and self-organized way. For instance, mobile clients connected to either IEEE 802.11 APs or UMTS BSs could offer Internet connectivity by acting as opportunistic ad hoc bridges to the traditional network infrastructure. In addition, mobile peers could dynamically form ad hoc networks and route packets between them. By fully exploiting the potential of the integration of infrastructure-based and self-organized peer-to-peer wireless networks, it is possible to enable powerful deployment environments where nodes can reach the traditional Internet via their most suitable Internet connectivity opportunities at any time. In fact, the simultaneous exploitation of different wireless interfaces potentially permits the dynamic creation of multi-hop paths, possibly composed by heterogeneous sub-paths, e.g., Wi-Fi and Bluetooth single-hop links, toward Internet connectivity points. In addition, any client may exploit different multi-hop paths (multi-path) for the connectivity needs of different applications, possibly with different service-specific requirements. In other words, the advancements in node capabilities and wireless communications enable the dynamic establishment of application-layer overlays that simultaneously exploit multi-hop multi-path heterogeneous connectivity, in both infrastructure and ad hoc modes.

In short, the ABS scenario is characterized by the simultaneous availability of highly heterogeneous context sources and networking opportunities. This heterogeneity has to be regarded as an opportunity to provide final users with richer services, by adapting mobile client behavior to their current context. At the same time it is not possible to leave to the user or the application developer the complexity of monitoring and managing the very different context information and wireless interfaces involved. Instead, we claim there is the need of a middleware-based solution providing autonomous monitoring and controlling capabilities. Based on the middleware mediation, application developers and users can exploit in a simple manner the many opportunities mobile clients offer. For example, an application can simply require to use always the most performing wireless interface; it is the middleware which is in charge of monitoring available networking opportunities, evaluating their suitability and, eventually, automatically switching among interfaces when a throughput degradation occurs, even by exploiting more than one interface at a time.

2.2 A Unifying Architecture Model For CAMPO Solutions

The previous section shows that the ABS scenario is characterized by simultaneous availability of many heterogeneous context sources and wireless interfaces on the mobile client coupled with the capability to both get and provide connectivity. These heterogeneous opportunities requires a middleware solution able to easily support applications and the final user to dynamically select the most suitable connection. However, the ABS scenario is only a specific case of possible exploitation of available communication interfaces. In fact, while we envision the ABS scenario as one of the most convincing and promising research field dealing with such heterogeneity, many literature contributions have recently addressed similar issues in a different manner.

In particular, from a wider point of view, we define Context-aware Autonomic Management of Preferred network Opportunity (CAMPO) the integrated management, with full context visibility, of all connectivity solutions dynamically available at clients. For instance, a CAMPO system should be capable of enabling a single (the most suitable) network interface at a client in some context conditions, e.g., in low-battery situations, while deciding to dynamically exploit multiple interfaces simultaneously for different running applications in a different context.

Lots of recent research papers and activities, from both academia and industry, can be regarded as relevant but partial contributions to the wide CAMPO area: although state-of-the-art related papers share the same ultimate goal of smartly exploiting the set of dynamically available connectivity options, they exhibit many differences. In particular, they assume some specific network architectures and/or wireless technologies, by missing to identify uniform/similar aspects that could bring to a unifying perspective in this research field. In addition, they tend to adopt different vocabulary and, even worse, the same terms with different meanings. An illustrative example is the plethora of words exploited to describe envisioned CAMPO solutions: 4th Generation (4G), Beyond 3G (B3G), 3G and Beyond (3GB), Always Best Connected (ABC), and Always Best Served (ABS) are frequently used as synonyms, even if we claim that they should identify different forms, with specific aspects and characteristics, of possible CAMPO deployment scenarios. The lack of both a unifying perspective and a unique shared vocabulary represent non-negligible obstacles for beginners to orientate themselves in the CAMPO research area and for researchers/expert practitioners to correctly position their work in the field.

The proposed model permits, on the one hand, to precisely and univocally define basic terms, such as horizontal and vertical handover, and, on the other hand, to provide a very general and coarse-grained grouping of CAMPO solutions based on their architecture principles, as better detailed in the following.

Let us stress that this model represents an original, and we hope valuable, generalization effort if compared with already available tutorial papers in the literature, all focusing on specific aspects of either 4G or ABC networks, as extensively detailed in Chapter 3 about related work. In addition, differently from already published work, this model provides a unifying overview and a thorough classification that can include both infrastructure-based deployment scenarios, with pre-deployed network attachment equipment (IEEE 802.11 APs, GPRS/UMTS BSs, …), and peer-to-peer scenarios where connectivity options are offered by opportunistically encountered peers, e.g., smart phones working as Bluetooth modems to the UMTS network.

2.2.1 The CAMPO Model

The lack of both a common description model and a shared vocabulary relevantly complicates the description of currently available CAMPO systems. In state-of-the-art papers, researchers are often involved in defining and re-defining even basic wireless terms, such as vertical handover. For these reasons, we claim the need for a simple but powerful model to represent a general CAMPO architecture, to classify both more usual cases of infrastructure-based wireless connectivity scenarios and emerging cases of opportunistically discovered peer-to-peer access to the Internet.

We propose the comprehensive general model reported in Figure 2.2. Any CAMPO system is modeled in terms of relationships between three types of entities only: *applications*, *interfaces*, and *connectors*. Each *application* represents a running service client at a user terminal and actively requests connectivity to fulfill its applicative goals, e.g., to download a hypertext from a Web server. **Interfaces** model the wireless hardware equipment available at the client side, e.g., Wi-Fi and Bluetooth client cards. Interfaces can be active/inactive (switched on/off, not stripped/stripped in the figure). **Connectors** are the entities actually providing mobile clients with connectivity by interworking with client-side active interfaces (graphically, in the figure, a connector is compatible with an interface if the profiles of the two representing shapes match). Connectors include both fixed AP equipment of infrastructure-based networks (IEEE 802.11 APs, Bluetooth APs, GPRS/UMTS BSs, …) and mobile wireless peers, namely peer connectors, either offering Internet connectivity or temporarily playing the role of servers in a peer-to-peer interaction.

We claim that any deployment scenario where CAMPO solutions operate can be modeled in terms of two kinds of relationships: between applications and interfaces (**interface selector**) and between interfaces and connectors (**connector selector**). On the one hand, when requiring connectivity, an application should associate with an active interface. In simple and widespread current environments, an application statically associates with a single interface, already active before application launch (the interface selector relationship is static and 1-to-1). On the other hand, for any active interface, more than one connector may be available. For instance, a client with its switched-on Wi-Fi card can have visibility of more than one Wi-Fi connector (near Wi-Fi APs or peers in ad hoc mode). Also in this case, nowadays most environments associate one interface with one single connector in a static way (the connector selector relationship is static and 1-to-1).

According to the model, we call **channel** the triple *{application, interface, connector}* that describes both the interface and connector relationships currently established for a running networked application client. Therefore, in traditional systems one application is usually involved in only one channel. We will see that all CAMPO solutions have the ultimate common goal to properly manage multiple channels for an application (or a set of applications) and of dynamically updating those channels depending on context.

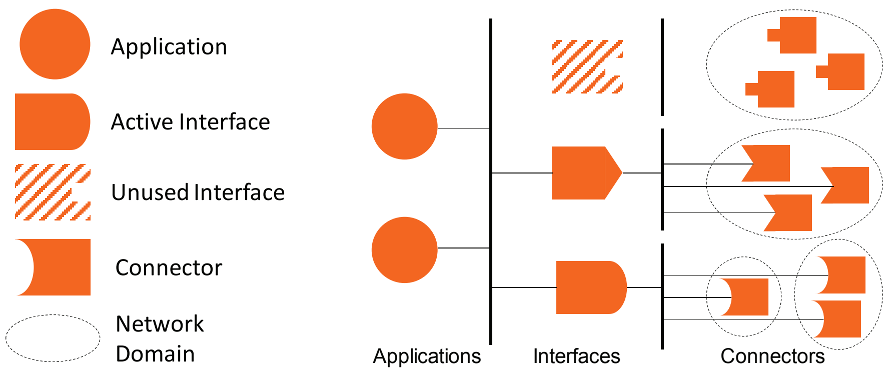

Figure 2.2 The proposed architecture model for CAMPO solutions.

Let us sketch an ideal provisioning scenario: we would like to have the most proper and dynamic channel selection (and consequent runtime update) with no impact on the design and implementation of both service clients and servers, which should focus only on the application logic, by possibly integrating also with legacy service components. In addition, it is desirable that also interface and connector implementations are not affected by relationship establishment and runtime modifications, in order to exploit any already deployed equipment and support. In other words, a CAMPO system is expected to be responsible for both interface and connector selections (multiplexing vertical black lines in the figure), possibly in a completely transparent way for application clients/servers, interfaces, and connectors.

At least any CAMPO solution should select the most proper interface(s) and connector(s) for an application statically, i.e., at application launch. Obviously, the context conditions determining channel determination may vary during service provisioning. A crucial aspect is, therefore, how a CAMPO solution reacts to dynamic variations in context conditions involving active channels (abrupt bandwidth/latency/jitter degradation due to congestion or client mobility, availability of new connectors with better quality, ...). Simpler CAMPO solutions are only static and take interface/connector selection decisions simply at channel instantiation, without any further control and action. Sometimes, it is impossible even to change interface/connector when the channel is either broken or so degraded to produce service interruption. More flexible and relevant CAMPO systems, instead, are dynamic, i.e., they manage active channels at runtime, by modifying relationships among applications, interfaces, and connectors, in order to exploit available networks at best at any time. In that case, a crucial point is to avoid service interruptions and/or loss of service sessions while re-configuring active channels at provisioning time. Let us note that, as in any management system that performs corrective operations in response to variations of monitoring indicators (here, the possibly wide set of context conditions of interest), CAMPO solutions should carefully consider the tradeoff between responsiveness in channel updating and related costs, avoiding the risk of possible thrashing due to continuous relationship modifications.

Before next section thoroughly discusses how the proposed model can permit to better catch similarities and differences between 4G and ABC solutions, let us rapidly observe that the common meaning assigned to usual terms, such as horizontal/vertical handover and micro/macro-mobility, can be clearly defined and better understood by means of our model. For instance, our model permits to univocally define horizontal handover as the process of updating a channel by modifying an interface-connector relationship while maintaining the same interface (the application-interface relationship is unchanged). In particular, in intra-horizontal handover (or micro-mobility) situations, a CAMPO system should replace the origin connector with a new destination connector in the same network domain (dashed oval in Figure 2.3, part a), i.e., in the same sub-network under the same administration realm. This is the usual behavior embedded in Wi-Fi client cards that automatically switch between different APs belonging to the same domain depending on the value and time-evolution of Received Signal Strength Indicators (RSSIs) from all APs in visibility. Differently, we define inter-horizontal handover (or macro-mobility) the situation, always without interface modification, where the change of connector also produces a domain modification, e.g., among Wi-Fi APs of different WLANs as in Figure 2.3, part b. Vertical handover, instead, is a channel update where the exploited interface is changes, thus usually forcing to a modification also in the selected connector(s), as depicted in Figure 2.3, part c.

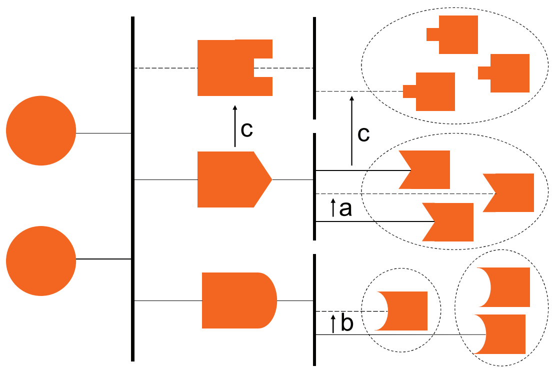

Figure 2.3 Intra-horizontal (a), inter-horizontal (b), and vertical (c) handover represented according to our proposed CAMPO model.

Note that in current wireless systems the underlying network equipment usually manages inter-horizontal handover in an application-transparent and embedded way, e.g., in IEEE 802.11 or UMTS roaming, by adopting not modifiable strategies/mechanisms for connector selection. On the opposite, intra-horizontal and vertical handovers often require additional network/service management actions (external to the wireless equipment implementation) for channel re-configuration, e.g., client IP address change, restart of client Authentication Authorization Accounting (AAA), and seamless transfer of service sessions. In particular, in infrastructure-based deployment scenarios, IEEE 802.11 APs and GPRS/UMTS BSs support inter-horizontal and vertical handover through re-routing mechanisms usually transparent to mobile clients, while in peer-to-peer connectivity there is typically the need for application peers to participate in an explicit end-to-end re-addressing. In the following, we will use the term *continuity management* to indicate the wide set of mechanisms, tools, and algorithms, from re-addressing to re-routing, from AAA to session transfer, that provide the building blocks to enable dynamic channel update without perceivable interruptions in service provisioning, possibly in a seamless way for wireless clients and final users. Given the wide heterogeneity of involved wireless technologies, in terms of both interfaces and connectors, there is a large spectrum of different continuity management mechanisms in the CAMPO literature. Our model also tends to give a unifying perspective on that plethora of support mechanisms, by identifying common goals and interworking opportunities/requirements, as better detailed in the following chapter.

2.2.2 Exploiting the CAMPO Model to Characterize 4G and ABC Solutions

With the goal of exemplifying the application of our model to notable CAMPO cases in the literature, this section analyzes which are the main differences and points of contact between 4G and ABC systems by modeling them according to our proposal. That should contribute to reduce the confusion and imprecision in the current usage of the two terms.

Figure 2.4 represents the simplest, and today most common, working environment for a CAMPO solution. Even if several interfaces are potentially available for clients, only one of them is usually active at a time. That active interface is associated with any running application and with only one connector that is selected, among the ones available for that interface, according to an enforced strategy. In other words, if the running applications at client are N, all of them exploit the same interface, and that interface associates with only one connector, the same for every channel. In the following, we will use the expression **<N:1:1>** for the channel triple to indicate the fact that the interface selector relationship is N-to-1 and the connector selector relationship is 1-to-1.

In that scenario, the burden of interface selection is usually delegated to the user who can manually switch on/off the interfaces mounted on her client terminal. In this case the only CAMPO role is the choice of the most proper connector for the switched-on interface. Often, that choice is directly embedded in the implementation of client connectivity equipment, e.g., the client card implementation of the IEEE 802.11 MAC layer that may select the Wi-Fi AP with maximum RSSI. Typically there is no support to seamless inter-horizontal and vertical handover. Only intra-horizontal handover is usually allowed with no impact on application implementation: its support is completely delegated to wireless client cards and to special-purpose infrastructure-side components for re-routing and/or re-addressing, e.g., via standard signaling protocols between IEEE 802.11 APs.

Let us observe that the above simple case represents a poor CAMPO working environment, which does not exploit at all the wide set of possibilities offered by multiple connectivity technologies possibly available at the client side. For instance, in that scenario, even client connectivity depends on users in charge of switching on the suitable interface depending on locally available networks. On the contrary, we claim that a CAMPO solution should be capable of exploiting the different wireless interfaces available at clients at best, even in a synergic way, by automatically selecting the most suitable channel for each application depending on current context conditions and by supporting seamless channel switch during service provisioning.

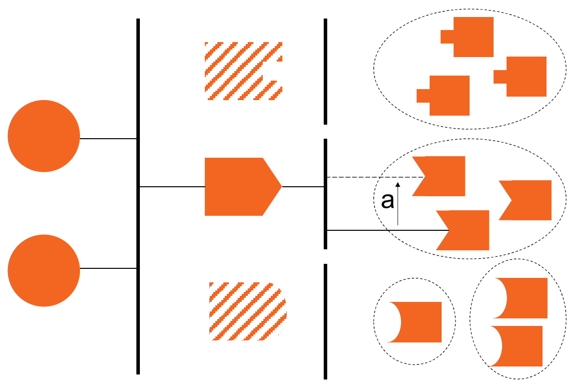

Figure 2.4 The simplest and most traditional <N:1:1> wireless environment.

The simple environment depicted in Figure 2.4 is still usual in current wireless systems. To overcome its limitations, several recent research activities, from both industry and academia, have proposed a large number of CAMPO solutions, with different characteristics and responsibilities, as extensively detailed in the next chapter. Nowadays, 4G and ABC are the terms most frequently used to generally indicate CAMPO systems that go beyond the above simple case. However, in several state-of-the-art papers, the two terms are used in a confused way, sometimes as synonyms, without stressing their peculiar characteristics and their differences. We claim that 4G and ABC should identify two specific and differentiated families of CAMPO solutions and their precise definitions could benefit from our CAMPO architecture model to clearly point out specific properties and assumptions.

In particular, by adopting our model, it is easy to identify that, differently from ABC solutions, **4G** systems respect the constraint that all applications at a 4G client have to exploit only one interface at a time (see Figure 2.5). CAMPO systems for 4G are in charge of two main management actions: i) selecting the interface to activate at any time and ii) performing vertical handover, i.e., commanding the continuity management support to seamlessly update all active channels when there is the need to change the currently switched-on interface. That typically occurs because there are no alternate connectors available for the previously activated interface, for instance, because a Wi-Fi-connected client is moving and entering an area not covered by IEEE 802.11 APs.

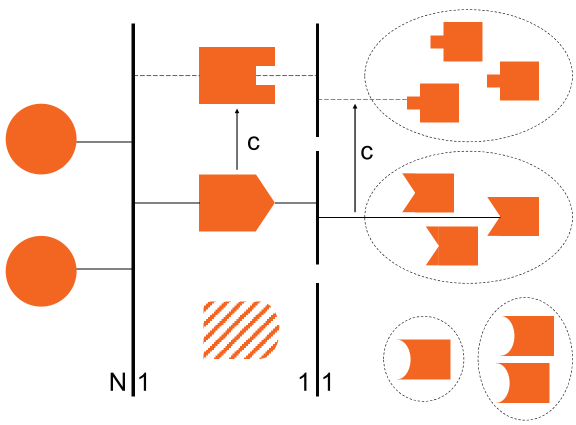

Figure 2.5 Usual 4G solutions can be modeled as <N:1:1> wireless environments that can dynamically change the activated interface.

To perform vertical handover, 4G clients and infrastructure-side support components collaborate to gather the context information required for proper interface selection, e.g., signal quality, level of congestion, and estimated jitter. 4G selection strategies may pursue either a local client-specific goal, e.g., continuous channel availability or QoS maintenance, or a global goal, e.g., load balancing among overlapped cells. Occasionally, 4G systems also aim at providing support for inter-horizontal handover, while intra-horizontal handovers are generally performed by network equipment in a transparent way in this kind of CAMPO solutions. On the contrary, connector selection (typically considering only infrastructure-based equipment, such as APs or BSs) is completely embedded in interface implementation and application-transparent.

In summary, in any case 4G solutions associate all applications running at a client with the same interface and, therefore, typically consider mobile client requirements as a whole to choose only the most suitable interface. The cardinality for interface/connector selector relationships in **4G is <N:1:1>** as in the simplest case of Figure 2.4, but these systems add the primary capability of **dynamically changing the activated interface**.

On the contrary, we claim that the **ABC** term should be used to indicate more flexible CAMPO solutions where **multiple interfaces may be simultaneously active** and the main goal is to activate and update the most suitable channels for any running application. The motivation is that different applications usually have different service-specific requirements in terms of bandwidth, latency, and sustainable discontinuity intervals. Thus, the possibility to have channels with different interfaces for different applications at the same time can significantly improve the exploitation of available networking options.

By exploiting our CAMPO architecture model, it is possible to identify two main categories of ABC solutions depending on the cardinality of the connector selector relationship. On the one hand, there are ABC systems where each activated interface associates with a single connector (<N:M:M> ABC systems). <N:M:M> solutions are responsible for activating the proper interfaces among the set of available ones and for selecting/updating the most suitable connector for each of them. On the other hand, more complex ABC systems additionally consider the possibility of associating multiple connectors with each active interface, thus enabling applications that exploit the same interface but different connectors (<N:M:L> solutions).

Figure 2.6 depicts the case of **<N:M:M>** CAMPO systems. Differently from 4G, here the interface/connector selection strategies should consider the requirements of any application and not of the client as a whole. In addition to interface activation and initial connector selection for each active interface, <N:M:M> systems are typically in charge of channel updating when new interfaces become more suitable than the currently used one. Therefore, these CAMPO systems generally include continuity management mechanisms for per-channel re-addressing and for communicating end-point modifications to application clients/servers. On the opposite, <N:M:M> systems usually do not automatically update the choice of the connector for each interface, by delegating that selection to the embedded behavior of network equipment. For instance, once activated, the client UMTS card autonomously selects the BS where to attach among the ones in visibility depending on an embedded strategy.

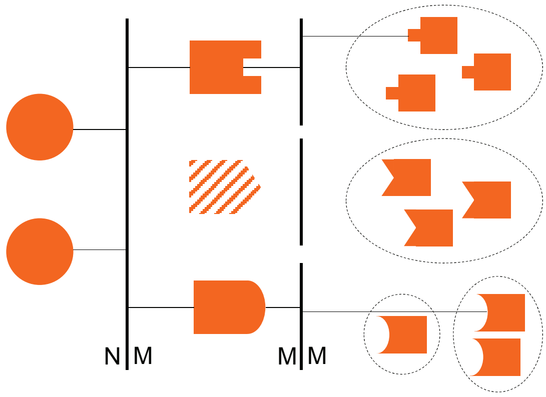

Figure 2.6 Usual ABC solutions can be modeled as <N:M:M> wireless environments.

<N:M:L> ABC solutions are even more flexible and can actively **select** not only the interface but also **the connector for each channel** (see Figure 2.7). In addition, these CAMPO systems are often able to consider an enlarged set of potential connectors. They can take into account not only traditional infrastructure-side network components, such as IEEE 802.11 APs and GPRS/UMTS BSs as in 4G, but also nearby peers in wireless ad hoc connectivity mode behaving as connectivity bridges. Notwithstanding that flexibility, in some real <N:M:L> systems the possibilities of connector selection may be reduced due to possible limitations in communication technology capabilities. For instance, Bluetooth allows concurrent interworking with connectors residing in different networks (scatternet) and ad hoc-configured IEEE 802.11 permits to simultaneously interact with different connectors in the same ad hoc network, provided that the nodes have the same Extended Service Set IDentifier. On the contrary, IEEE 802.11 in infrastructure mode and most cellular technologies provide very limited or null capabilities to control connector selection, thus forcedly limiting the selection space available for CAMPO systems developed on their top.

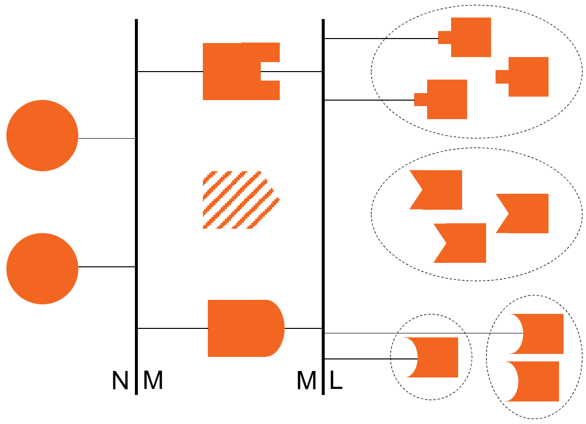

Figure 2.7 Flexible ABC solutions support <N:M:L> cardinality for interface/connector selector relationships.

Similarly to <N:M:M> systems, <N:M:L> solutions are in charge of implementing continuity management mechanisms for per-channel re-addressing, even if in these systems mobile nodes are often responsible for invoking these mechanisms at channel updates (end-to-end visibility of channel modifications). For instance, mobile nodes may be requested to notify their new IP addresses after a channel update by exploiting CAMPO signaling facilities. In addition, differently from other ABC systems, <N:M:L> solutions should be able to manage the increased complexity stemming from the need to simultaneously monitor a large set of connectors for different interfaces in order to command channel updates when better connectors are available. Moreover, these solutions should address the challenging issue of considering and evaluating the possible unreliability of peer connectors to operate proper channel decisions: differently from APs and BSs, peer connectors may abruptly become unavailable due to mobility or power shortage, thus forcing frequent channel update operations (and the consequent overhead).

Let us finally stress that we use the ABS term instead of ABC to indicate not only the capability to exploit several paths simultaneously but also that these paths can be eventually composed of several heterogeneous hops, based on hybrid infrastructure/ad hoc connectivity. In our opinion this is a remarkable difference that state-of-the-art ABC contributions do not explicitly consider.

2.3 Applying Design Guidelines to ABS Scenario Use Cases

To better clarify the motivations and the potential advantages behind the idea of a CAMPO middleware specifically designed for ABS scenarios, let us rapidly sketch two examples of envisioned ABS applications. They depict i) the provisioning of LBSs exploiting the many available positioning systems and ii) the provisioning of seamless Internet connectivity based on both infrastructure and ad hoc connectivity based on context information. Our purpose is twofold: on the one hand, we aim to demonstrate how the adoption of a middleware solution based on the proposed guidelines may enable the adoption of the ABS scenario; on the other hand, we want to identify some simple design rules which may greatly simplify the development of new ABS applications.

2.3.1 A Translucent Access to Heterogeneous Context Sources

To make practical examples of multiple heterogeneous context information usage scenarios, consider the case of Alice on vacation in a foreign city and moving from a square to a museum while accessing a LBS providing historical information. While the GPS location information accuracy is really high in the middle of the square, it lowers when approaching the edge of the square due to high buildings partially covering signals of GPS satellites; finally, the GPS receiver becomes completely useless when Alice enters the museum building. At the same time, inside the museum the Ekahau positioning system starts providing location information exploiting RSSI values gathered via a newly discovered Wi-Fi network. While moving from the square to the museum Alice has the willingness to seamlessly and transparently switch from a positioning system to another depending on availability and suitability, i.e., GPS outdoor and Ekahau indoor.

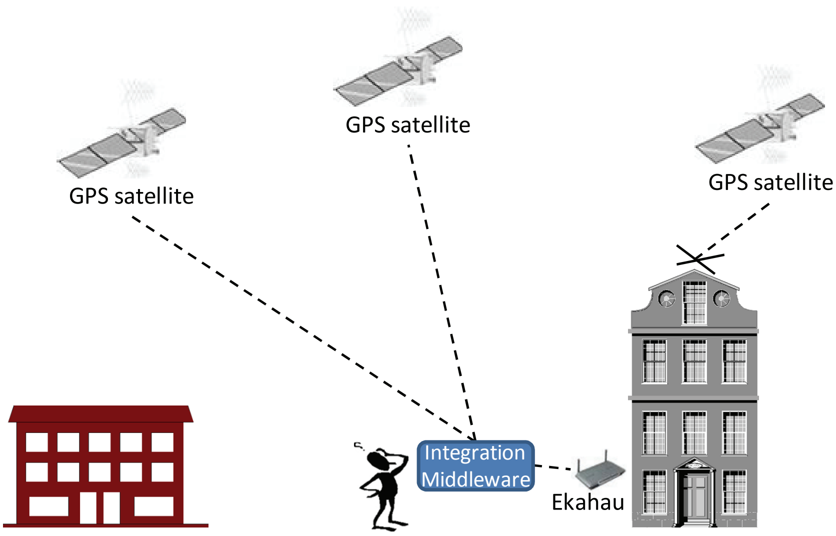

Figure 2.8 From GPS to Ekahau switching when approaching a building.

To this purpose the application should associate, at any time, with the positioning technique that best fits the execution context, possibly by leaving that choice even to the operating system, e.g., the positioning system with lower power consumption or the one with greater precision/update frequency. In addition, when several positioning systems can concurrently work, the application should either perform positioning data merging/fusion, e.g., according to context-aware requirements about robustness and confidence, or propagate a suitable view of all the location data produced by simultaneously working positioning systems to enable application-level choices on which positioning information to exploit. Let us note that proper management decisions could depend on synergic considerations deriving from the whole set of both running LBSs and positioning systems available at a client. For instance, if a positioning system is switched on because of LBS_1 requirements, it makes sense to exploit that positioning technique also for LBS_2, even if LBS_2 accuracy requirements are satisfied also by other positioning systems with lower energy consumption.

LBS developers currently have to know the details of the positioning system that will be available when deploying their services; LBS implementation is typically not portable and depends on the characteristics of the target positioning system (sometimes on the specific implementation of that positioning solution). Therefore, also due to the fact that current wireless clients tend to simultaneously host several wireless technologies useful for positioning (e.g., terminals with Wi-

Fi and/or Bluetooth connectivity and/or equipped with GPS), there is a recent and emerging research trend in support infrastructures for uniformly integrating heterogeneous positioning techniques.

The ultimate goal is not only the easy LBS portability over different positioning solutions dynamically retrieved at LBS provisioning time but also the exploitation of these context information for networking opportunity smart management. In other words there is the need for novel context-aware middleware solutions capable of propagating differentiated levels of visibility up to the application level and of synergically managing heterogeneous positioning systems and communication interfaces depending on application requirements, user preferences, device characteristics, and overall system state.

We claim the middleware has to provide integrated management of heterogeneous low-level components by adopting a **translucent approach**, intended as the simultaneous **provisioning to the application layer of both high- and low-level API**. Thanks to the translucent approach, applications aiming to interact with low-level components in a simplified manner, namely simple ABS applications, can get a transparent access via high-level API, thus perceiving the underlying available low-level components as a unique multi-behavior facility. On the contrary, applications willing to have direct visibility and to manage peculiar information/features of positioning systems, namely smart ABS applications, can interact in a middleware-mediated but fully aware fashion, via low-level API.

It is possible to summarize previous considerations defining the two following middleware design rules:

Design rule 1: **differentiated access to low-level details**. The translucent approach allows LBSs built on top of the middleware to get a uniform and aggregated access to all the characteristics of integrated positioning systems and communication interfaces. On the one hand, it provides applications with a uniform API independently of the specific positioning solution, e.g., to reduce overhead it is possible to limit the accuracy of Ekahau-based and BTProximity-based positioning in the same way. On the other hand, it permits to access/configure all the available context sources aggregately, e.g., to gather all the data about current accuracy from all activated positioning systems, with no need to interact with each positioning system separately. In this manner, applications can achieve a uniform aggregated (and thus simplified) access to lower layers. Let us note that other different research fields use “translucency” to name the flexible combination of both visibility and transparency: for instance, [Ramamurthy et al. 1999] and [Shen and Tucker 2007] adopt the translucent term to indicate similar hybrid visibility in the area of optical networks.

Design rule 2: **differentiated control of low-level components**. Furthermore, the middleware should not only work to expose the context data uniformly to the application level, but should originally permit to **control** positioning systems and communication interfaces behavior **with different levels of opportunities**. For example, the user could be interested in changing the location update frequency to decrease power consumption. Based on a middleware solution, the user could simply notify her requirement to the integrating middleware, thus delegating to it the burden of actually interacting with underlying positioning systems. Without the availability of an integrating middleware the user should personally and separately control every positioning system behavior. While most state-of-the-art integration middlewares limit their efforts in merging heterogeneous systems to provide a uniform static interface for location gathering and communication interface selection, a CAMPO middleware should actually put together low-level components to enable the integrated synergic control of their behavior by considering them aggregately. For instance, an LBS could command to simultaneously lower the power consumption of every positioning system just specifying to set the `PowerConsumption` control feature to `low`. About translucency in low-level component control, simple applications only have the burden of specifying desired behaviors, by delegating the middleware for any required action. For instance, the middleware could provide a set of pre-defined declarative policies and interface evaluation metrics that simple applications can only decide to de/activate. Note that the opportunity to control positioning systems via declarative policies and communication interfaces via differentiated metrics greatly facilitates application development because applications leave the burden of any required monitoring/control action to the middleware. Smart applications, instead, can directly control each low-level component features and capabilities in a fully-aware manner, via uniform middleware-mediated API. In this case, applications can access low-level API to interact with and control each low-level component separately, e.g., for the purpose of switching on/off and configuring a specific component.

2.3.2 Mobility-aware Heterogeneous Connectivity Provisioning

To understand how the availability of multiple interfaces on a mobile client can enhance the user experience suppose that Alice is at the University campus with her laptop, which hosts three different wireless cards for Bluetooth, Wi-Fi, and UMTS-based connectivity (see Figure 2.9). Alice is willing to browse the Web to download the files with today's lesson slides. To that purpose, differently from what today's solutions require, Alice simply opens her Web browser without specifying which wireless connectivity to exploit. In fact, it is the underlying CAMPO middleware that seamlessly discovers that there are currently two connectors available: a free Wi-Fi AP of the campus network and a UMTS BS of the telecom provider Alice is subscribed to. Depending on Alice's preferences, interface and connector selector middleware components transparently decide either to establish a channel based on the free AP if the reduction of economic costs is the priority (4G <N:N:1> scenario) or to simultaneously exploit two channels (based on Wi-Fi and UMTS) if Alice desires maximum bandwidth (ABC <N:M:M> scenario). Then, Alice meets her colleague Bob who carries a smart phone with Bluetooth and UMTS interfaces. Bob is willing to download Web files as well and opens an ftp client on his phone. This time the middleware, by exploiting the applicable context with Bob's preferences of prioritizing free connectivity, transparently establishes a Bluetooth-based channel towards Alice's laptop, which works as a collaborative peer to forward packets via the Wi-Fi AP.

Afterwards, Alice moves towards a green recreational area where there is no campus Wi-Fi coverage; there, she requests streaming an audio file from an Internet server. Since the applicable context specifies to prefer free connectivity opportunities and the middleware estimates that Carol's laptop is still relatively to Alice, the middleware now decides to establish a channel via ad hoc Wi-Fi towards Carol's laptop (behaving as a peer connector) who is studying at the park boundaries covered by the campus Wi-Fi signal. Finally, Alice decides to go home by continuing to listen to the streaming audio service: when Carol's connectivity is lost, the middleware automatically re-qualifies Alice's channel by exploiting the UMTS connector during the way back and the domestic (less expensive) Wi-Fi AP when entering home. If Alice was simply interested in sending non-urgent (delay-tolerant) emails while moving back home, the middleware could decide to temporarily store the messages and forward them only when encountering free connectors; temporary channel interruptions are not an issue for this service.

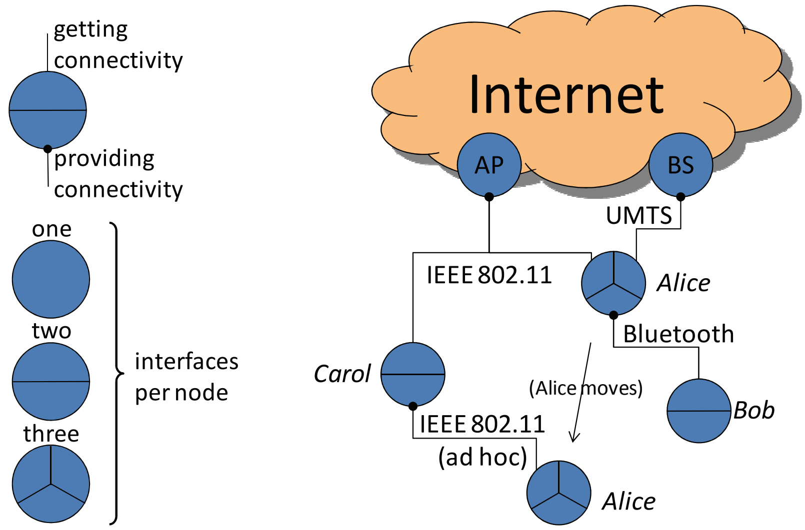

Figure 2.9 A simple example of envisioned application in an ABS scenario.

Note that Alice, Bob, and Carol mobility may relevantly reduce the reliability of these self-organizing connectivity opportunities; depending on application-specific requirements, there is the need to favor the selection of connectivity opportunities with compatible durability, which should be estimated based on practical, lightweight, and effective mobility awareness.

The previous example can be further evolved to a more complex one characterized by stronger heterogeneous ad hoc multi-hop multi-path connotation. When moving from city to city by train, Carol should be able to exploit connectivity opportunities offered by other passengers, possibly in other wagons, reachable via multi-hop heterogeneous paths, and connected to the Internet via Wi-Fi/Wi-MAX APs, such as node F in Figure 2.10. In this case the nodes tend to move together (joint mobility) and connectivity opportunities have similar expected durability. Therefore, connectivity selection should not only be mobility-aware, but also consider application-specific quality requirements, e.g., expected throughput. Thus, there is also the need for practical, lightweight, and effective ways for coarse-grained estimation of the quality of available connectivity opportunities. Note that the synergic management of multiple connectivity opportunities can also improve handover effectiveness in the case of abrupt path unavailability, e.g., by seamlessly re-routing traffic to a different collaborating tourist with Bluetooth/UMTS when the one with Wi-Fi/WiMAX leaves the train at a station.

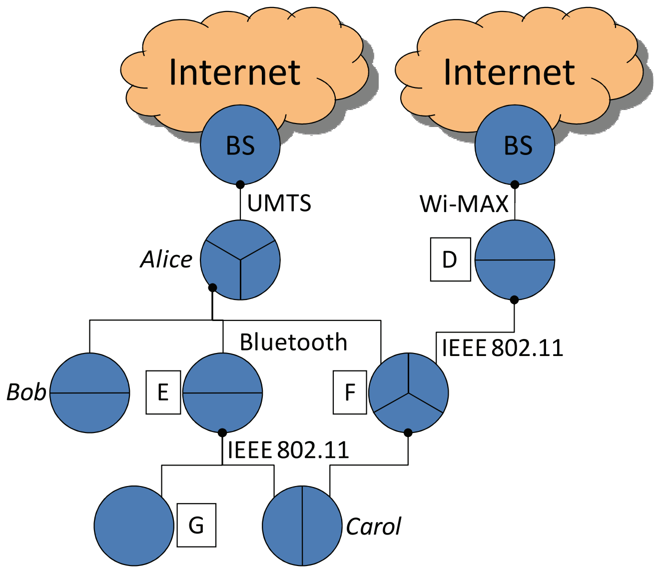

Figure 2.10 Multi-hop multi-path ABS example.

The above application scenario, even if addressing the simplified and easy understandable case Internet connectivity, e.g., excluding peer-to-peer non-Internet-based services, clearly shows the potential complexity of the targeted problem and the unsuitability of delegating final users to take proper channel management decisions. There is the need for innovative and effective middleware to properly handle the numerous technical challenges involved, from heterogeneity of interfaces/connectors and their proper combination for channel establishment, to context-dependent channel re-qualification at runtime.

The crucial design aspect is the adoption of proper simplifying assumptions on monitoring indicators with the goal of achieving the most effective tradeoff between management overhead and optimality of managing a wide set of networking opportunities. We claim that the design rules we propose below relevantly affect the achievable performance/overhead tradeoff and provide a useful methodological how-to for an **effective support of both infrastructure and peer-to-peer connectivity**.

Design rule 3: **tradeoff between local and global management**. Let us start by noting that some monitoring information, such as RSSI of visible Wi-Fi APs and mobile peers, are anyway locally provided by wireless interfaces, with no additional management costs, thus allowing some forms of local coarse-grained evaluation of single-hop paths with minimum intrusion. On the opposite, in general, any evaluation of multi-hop paths requires, to some extent, the distributed coordination and transfer of non-local monitoring data. More global the visibility of monitoring data, more intrusive the management operations and closer to optimality the connectivity decisions. That forces, first of all, to clearly identify what the middleware should perform locally and what with a global management perspective.

On the one hand, we propose to exploit locally available monitoring data, i.e., for each wireless interface the set of fixed/mobile devices offering connectivity to take local decisions about the suitability degree of available collaborating devices. Therefore, at any node a middleware solution should determine a limited subset of neighbors to activate a single-hop connection with. Limiting the subset to a very small cardinality of 2 or 3 is sufficient in most application scenarios, with positive effects in terms of management overhead reduction. On the other hand, we claim the need also for a second-step global phase, where the middleware collects additional monitoring data only for the potential multi-hop paths enabled by the subset of activated single hops. Note that the local phase can work also before any single-hop connection is established, without introducing any additional communication overhead, while the global phase includes distributed management operations, such as IP routing updates.

Design rule 4: **tradeoff between single- and multi-path granularity**. A middleware solution should decide the most suitable level of granularity to allow in the relationships between applications/traversing-flows and potentially available paths (one path for all applications/flows, one path for each of them, or multiple paths even for the same one). We use the term single-path granularity for a node to indicate the case where packets from/to different clients and applications are routed all in the same way at that node. At the other extreme, multi-path granularity identifies the more flexible case where the middleware can manage multiple paths even for each application/flow, by increasing middleware complexity but taking full advantage of the available multi-hop multi-path opportunities. We propose a middleware that can be dynamically configured to work in either single- or multi-path granularity modes depending on performance/overhead requirements at runtime deployment.

Design rule 5: **tradeoff between static and dynamic responsiveness**. We call static management the middleware behavior of re-evaluating networking opportunities and possibly changing routing rules only when either the exploited paths are broken or there are new single-hop paths available. Dynamic management, instead, indicates the more aggressive polling-based monitoring of the wireless environment, e.g., the estimated throughput for any potential multi-hop path and the consequent modification of routing choices at each node. Of course, dynamic management requires additional computing, communication, and power costs, but more promptly adapts to runtime variations of path durability/throughput. We propose to carefully consider static/dynamic management tradeoff depending on local/global management phases and their related costs.

In conclusion, we propose to adopt a context-aware middleware solution to support the easy development and deployment of applications taking full advantage of the ABS scenario. We believe that based on the above five design rules, it is possible to achieve a proper tradeoff among the complexity of monitoring and controlling procedures and the capability to exploit meaningful and up-to-date context information to take the most suitable control decision. In addition, it is possible to manage the great complexity deriving by the simultaneous exploitation of multiple heterogeneous wireless interfaces in both infrastructure and ad hoc fashion, even by handling management modifications during service provisioning.

For readers' convenience we summarize together the five design rules:

1) **differentiated access to low-level details**, that is a translucent access to low-level components, providing the capability to interact both in a simplified and direct way;
2) **differentiated control of low-level components**, since there is the need of permitting to control the behavior of low-level components both in a middleware-mediated and fully-aware fashion;
3) **tradeoff between local and global management**, considering that global visibility of monitoring data delves into more complex and expensive management operations but even more suitable connectivity decisions;
4) **tradeoff between single- and multi-path granularity**, selecting the proper granularity degree in relation to the need of providing a per-client/per-application/per-flow differentiated quality of service;

5) **tradeoff between static and dynamic responsiveness**, carefully considering that greater is dynamicity, more promptly reacts the middleware to environment changes, but greater is the computational, power, and communication cost.

The presented design rules will guide the definition of the general architecture and the work of implementation of our middleware solution able to synergically integrate and manage multiple and heterogeneous positioning systems and communication interfaces, which will be extensively presented in Chapters 4 and 5. Prior to the description of our novel middleware, Chapter 3 will present an in-depth analysis of the state-of-the-art literature contributions considering already proposed solutions for context sources integration, dynamic remote service discovery, and context-aware management of communication opportunities.

Chapter 3 - A novel Taxonomy to Model and Classify the CAMPO Area

As already presented in the previous chapter, the CAMPO model includes a wide area of research activity, from 4G systems for vertical handover management to ABC/ABS ones for ad hoc networking self-organization. In other words the CAMPO model permits to effectively categorize all the contributions in the literature that manage networking opportunities in a context-aware way. This chapter provides a description of the state-of-the-art in the CAMPO area based on a novel taxonomy that considers both context information gathering and networking opportunity evaluation/management.

As already emerged by the previous design rules, we believe that the exploitation of the many valuable context information available on the mobile client is a primary issue that CAMPO solutions have to address. The literature has mainly focused on the capability to provide a homogeneous access to this type of context source and information. For this reason the state-of-the-art analysis related to context gathering has the primary goal of comparing solutions for the integration of heterogeneous context sources, in particular positioning systems. Their common objective is to merge provided information and gain an integrated synergic exploitation of several context sources simultaneously. Section 3.1 first presents research efforts on **integration middlewares, ordered according to the level of visibility propagated to LBSs**, from transparent solutions that hide any low-level positioning system detail, to contributions with partial visibility and control for LBSs built on their top. The second part of Section 3.1, instead, focuses on JSR-179 that represents the most notable standardization effort for Java-based LBSs on mobile phones [JSR-179]. JSR-179 provides a standardized API to perform coarse-grained integration and some limited forms of control of underlying positioning systems.

Several CAMPO contributions do not consider context gathering as a primary issue. Instead, they are specifically focused on the evaluation of networking opportunities based on heterogeneous context information and on the management of connectivity continuity when mobile clients perform a handover procedure from an origin to a destination connector. For this reason, the chapter provides an in-depth analysis of CAMPO systems proposing novel solutions to evaluate networking opportunities and supporting the dynamic change of connector. In particular, Sections 3.2 and 3.3 aim to achieve the twofold goal of i) **identifying the primary design choices** emerging from available solutions, together with corresponding tradeoffs, and ii) **better positioning the wide variety of related work in a single original taxonomy**. In particular, we propose a novel classification able to cluster any state-of-the-art solution along three directions: deployment scenarios, evaluation process, and continuity management. In short, deployment scenarios represent and define the characteristics of target execution environments. The evaluation process specifies how to instantiate and update channels (interface and/or connector selection). Continuity management identifies the set of mechanisms, algorithms, and tools to actually operate horizontal/vertical handovers, possibly in a seamless way notwithstanding runtime client mobility.

3.1 Context-source Integration Middlewares

Several research activities have recently started to address the emerging field of the integration of heterogeneous context sources and, specifically, positioning systems. The main goal is to put together and suitably merge information from different sources, by providing a uniform interface that applications can easily exploit independently of the context sources and positioning solutions available at runtime in their deployment environments. However, most research activities only concentrate on the issues of uniform access to context sources and fusion of information, without giving possibilities either for synergic management of heterogeneous context sources or for exploitation of context awareness to guide management decisions. On the opposite we claim that these are crucial aspects for spreading the adoption of advanced context-aware applications.

In particular, as the next section will show, already proposed integration middlewares lack in dynamicity, management capabilities, and extensibility. First of all, already proposed solutions usually provide hard-coded algorithms for information fusion and hard-coded policies for dynamic change of context sources, e.g., exploiting always the positioning system with greatest accuracy despite the imposed power consumption. Secondly, traditional middlewares do not provide any capability to change context sources behavior from the application layer. The only possible interaction is collecting information in a bottom-up fashion; applications are prevented from any top-down interaction to control underlying components behavior. Finally, the contributions in the literature support the delivery of only pre-defined information, e.g., location information as a latitude, longitude, altitude triple; context sources are not allowed to provide additional information specifically considered by the overlaying integrating middleware, e.g., the expected accuracy degree of the provided location information.

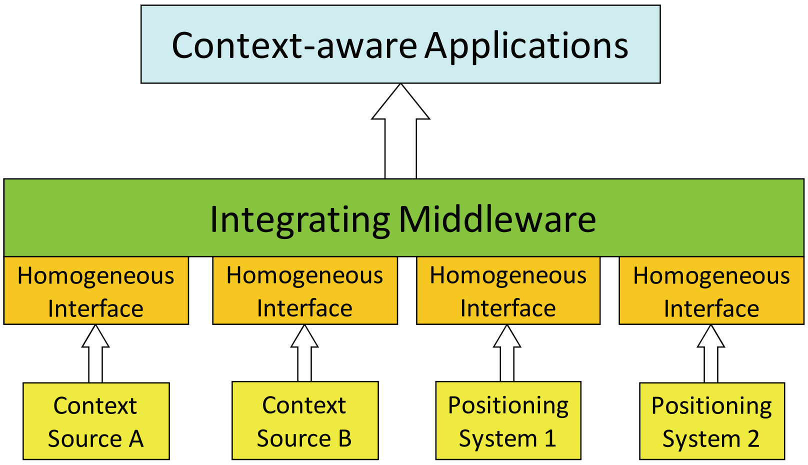

Figure 3.1 Usually adopted integrating middleware solution.

As Chapter 4 better details, our purpose is not confined to supporting a homogeneous access to positioning systems and integrating available context sources to merge their provided information. In fact, we envision as crucial the designing and development of novel middleware solutions able to dynamically control integrated context sources behavior, e.g., to switch on/off available ones depending on their availability and application requirements.

3.1.1 The Notable Case of Positioning Integration Middlewares

As already stated, a main property to differentiate positioning integration solutions in the literature is the degree of visibility propagated up to the LBS application level. Contributions in the field span from completely transparent approaches hiding LBSs from the complexity of direct interaction with positioning systems but not providing any control capability, to integration solutions allowing limited controllability but complicating the development of LBSs, which have to statically embed details about the exploited positioning techniques directly in their application logic.

In order of increasing level of visibility, the Alipes architecture focuses on the integration of heterogeneous positioning systems through appropriate wrappers to provide LBSs with a uniform API [Nord et al. 2002]. The goal is to force the exploitation of the available positioning system that best fits LBS accuracy requirements, by possibly performing location data fusion in order to achieve the required robustness of positioning data. Moreover, Alipes provides user-controlled privacy, by requesting explicit user permission before disclosing location information. The integration system proposed in [Hosokawa et al. 2004] has the primary goal of seamless navigation via uniform map-based interfaces, regardless the actually exploited positioning system. Its main solution guideline is to exploit middleware components, called mediators-wrappers, to abstract from specific peculiarities of used positioning systems and maps. In addition, [Hosokawa et al. 2004] permits to dynamically switch exploited positioning system in a completely transparent way. The integrated Platform for Location-based services (PoLoS) offers an API to facilitate the development of new LBSs [Spanoudakis et al. 2003]. It also supports the introduction of new positioning systems through a plug-in architecture; the middleware interacts with positioning systems in a standardized way via OSA/Parlay. Similarly, the Framework for Location Aware ModElling (FLAME) is a transparent integration middleware: it bases its positioning abstractions on a multi-step architecture for location data fusion, generation of geometric relationships, and event-based location data disclosure [Coulouris et al. 2002]. Finally, the Location Operating REference model (LORE) originally proposes different abstracting steps to provide high-level location data, independently from low-level details: positioning, modeling, fusion, query tracking, and intelligent notification [Chen et al. 2004]; in addition, it ensures privacy and security management, by controlling information disclosure, similarly to Alipes. Positioning system integration in LORE is achieved by the Common Adapter Framework that provides a standard API to fetch location information.

The above middlewares integrate positioning systems with the primary goal to facilitate LBS development. They tend to propose **transparent approaches that hide LBSs from positioning complexity**, but do not support any application-specific form of configuration, control, and management of positioning techniques. The main contribution of those proposals is to offer a framework to quickly prototype and deploy LBSs. However, they relevantly limit the capabilities of advanced LBSs, often interested in performing context-aware, cross-layer, and portable positioning management operations.

Only a few proposals have recently started to provide some forms of visibility of low-level features/characteristics, by **introducing the partial possibility of cross-layer approaches and limited control**. This demonstrates that it starts to be recognized the need for mediated visibility of underlying positioning systems, in order to achieve effective, application-specific, and context-aware management decisions, even if risking to complicate and slow down the realization of LBSs.

In particular, MiddleWhere provides LBSs with some low-level positioning details, such as location resolution, confidence, and freshness [Ranganathan et al. 2004]. Adapter components act as device drivers, thus permitting to MiddleWhere to communicate with positioning system implementations: each adapter makes location descriptions uniform by hiding positioning system implementation peculiarities. The Location Services Module (LSM) supports not only positioning data merging but also some forms of control of heterogeneous positioning systems [Agre et al. 2002]. However, it performs merging and control in a hard-coded and not flexible manner: to achieve visibility of data/control features for a specific positioning system, LSM-based LBSs should have full static knowledge of positioning characteristics, e.g., should know name and syntax of positioning-specific control functions. Location Stack represents a state-of-the-art model of solution for location (and also context in general) data fusion [Hightower et al. 2002]. It identifies several middleware components, deployed in layers, which can sequentially (as stages of a pipeline) provide increasing levels of abstraction: Sensors, Measurements, Fusion, Arrangements, Contextual Fusion, Activities, and Intentions. However, the first implementation of it, namely the Unified Location Framework (ULF), has shown that such a highly-layered system is unsuitable for properly propagating the visibility of low-level data such as accuracy and precision, often useful for application-level LBS decisions [Graumann et al. 2003]. In other words, the ULF implementation experience points out the need for cross-layering to expose low-level details to LBSs and to activate direct control of positioning features from application logic.

In conclusion, most proposals in the literature only address the positioning integration issue while hiding low-level details depending on positioning technique and system implementation. MiddleWhere, LSM, and ULF are the only ones that offer partial visibility of positioning data characteristics and control features, but in a statically pre-determined way.

3.1.2 The JSR-179 Location API for J2ME

In the last years, the industrial research activity has primarily focused on the development of standards to address the wide heterogeneity of available positioning systems. The JSR-179 API [JSR-179], also known as Location API for J2ME, represents the most notable result of that standardization effort for Java-based LBSs on mobile phones. JSR-179, inspired by the usual and widespread interface of the GPS solution, provides **a standardized API to perform coarse-grained integration and control of positioning systems** (location providers according to the JSR-179 terminology). To better understand how JSR-179 provides location information, here we rapidly report its main characteristics and offered functions.

The `LocationProvider` class is the JSR-179 API entry point. Applications invoke the `getInstance()` method of `LocationProvider` to retrieve an actual location provider implementation among the currently available ones. The actual location provider is the selected positioning system that returns location information to applications.

When invoking the `getInstance()` method, an application optionally specifies particular criteria (`Criteria` class) that the actual location provider must satisfy. If several actual location providers are compatible with the passed criteria, `LocationProvider` selects the one which best fits the requirements according to a pre-determined strategy. Criteria can specify that the actual location provider must supply speed and altitude, and/or that the provided horizontal/vertical coordinates have to respect a minimum accuracy level. Moreover, it is possible to specify the desired power consumption (low, medium, or high). Let us notice that the passed criteria are exploited only at the moment of the selection of the actual location provider; they are completed neglected at provisioning time.

Figure 3.2 depicts an example of application that requests an actual location provider implementation, by specifying the desired selection criteria. The result is the activation of the positioning system best fitting the criteria among the currently available ones (Location Provider 2 in the figure). Location Provider 2 is associated with the application until a new explicit request of location provider selection to the JSR-179 API.

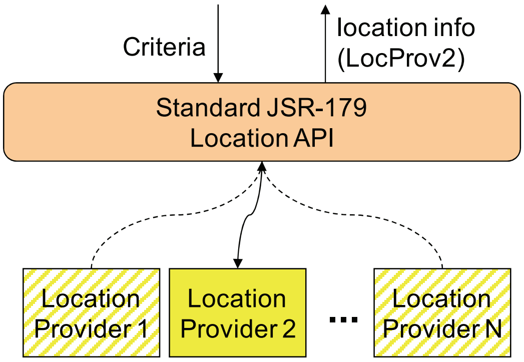

Figure 3.2 The JSR-179 API for criteria-based selection of an actual Location-Provider implementation.

Location providers return location data in three different ways:

- on demand, via the `getLastKnownLocation()` and `getLocation(timeout)` methods, which respectively provide cached and just updated location information, the latter actively requesting for new data to the underlying positioning system;
- periodically at fixed time intervals, via the method `setLocationListener (listener, interval, timeout, maxAge)`. Only one periodical listener at a time can be registered with each location provider instance;
- in an event-driven fashion via the `addProximityListener (listener, coordinates, proximityRadius)` method. The only triggering event that can be exploited in JSR-179 is the proximity of the located client to specified coordinates. Several proximity listeners may contemporarily indicate multiple coordinates close to which a location provider triggers the events.

The provided location information specifies qualified coordinates (physical location), address info (symbolic location), or both. Moreover, it may include additional data such as speed, timestamp, and the technology of the actual location provider.

JSR-179 is a good example of standardization effort in the industrial research area to leverage the adoption of positioning systems and LBSs. Its architecture and API have the goal of representing a standardized model for every developer willing to provide new positioning systems or LBSs. However, we claim that JSR-179 does not provide a sufficiently expressive API to perform efficient integration and control of positioning systems. In particular, it supports neither the dynamic management of multiple location providers nor the provisioning of low-level system-specific details to the application level as required by many LBSs.

First of all, **it does not support the dynamic and flexible management of dynamically retrieved location provider implementations**. On the one hand, JSR-179 only permits to exploit one location provider at a time among the ones currently available at a client, even if several of them satisfy the specified criteria. On the other hand, according to the JSR-179 specification, LBSs have the full duty of monitoring the performance of the selected location provider and of taking suitable management operations consequently, e.g., requesting for a new location provider selection in response to accuracy degradation. In other words, once JSR-179 has selected a location provider, the specified criteria are no more considered even if the capabilities of the actual location provider do not satisfy the LBS requirements any more or if a new more suitable location provider becomes available at the client.

In addition, the JSR-179 API assumes that the **characteristics of location providers are statically identified and do not considerably change over time**: that is partially true for static features, e.g., ability to provide speed/altitude or not, but not applicable to dynamic characteristics such as horizontal/vertical accuracy. For example, GPS accuracy may abruptly decrease when the user moves from an outdoor to an indoor environment. Moreover, JSR-179 has dynamicity and flexibility limitations also due to its impossibility to accommodate new positioning systems newly introduced at service provisioning time. The actual location provider implementation is determined only once at the moment of location provider instantiation; JSR-179 does not consider any context change after that instantiation, until a new LBS request for actual location provider determination. Another limitation of JSR-179 is that selection criteria are limited to few and statically predetermined elements. It is possible to specify as requirements only the features defined in the criteria class before service provisioning. Moreover, also the event handling functions of JSR-179 exhibit non-negligible limitations, as already pointed out: only one type of triggering event is supported, the one related to proximity to a fixed location.

But, according to our opinion, corroborated by our experience in developing and prototyping LBSs, JSR-179 exhibits the most relevant lack in its **limited capabilities to propagate the visibility of low-level details of underlying location providers** when needed. In fact, the only state information available about location providers is their availability status (available, temporarily unavailable, or out of order). This full and uniform transparency of low-level positioning system features does not always fit the requirements of application-level visibility typical of LBSs. For example, a LBS would get and control peculiar positioning system functions, such as to get and possibly change the location provider privacy level.

The academic research on the extension of JSR-179 capabilities to achieve greater flexibility and dynamicity is still at its very beginning, also due to the novelty of the standardization effort. [Di Flora et al. 2005] proposes the integration and management of multiple positioning systems via a JSR-179 fully compliant API. It tries to increase dynamicity by transparently switching among available positioning systems: in particular, it alternatively exploits either GPS/Bluetooth-based positioning dependently on client outdoor/indoor location. However, the proposal does support neither the dynamic change of positioning selection criteria (only system availability), nor the integration with new positioning systems at provisioning time. Moreover, it does not provide any function at all to control integrated positioning systems from the application layer.

3.2 An Original Comprehensive Taxonomy for CAMPO Solutions

CAMPO systems may pursue goals at different abstraction levels and with different flexibility, by considering even very diverse assumptions on their working environments. Here we aim to provide an exhaustive description of all the possible differentiated characteristics of CAMPO systems, organized in a structured classification that permits to clearly position all CAMPO contributions in the literature.

Figure 3.3 graphically depicts the proposed comprehensive and original taxonomy for CAMPO solutions. Our taxonomy is structured according to three main categories: deployment scenarios, evaluation process, and continuity management. In addition, it shows how our taxonomy refines each primary category by identifying first-, second-, and third-level sub-classes, structured in a hierarchical tree organization. The characteristics, relevance, and CAMPO system coverage for each subclass will be described in the following sub-sections.

We claim that our articulated taxonomy can clearly classify the whole set of CAMPO systems, by facilitating the identification of similarities/differences between the various contributions. In fact, on the one hand, the three primary categories address all the primary aspects of state-of-the-art CAMPO solutions, even if those aspects are sometimes identified with different terms in the literature. On the other hand, they represent the three crucial families of design choices to consider for researchers and developers when designing novel CAMPO solutions. Other support functions, e.g., quality management, that may be employed in CAMPO systems but are not specific of the CAMPO research field, are outside the taxonomy, which aims to center the specificity of the area.

To permit the full understanding of the sub-class structuring detailed in the following sub-sections, let us first introduce what we intend exactly for the three taxonomy categories. We define **deployment scenario as the set of assumptions and related constraints** that a CAMPO solution adopts depending on its expected and/or current working environment. In other words, the deployment scenario category relates to working environment characteristics such as the set of selectable channels (possibility to choose either interface or connector or both), the number of interfaces that may be simultaneously active at clients, and the role/location of support components, e.g., to locally/remotely trigger handovers and to perform session transfer.

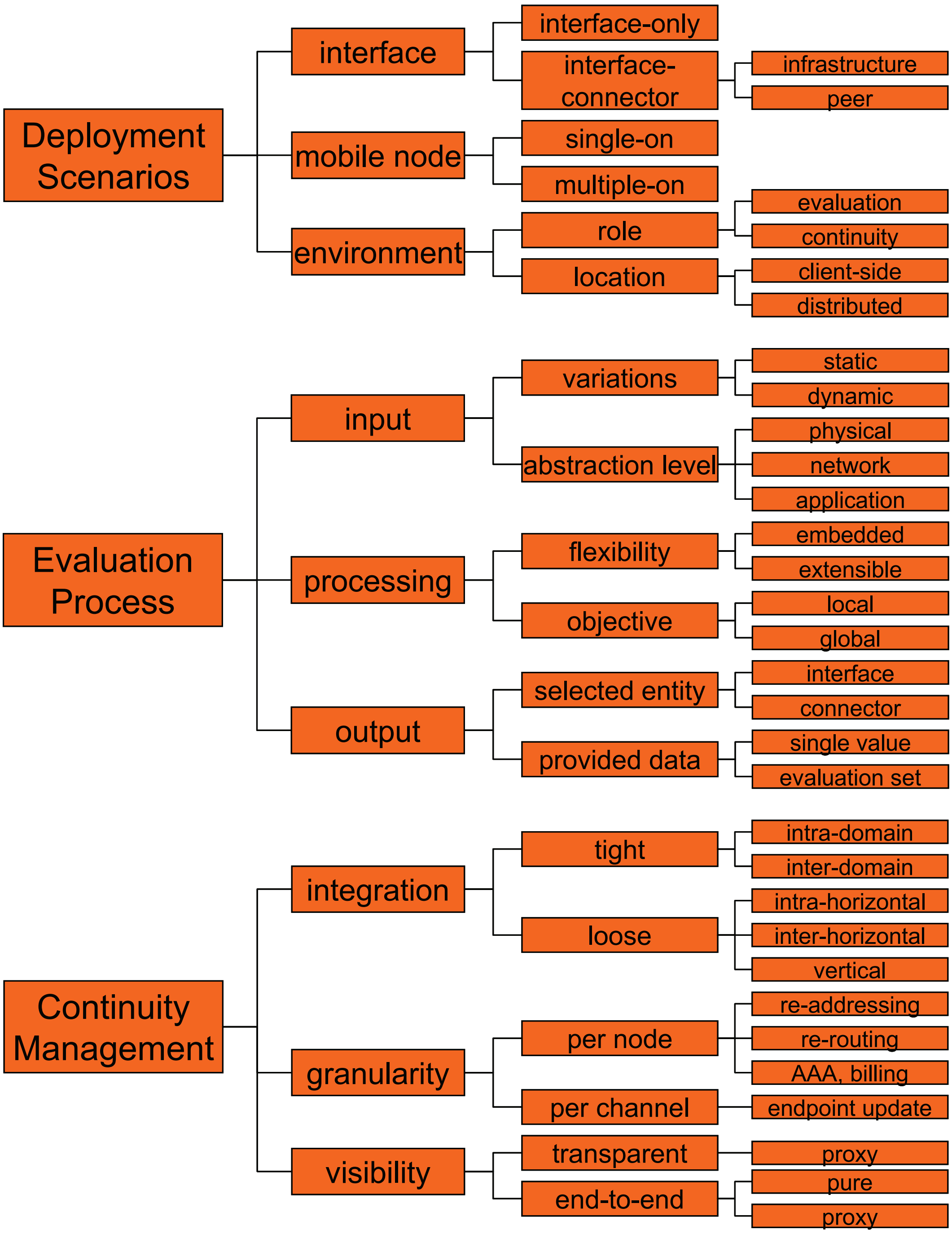

Figure 3.3 The proposed CAMPO taxonomy.

While the deployment scenario identifies the addressed working environment, evaluation process and continuity management categorize the primary operations that a CAMPO system must perform to provide final users with suitable and updated channels. In particular, **the evaluation process is in charge of gathering context information** and consequently **providing a quantitative estimation of the suitability of available channels** according to an adopted metric. Any metric consists of three main parts, which determine its complexity and expressiveness: input, processing method, and output. The input sub-class defines the level of abstraction and variability of the exploited context information, e.g., the continuously changing bandwidth offered by a connector or the more static user-level indications about preferred interfaces. The processing method mainly influences metric flexibility (ability to vary the adopted metric at provisioning time) and the scope of CAMPO goals (e.g., the global objective of optimal load sharing among all access networks). The output sub-class defines the granularity of selected entities and the result type provided by the evaluation process, e.g., the most suitable interface or a set of values quantifying the suitability of any potentially available connector.

Finally, **continuity management relates to all the mechanisms, tools, and strategies to actually perform the update of active channels at provisioning time** without user-perceivable service interruptions. Continuity management not only decides when and how to update channels depending on the evaluation process output (trigger component) but also provides the support mechanisms to seamlessly switch among interfaces and/or connectors at runtime (switcher component). Let us note that first simple CAMPO systems did not include continuity management facilities at all, or sometimes a very limited capability subset, by leaving application developers the burden of addressing the challenging continuity issues associated with channel update. We claim that, nowadays, the availability and flexibility of continuity management solutions are crucial aspects of CAMPO systems because of the growing relevance of mobile continuous services, such as mobile multimedia. That is the reason why we have decided to originally devote one whole specific category of the taxonomy to that demanding aspect.

Most state-of-the-art CAMPO systems only concentrate on one of the three categories and adopt solutions proposed by other CAMPO researches for the remaining two. For instance, many contributions have focused on identifying original strategies for interface/connector selection by delegating to Mobile IP the burden of partially solving continuity issues. Other CAMPO systems propose innovative continuity management solutions by leaving to mobile nodes the burden to evaluate when, where, and which type of handover should be performed. Moreover, as better detailed in the following, some CAMPO proposals provide seamless horizontal/vertical handover of service sessions by imposing specific constraints on deployment scenarios, i.e., by assuming the presence (and full visibility) of needed support components in specific locations. Anyway, in general, even if the three categories are not strictly correlated, some assumptions in a category may limit the space of potential design choices for the remaining two. In other words, the chosen deployment scenario subclass may limit the available design choices for CAMPO developers by making some evaluation and continuity solutions not viable. For instance, it is hard to pursue a global objective in a deployment scenario where CAMPO components are located only client-side, global context information is often not available, and continuity management is usually performed in an end-to-end way.

The following sub-sections extensively describe and discuss the three adopted taxonomy categories, by pointing out the associated design options for CAMPO developers.

3.2.1 CAMPO Deployment Scenarios

The variety of state-of-the-art CAMPO solutions also depends on the fact that they focus on **different target deployment scenarios**. It is possible to identify three main categories of deployment scenarios where CAMPO solutions take decisions based on i) the only information about available interfaces (and possibly associated connectors) at each client node, ii) the additional data about working channels and overall node capabilities/requirements for each client node, and iii) the additional knowledge about the whole execution environment, including the infrastructure side, such as the presence of auxiliary CAMPO-related support components.

CAMPO systems in the first category (**interface scope**) establish and update channels on a node depending on interface scope, i.e., the set of static/dynamic characteristics of the interfaces (either active or not) available at the node. These CAMPO solutions can only consider aspects such as possibilities of interface control and programmability: in fact, some network interfaces only allow to be switched on/off, while in other cases dynamic interface management is possible with a finer control degree, e.g., by deciding when and to which connector to command the horizontal handover of an interface. By focusing on the second sub-category (**mobile node scope**), these CAMPO solutions can consider the whole client node environment to guide their decisions. Depending on the whole set of not only active/inactive interfaces (and their related connectors) but also of already established channels and node requirements, e.g., a constraint on overall maximum power consumption, they could exploit one interface at a time (single-on) or even multiple interfaces simultaneously (multiple-on). Finally, with a higher degree of visibility of the deployment scenario, CAMPO solutions in the third category (**environment scope**) can consider the whole deployment environment, including the availability of external support components, their location, and potential role. For instance, they can update channels depending on user-specific continuity requirements by considering whether infrastructure-side components for continuity support are available, such as proxies bi-casting traffic to origin and destination networks during handovers [Shenoy and Montalvo 2005].

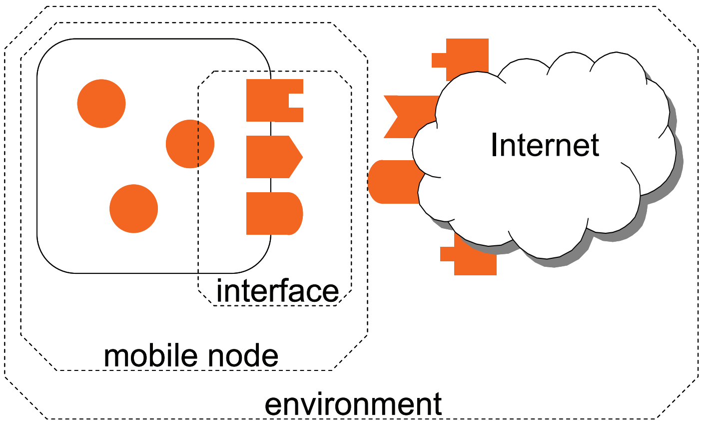

Figure 3.4 The three subclasses for deployment scenarios with different visibility scopes.

Figure 3.4 depicts the three deployment subclasses, by pointing out that they identify scenarios with enlarging scopes (and consequently increasing complexity). The decision of which deployment scenario sub-category to address strongly contributes to determine the capabilities and limits of a CAMPO solution. For instance, a deployment limitation may impact on the possibilities to select connectors, e.g., a specific deployment environment (mobile node scope, single-on) may not permit to consider a connector, which is instead available in multiple-on environments. In addition, some CAMPO design choices may impose constraints on target deployment environment, e.g., the minimization of power consumption may force to consider only single-on scenarios. Therefore, in the following we try to clearly delineate the three sub-categories to simplify both the understanding of CAMPO capabilities/limits and the associated design choice possibilities.

3.2.1.1 Interface Scope

Interface scope CAMPO solutions may be classified into two primary types: interface-only and interface-connector. **Interface-only** systems can select only the interface to exploit (possibly by switching on inactive interfaces), by delegating any other interface management action to the embedded and not-modifiable firmware/hardware of the selected interface. In other words, interface-only CAMPO systems address deployment scenarios where, given an interface, the selection of its associated connector is outside control. On the contrary, **interface-connector** CAMPO systems can additionally select the connector to exploit via interface-embedded capabilities, which could depend on client card implementations. For instance, UMTS does not enable the choice of a specific BS (interface-only), while ABC solutions usually permit to switch to a selected connector (interface-connector).

By focusing on related design choices, the decision of assuming a target deployment scenario with interface scope significantly limits the design possibilities for CAMPO developers. In very simple CAMPO solutions of this class, the choice of the connector for a given interface is completely delegated to the embedded behavior of low-layer communication components. For instance, the selection of a specific IEEE 802.11 AP among the set of APs in visibility is typically embedded in Wi-Fi card implementations. Only very simple metrics can be specified: for instance, if the most important requirement is to minimize power consumption, a CAMPO system gives priority to the activation of low consumption interfaces, such as Bluetooth, without considering any other context indicator. In slightly more flexible cases, interface scope CAMPO solutions can also decide which connector to exploit among the available ones for a given interface, for instance the AP connector with minor congestion among the trusted ones. That is obviously possible only when interface implementation also supports connector selection. In these cases, CAMPO systems are in charge of comparing not only interface capabilities (often statically pre-defined) but also connector-related performance indicators, such as currently offered QoS level, thus calling for resource-demanding runtime monitoring functions.

Let us note that interface scope CAMPO solutions usually consider no more than infrastructure-based connectors, such as IEEE 802.11 APs or UMTS BSs. Only recently, first CAMPO proposals have started to work on the possibility to exploit also nearby client nodes as peer connectors. A peer connector can either behave as a bridge between the client and the fixed Internet infrastructure or directly offer services in a peer-to-peer fashion. The possibility to consider peer connectors provides a deployment scenario that greatly differs from traditional infrastructure-based environments, by requiring to take into account peer characteristics of various nature. In fact, while infrastructure-based connectors are usually fixed and reliable, peer nodes may be mobile, statically unknown, and with variable levels of trust. Moreover, clients and peer connectors should share a common goal to make their connectivity convenient for nearby peers. In addition, peer connectors typically provide connectivity in an intermittent way since i) their offer produces additional computational, networking, and energetic loads for them, and ii) their possible mobility increases the probability of connectivity loss for their associated clients. These challenging aspects represent state-of-the-art open issues and call for original approaches to manage novel forms of context, e.g., peer mobility indicators.

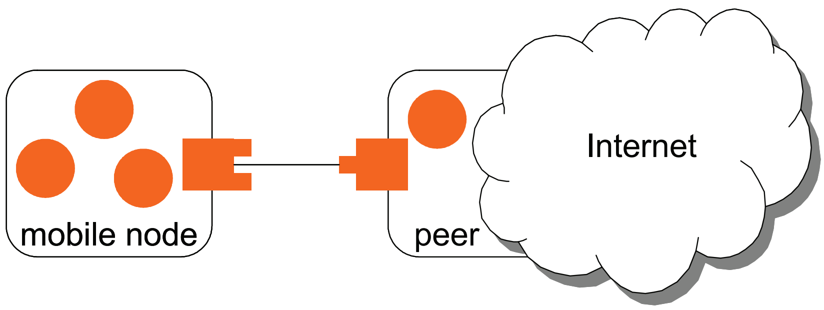

Figure 3.5 An example of deployment scenario with peer connectors.

Finally, let us rapidly observe that the CAMPO architecture model presented in Chapter 2 can also apply to the case of peer connectors, which in their turn can exploit channels to connect to the Internet or to other peer connectors. In the simple example of Figure 3.5, indeed, a client hosts only the Bluetooth interface, which is used to connect to a nearby peer connector offering Internet connectivity, e.g., via its Wi-Fi interface connected to a free IEEE 802.11 hotspot (second connector, in pipeline with the first one).

3.2.1.2 Mobile Node Scope

By focusing on the second category of our deployment scenario classification, CAMPO systems with mobile node scope may concentrate on exploiting either only one network interface at a time (single-on) or several interfaces simultaneously (multiple-on). For instance, 4G solutions usually exploit only one interface at a time: every channel uses the same interface, the currently active one. On the contrary, ABC solutions can take advantage of several interfaces simultaneously and can establish different channels by exploiting different, concurrently activated interfaces.

The choice between single-on and multiple-on deployment scenarios is probably the most important aspect affecting CAMPO design choices. Sometimes that decision is forced by the addressed execution environment. In fact, the possibility of having multiple active interfaces at the same time usually depends on client capabilities. Limited mobile nodes often prefer simple, lightweight, and energy-preserving CAMPO solutions, e.g., there should be only one active interface at a time, selected according to a static priority order. As resource availability on common clients is increasing, CAMPO systems start to favor smarter and more resource-consuming solutions, e.g., which exploit several interfaces at the same time to widen the available bandwidth or to provide continuous connectivity via duplicated data flows during vertical handovers [Shenoy and Montalvo 2005].

In addition, the single-on/multiple-on deployment category influences other relevant design choices. For instance, CAMPO single-on solutions have to choose the only one interface to activate at a node by taking into consideration the requirements of the whole set of running applications and trading among them. Multiple-on solutions, instead, are inherently more complex because they must allow the simultaneous management of multiple interfaces. Anyway, in single-on solutions the decision to change the activated interface implies evaluating the costs to update all channels, while multiple-on solutions can activate new interfaces independently of other working channels.

3.2.1.3 Whole Environment Scope

To the purpose of proper channel selection/update, CAMPO solutions may additionally consider the possible availability of support components with different roles (evaluation process and continuity management) in the execution environment, located either on the client side or distributed on both client and infrastructure sides.

About the role of support components in the execution environment, the variegated set of mechanisms and tools of interest for CAMPO systems span from QoS channel monitoring to more general context gathering, from metric evaluation to the triggering of continuity management operations, from context transfer in response to handovers to AAA support. Here, we do not provide the exhaustive description of CAMPO support mechanisms because they will be extensively detailed in the following sections devoted to evaluation process and continuity management.

About the location where support components execute, some CAMPO solutions privilege the exploitation of only client-side components, while other approaches include components distributed both at clients and in the network infrastructure. Client-side support components are generally simpler and focused on switching between available interfaces. Usually, both client- and infrastructure-side components interwork to collect monitoring information about connectors, especially in flexible CAMPO systems where monitored context includes data from the whole execution environment. In addition, when infrastructure-side components have also the role of interface/connector selectors, they can take into account not only client capabilities/requirements but also global network objectives, such as load balancing. Continuity management components are usually deployed on both the client side, e.g., pre-fetching buffers to sustain streaming continuity during handovers, and the infrastructure side, e.g., Session Initiation Protocol (SIP) servers for session transfer.

As for the other deployment scenario categories, also the role and location of support components in the execution environment may depend on deployment constraints and are interrelated with CAMPO design choices. For instance, by referring to the already examined notable examples of 4G and ABC, 4G systems usually aim to maintain mobile nodes simple, by delegating to infrastructure-side components as many tasks as possible, e.g., vertical handover triggering and traffic re-routing. ABC solutions, instead, tend to favor limited coupling between mobile nodes and the network infrastructure: clients are usually smarter and directly manage interfaces/connectors. More generally, on the one hand, deploying special-purpose support components on mobile nodes is often the simplest way to CAMPO realization from the perspective of network providers because it does not force to modify network infrastructures, by imposing all the burden of hardware/software adoption to interested end points. However, given the often limited capabilities of mobile nodes, the client-side approach does not enable flexible CAMPO solutions. On the other hand, the availability of a highly open and dynamic network infrastructure would permit to deploy special-purpose components on powerful infrastructure nodes, where and when needed, with obvious potential advantages in terms of performance. For instance, it would allow IEEE 802.11/UMTS integration by dynamically injecting the needed continuity support capabilities in the UMTS network equipment at the edges with the Wi-Fi LANs of current interest. Network infrastructures, especially today's telecommunication ones, often lack the needed degree of openness and dynamicity to achieve that goal, and there is the need of interacting with external support components, such as Mobile IP servers outside the UMTS infrastructure, to enable seamless network switching.

3.2.2 CAMPO Evaluation Process

The goal of CAMPO evaluation process is to **quantitatively measure**, in a homogeneous and comparable manner, **the current suitability of possibly heterogeneous interfaces and/or connectors** to be included in active channels. That comparison is necessary whenever a decision is important, either at channel set up time (when an application starts and requires the instantiation of a new channel) or to update active channels during service provisioning (change of interface and/or connector). Figure 3.6 gives a high-level simple representation of the CAMPO evaluation process as the pipeline of three successive steps: the collection of input values describing the current and applicable context, the evaluation of a metric that processes context input, and the delivery of the output result produced by the processing step. Input, processing, and output are the three sub-classes of the evaluation process category of our taxonomy and will be extensively described in the following sub-sections.

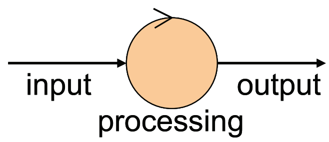

Figure 3.6 CAMPO evaluation as a three-step process.

More traditionally, it is possible to define the evaluation process as the set of operations needed to evaluate an adopted metric, represented by the cost function:

resultType C (inputType1 staticContext, inputType2 dynamicContext)

where the staticContext and dynamicContext input parameters are current values of entities of interest in the applicable context, respectively with less frequent or more frequent modifications, and resultType can largely vary in nature and represents the recommendation for channel instantiation/update, as better explained in the following. The distinction of input parameters depending on their expected change rate is crucial to enable effective CAMPO solutions, e.g., which continuously monitor only a limited set of context indicators while assuming fixed per-session values for static context parameters.

The definition of a suitable evaluation process for a CAMPO system is a complex task. A primary issue is to properly choose the context input parameters to consider. Input parameters should be easily measurable and comparable for the different interfaces/connectors available at runtime. In fact, to guarantee openness and easy extensibility, the evaluation process should not depend on any particular communication technology (and should be applicable also to newly introduced connectivity solutions). In addition, different evaluation processes may exhibit very differentiated complexity, changeability, and expressiveness, with a significant impact and correlation with the addressed deployment scenario. Roughly speaking, very expressive processing with complex goals, such as fairly distributing bandwidth to clients in a network domain, may require frequent interactions with the deployment environment to collect monitoring indicators (to be included in the context input) and call for frequent re-evaluations (output result updates). On the contrary, simple evaluation processes, such as minimizing client power consumption by always exploiting the least consuming interface, can be provided very easily with a pre-defined priority order for interfaces/connectors, with minimum overhead.

3.2.2.1 Input

Input data may have different degrees of dynamicity and be at different layers of abstraction (variations and abstraction level sub-classes of our taxonomy). About variations, input data can be classified as either static or dynamic. Static context information, such as average power consumption for an interface, maximum data throughput for a connector, user preferences and application requirements, either slowly changes or does not vary at all for a given channel during a service session. Dynamic context data, such as the bandwidth/delay/jitter currently provided by a connector, tends to vary frequently, also within the same service session, thus requiring proper context monitoring support.

About levels of abstraction, context input may include physical-, network-, or application-layer information. Physical context data usually relates to dynamic monitoring information such as RSSI or Signal to Noise Ratio (SNR). In addition, it may include static properties of a communication technology, e.g., average power consumption or maximum data throughput. Network-layer data comprise currently provided bandwidth, delay, and jitter, which typically are very dynamic context indicators. Application-layer data, instead, include user-related context (user requirements, priorities, history, …), terminal-related capability context (client hardware and software capabilities), and connectivity provider-related context (such as the enforced pricing model).

As already stated for other CAMPO taxonomy categories, the choice of input parameters for the evaluation process depends on the target deployment environment and may strongly influence several CAMPO design choices. Since most gathering/processing activities are performed at clients in usual CAMPO solutions, simple evaluation processes exploiting static input parameters better fit mobile nodes with limited capabilities. Clients with richer resource availability permit more dynamic and expressive input data, so to promptly react to even complex context variations, such as changes in client location and speed. Open/rich deployment scenarios with infrastructure-side components usually permit to monitor a larger set of state information about eligible connectors, e.g., the amount of currently served mobile nodes or the available bandwidth. In an evolutionary historic perspective, let us note that first CAMPO approaches simply consider low-layer context information, by basing their choices over traditional and locally measurable input values, such as network latency and RSSI, e.g., by choosing always the IEEE 802.11 AP with greatest RSSI. The trend in advanced CAMPO solutions is to include higher-layer input data to perform channel management decisions with full awareness of more expressive context information.

3.2.2.2 Processing

The choice of the processing method to adopt remarkably influences the expressivity and complexity of the evaluation process. Our claim is that the most important processing aspects are the level of flexibility (the capability to modify and adapt processing methods at runtime, either embedded or extensible) and the scope of the pursued objective (either local or global).

About flexibility, embedded processing methods permit to define how to combine input data only before starting CAMPO execution; any metric variation requires stopping and restarting the CAMPO system. Extensible processing methods, instead, are modifiable also during service provisioning. For this reason, most state-of-the-art proposals offer extensible processing solutions, either function-based or policy-based. Function-based metrics typically calculate output as a linear function of context input data and of weights, which may be adaptively configured also at service provisioning time. To further increase processing flexibility, some CAMPO solutions propose general frameworks to define metrics depending on high-level declarative obligation policies. Policies are activated by specified values of context input and may be specified/modified at runtime, with no impact on the implementation of CAMPO system and of service logic.

About the processing objective scope, in the case of local goal, the processing method has visibility and tries to fit only the local requirements expressed by the client node, such as minimizing client power consumption and maximizing network throughput. In the case of global objective, the processing method aims to achieve a network-wide objective, such as balancing network load among available connectors at all clients and maximizing the global throughput of a network, e.g., by preferring connectors currently involved in the minimum number of channels.

Also in this case, the flexibility and the objective scope of the processing method affect CAMPO design choices and are influenced by the addressed deployment scenario. Even if less intrusive from the overhead point of view, embedded solutions seem unsuitable for highly dynamic CAMPO scenarios; some recent research activities even claim the impossibility of adopting effective static definitions of general-purpose processing methods because it starts to be recognized that metric suitability also depends on user-/application-level requirements. Usually there is the need for a high level of processing expressivity when exploiting rich input data, in order to exploit at best the deep context visibility to achieve sophisticated interface/connector selections. Simpler processing methods are usually coupled with less complete context awareness and the willingness to achieve lightweight CAMPO solutions. In general, more complex and flexible the processing method, more context information needed (by generally requiring the execution of monitoring components with non-negligible overhead), and more expressive the resulting output, as detailed in the following.

3.2.2.3 Output

The evaluation process output may be of various nature. On the one hand, CAMPO solutions may differ on the type of entity provided as processing method result type (either interface or connector). On the other hand, and orthogonally, they can either directly provide their result as the best interface/connector/channel (value sub-class) or produce a set of values that continuity management can then exploit for channel update decisions (evaluation set sub-class). For instance, that set of output values may be the list of available interfaces in prioritized order, the list of available connectors in prioritized order, or a set of numerical values quantitatively evaluating any potentially available channel.

Let us rapidly note that, in any case, the processing output is only the input data for the following continuity management stage and may differ from the final handover decision taken by the CAMPO system. For instance, the continuity trigger component, described in the following section, can decide not to command the switching towards the first channel in the output prioritized list because that channel was recently used; CAMPO continuity management solutions may adopt hysteresis time intervals to prevent channel update bouncing.

The chosen sub-class of evaluation process output affects the design and implementation complexity of a CAMPO system. In the case of interface selection, the output is simply the result of the evaluation of all the interfaces available at the client node, by considering or not the currently switched off ones. In the case of connector selection, the set of possibilities to take into account is noticeably larger: CAMPO systems have to evaluate all connectors associated with any interface (in most CAMPO solutions also non-active ones). In addition, the ability to select connectors tends to increase the CAMPO complexity because it requires both managing the heterogeneity of interface features and deeply interacting with them, e.g., to extract context information about both IEEE 802.11 APs and GPRS/UMTS BSs. Moreover, the choice of an output result type may affect the level of abstraction of the context information to consider: in the case of interface selection, physical- and network-layer context is usually considered adequate; when the result type also includes connectors, there is usually the need for full visibility of more complex and articulated application-layer context, thus pushing towards more flexible CAMPO design choices.

3.2.3 CAMPO Continuity Management

After the initial phase of channel set up, there is the need to reconsider and possibly update working channels depending on context variations, including interface/connector availability. In CAMPO solutions, **continuity management components are in charge of** receiving the output of the evaluation process, of consequently deciding channel updates, and of **seamlessly performing interface/connector switches at service provisioning time**, for instance by ensuring content streaming continuity in despite mobile node movements between a Wi-Fi covered area and another location with only UMTS connectivity.

Continuity management components can be grouped into two main classes: triggers and switchers. On the one hand, triggers may reside on either the client- or the infrastructure-side and play the role of monitoring active channels, by commanding channel update operations to switchers when needed. Triggers are clients of the evaluation processing output: they either gather evaluation results via polling or wait for event notifications related to context variations. On the other hand, when commanded by triggers, switchers perform all network/service management operations to maintain service session continuity notwithstanding runtime channel modifications. Trigger classification and design choices are strongly interrelated with evaluation output sub-classes and the paper has already discussed that point in the previous section. For that reason, here we focus on switcher mechanisms, crucial for continuity management, and on their classification and deriving design choices.

Switcher solutions have been and still are one of the primary open points of investigation in CAMPO research activities. Switcher proposals in the literature may be classified along three primary directions: level of integration with the execution environment, granularity, and level of client visibility. **Integration represents the relationship among origin and destination connectors** when performing handovers. In particular, from the integration point of view, connectors may be either tightly or loosely coupled. **Tightly integrated connectors are deployed in the same administrative domain**, are supposed to know each other, and can communicate directly, e.g., via special-purpose protocols, thus reducing the complexity and the duties of switcher mechanisms. **Loosely integrated connectors**, instead, are deployed in **different administrative domains**, typically communicate via standard IP-based channels, and require intra-domain agreements in order to cooperate effectively.

Granularity defines the target of the continuity management process. When triggered, **per node switchers migrate every active channel** between connectors involved in the handovers, by adopting a coarse-grained approach and exploiting the most suitable interface/connector for the whole mobile node. On the contrary, **per channel switchers can migrate even only one channel per time**, in a finer-grained manner, thus enabling each channel to exploit its most suitable interface/connector.

Client visibility permits to identify the degree of **involvement of mobile clients in the continuity management process**. **Transparent** CAMPO continuity solutions usually perform their management actions **on the infrastructure side**, without any direct client involvement, thus enabling simple and lightweight mobile clients. On the contrary, **end-to-end approaches** tend to minimize infrastructure-side requirements, by delegating the **needed continuity management operations to mobile clients**.

3.2.3.1 Tight/Loose Integration

Network providers, especially cellular operators, have recently spent significant efforts to address CAMPO integration issues, in particular with the specific challenge of WLAN/cellular integration in mind. Possibly biased by the experience of telecommunications providers, these solutions aim to extend the operator network infrastructure, often in a close and proprietary way, to seamlessly include additional connectivity opportunities, such as Wi-Fi. On the contrary, there is a recent emerging trend in integration systems that aim to provide seamless interworking of different connectivity solutions without deploying any novel equipment on the operator-side part of the network infrastructure. The advantages are obvious in terms of dynamicity and easy deployment.

Guided by the two exemplifications above, we have decided to classify continuity management CAMPO solutions in two integration sub-classes: tight integration solutions, e.g., where WLAN APs should be deployed as novel equipment inside a proprietary cellular network, or loose integration solutions, e.g., where WLAN APs are deployed outside a cellular network, typically at the boundaries between the cellular network and the traditional Internet, with no impact on the already installed network equipment [ETSI 2001]. Only to mention a notable example, according to the proposed taxonomy, IEEE 802.11 APs are classified as tightly integrated within a GPRS/UMTS network when deployed as a part of the GPRS/UMTS infrastructure and seen as proprietary network equipment belonging to the telecommunication provider infrastructure. On the contrary, the same APs could be deployed in a loosely integrated way, when installed outside the operator infrastructure and possibly managed by a third party.

First CAMPO solutions were focused on tightly integrated scenarios. By adopting primarily the network operator point of view, the most relevant issues they addressed was to find the optimal location where to deploy and integrate connectors inside the already available telecommunication infrastructure (usually WLAN connectors in a cellular network) in order to provide continuity management capabilities.

Most recent CAMPO literature is changing the point of view. It mainly focuses on loosely-coupled integration of heterogeneous networks, by assuming few or no capabilities at all to intervene on already deployed proprietary networks. That change of focus has also influenced a terminology change: first CAMPO solutions classified handovers as intra/inter-domain depending on the possible change of administrative domains between origin and destination networks, usually managed in a proprietary way by their operators; currently, most CAMPO papers use the terms intra-horizontal, inter-horizontal, and vertical handover (see the definitions in Chapter 2) to move the accent on the characteristics of evaluation process and continuity management.

The set of needed switcher actions strongly depends on the integration relationship between origin and destination connectors. In tightly integrated scenarios, intra-domain handovers require a limited set of support mechanisms. In fact, since connectors belong to the same administrative domain, it is possible to assume that network-related client characteristics do not change, first of all client IP address. The only needed action is to update client location, i.e., to propagate the identity of the currently accessed connector in order to correctly re-route packets; that usually involves only lower layers of the OSI protocol stack. On the contrary, inter-domain handovers usually force to change several network characteristics and require the coordination of connectors deployed in different administrative domains, probably handled by different network operators. For instance, to support seamless connectivity for final users, origin and destination connectors have to agree on a common AAA mechanism to transparently migrate client credentials; in this case, packet re-routing usually interests also the higher layers of the OSI stack. In any case, in tightly integrated scenarios, all switcher operations on the client-side are delegated to embedded firmware, which transparently migrates active channels among interfaces. In addition, in loosely integrated scenarios, intra-horizontal handovers require the same mechanisms adopted for intra-domain handovers, while inter-horizontal and vertical handovers are similar to inter-domain ones. However, let us note that in the case of vertical handovers switcher mechanisms on the infrastructure-side do not change considerably while on the client-side there is the need to perform several additional actions, specifically designed to effectively and seamlessly migrate active channels without affecting the lower layer implementation of the exploited interfaces.

Also the decision of tight/loose integration relevantly influences the design choices available for effective CAMPO solutions. Tight integration solutions usually require low-layer interface integration and are characterized by little or no handover control capabilities propagated up to the application level. Both interface and connector selection are usually delegated to embedded solutions, e.g., interface firmware. In addition, tightly integrated environments are possible only if promoted by network infrastructure providers (for instance, by cellular operators) because they require the exploitation of special-purpose integration equipment directly deployed in the cellular network. However, tightly integrated solutions are usually characterized by better performance, e.g., smaller handover completion times, because origin and destination connectors can benefit from the homogeneity deriving from belonging to the same administrative domain. On the contrary, usually loosely integrated CAMPO solutions only require the deployment of special-purpose components on mobile nodes, e.g., to evaluate when and to which network to make a handover, and/or on auxiliary distributed support components added with no impact on the cellular network, e.g., intermediary proxies working as care-of-addresses for mobile nodes. In addition, loosely integrated solutions may take advantage of already available standard supports for continuity management, e.g., Mobile IP and SIP, thus facilitating and accelerating the deployment of CAMPO systems. Notwithstanding the potential limitations in terms of achievable performance, flexibility and easy deployment make loosely integrated CAMPO solutions definitely more suitable in the execution environments envisioned for the near future.

3.2.3.2 Granularity

From the granularity point of view, switching operations may be categorized in two main sub-classes: per node and per channel. Shortly, in per node CAMPO solutions the main continuity management goal is to seamlessly re-route the whole node traffic from the origin to the destination network. In the case of per channel switching, instead, the CAMPO continuity support is in charge of updating any single node channel, possibly independently from one another.

Per node switching requires reconsidering and possibly reconfiguring the whole set of resource bindings of a mobile node transparently, e.g., to autonomously discover a new printer in the destination network and to maintain updated references to remote resources such as network file systems. In addition, there is the need to disseminate information about node location, since a network change usually delves into location-related modifications such as IP address change.

Moreover, there is the need to manage traffic flows during handovers, e.g., by maintaining streaming continuity at the expense of additional computing/network overhead via bi-casting to both origin and new destination networks. Finally, the provisioning of common AAA/billing mechanisms could be indispensable for seamless handovers between networks owned/administered by different operators.

Per channel switching tends to require only a subset of the above continuity management operations. First of all, there is no need to consider possible updates for any active channel. In addition, since several interfaces may be simultaneously active, in the case the channels exploiting one of them become abruptly interrupted, CAMPO solutions can quickly update them with one of the other active and working interface. The main continuity management issues are how to correctly update channel references at end-points and how to perform per channel switching by minimizing packet loss while limiting computing/traffic overhead.

The adoption of either per node or per channel solutions is strongly interconnected with the chosen target deployment scenario, in particular with the adopted mobile node scope, and relevantly influences the CAMPO design choices. Single-on CAMPO systems exploit only one interface at a time and the adoption of per node continuity management is the only possible choice for them. Deployment scenarios based on more powerful client nodes often push for the adoption of multiple-on solutions, and that is often connected with the possibility of having per channel continuity management granularity. For instance, per node granularity is the usual solution in 4G single-on scenarios where the evaluation output generally triggers the interface change for all the channels active at a node. That is typically achieved by exploiting infrastructure-side support components, e.g., Mobile IP to achieve care-of-address and traffic re-route. Per channel granularity, instead, is more common in ABC scenarios where continuity management support performs channel re-addressing and must inform involved clients of any change in channel end-points, e.g., by exploiting SIP as the signaling protocol.

3.2.3.3 Client Visibility

By focusing on the perspective of client involvement in continuity management switching operations, it is possible to identify two extremes: either transparent or end-to-end switching. Transparent continuity solutions completely hide mobile nodes from the actions performed to update channels. For instance, infrastructure-side switchers may redirect traffic from the origin to the destination interface, without requiring any client awareness. On the contrary, end-to-end continuity management solutions request full client visibility: for instance, peers with active channels usually have to inform their clients in the case of interface change, possibly by communicating also how to reach them after the modification. Most continuity solutions lie between these two extremes: often infrastructure-side components operate transparent traffic re-routing, while mobile nodes are partially involved in triggering handover procedures.

To achieve the goal of decoupling client actions as much as possible from continuity operations, several recent CAMPO solutions, with both transparent and end-to-end visibility, are based on the adoption of intermediary support components, i.e., proxies, especially to handle the case of vertical handovers. For instance, continuity management proxies are used to predict vertical handovers in order to anticipate the associated management operations. The goal is to accelerate handover completion as much as possible to better support time-continuous service provisioning, such as interactive multimedia streaming. Some proxy-based continuity management proposals simply adapt and exploit support mechanisms non-strictly designed for CAMPO, e.g., by exploiting DNS or SIP to communicate new IP addresses to mobile nodes. Other solutions adopt Mobile IP for care-of-addresses and dynamic traffic re-routing.

The adoption of either a transparent or an end-to-end continuity management solution primarily depends on CAMPO objectives and considerably affects its design choices. For instance, the main goal of a network operator could be to provide seamless handovers between its cellular network and other connectivity solutions in order to provision services despite the actually exploited network interface. In this case, transparent continuity management is desirable and to that purpose network operators may deploy special-purpose continuity management components on the infrastructure side, thus minimizing the needed modifications at clients to maximize the immediately available market of potential users. On the contrary, if it is impossible to modify the core cellular network infrastructure or in the perspective of CAMPO service providers, the most suitable solution is end-to-end or proxy-based. Let us note that, in addition to the already sketched advantages in terms of flexibility and dynamicity, continuity management solutions close to the end-to-end extreme of our classification can relevantly reduce the required initial CAMPO investments: in fact, these approaches tend to share costs among interested users, who are requested to buy new client devices or to extend/update them in order to benefit from continuity management support.

3.3 Classifying the Campo Literature According to the Proposed Taxonomy

The CAMPO area is currently one of the most interesting and actively investigated fields in wireless research. Nowadays, as the continuously emerging novel CAMPO proposals demonstrate, the complex and articulated issues of seamless integration of heterogeneous wireless technologies are not yet fully addressed. Despite the still dynamic evolution of the area, some first survey papers have been published. However, as better clarified in the following, they are focused on either some specific challenges or categories of solutions, without the ambition of providing a comprehensive unifying CAMPO overview.

For instance, most proposed classifications only focus on infrastructure-based 4G systems, by missing to point out the similarities with peer-based connector solutions, and only provide a coarse-grained category classification. [Akyildiz et al. 1999] represents a relevant seminal contribution about heterogeneous network integration: it is the first paper to analyze handover management by identifying three phases, handover initiation, generation of new connections, and data flow control. [Pahlavan et al. 2000] proposes another simple handover taxonomy based on two dimensions: handover architecture (which component is in charge of handover decision and which is the supported degree of continuity?) and evaluation processing methods (which metric, by exploiting which handover indicators?). [Nasser et al. 2006], instead, proposes a handover classification based on network types, number of connections, number of administrative domains, and possibilities of user control, but not considers at all continuity management solutions. [McNair and Fang Zhu 2004] describes handovers as three-stage processes: handover decision, link transfer, and channel assignment. [Kappler et al. 2007] provides a cellular operator point of view, by delineating five primary steps to integrate heterogeneous networks: wireless medium sensing, discovery, establishment of security and internetworking relationships, composition negotiation, and composition realization. [Niyato and Hossain 2005] only partially relates to CAMPO systems and specifically presents a survey about call admission control in heterogeneous wireless environments.

In addition to the above papers about infrastructure-based 4G solutions, a very few other contributions, considering also ABC and with some survey content, exist in the literature. [Gustafsson and Jonsson 2003] discusses user experience and business relationships in ABC scenarios: its main contribution is the decomposition of solutions in functional blocks but it does not provide any CAMPO classification. [Ferreira et al. 2005] presents a high-level integration analysis by focusing on the simultaneous usage of services from different systems and operators. [Cavalcanti et al. 2005] is partially devoted to discuss open issues to integrate cellular networks, WLANs, and mobile ad hoc networks: the paper briefly and simply classifies the related literature according to different layers of abstraction.

Some other papers with survey content specifically concentrate on continuity management, without considering deployment scenarios and evaluation process issues, thus providing a very partial CAMPO perspective. [Reinbold and Bonaventure 2003] overviews main differences between micro- and macro-mobility, by only comparing primary micro-mobility protocols. [Saha et al. 2004] provides a slightly more articulated classification, by differentiating contributions into micro-, macro- and global-mobility. Other contributions adopt traditional layering classifications: [Akyildiz et al. 2004] distinguishes continuity management solutions in link-, network-, and cross-layer approaches, while [Banerjee et al. 2003] additionally takes into consideration transport and application layers. [Roberts et al. 2006] and [Lampropoulos et al. 2005], instead, focus on cellular networks: the former surveys cellular technologies by pointing out main issues for link- and network-layer integration; the latter specifically addresses WLAN-cellular handovers by pointing out how required management actions strongly depend on network integration levels.

Let us stress that, if compared with the above papers, **our survey originally provides a unifying architecture model** and a more comprehensive and fine-grained CAMPO system classification. In particular, **the evaluation process** category and **the consideration of both infrastructure- and peer-based connectors** represent relevant innovative aspects of our taxonomy. In the following, the chapter describes the most important CAMPO contributions in the literature, by presenting them in the grid of our taxonomy and by focusing on the specific aspects that make the examined proposals exemplar for their categories.

3.3.1 Deployment Scenarios

CAMPO-related papers usually do not explicitly point out the deployment scenario they specifically address. However, each contribution has some relevant aspects that implicitly position it in a specific deployment sub-category of our taxonomy.

3.3.1.1 Interface Scope

In the perspective of interface scope, available CAMPO solutions can be classified according to their capability to select among either interfaces [Stemm and Katz 1998; Minji Nam et al. 2004] or interface-connector pairs [Cheng Wei Lee et. al 2005]. The latter case also includes CAMPO systems that can consider peer connectors [Hung-Yu Wei and Gitlin 2004; Lera et al. 2005; Luo et al. 2003; Frattasi et al. 2006; Seung-Seok Kang and Matt W. Mutka 2005; Chunyan Fu et al. 2006].

At the cost of imposing precise assumptions on deployment requirements, **some CAMPO solutions give mobile nodes the only possibility to select interfaces**. For instance, the scenario proposed in [Stemm and Katz 1998] is based on the wireless overlay network assumption: the network technology with wider coverage provides connectivity everywhere and continuously, usually with limited bandwidth; other wireless technologies cover only limited and eventually spatially discontinued areas, but providing a larger bandwidth. [Stemm and Katz 1998] has the main goal of always exploiting the available network with larger bandwidth. As a consequence, vertical handover is performed only depending on the availability/unavailability of networks themselves, in an interface scope manner; it takes into account finer RSSI-based connector scope considerations only in horizontal handovers. Wise Interface SElection (WISE) adopts a slightly widened scope, by primarily concentrating on interfaces and secondarily on connector scope [Minji Nam et al. 2004]. In particular, WISE considers, with decreasing priority, mobile node requirements (with an interface scope, to minimize power consumption according to interface nominal characteristics) and network infrastructure requirements (with a connector scope, to redistribute network load when performance degrades).

Most common CAMPO solutions in the literature offer both interface and connector selection. For instance, [Cheng Wei Lee et. al 2005] monitors performance indicators of the currently used network and of other current connectivity (WLANs and cellular networks). That enables mobile nodes not only to choose the most proper interface but also the most proper interface-connector pair, at the expense of maintaining an almost updated view of context indicators describing the current state of the execution environment.

The exploitation of peer connectors requires additional deployment scenario capabilities and introduces further complexity in context evaluation. [Hung-Yu Wei and Gitlin 2004] proposes a two-hop-relay architecture, based on Relay Gateway (RG) nodes that can behave both as usual mobile nodes and as cellular gateways. They can seamlessly switch interfaces depending on network availability. In addition, they can improve WLAN coverage by exploiting cellular interfaces where WLAN connectivity is not available. Mobile nodes have to explicitly request for RG-based connectivity, in a non-transparent way. [Lera et al. 2005] proposes a similar approach based on two-hop paths towards the fixed Internet infrastructure. Other CAMPO solutions propose more flexible and complex multi-hop organizations. In [Luo et al. 2003] a peer connector, namely the Proxy Client, can interwork with both cellular and IEEE 802.11 ad hoc networks. Differently from previous examples, in [Luo et al. 2003] mobile nodes can interact with Proxy Clients not only directly but also via intermediate peer connectors, namely Relay Clients, in a multi-hop ad hoc manner. [Frattasi et al. 2006] provides several deployment scenarios for peer- and infrastructure-based connectors, by identifying associated service classes of primary relevance: for instance, cooperatives services, such as resource sharing, that exploit single-hop peer coverage; local retransmission services where peer connectors operate local multicast retransmissions; improved QoS support services where multimedia streams are split into several parts, each one processed at a distinct peer.

With still a greater degree of peer involvement, there are a very few CAMPO proposals aiming at **the coordination of a set of mobile nodes to create Mobile Ad hoc Network (MANET)** connectivity opportunities. Cooperating ad Hoc networking to sUpport Messaging (CHUM) dynamically elects one node to play the role of gateway between MANET and the fixed network infrastructure [Seung-Seok Kang and Matt W. Mutka 2005]. In particular, CHUM exploits WLAN connectivity on the MANET side and 3G on the infrastructure side. [Chunyan Fu et al. 2006] provides a similar example of MANET-3G integration, where SIP is exploited as the node-gateway signaling protocol.

In short, interface-only CAMPO solutions are certainly simple, e.g., because they exploit only rather static context information such as interface nominal capabilities. First CAMPO systems exploited this approach to accelerate their prototyping, to simplify their implementation, and to accelerate their adoption. However, the need of a deeper context consideration and of a full exploitation of interface capabilities has rapidly emerged. In fact, most current CAMPO proposals belong to the interface-connector deployment scenario category. In particular, most of them exploit only infrastructure-based connectors. However, increasing mobile node capabilities are opening new scenarios where mobile nodes are also exploited to offer connectivity opportunities, and thus the class of peer-based interface-connector solutions is gaining relevance. We claim the crucial importance of considering also peer connectors in envisioned wireless environments and, for this reason, here we have devoted relevant space to the few related proposals in the literature. Most state-of-the-art papers focus instead on more traditional infrastructure-based connectors, and will be extensively described also in the rest of the section.

3.3.1.2 Mobile Node Scope

By focusing on mobile node scope, the crucial decision point for CAMPO systems is the capability to exploit either only one interface at a time (single-on solutions such as [Buddhikot et al. 2004]) or several interfaces simultaneously (multiple-on solutions such as [Ylitalo et al. 2003; Chebrolu and Rao 2006; Kristiansson and Parnes 2006]).

[Buddhikot et al. 2004] is an exemplar case of **CAMPO single-on solutions**. It supports a Simple IP operating mode where each mobile node can activate a single interface at a time; there is no support at all for continuity management and ongoing sessions are lost whenever a vertical handover occurs. In addition, it offers a richer Mobile IP operating mode where, even if only one interface is actually exploited for communication purposes, multiple interfaces are kept active to prepare handovers in background. In that way, it is possible to proactively perform handover management, by anticipating new IP address requests to destination networks in order to accelerate handover completion.

Multiple-on solutions offer the additional capability of allowing each running application to **exploit the available interface that best fits its specific requirements**. For instance, [Ylitalo et al. 2003] uses multi-homing (see Section 3.3.3.3) to simultaneously enable multiple interfaces, even for the same application, and to update interface selection for each channel independently. In Bandwidth AGgregation (BAG) the focus is on widening the bandwidth available for an application by splitting its traffic over different channels with simultaneous exploitation of different interfaces [Chebrolu and Rao 2006]. In addition, at the same time BAG simplifies seamless node mobility and enhances channel reliability because applications can send multiple packet copies through different channels.

Finally, Simulcast represents an original hybrid solution between single- and multiple-on CAMPO systems [Kristiansson and Parnes 2006]. When different networks with similar suitability (according to a Simulcast embedded metric) are available, an application can exploit several interfaces simultaneously, even for the same traffic flow. On the contrary, Simulcast forces the exploitation of a single channel when it is evaluated as the most suitable one (largely better than the others according to the metric). Simulcast vertical handovers may be time-consuming because the system does not perform any handover prediction operation. Other multiple-on solutions are similar to the ones above and are listed here for the sake of completeness [Gazis et al. 2005; Fodor et al. 2003; Xing and Venkatasubramanian 2005; Adamopoulou et al. 2005; Jun-Zhao Sun et al. 2005].

In the perspective of identifying related trends in CAMPO system evolution, similarly to infrastructure-based connector proposals examined in the previous sub-section, single-on CAMPO solutions are certainly the most common ones in the literature. The primary reason is of limiting both power consumption and development complexity (e.g., multiple-on solutions require finer-grained and more complex channel management). We claim that multiple-on CAMPO systems will be the most adopted ones in future wireless environments. In fact, the increasing mobile node capabilities and the always growing bandwidth/reliability/continuity requirements of envisioned applications push towards the research and the industrial realization of multiple-on systems. For this reason, we have devoted wide space in this sub-section for the overview of the few multiple-on systems currently available.

3.3.1.3 Environment Scope

When considering the environment scope - the wider degree of visibility in the deployment scenario category - CAMPO proposals may differ in terms of adopted components, their deployment location, and their role. Here, we present most representative related contributions, by starting with solutions deployed only on the client-side ([Jun-Zhao Sun et al. 2005; Minghui Shi et al. 2004]), then describing distributed CAMPO components ([Vidales et al. 2005; Akyildiz et al. 2005]), and finally presenting proposals based on deployment in the cellular infrastructure ([Inoue et al. 2004; Karetsos et al. 2005]).

Connectivity Middleware Management provides an API to enable **adaptive connectivity in an end-to-end fashion** by working primarily on mobile clients [Jun-Zhao Sun et al. 2005]. In particular, it offers functions for event-based monitoring of available interfaces and for commanding channel switching. [Minghui Shi et al. 2004] belongs to the same category but specifically deals with authentication management, by proposing an end-to-end scheme to maintain communication privacy notwithstanding the change of exploited access network.

Other CAMPO solutions deploy their components on both mobile clients and the access network infrastructure. Policy-based system to ROam Transparently among Overlay Networks (PROTON) deploys highly demanding processing tasks related to policy specification/deployment in the infrastructure, while it operates policy enforcement on mobile nodes [Vidales et al. 2005]. Similarly to PROTON, Architecture for ubiquitous Mobile Communications (AMC) offers functions for evaluation process but originally adds support for continuity management [Akyildiz et al. 2005]. In particular, continuity management support functions are provided by infrastructure-side components for interoperability between heterogeneous networks of different operators, including authentication, accounting, and billing facilities. AMC performs evaluation in a distributed way: client-side components gather information about network conditions and determine which is the best channel depending on exclusively local considerations; infrastructure-side components have a global state view and may override local choices for optimal traffic distribution.

Differently from the above proposals, [Inoue et al. 2004] and [Karetsos et al. 2005] are based on **component deployment on the infrastructure side**. [Inoue et al. 2004] performs handover triggering, network discovery, and AAA operations by exploiting ad hoc components installed in the cellular infrastructure, transparently from the client point of view. Mobile nodes only have the burden to register themselves to request for seamless handover-transparent channels, to update their preferences, and to notify their location changes. In [Karetsos et al. 2005] infrastructure-side components are organized hierarchically and provide both evaluation process and continuity management. In the case of network degradation, e.g., traffic congestion, [Karetsos et al. 2005] tries to locally manage the situation by exploiting support components in the local network; if not enough, it involves global infrastructure-side components with multi-network visibility, e.g., to redistribute traffic among overlapping heterogeneous networks.

In short, with regards to role and location of CAMPO components, we do not observe any major evolution trend in state-of-the-art contributions. Client-side CAMPO solutions require greater resource availability on mobile nodes, in terms of both computing and energy power. More distributed CAMPO solutions require a certain degree of control on the network infrastructure, not always easy to achieve especially in cellular networks. However, the growth in mobile node capabilities coupled with the proliferation of heterogeneous wireless networks is pushing for privileging client-side CAMPO systems, mainly to facilitate CAMPO development and deployment over open wireless environments. In particular, recent solutions are encouraging client-side evaluation process to achieve scalability, while continuity management is often performed in a distributed way because, to some extent, some level of infrastructure-side involvement is useful to provide effective seamless connectivity, e.g., to bi-cast packets to both origin and destination connectors during handovers.

3.3.2 Evaluation Process

The evaluation process is one of the most important, characterizing, and widely investigated aspects of CAMPO systems. Within the evaluation process category, most CAMPO proposals focus on processing methods, in particular by identifying solutions with growing degrees of flexibility; a smaller set of solutions concentrate on local/global objective distinction. On the contrary, a very few papers in the literature explicitly point out their characteristics in terms of input and output sub-categories.

3.3.2.1 Input

As already stated, most CAMPO solutions do not clearly describe which are the characteristics of their evaluation process input, by rapidly and implicitly assuming it depending on the exploited processing method. However, we claim the relevance of the input sub-category to understand the complexity of input gathering and the correlated monitoring costs. For this reason, we have thoroughly examined the CAMPO literature and chosen to present here some exemplar contributions exploiting simple input information ([Stemm and Katz 1998]), physical-level input about the status of wireless communications ([Minji Nam et al. 2004; Mohanty and Akyildiz 2006]), and more expressive, complex, and innovative input at the application level ([Cheng Wei Lee et. al 2005; Qian Zhang et al. 2003]). Finally, the section describes two CAMPO contributions explicitly working on input characteristics ([Stavroulaki et al. 2006; Balasubramaniam and Indulska 2004]).

Simplest CAMPO solutions, based on the overlay network assumption, exploit a **static priority order among available interfaces**; the only dynamic input data to consider is network availability, often based on beacon frames [Stemm and Katz 1998]. Most evaluation process proposals, instead, exploit more dynamic input information [Minji Nam et al. 2004; Mohanty and Akyildiz 2006; Cheng Wei Lee et. al 2005; Qian Zhang et al. 2003]. Some contributions focus on a small set of physical-level network parameters. [Minji Nam et al. 2004] primarily considers power consumption and, secondarily, network conditions. In particular, the evaluation process performed at mobile nodes exploits visibility of static physical information, such as interface power consumption in transmit/receive/idle state, and of variable communication states (mainly network residual bandwidth). Similarly, Cross-layer Handoff Management Protocol (CHMP) is primarily based on physical context input, i.e., current RSSI of APs in visibility and interface-embedded RSSI thresholds for handover triggering [Mohanty and Akyildiz 2006]. Differently from [Minji Nam et al. 2004], CHMP presents a more articulated handover mechanism taking into account different signaling delays for intra- and inter-domain handovers. Moreover, threshold values can vary dynamically, depending on mobile node speed and on handover failure probability requirements.

Other CAMPO solutions consider **more expressive and complex input data** [Cheng Wei Lee et. al 2005; Qian Zhang et al. 2003]. [Cheng Wei Lee et. al 2005] dynamically evaluates performance indicators for both WLAN APs and cellular network BSs. In the case of WLAN, it monitors RSSI variations (physical layer) and residual bandwidth (network layer), which are derived from direct measurements of throughput, channel utilization, and frame loss rate (the last two indicators available in QoS BSS beacon frames [IEEE 802.11e 2005]). In the case of cellular networks, it exploits statically defined nominal values. [Qian Zhang et al. 2003], instead, applies the Fast Fourier Transform to RSSI values of APs in proximity to quickly and accurately detect signal decay. In addition, it exploits network-level information: the Network Allocation Vector provided by IEEE 802.11 APs is used to infer bandwidth and access delay. In that way, given a set of eligible APs with RSSI over a threshold, it can select the one currently less loaded.

Finally, **only a very few proposals explicitly describe their context input**. [Stavroulaki et al. 2006] considers as input data user profiles (subscribed services, corresponding QoS requirements, and maximum price allowed), terminal profiles (client device hardware/software capabilities), network offers (currently available services, supported QoS levels, and corresponding costs), and configuration costs (time and, more generally, resources required to perform channel reconfiguration). Let us note that, while user and terminal profiles are rather static, network offers and configuration costs may be very dynamic indicators. [Balasubramaniam and Indulska 2004], instead, presents a precise description of context input data, classified into static and dynamic. Static information relates to the mobile node as a whole and covers different abstraction layers, from device capabilities to personal settings such as user-defined interface priorities. Dynamic information can be either associated with the whole mobile node, e.g., user location, or differentiated for each channel/interface, e.g., currently available bandwidth.

By summarizing, only very first CAMPO solutions proposed the adoption of exclusively static input parameters, based on the overlay network assumption. In fact, the relevance of exploiting dynamic input to perform connector evaluation has rapidly emerged and is now widely recognized. Several CAMPO solutions still adopt only physical input data, e.g., RSSI and SNR, mainly inspired by the traditional evaluation processes for horizontal handovers. To provide effective evaluation processes with a proper tradeoff between complexity of input gathering and expressiveness, some systems have claimed the inclusion of traditional QoS indicators in context input. Most recent solutions additionally exploits application-level input information, which better describes the characteristics of terminal, user, and environment context. However, at the current stage it is not yet fully addressed the issue of efficiently retrieve from context and exploit in the evaluation process information at a high level of abstraction.

3.3.2.2 Processing

The processing method is certainly the most characterizing aspect of the evaluation process. Here we present CAMPO contributions taking into primary consideration the flexibility of their processing methods: first embedded methods ([Stemm and Katz 1998; Hongyang Bing et al. 2003; Hou and O'Brien 2006; Wei Song et al. 2005]), then dynamically configurable approaches, either function-based ([Adamopoulou et al. 2005; Hasswa et al. 2005]) or not ([Xing and Venkatasubramanian 2005; Gazis et al. 2005; Qingyang Song and Jamalipour 2005; Ahmed et al. 2006; Luan Huang et al. 2006]), finally more flexible policy-based solutions ([Ylitalo et al. 2003; Vidales et al. 2005; Wei Zhuang et al. 2003; Jun-Zhao Sun et al. 2004]). Let us note that the section is organized in terms of flexibility degree and considerations about the objective scope of processing methods, either local or global, are presented along the overall description.

Embedded processing methods are usually based on a fixed priority order among available interfaces, as in CAMPO systems based on the overlay network assumption, thus providing limited flexibility. For instance, [Stemm and Katz 1998] triggers an upward vertical handover, from the current to an alternative wireless technology with smaller bandwidth and wider coverage, whenever the current interface becomes unavailable; similarly, downward vertical handovers are triggered whenever new interfaces with better performance and more limited coverage become available. Instead, the RSSI comparison is exploited to trigger horizontal handovers. [Hongyang Bing et al. 2003] adopts a slightly more complex processing method by taking into account also some simple indicators to estimate QoS. In particular, it considers RSSIs of APs and BSs in visibility and their distance from mobile nodes. A vertical handover from UMTS to WLAN occurs when the target AP is close to the client and its RSSI overcomes a threshold; WLAN-to-UMTS handovers are triggered whenever the AP RSSI is below a threshold, a UMTS BS is close to the client, and its RSSI overcomes a threshold. UMTS RSSI is not considered at all because the proposal always prefers WLAN connectivity if available. [Hou and O'Brien 2006], instead, provides an embedded but slightly configurable solution. It enables different vertical handover strategies between two extremes: the former triggers a handover as soon as the corresponding channel is interrupted, the latter waits for a fixed time interval after channel interruption before starting handover execution. [Hou and O'Brien 2006] decides which handover strategy to perform and with which delay by adopting fuzzy logics to model input uncertainty, e.g., probability of link interruption, probability of handover failure, size of unsent messages.

Let us note that the goal of [Stemm and Katz 1998; Hongyang Bing et al. 2003; Hou and O'Brien 2006] is local: they aim to select the most suitable channel from the mobile client point of view, by considering only context input evaluated at mobile nodes. Always within the embedded processing sub-category, other CAMPO solutions target a global objective, e.g., monitoring the performance of each considered network and optimally distributing load. For instance, [Wei Song et al. 2005] provides call admission control capabilities by taking into account network residual bandwidth and by differentiating handover management actions depending on traffic type, either voice or data: voice calls are preferably allocated to cellular networks, more suitable in terms of delay and coverage; data traffic is directed to WLANs, preferred for their larger bandwidth.

To deal with dynamically variable requirements and network conditions, **most state-of-the-art CAMPO proposals exploit general-purpose function-based processing methods**. The processing function in Terminal Management System (TMS) considers quality and cost of eligible connectors and a user-specified priority order among operators and interfaces [Adamopoulou et al. 2005]. TMS evaluates the processing function for any available connector: each function term is weighted according to user-specified priorities, which can change at service provisioning time. Also Vertical Handoff Decision Function (VHDF) provides users with the capability to specify a priority order among different network characteristics, by defining a proper weight set [Hasswa et al. 2005; Nasser et al. 2006]. The VHDF processing function exploits the weight set to evaluate a linear combination of network conditions, network performance, service cost, power requirements, security, proactive handoff, and client speed for each active interface.

Other CAMPO proposals exploit **more complex processing methods**, based on the **knapsack** algorithm ([Xing and Venkatasubramanian 2005; Gazis et al. 2005]) **and on the Analytic Hierarchy Process** (AHP) [Qingyang Song and Jamalipour 2005; Ahmed et al. 2006]. Knapsack-based solutions try to address the channel-to-interface or channel-to-connector assignment issue in a per-channel fashion. For instance, [Xing and Venkatasubramanian 2005] exploits the knapsack algorithm to minimize average power consumption and user dissatisfaction in terms of distance from traffic requirements. Each traffic flow is modeled as the set of its associated bandwidth/delay requirements and a partitionability flag; any available network is represented by its maximum bandwidth, maximum delay, and power consumption indicators. [Qingyang Song and Jamalipour 2005] exploits AHP to decide weights for the considered criteria and Grey Relational Analysis (GRA) to rank channel alternatives. AHP splits a complex problem, in this case the provision of the best QoS as a local objective, into a number of decision factors: availability (decomposed in RSSI and coverage area), throughput, timeliness (delay, response time, and jitter), reliability (BER, burst error, and average retransmissions per packet), security, and cost. On the one hand, GRA normalizes and compares UMTS and WLAN QoS parameters; on the other hand, it exploits AHP to determine Grey Relational Coefficients and thus to chose the most suitable interface. Also [Kibria and Jamalipour 2007] exploits AHP and GRA, while [Bari and Leung 2007] adopts the TOPSIS algorithm to perform connector ranking by considering the distance between evaluated metric parameters and their desired values. The above function-based processing methods pursue a local objective, often dependent on user requirements represented by weight sets. On the contrary, [Luan Huang et al. 2006] aims at a global objective: while the problem statement is similar to the knapsack one, [Luan Huang et al. 2006] tries to aggregately maximize the set of utility functions representing the level of satisfaction of every user currently connected to the system.

To further improve flexibility and extensibility, **some CAMPO systems propose the adoption of policy-based processing methods**. [Ylitalo et al. 2003] continuously monitors context input and checks whether there are conditional clauses that apply in the current set of enforced policies; the verification of a condition clause triggers a management action to perform, and each action associates with an ordered list of network interfaces. PROTON is a more complex and articulated example of policy-based CAMPO solution [Vidales et al. 2005]. PROTON exploits policies as event-condition-action rules, i.e., declarative rules that specify actions to execute whether conditions apply, with events that trigger condition evaluation. Context input data are not considered aggregately, as in many function-based metrics: PROTON breaks down context into fragments and allows the specification of independent normalization functions for any fragment. In particular, PROTON permits to define tautness functions that evaluate how tautly a condition fits to an event: the closer is the returned value to 0, the tauter a condition is to a specific event.

Other relevant policy-based solutions are [Wei Zhuang et al. 2003], [Wei Song et al. 2007] and [Jun-Zhao Sun et al. 2004]. [Wei Zhuang et al. 2003] proposes differentiated policy-based management depending on the integration degree between origin and destination networks. [Wei Song et al. 2007] performs policy decision and enforcement on both the client- and the infrastructure-side; the primary overall objective is to balance networking load among overlapping cellular and WLAN networks. [Jun-Zhao Sun et al. 2004] adopts a context-aware policy solution that permits the definition of policies with different scopes (interface, channel, or application). Let us finally note that, independently of the fact they involve or not infrastructure-side components, [Ylitalo et al. 2003], [Vidales et al. 2005], and [Jun-Zhao Sun et al. 2004] pursue a local objective, while [Wei Zhuang et al. 2003] and [Wei Song et al. 2007] target a global objective.

By summarizing, the processing method objective scope, either local or global, simply represents a design choice that depend only on developer purposes. On the contrary, the evolution trend in CAMPO processing flexibility is worthwhile of some additional considerations. Initial CAMPO research efforts adopted embedded processing solutions: they share the common non-negligible limitation of not allowing to change processing method requirements at runtime. That imposes strict static constraints that usually prevent taking optimal choices: for instance, the overlay network assumption usually does not apply, thus making impossible the adoption of a static priority order [Vidales et al. 2005]. Embedded processing solutions are still of some limited interest because of their efficiency and scarce overhead, especially for mobile nodes with limited capabilities; however, their decreasing relevance is a clear trend.

To provide greater flexibility, most recent CAMPO systems are proposing function-based solutions, which mainly differentiate in relation to the exploited mechanisms (from simple sums of addends to knapsack and AHP/GRA algorithms). Their objective is to achieve the optimal tradeoff between flexibility and computational complexity. For instance, in most cases function-based processing permits to change parameter weights at runtime but statically imposes the number and type of parameters. Recently proposed policy-based solutions offer even greater flexibility, but are still in their infancy by often providing valuable policy frameworks but omitting to propose actually effective metrics in terms of imposed overhead. Our opinion is that the additional complexity of policy frameworks and the potentially limited efficiency imposed by policy management and evaluation at runtime are not sufficiently justified by current practical deployment scenarios. Therefore, we expect that function-based processing methods will continue to be the most adopted ones also in next generation CAMPO systems.

3.3.2.3 Output

Even if the current CAMPO literature usually skips over the explicit description of the main evaluation output characteristics, it is possible to identify two major groups of contributions: a category of CAMPO solutions provide **quantitative values** that measure the suitability of eligible channels; another category of proposals **directly returns the most suitable channel** to either activate or switch to.

Function-based processing methods ([Adamopoulou et al. 2005, Hasswa et al. 2005]) and AHP-based solutions ([Qingyang Song and Jamalipour 2005; Ahmed et al. 2006]) typically belong to the first group. Sometimes the set of output values is available to the application level, e.g., to enable direct decisions by the running applications; in other cases, at default, the output result is only the identification of the best channel, but applications can optionally request the output values describing all the currently available connectivity opportunities. Embedded and policy-based processing methods usually belong to the second group: applications on their top typically do not have any visibility of output values to compare eligible channels.

Let us note that most current CAMPO solutions are in this second group and completely hide the application layer from low-level evaluation process details. While a transparent approach is commonly desired to minimize application development complexity, at the same time it may significantly reduce application capabilities, thus making difficult to develop context-aware services. In particular, we claim the suitability of middleware-level CAMPO solutions that manage low-level CAMPO implementation details but with some advanced and hybrid forms of visibility propagation up to the application layer, as processing methods that offer output values do.

3.3.3 Continuity Management

Several CAMPO contributions address the increasingly relevant and challenging issue of continuity management in order to fully support seamless node mobility in heterogeneous networks even while accessing continuous services. State-of-the-art solutions greatly differ in terms of integration level, granularity, and visibility. Research activities coming from the telecommunication area often adopt a tightly integrated perspective, while loosely-coupled CAMPO solutions are recently becoming more and more popular. Granularity aspects include both per-node continuity management mechanisms, mostly adopted in 4G systems, and per-channel ones, more appropriate for ABC solutions. The visibility category, instead, mainly affects the architecture of continuity management CAMPO solutions, i.e., end-to-end, proxy-based, or mainly transparent.

3.3.3.1 Integration

The industrial efforts accomplished in the last years to provide a standardized architecture for heterogeneous connectivity integration have primarily adopted a cellular operator point of view. The main issue was to **tightly integrate** heterogeneous wireless networks by effectively **including WLANs inside cellular networks**. In particular, [3GPP 2002] identify six differentiated environments with a rising level of requirements: i) common billing and customer care, ii) common access control and charging, iii) WLAN-accessible cellular services, e.g., a Wireless Application Protocol (WAP) service available via WLAN connectivity, iv) service continuity, e.g., the possibility to access the same WAP service after cellular-WLAN handovers, v) seamless service, i.e., service continuity achieved in a completely user-transparent way, and vi) access to traditional circuit-switched cellular services via WLAN.

By concentrating on **loosely-coupled integration proposals, they share the primary goal of minimizing required modifications** to GPRS/UMTS and WLAN deployment environments. For instance, to this purpose [Jyh-Cheng Chen and Hong-Wei Lin 2005] exploits special-purpose gateways deployed at the boundaries between GPRS and WLAN networks. Similarly, [Buddhikot et al. 2004] proposes the deployment of gateway integrators over WLANs and connected to the integrated cellular network via Internet, while [Bernaschi et al. 2005] exploits externally deployed Mobile IP support.

Several recent CAMPO contributions tend to position themselves somewhere in the middle between the two extremes of tightly- and loosely-coupled integration. They usually implement hybrid solutions where there is the possibility to choose the level of integration at system deployment time [Salkintzis et al. 2004; Lera et al. 2005]. For instance, the [Salkintzis et al. 2004] tightly-integrated operating mode enables seamless service continuity independently of WLAN/GPRS roaming, by also enabling the reuse of GPRS AAA. The architecture is mainly based on two components, one deployed at mobile nodes (WLAN Adaptation Function, WAF) to transport GPRS signaling/data over IEEE 802.11 WLANs; the other on the infrastructure side (GPRS Interworking Function, GIF) to offer a standardized interface to the GPRS core network, thus hiding WLAN peculiarities. By exploiting that tightly-integrated architecture it is possible to implement the first five scenarios identified in [3GPP 2002]. In addition, [Salkintzis et al. 2004] can work in a loosely-integrated mode by exploiting the possible availability of an IP network internal to the cellular operator network: in that case, WLAN data traffic is carried by the internal IP network and does not pass through the GPRS core infrastructure as it would in tightly-integrated CAMPO solutions. [Salkintzis et al. 2004] exploits Mobile IP for service continuity; it faces AAA and billing issues with an operator perspective, by introducing authenticator components for WLAN users that permit to reuse the same AAA/billing mechanisms exploited in the cellular network. [Lera et al. 2005] is another valuable example of hybrid approach where both tight and loose integration are supported. The integration is based on a special-purpose UMTS/IEEE 802.11 gateway accessing the Internet via UMTS and providing mobile clients with IEEE 802.11 connectivity.

However, differently from [Salkintzis et al. 2004], that gateway may also be a peer connector. In the case of tight integration, [Lera et al. 2005] deploys the gateway inside the UMTS network, as if it were an UMTS device; in the loosely-coupled integration case, the gateway acts as an IP router in its WLAN and does not require any intervention on the UMTS network.

To briefly sum up the above aspects, first research work on CAMPO solutions, especially coming from the industrial telecommunication field, has primarily followed a cellular operator perspective, by consequently adopting a tight level of integration. In particular, tightly-coupled CAMPO solutions were mainly motivated to minimize modifications needed at the client side, thus allowing operators to be the primary actor pushing for innovation. The current evolution trend, instead, is assigning growing and growing relevance to loosely-coupled integration, since it can enable the integration of several heterogeneous networks in an easier and more open way, by favoring heterogeneity also in terms of involved network operators. Let us note that most state-of-the-art CAMPO systems in the previous sub-sections tend to position themselves closer to the loosely-integrated extreme: they typically assume limited or no capability to intervene on infrastructure-based connectors, which are managed as not modifiable legacy components.

3.3.3.2 Granularity

The granularity continuity management category discriminates per-channel and per-node CAMPO solutions, respectively better fitting ABC and 4G systems.

Most spread **per-channel ABC systems propose re-addressing mechanisms for continuity management**, especially for application-specific provisioning environments, such as multimedia streaming. The Multipath Smooth Handoff scheme activates multiple channels along multiple paths from the stream sender to its receivers [Yi Pan et al 2004]. The primary idea is to exploit multiple paths simultaneously and, once evaluated path performance in an end-to-end fashion, to select the most suitable paths depending on service requirements. The main support components related to continuity management are i) a path management module running at both sender and receiver, which exploits Mobile IP simultaneous binding [Perkins 2002] and route optimization [Johnson and Perkins 2001], and ii) rate control modules at each path endpoints that perform the on-line bandwidth monitoring for each channel. [Luo et al. 2003], instead, is a per-channel

ABC solution specifically focused on re-routing capabilities. It supports both infrastructure-based connectivity (assumed as always available but possibly with a low data rate) and ad hoc multi-hop networking (to explore the possibility of higher data rates, when needed, via peer connectors to the cellular infrastructure). Whenever an ad hoc path is broken, e.g., due to an intermediate node failure or movement, the interested client passes to infrastructure-based connectivity while starting the simultaneous discovery of other possible paths with greater QoS. Finally, an interesting per-channel ABC proposal is presented in [Ghini et al. 2005], where it is possible to exploit re-routing among not only different interfaces of the same mobile node, but also different mobile nodes belonging to the same group, e.g., from a specific user's PDA to her car radio.

Notwithstanding the relevance of the above ABC solutions, **most CAMPO systems still provide simpler and less flexible per-node granularity**. That is the case of 4G CAMPO contributions, despite their proposed architecture is either loosely or tightly integrated. Several 4G continuity management mechanisms deal with packet forwarding, AAA, and billing [Marques et al. 2005; Ahmavaara et al. 2003; Hui Luo et al. 2003; Koien and Haslestad 2003]. As a general property, however, per-node CAMPO solutions exhibit a greater maturity if compared with the analogous continuity management support mechanisms available in per-channel systems.

In this sub-section we have devoted more space to the overview of continuity management aspects related to per-channel CAMPO systems, while continuity management support in 4G solutions will be more extensively described in the following sub-section. In fact, the family of 4G CAMPO systems is relevant not only because it is a notable example of per-node granularity but also for the associated degree of continuity management client visibility. Let us finally note that, at the moment, CAMPO solutions with per-channel granularity are only sporadically adopted. However, as previously motivated in different points of the paper, we claim the suitability and the growing diffusion of ABC systems in the heterogeneous and open wired-wireless integrated networks of the near future, thus increasing the importance and calling for novel solutions to efficiently deal with per-channel continuity management.

3.3.3.3 Client Visibility

Client visibility may greatly vary from end-to-end continuity management solutions where mobile nodes directly perform the needed management operations with limited or no external help ([Chuanxiong Guo et al. 2004; Snoeren and Balakrishnan 2000; Li Ma et al. 2004]), to proxy-based solutions at different levels of transparency for mobile nodes ([Bellavista et al. 2005a; Bellavista et al. 2005b; Politis et al. 2004; Qi Wang and Ali Abu-Rgheff 2006]), or even to fully transparent systems where infrastructure-side components perform all continuity management actions ([Shenoy and Montalvo 2005; Akyildiz et al. 2005]).

In the **end-to-end** category, [Chuanxiong Guo et al. 2004] concentrates on how to split end-to-end handovers into two distinct phases: localization of mobile clients and channel continuity maintenance. DNS and peer-to-peer information distribution are suitable for the former phase, while a Subscription/Notification service is adopted for the latter one. Even [Snoeren and Balakrishnan 2000] exploits DNS for mobile node tracking, but proposes modifications to the standard TCP protocol to allow IP address change while ensuring channel continuity. [Li Ma et al. 2004], instead, is primarily based on the adoption of standard protocols, i.e., Stream Control Transmission Protocol (SCTP) [Stewart et al. 2004] and Mobile SCTP (mSCTP) [Riegel and Tuexen 2006]. The primary SCTP feature for continuity management is multi-homing, which enables an SCTP session to be established over multiple interfaces identified by multiple IP addresses; mSCTP provides the additional capability to add, delete, or change IP addresses during active SCTP associations. In particular, [Li Ma et al. 2004] supports two possible end-to-end handover procedures depending on the fact that the addressed deployment environment enables either single-homing or dual-homing. In the first case, [Li Ma et al. 2004] does not provide seamless mobility and the channel is interrupted during handover. On the contrary, in the second case, it provides seamless mobility at the cost of traffic replication. In addition, it permits to deploy proxies with mSCTP capabilities that can perform as gateways for legacy fixed servers without mSCTP capabilities, thus providing a continuity management solution transparent also from the server point of view.

As already mentioned, some CAMPO solutions are neither completely end-to-end nor transparent because they exploit both **intermediate proxies deployed on**

the infrastructure and special-purpose components on mobile nodes. For instance, [Bellavista et al. 2005a] and [Bellavista et al. 2005b] are based on shadow proxies and client stubs. Mobile agent-based shadow proxies run on the fixed network and dynamically migrate close to the APs currently providing connectivity to their associated mobile clients; client stubs run at mobile nodes and transparently interface with possibly legacy client applications. Both components adaptively resize their buffers depending on handover prediction, with the goal of supporting seamless mobility while minimizing memory and computing overhead.

Other CAMPO solutions exploit well-standardized proxy-based solutions, e.g., Mobile IP and SIP, adapted for seamless continuity management. The Enhanced Mobility Gateway (EMG) uses both Mobile IP and SIP, respectively for non-real-time and real-time services [Politis et al. 2004]. On the one hand, Mobile IP is not appropriate for applications with strict real-time constraints, such as Voice over IP, because of the delay imposed by triangular routing; routing optimization enhancements could solve the problem, but requiring modifications of the standard IP stack over mobile clients. On the other hand, SIP natively supports only UDP because it has been designed mainly by considering streaming applications and is not suitable for highly reliable traffic. EMG components are deployed at the edges between wireless networks and the fixed Internet, by acting as both Mobile IP Foreign Agents and SIP proxies. Also [Qi Wang and Ali Abu-Rgheff 2006] exploits Mobile IP and SIP, by proposing two alternative solutions: i) it decomposes standard Mobile IP and SIP facilities to effectively merge them together in an original integrated support that eliminates redundancies and maximizes efficiency; and ii) it provides a continuity management support fully compliant with standard Mobile IP and SIP, but imposing a greater overhead. In addition, [Cheng Wei Lee et. al 2005] exploits Mobile IPv6, while [Nursimloo and Chan 2005] exploits Fast Mobile IPv6 and SIP to support real-time mobility. [Wi Wu et al. 2005] analyzes the delay due to the usage of SIP for vertical handover among WLAN and UMTS networks. [Sarikaya 2006] delineates five possible architectures to integrate WLANs and cellular networks that differ for the location where Mobile IP home agents are placed.

In the transparent continuity management class, the common goal of [Shenoy and Montalvo 2005] and [Akyildiz et al. 2005] is to provide seamless mobility, the former focusing on traffic re-routing, the latter on interoperability. [Shenoy and Montalvo 2005] supports seamless vertical handover in a transparent manner, by offering signaling capabilities among origin and destination networks. When a cellular-to-WLAN handover is required, infrastructure-side components proactively perform both mobile node authentication and channel allocation; mobile clients have to wait for handover notification messages, to notify the starting of actual handovers, and to perform local channel update. [Akyildiz et al. 2005] specifically aims to support transparent heterogeneity management, by using IP as the gluing protocol: its main components are a Network Interoperating Agent (NIA) running in the fixed Internet and several Interworking Gateways (IGs) residing on the integrated and heterogeneous wireless networks. The centralized NIA provides interoperability capabilities between IGs, thus eliminating the need of direct service level agreements between each pair of involved networks.

By trying to identify a visibility evolution trend, in order to promote CAMPO diffusion, initial solutions have proposed the adoption of intermediary proxies based on standard protocols, e.g., Mobile IP and SIP. However, their limited performance has motivated novel proxy-based solutions specifically designed to support continuity management. About transparent continuity management, this class of solutions usually take advantage only of infrastructure-side components, thus providing also legacy mobile nodes with seamless handovers. Since there is no CAMPO system providing complete transparency, we have included in this category all the contributions that minimize the deployment of newly added components on mobile clients (these CAMPO solutions often rely on a tightly-integrated architecture). End-to-end continuity management has the primary advantage of not requiring any additional CAMPO component running on the infrastructure side, thus being immediately deployable and distributing to interested users the burden/cost of installing and executing required components on mobile nodes.

Currently, neither proxy-based nor transparent nor end-to-end solutions have clearly emerged as the most promising ones. We envision that cellular operators will continue pushing mainly transparent solutions for continuity management, often by exploiting special-purpose supports. However, the spreading of wireless networks based on unlicensed technologies, e.g., IEEE 802.11 WLANs and Bluetooth PANs, should promote the adoption of solutions based on proxies that are not directly managed by communication operators. Finally, at the moment, end-to-end proposals seem the most promising ones for QoS management purposes because of their direct involvement of endpoint nodes.

3.3.4 Overall Considerations and Emerging Trends in CAMPO Literature

In the above overview of the CAMPO literature we have tried to point out the most important aspects of each contribution, by using those primary aspects to position CAMPO systems in our articulated taxonomy. Table 3.1 has a twofold goal: on the one hand, it concisely summarizes what already presented for the most relevant CAMPO systems; on the other hand, it additionally indicates how each of these systems relates about the other categories of our classification, even if by providing partial contributions of minor relevance. Generally speaking, the table shows that **many CAMPO systems provide only partial solutions** concentrated on a specific subset of CAMPO issues. That is the case, for example, of most papers focused on tight/loose integration [Lera et al. 2005; Buddhikot et al. 2004; Jyh-Cheng Chen and Hong-Wei Lin 2005; Salkintzis et al. 2004], which do not provide any support mechanism for the evaluation process.

By delving into finer details, it is possible to observe some other interesting trend. About deployment scenarios, **the connector scope is currently the most adopted one**, especially based on infrastructure components; peer connectivity is not yet commonly accepted. Moreover, **single-on solutions are the most spread**, while there are still a few multiple-on proposals, some of them for special-purpose mobile clients. By focusing on the evaluation process, the dynamicity of context input is growing in recent proposals and the considered input increasingly includes different abstraction levels. The objective scope is primarily local despite the degree of flexibility, and the selected entity is usually a connector. Finally, about continuity management, it is possible to observe that most CAMPO contributions that do not adopt a cellular operator perspective exploit a loosely-integrated approach. **The most common granularity is per-node**, while only few contributions provide per-channel continuity management. The most usual approach is proxy-based, independently of the adopted visibility sub-category.

Table 3.1 The CAMPO literature classified according to the proposed taxonomy (secondary aspects in brackets).

CAMPO System	Deployment Scenario			Evaluation Process			Continuity		
	interface	mobile node	envi-ronment	input	pro-cessing	output	inte-gration	granu-larity	vis-ibility
Stemm and Katz 1998	interface	mainly single-on	eval on client, cm on infra	static, phy	embedded, local	interface, single value	loose	per node	proxy
Wei Song et al. 2005	infrastructure	single-on	eval on infra	static, net	embedded, global	connector, single value	na	na	na
Minji Nam et al. 2004	interface (infrastructure)	single-on	eval on client, cm on infra	primarily static, phy	embedded, local (global)	both, single value	tight	per node	tran-sparent
Hongyang Bing et al. 2003	infrastructure	single-on	eval on client	dynamic, phy	embedded, local	connector, single value	tight	per node	tran-sparent
Hou and O'Brien 2006	infrastructure	single-on	eval on client	dynamic, phy	embedded, local	connector, single value	na	na	na
Cheng Wei Lee et. al 2005	infrastructure	single-on	eval on client, cm on infra	dynamic, phy/net	embedded, local	connector, single value	loose	per node	proxy
Chebrolu and Rao 2006	infrastructure	multiple-on	eval on client, cm on both	dynamic, net	embedded, local	connector, single value	loose	per channel	proxy
Mohanty and Akyildiz 2006	infrastructure	single-on	eval on client, cm on both	dynamic, phy	embedded, local	connector, single value	loose	per node	proxy
Qian Zhang et al. 2003	infrastructure	single-on	both on client	dynamic, phy	embedded, local	connector, single value	loose	per node	end-to-end
Adamopoulou et al. 2005	infrastructure	single-on	eval on client	dynamic, phy/net/app	function, local	connector, value set	na	na	na
Nasser et al. 2006	infrastructure	single-on	eval on client	dynamic, phy/net/app	function, local	connector, value set	na	na	na
Xing and Venkatasubramanian 2005	infrastructure	multiple-on	eval on client	dynamic, phy/net/app	function (kn.), local	connector, single value	na	na	na
Gazis et al. 2005	infrastructure	multiple-on	eval on client	dynamic, net/app	function (kn.), local	connector, single value	na	na	na
Qingyang Song and Jamalipour 2005	infrastructure	single-on	eval on client	dynamic, net	function (ahp), local	connector, value set	na	na	na
Kibria and Jamalipour 2007	infrastructure	single-on	eval on client	dynamic, phy/net	function (ahp), local	connector, value set	na	na	na
Bari and Leung 2007	infrastructure	single-on	eval on infra	dynamic, phy/net/app	function (topsis), local	connector, value set	na	na	na
Balasubramaniam and Indulska 2004	infrastructure	single-on	both on infra	dynamic, phy/net/app	function, both	connector, single value	loose	per node	proxy
Luan Huang et al. 2006	infrastructure	single-on	eval on infra	dynamic, net	function, global	connector, single value	na	na	na
Vidales et al. 2005	infrastructure	single-on	eval on both, cm on infra	dynamic, phy/net/app	policy, local	connector, single value	loose	per node	proxy
Jun-Zhao Sun et al. 2004	infrastructure (path)	multiple-on	both on client	dynamic, phy/net/app	policy, local	connector, single value	loose	per channel	end-to-end
Ylitalo et al. 2003	infrastructure	multiple-on	eval on client, cm on infra	dynamic, phy/net/app	policy, local	connector, single value	loose	per node	proxy
Jun-Zhao Sun et al. 2005	infrastructure	multiple-on	both on client	dynamic, phy/net/app	policy, local	connector, single value	loose	per channel	end-to-end

Wei Song et al. 2007	infrastructure	single-on	eval on both	dynamic, phy/net	policy, global	connector, single value	na	na	na
Wei Zhuang et al. 2003	infrastructure	single-on	eval on both	na	policy, global	connector, single value	na	na	na
Hung-Yu Wei and Gitlin 2004	peer (single-hop)	single-on, RG multiple-on	cm on client	na	na	na	na	na	na
Luo et al. 2003	peer (multi-hop)	single-on, PC multiple-on	cm on client	na	na	na	na	na	na
Chunyan Fu et al. 2006	peer (MANET)	single-on, CGW multiple-on	eval on client, cm on both	na	na	na	na	na	na
Lera et al. 2005	peer (single-hop)	single-on, GTW multiple-on	cm on both	na	na	na	both	per node	tran-sparent
Buddhikot et al. 2004	infrastructure	single-on	cm on both	na	na	na	loose	per node	proxy
Kristiansson and Parnes 2006	infrastructure	multiple-on (single-on)	cm on client	na	na	na	loose	per channel	end-to-end
Akyildiz et al. 2005	infrastructure	single-on	eval on both, cm on infra	na	na	na	loose	per node	proxy
Inoue et al. 2004	infrastructure	single-on	cm on infra	na	na	na	tight	per node	proxy
Jyh-Cheng Chen and Hong-Wei Lin 2005	infrastructure	single-on	cm on infra	na	na	na	loose	per node	proxy
Salkintzis et al. 2004	infrastructure	single-on	cm on infra	na	na	na	both	per node	tight tran-sparent, loose proxy
Yi Pan et al 2004	infrastructure	single-on	cm on client	na	na	na	loose	per channel	end-to-end
Snoeren and Balakrishnan 2000	infrastructure	single-on	cm on client	na	na	na	loose	per channel	end-to-end
Li Ma et al. 2004	infrastructure	single-on	cm on client	na	na	na	loose	per node	end-to-end
Bellavista et al. LNCS 2005	infrastructure	single-on	cm on client	na	na	na	loose	per node	end-to-end
Bellavista et al. ICDCSW 2005	infrastructure	single-on	cm on infra	na	na	na	loose	per node	proxy
Politis et al. 2004	infrastructure	single-on	cm on infra	na	na	na	loose	per node	proxy
Shenoy and Montalvo 2005	infrastructure	single-on	cm on infra	na	na	na	loose	per node	tran-sparent
Qi Wang and Ali Abu-Rgheff 2006	infrastructure	single-on	cm on both	na	na	na	loose	per node	proxy

3.4 Conclusive Remarks on the CAMPO Taxonomy

The CAMPO area has recently gained relevant interest from both academic and industrial researchers. In fact, the wide availability of several wireless communication technologies, coupled with increased resource availability at mobile clients, asks for novel support solutions to take full advantage of the new network/client capabilities. These research efforts have already provided several interesting contributions, differing not only for the specifically addressed issue, e.g., seamless vertical handover, but also in relation to assumptions made about the targeted execution environment, e.g., mobile node capabilities and available networks. However, common models and frameworks that permit to clearly describe CAMPO contributions and to easily compare them are still missing.

This chapter aims to two primary goals. On the one hand, **it originally identifies three main categories for their classification**, i.e., deployment scenario, evaluation process, and continuity management. The proposed taxonomy is exploited to clarify the characteristics of the two main families of CAMPO solutions, i.e., 4G and ABC. On the other hand, it **deeply analyzes the state-of-the-art** in the field, not only classifying the existing literature according to the proposed taxonomy, but also pointing out current trends of evolution of the CAMPO research area.

While already proposed contributions demonstrate ABS feasibility and potential benefits, there is still the need to face up some open challenges to leverage and accelerate the widespread ABS scenario adoption. By analyzing the evolution trends of not only ABS but also current CAMPO literature, we claim that future proposals should have the primary goal of **integrating several technologies managed by multiple operators**. These proposals should provide effective mechanisms to enable interoperability among heterogeneous components in an open and dynamic way. In particular, we claim that ABS researchers should mainly focus their current investigation efforts on:

1. **context-aware evaluation processes**, with an optimal trade-off between evaluation flexibility/effectiveness and expressive power of context representations. Evaluation processes should exploit context awareness to dynamically adapt their behavior to execution environments. Moreover, they

should primarily work on mobile nodes to improve scalability and to reduce dependence on external support;

2. **hybrid deployment scenarios**, which include both infrastructure and peer connectors, the latter with possibly complex coordination capabilities as in multi-hop multi-path heterogeneous connectivity scenarios. In particular, we envision a growing involvement of mobile nodes to take full advantage of their increasing capabilities, e.g., by adopting multiple-on solutions;
3. **open and highly decentralized continuity management solutions**. In fact, continuity management is the crucial issue not yet satisfactorily addressed by ABS proposals. We claim that next generation continuity management should be as much technology-independent as possible. While cellular/IEEE 802.11 tight integration has represented a valuable first step towards seamless connectivity, continuity management should be performed in a more distributed and open way. Open CAMPO systems should exploit third-party support components instead of ad hoc ones as in current tightly integrated scenarios. In addition, the distribution of continuity components on both client and infrastructure sides will permit to integrate heterogeneous technologies for wireless connectivity in a more effective and economically-efficient manner.

Chapter 4 - Translucent and Context-aware Integrated Management of Heterogeneous Positioning Systems

The primary goal of this work is to take full advantage of the different connectors, channels and paths provided by the envisioned ABS scenario. To correctly evaluate the suitability degree of these networking opportunities there is the need to consider many context information. The activity has allowed to consider and analyze several heterogeneous context sources, e.g., user mobility degree and mobile client handover prediction as Chapter 5 will show. However, we have identified user location as one of the most valuable (and currently the most exploited) context information, as several location-based applications already available demonstrate. For this reason the first part of the PhD activity has been devoted to the analysis and integration of several heterogeneous positioning systems. This chapter focuses on positioning systems and geographical location. However, the proposed architecture and developed middleware for positioning system integration is only a first step toward a generalization to wider set of context sources and information.

In particular, we have followed the design rules of translucent and cross-layer control to design and develop our **Positioning System Integration and Management (PoSIM)** middleware for the integrated management of heterogeneous positioning systems. PoSIM has the twofold goal of enabling both the mediated visibility of all the information provided by underlying positioning systems and the mediated control of their configurable characteristics for a synergic context-dependent management. In particular, PoSIM provides the application layer with mediated and facilitated access to either low- and high-level API: LBSs can interact with underlying positioning systems in either a visible or transparent manner, respectively. Nevertheless, PoSIM can appear to the application level as a middleware offering a single, multi-faceted, and flexible API, thus simplifying its usage and potentially leveraging its adoption.

Figure 4.1 depicts our PoSIM middleware architecture. To interact transparently with positioning systems, simple LBSs can exploit the Policy Manager (PM) and Data Manager (DM) high-level API to respectively control positioning systems behavior and get their location information. To interact in a more visible and flexible way, smart LBSs can exploit the Positioning System Access Facility (PSAF) low-level API to directly access the Positioning System Wrappers (PSWs) for the currently available and integrated positioning systems.

About control (top-down colored arrows) and data (bottom-up white arrows) flows in Figure 4.1, let us anticipate that i) PM is the middleware component devoted to control and enables application-level management capabilities based on context information gathered from PSAF, and ii) DM, instead, plays the role of exposing location information according to dynamically configurable differentiated modes. PSAF, instead, can provide LBS with both control capabilities and positioning data. Let us stress PM and DM exploits low-level PoSIM API, i.e., PSAF methods, to offer an encapsulated high-level API with more articulated and easy-to-use services at a higher level of abstraction.

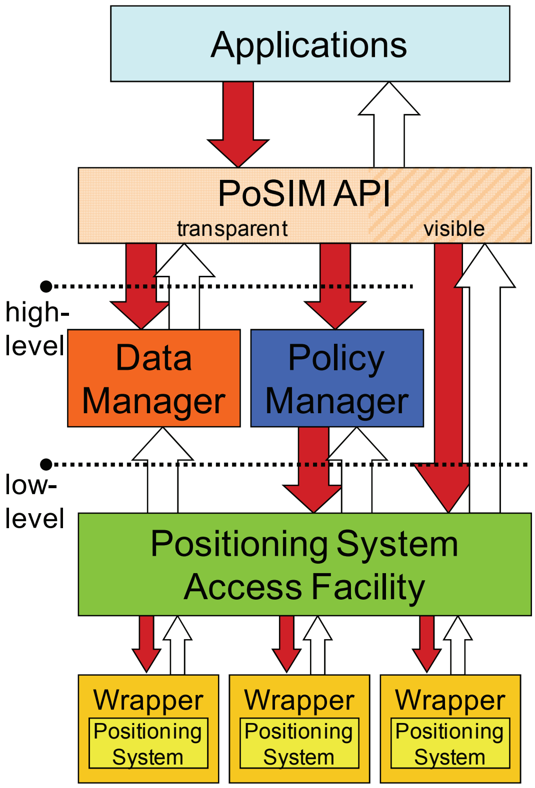

Figure 4.1 The PoSIM architecture (white arrows represent data flows, colored arrows are control flows).

PoSIM does not rely on any particular statically predefined ontology and on syntactic/semantic conventions on how to represent control capabilities and positioning data. It only defines a simple model distinguishing between positioning system features and infos. **Features describe positioning system characteristics and capabilities, possibly with settable values** useful for control/configuration, e.g., power consumption or ensured privacy level. **Infos are not configurable location-related data**, e.g., positioning information and its accuracy. Infos are the only data provided to simple LBSs while smart LBSs have visibility of both features and infos.

In the following, a relevant part of this chapter is dedicated to the main design and implementation guidelines of the PoSIM middleware. For each PoSIM component, we provide an overview of its functions and offered API, some practical usage examples to show how to take full advantage of its capabilities, and design/implementation insights. In addition, the final part of the chapter is devoted to our novel Privacy Enabler solution, which supports the management of the user privacy considering both user and LBS requirements.

4.1 Policy Manager

The **Policy Manager** (PM) is the PoSIM component responsible for enforcing the policies for **dynamic control and management of heterogeneous positioning systems**. In particular, the PM API allows simple LBSs to ask for pre-defined behaviors specified via default policies. PM is in charge of autonomously and dynamically interacting with positioning systems to transparently satisfy LBS requirements. Let us point out that PM provides a context-aware control of positioning systems: it can take into account both application-level requirements, e.g., minimum power consumption, and current system state, e.g., by avoiding to turn off a positioning system in the case it is the only one switched on and there is at least one LBS calling for positioning data.

Via the high-level and transparent PM API, LBSs can actively control positioning systems by simply specifying the desired behavior with no visibility of any low-level positioning detail. In particular, the PM provided methods are:

- `insert(newBehavior)/delete(aBehavior)`, to add/remove a new/existing PoSIM behavior;
- `activate(aBehavior)/deactivate(aBehavior)`, to effectively require the activation of a behavior among the already defined ones.

Behaviors are implemented as declarative policies, i.e., set of actions that PM must perform whenever conditions specified in the policy apply. Conditions are relational expressions related to positioning system infos/features; actions are management operations that PM performs over positioning system features. Let us observe that PoSIM allows not only to enable/disable a given behavior at service provisioning time by de/activating declarative policies, but also to introduce novel behaviors by adding new policies. In addition, any activity related to behavior definition and de/activation is independent from the actual implementation of both PoSIM and positioning system components below the PM level. In this manner, on the one hand, changes in integrated positioning systems cannot affect behaviors; on the other hand, LBSs can actively specify the desired control behavior transparently, thus facilitating and leveraging their development.

```
policy ::= [salience] name policy type
policy_type ::= isolated | ordering

isolated ::= conditions actions
ordering ::= ord_data bestN bestAct worstAct

bestAct ::= actions
worstAct ::= actions

conditions ::= cond | cond conditions
actions ::= action | action actions
cond ::= data value operator
action ::= Feature value
ord_data ::= numeric data

data ::= Info | Feature
bestN ::= non negative integer
salience ::= integer
name ::= string
value ::= string | integer | double
operator ::= =|!=|<|>|<=|>=|eq|neq
```

Figure 4.2 PoSIM policy representation.

As Figure 4.2 shows, PoSIM supports the specification and activation of two types of policies: isolated and ordering policies. **Isolated policies separately apply the same condition-action rules to each positioning system** retrieved at runtime in the execution environment. Conditions are a set of relational expressions, each one described with a data name/value and a relational operator. Supported relational operators include `=`, `!=`, `<`, `>`, `<=`, `>=`, and `'eq'/'neq'` (i.e., `=/!=` among strings). Actions are a set of operations on modifiable features, each one with an associated name and value. Given a positioning system, if all conditions are satisfied, the policy is triggered, namely fired, and all the features in actions are set to the values indicated in the policy, i.e., the policy actions are enforced. For instance, a PoSIM isolated policy could turn off the positioning systems with higher energy consumption if that does not endanger application-specific requirements about positioning precision and accuracy.

```
name:lowPowerConsumption
conditions:
    Feature(name:Power, value:8) op:>
    Info(name:Accuracy, value:5) op:<
actions:
    Feature(name:State, value:off)
```

Figure 4.3 The lowPowerConsumption isolated policy.

Figure 4.3 reports the `lowPowerConsumption` policy that switches off a currently available positioning system if its power consumption is greater than 8 and its accuracy below 5 (rapid notes about the mapping between power/accuracy values in the policy and their actual, possibly proprietary, counterparts in the integrated positioning systems are in Section 4.4).

```
name: onBestAccuracy
ord_data:
    Info(name:Accuracy)
bestN:
    1
best actions:
    Feature(name:State, value:on)
worst actions:
    none
```

Figure 4.4 The onBestAccuracy ordering policy.

Ordering policies, instead, can **compare available positioning systems** in order to sort them according to a desired indicator, e.g., listing positioning systems from the best to the worst one in terms of accuracy. In other words, in a sense the scope of ordering policies is wider than that of isolated ones, since ordering policies tend to intrinsically manage positioning systems aggregately. Ordering policy actions consist of two sets of features, best and worst: PM enforces best actions for the best `bestN` positioning systems, while it executes worst actions for the remaining ones. For instance, an ordering policy could request to always turn on the positioning system with best accuracy. Figure 4.4 depicts the `onBestAccuracy` policy that sorts positioning systems in relation to provided accuracy, and turns on the one with maximum accuracy.

In addition to the above examples, we have specified default policies in PoSIM, ready to be activated by simple LBSs. PoSIM already includes the following isolated policies of common usage:

- `onlyPhysical/onlySymbolic`, which activates only the positioning systems that offer physical/symbolic location information;
- `highAccuracy(threshold)`, which switches off all positioning systems whose accuracy is below `threshold`;
- `highPrivacy(threshold)`, which sets the privacy level of available positioning systems (at least) to the `threshold` value.

In addition, PoSIM defines the following ordering policies of common usage:

- `onlyBestAccuracy(bestN)`, which activates the `bestN` positioning systems (in terms of accuracy) by switching off all the others;
- `onlyBestConsumption(bestN)`, which keeps active only the `bestN` positioning systems in terms of lower consumption.

Let us notice that isolated policies compare info/feature values gathered at runtime with thresholds: a given isolated policy can be concurrently fired on different positioning systems (its triggering condition could be verified for several positioning systems at the same time); that should be carefully considered when specifying policies to avoid undesired behaviors. For instance, the above described `lowPowerConsumption` isolated policy is badly defined for most deployment environments because it could turn off all available positioning systems, thus making impossible to obtain any updated positioning information. This is one of the motivations why PoSIM also integrates ordering policies that provide the additional capability to manage positioning systems comparatively. In fact, ordering policies can enforce different actions depending on positioning system order and do not require specifying threshold values, which may be a hard task in many real-world deployment scenarios.

When different policies are simultaneously fired, in general there is also the possibility of conflicting actions. For instance, in the case of `lowPowerConsumption` and `onBestAccuracy` firing in the same time interval, the former may request switching off every positioning system, while the latter would turn on the positioning solution with highest accuracy. To help avoiding possible conflicts, any PoSIM policy is associated with a priority, either provided at development time (namely salience) or depending on policy activation order, e.g., recently activated policies are favored. Fired policies are enforced from the most prioritized to the least one, as further detailed in the following.

Let us also note that the definition of conflicting rules may not always be erroneous. For instance, consider again the simultaneous firing of `lowPowerConsumption` and `onBestAccuracy`, the former with less priority than the latter. If any integrated positioning system provides limited accuracy and imposes too high power consumption, `lowPowerConsumption` would turn off every positioning system. On the contrary, since `onBestAccuracy` has higher priority, certainly at least one positioning system will be maintained on, i.e., the one with best accuracy. In fact, as better detailed in the following, PoSIM recognizes conflicting actions (sets of operations working on the same positioning system features) and, in the case, only executes actions with higher priority.

Figure 4.5 depicts the PM architecture. The Policy Controller (PC) i) provides the capability to insert/delete and de/activate policies, ii) interacts with PSAF to get up-to-date info/feature values needed to evaluate the conditions of activated policies, iii) requests the Policy Engine (PE) to check for policy condition satisfaction and to execute the actions specified in fired policies.

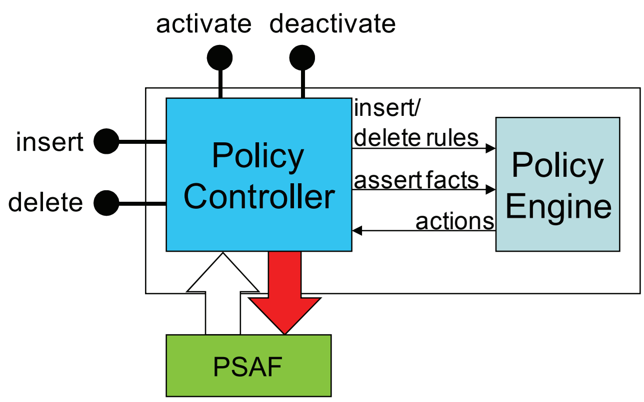

Figure 4.5 The architecture of the Policy Manager.

Delving into finer implementation details, the PoSIM PE exploits Jess [Jess], a rule engine based on the Rete algorithm [Forgy 1982]. PC automatically transforms new policies, described as Java classes, in Jess rules and, at their activation, provides PE with them. The Jess knowledge base includes only the infos and features that appear in at least one active policy condition, i.e., only infos and features relevant for currently activated policies. In that way, PC only retrieves the needed monitoring indicators from the underlying positioning systems, thus limiting the PoSIM middleware overhead.

By default, PE enforces policies by following the standard Jess "depth" (age-based) strategy, i.e., if several policies are simultaneously fired, PE performs the enforcement of the most recently activated one first and then fires the remaining ones in activation order. In addition, PoSIM administrators can add new policies by explicitly specifying a salience integer value, thus possibly affecting the order of policy enforcement. In particular, when specified, PoSIM policies are fired in relation to their salience, from the highest to the lowest: policies with the same salience value are fired with the Jess standard strategy (depth first); policy salience is set to 0 by default.

Let us observe that the PoSIM goal is not to specifically provide an original, powerful, and general-purpose policy management support. PoSIM simply exploits a subset of Jess existing capabilities, with the purpose of providing LBSs with the capability of dynamically adding and/or removing policies, even at service provisioning time, in an easy but conveniently flexible way. In fact, while in principle it could be possible to add in PoSIM whatever policy written in the Jess native language, we decided to limit the range of valid policies by imposing the mandatory structure reported in Figure 4.2. On the one hand, that simplifies the work of PoSIM administrators by providing a rigid but sufficiently expressive discipline for policy specification. On the other hand, that limitation reduces the risks of erroneous policy specification, by also paving the way to effective automatic tools for conflict identification and analysis. Moreover, Jess policies do not apply actions directly; in other words, Jess has no direct access to the PSAF component. When policies are fired, requested actions are not performed immediately but first ordered according to policy priorities. Then, if PoSIM recognizes conflicting actions, it only executes the actions related to the policy with highest priority, by inhibiting remaining actions. For instance, if both policy1 and policy2 are fired and the higher-priority policy1 requires to set power consumption to 3 while policy2 would set consumption to 5, then PoSIM sets power consumption to 3 by not considering policy2 at all. Finally, PM does not allow Jess loop rule activation, i.e., action enforcement does not produce the immediate re-evaluation of the conditions of all activated policies in a cyclic way, in order to simplify policy management and to limit enforcement costs.

4.2 Data Manager

The Data Manager (DM) is the PoSIM component responsible for offering an **aggregated view of positioning information** to the application level, thus providing differentiated context-dependent views of location data. In particular, DM aggregately provides PoSIM-based LBSs (specifying when and which positioning information they are interested in via the DM API) with the location info produced by the different integrated positioning systems and collected together in a single XML document. Let us stress that DM provides context-aware location information: PoSIM returns positioning data by taking into consideration both LBS requirements and positioning system information, e.g., by comparing the minimum accuracy required by an LBS with the accuracy level offered by each positioning system available.

In particular, LBSs can ask to be provided with the XML location data document in three different ways:

- on demand, exploiting either `onDemand()` or `onDemand(listener)` methods, which immediately provide the already estimated positioning data (last performed estimate). The latter method additionally applies LBS-specific filters, as better detailed in the following;
- at regular time intervals, exploiting the `periodical(interval, listener)` method, which commands a periodical delivery process to notify the listener every interval milliseconds;
- in an event-driven fashion, exploiting the `addEvent(event, listener)` method, which permits to specify a specific event to trigger future delivery of the location document.

LBSs can simply exploit easy-to-use **pre-defined conditions to trigger location data delivery**. For instance, the pre-defined `atLocation` condition triggers location notification only when the current symbolic location coincides with what specified as the invocation parameter. In addition, the **`addFilter(filter, listener)` method provides a simple way to filter positioning data**: for instance, the pre-defined `highAccuracy` filter automatically discards location information whose accuracy is below a given threshold. In addition, the proper exploitation of filtering rules permits to relevantly reduce the middleware overhead by avoiding useless notifications of non-relevant location changes. In summary, by exploiting the above methods, LBSs can specify both which information they are interested in and when they are interested in getting it without specific knowledge about the implementation details of the positioning systems they are using.

Let us notice that **declarative policies and filter rules have very different roles** in PoSIM and behave much differently. **Policies actively control positioning system behaviors**: they can modify positioning system features, which may impact on other features and on positioning info performance. For instance, the `lowPowerConsumption` policy deactivates positioning systems with a too high power consumption level, by possibly affecting positioning accuracy since some systems could be switched off by the policy enforcement actions. Moreover, a **policy activation impacts on any LBS** on top of PoSIM. For instance, the `highAccuracy` policy forces LBSs not to exploit positioning systems with low accuracy. On the contrary, **a filter rule just avoids to deliver positioning information considered useless by a specific LBS**, without any impact on positioning system working. Each LBS can declare its filtering rules, without any possible interference with other simultaneously working LBSs.

As rapidly mentioned, DM offers the access to any information generated by the integrated positioning systems, possibly added with context data from other sources, as an XML document. In that way, smart LBSs can have access to the wide set of location data and feature-related information available, in order to flexibly decide which information to exploit at the application level. In particular, the provided XML document consists of:

- a timestamp describing when DM created the XML document;
- a source for each exploited positioning system (embedded in a common sources parent tag);
- an info tag for each information provided by a source.

```
<Data>
    <timestamp time=docTS/>
    <sources>
        <source name="GPS">
            <info Location="xyz" />
            <info Accuracy="high" />
            <info Timestamp =locTS />
        </source>
        ...
    </sources>
</Data>
```

Figure 4.6 The structure of the PoSIM document with positioning data and their characteristics.

PoSIM describes a delivery triggering event as a triple including an info name, a value, and an `evaluate(...)` method, which returns true if the positioning info must be delivered, false elsewhere, usually depending on the evaluation of the positioning info itself. In particular, we have decided to implement two main triggering event categories of common usage: isolated and comparing. An isolated event exploits an `evaluate(Info threshold)` method that compares the current info value with a fixed threshold. A comparing event, instead, uses an `evaluate(Info currentValue, Info previousValue)` method to compare the current info value with the data provided in the previous delivery.

To clarify how these two event types can cover most common usage scenarios, let us rapidly present the following simple examples of pre-defined PoSIM events:

- `atLocation(loc)` is an isolated event that triggers data delivery only if the current symbolic location is equal to loc;
- `distance(dist)` is a comparing event that triggers data delivery only when the current location differs from the previously delivered positioning info for more than dist meters.

Let us note that the `distance(...)` triggering event has the result of providing positioning information by exploiting spatial variation as the triggering period ("space-periodical" positioning update). For instance, that could be useful for an advertising LBS interested in providing new information to the user whenever she moves more than 50 meters from the previous update location. In addition, PoSIM allows to specify and/or-aggregated events, i.e., sets of isolated/comparing events that trigger information delivery whenever all events occur (and modality) or at least one event occurs (or modality) during a specified time interval.

Filtering rules, instead, consist of an info name, a value, and an `evaluate(...)` method which, given the gathered info value, returns true if the info should be discarded, false otherwise. In other words, whenever an integrated positioning system (or any context source more generally) has the specified info and that info does not satisfy the given `evaluate(...)` method, DM discards the entire source (see Figure 4.6). For instance, one of the PoSIM pre-defined filter rule is called `onlyHighAccuracy(acc)` and discards every source whose accuracy is below acc (in a scale from 0 to 9, see Section 4.5.2).

Finally, let us note that **expert users**, such as PoSIM administrators, **can develop and deploy new policies, new triggering events, and new filtering rules** in a relatively easy way. In fact, all of them are implemented as Java classes that can be simply sub-classed to specify new specialized policies, events, and filters. In that way, the PoSIM behavior can be easily extended with impact on neither its implementation nor the application logic code. At the same time, simple LBSs and novice LBS developers can also work by only selecting their policies, events, and filters of interest among the set of pre-defined and most common ones already provided by default in the PoSIM distribution.

Figure 4.7 depicts the DM architecture. Data Builder (DB) collects infos from the currently exploited positioning systems and possibly aggregates them with context information of interest. DB periodically (every configurable polling period, 2 seconds is the default value) gets information from PSAF and provides gathered data as an XML document. Data Disclosure (DD) is the component that actually exhibits DM API, by exposing appropriate methods to specify how interested LBSs can get data. In other words, fed by DB monitoring information, DD delivers the XML document with positioning data to every registered LBS listener when either the polling period expires or an associated event occurs.

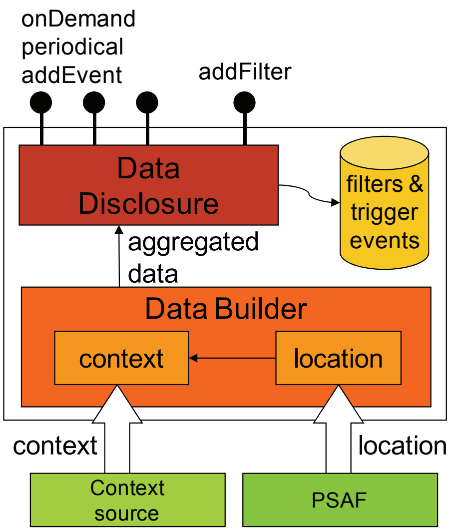

Figure 4.7 The Data Manager architecture.

The delivered XML document is the result of filtering the raw positioning data produced by the activated positioning systems with the filters specified by the interested listeners. Let us observe that each method of the DM API allows to specify a listener, apart from `onDemand()`; that increases the flexibility of our middleware solution if compared with other recently emerging proposals for positioning integration [Di Flora et al. 2005]. In fact, LBSs not only are able to simply gather location information with a one-shot interaction with PoSIM (`onDemand()` method), but also can ask for a more personalized delivery based on LBS-specific requirements implemented via the listener parameter. PoSIM can perform several articulated positioning data management actions, such as continuous location monitoring to verify if the available data are really of interest (`addFilter(...)` method) or if relevant events occur (`addEvent(...)` method). In that way, LBS development and deployment are greatly simplified; the only burden for LBS providers is to decide the triggering events, filtering rules, and time intervals for each of their listeners.

4.3 Positioning System Access Facility

Smart LBSs and PM/DM can directly control the integrated positioning systems by exploiting the lower level API of the **Positioning System Access Facility**

(PSAF). PSAF supports APIs to dynamically handle the registration/cancellation and to **retrieve/control infos/features of all the positioning systems locally available at the controlled mobile client**. In particular, the PSAF API allows:

1. to dynamically un/register a positioning system implementation in the set of locally available positioning solutions (the only constraint is that the registered positioning implementation offers a PSW-compliant interface, see the following);
2. to interact with registered positioning systems via the Query/Control interface.

The PSAF Query/Control interface enables the interaction with registered positioning systems in an aggregated and synergic way, by taking decisions depending on the whole set of available systems. In particular, the Query interface includes the following methods:

- `getInfos(posSysSet)/getFeatures (posSys-Set)`, which returns the set of info/features for the specified set of positioning systems;
- `getInfo(posSysSet,name)/getFeature (posSys-Set, name)`, which returns the value of a specific info/feature for the specified set of positioning systems;
- `getAvailable()`, which returns the list of the currently available positioning systems.

The Control interface, instead, offers the method:

- `setFeature(posSysSet, name, value)`, which changes the value of the name feature for the specified set of positioning systems.

For instance, in response to the invocation of `getInfos(null)`, PSAF provides all the info of every registered positioning system, while the invocation of `setFeature(GPS, State, off)` commands PSAF to change to off the value of the feature State of the positioning system named GPS.

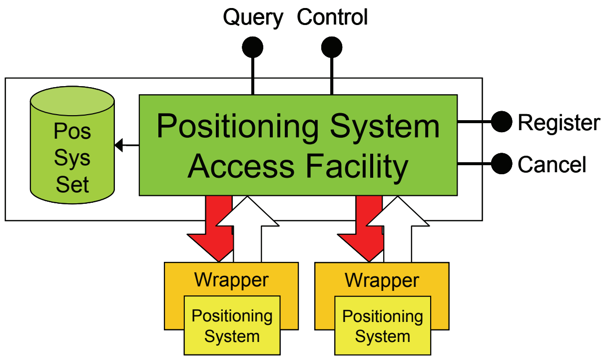

Figure 4.8 The architecture of the Positioning System Access Facility.

Smart LBSs and PoSIM middleware components can invoke the Query/Control methods; only PoSIM administrators, instead, can access the Register/Cancel interface. Let us stress that PSAF is the only way for higher middleware and application layers to access integrated positioning systems, thus guaranteeing controlled and system-safe accesses to low-layer positioning components, independently of their specific technique and implementation peculiarities. The only requirement is that positioning systems provide their infos/features via a specified interface; that interface is practically obtained by wrapping the implementations of positioning systems with PoSIM Positioning System Wrappers (PSWs). PSAF exploits Java introspection to dynamically determine and access the set of infos/features exposed by the wrappers and actually implemented by the underlying positioning systems that are currently available in its deployment environment.

4.4 Positioning System Wrapper

As already pointed out, the **Positioning System Wrapper** (PSW) is the crucial middleware component that **hides positioning system heterogeneity**. It exposes to the upper middleware layers a common API, independent of the wrapped positioning system and of its implementation details, by providing infos/features compliant to the exploited ontology for representing positioning-related data. For instance, if the ontology in use specifies that accuracy values are integers in the [0, 9] range, the PSW `getAccuracy()` method will provide location accuracy as an integer value. Any PSW component will interact with its wrapped positioning system, retrieve the associated accuracy value by exploiting positioning-specific awareness and syntax, and transform it accordingly to the adopted ontology, e.g., transforming a "high accuracy" string return value in the correspondent integer. That ontology is the only knowledge to be shared among the PoSIM components, which allows policies, triggers, and filters exploiting that ontology to be specified independently of the positioning implementation details.

Delving into finer details, PSW offers:

- a `getX()` method for each feature provided by the wrapped positioning system, where X is the name of the feature;
- a `setX(value)` method for each available modifiable feature, where value is the new value to be set for that feature;
- an `infoX()` method to read each location-related information provided by the wrapped positioning system, where X is the info name.

PSAF exploits Java reflection to correctly map its `getX()/setX()/infoX()` methods to the corresponding (sets of) lower-level invocations in the wrapped implementations of currently available positioning systems. For instance, given the wrapper of a particular positioning system, to get the current value of the Location info, PSAF invokes its `infoLocation()` method, while, to change the value of the `PowerConsumption` feature, PSAF invokes its `setPowerConsumption(newStrategy)` method, which changes the value of that feature to `newStrategy`.

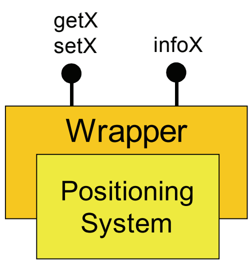

Figure 4.9 The Positioning System Wrapper API.

As already pointed out, the distinction between infos and features is the only assumption PoSIM performs on provided information. In fact, thanks to the adoption of Java introspection, PoSIM components are independent from the details of information representation. The integration of a new and unexpected type of positioning system into PoSIM only requires encapsulating it in a PSW that provides its infos and features through the above PSW interface.

We have already stated the flexibility stemming from not relying upon any statically specified ontology. In this manner it is possible to adapt PoSIM to any legacy component/application, even if not known at middleware development time. In the current PoSIM prototype implementation, we propose and adopt a simple ontology that should be taken into account when defining declarative policies, filter rules, and triggering events. The definition of such an ontology is not the specific scope of our research work and the currently exploited ontology can be easily modified/replaced without affecting the implementation of PoSIM components. In particular, the adopted ontology defines three main feature/info categories: mandatory, common, and peculiar. According to the ontology, any integrated positioning system must offer mandatory features/infos. We consider as mandatory:

- Location info, the last location information provided by the wrapped positioning system,
- Timestamp info, the time in which the provided location info has been estimated,
- PSState info, either on or off to indicate whether the positioning data has been obtained either correctly or not,
- Name feature, to get positioning system name,
- State feature, a modifiable feature to switch on/off a positioning system,
- ExploitedComm feature, e.g., IEEE 802.11 for Ekahau and Bluetooth for the GPS solutions using Bluetooth connectivity towards their clients, and,
- LocationType feature, whose value can be physical, symbolic or both.

The infos/features classified as common are optional (some positioning systems may decide not to implement them) but pre-defined as they are of common and frequent usage. For instance:

- Accuracy info, related to the provided location information,
- PrivacyLevel feature, to indicate if the positioned client can hide its location information,
- PowerConsumption feature.

Finally, we consider also the possibility to include other a-priori unknown infos/features, peculiar to a specific positioning system and thus not usually shared between PSWs. For instance:

- GPS FixType info, which can be 2D, 3D or no fix and that makes sense only when considering the GPS positioning system,
- the Ekahau Status feature, providing detailed Ekahau-specific information about the working status of the Ekahau Positioning Engine.

Let us stress again the differences between common and peculiar features/infos. For instance, PoSIM models the power consumption feature as common (not mandatory) because it is not always possible to evaluate power consumption for every positioning system but the feature is of common usage and LBS developers could be aware of its possible availability. When the feature is available, there is the need to agree on the measurement unit of its returned value, e.g., in mwatt or in a scale from 0 to 9, and that is specified in the ontology. On the contrary, peculiar features/infos can be added freely by PSW implementers, with no impact on the adopted ontology and without any requirement on returned value semantic.

Table 4.1 reports mandatory, common, and some examples of peculiar features/infos. In the implemented ontology, the physical location information is modeled in terms of latitude, longitude, and altitude, while symbolic location information is represented as a layered (hierarchical) location, e.g. [Italy, Bologna, EngSchool, Lab2] (additional details in Section 4.6). Accuracy is represented by an integer value between 0 (minimum) and 9 (maximum). The privacy level has a value between 0, uncontrolled location information delivery, and 9, stealth mode, i.e., only the positioned client has access to its own location. Power consumption is modeled with a value in the [0, 9] range, usually measured in a static way (see the PoSIM implementation insights in the following section).

Among the above listed infos/features, let us rapidly focus on two of them, State and PSState, to better explain their semantic. The State feature returns on/off depending on the fact that the positioning system is switched on/off, thus being exploitable or not to obtain positioning data. Even if a currently switched off positioning system cannot provide localization info, the correspondent PSW can continue to offer old positioning data based on previous values, implicitly specifying they are history-based estimations via the timestamp info. Also PSState is either on or off, representing if the positioning operations of a switched-on positioning system have been performed in a correct way in the last time interval. For instance, even if a GPS device is active (State is on), it could not be able to provide a correct location information (PSState is off) since there are not enough satellites in line of sight (no fix according to the GPS terminology).

Table 4.1 Mandatory, optional, and peculiar infos/features as defined in the default PoSIM ontology.

<table>
<tr><th colspan="2">Category</th><th>Name</th><th>Modifiable</th></tr>
<tr><td rowspan="7">Mandatory</td><td rowspan="3">Info</td><td>Location</td><td>n.a.</td></tr>
<tr><td>Timestamp</td><td>n.a.</td></tr>
<tr><td>PSState</td><td>n.a.</td></tr>
<tr><td rowspan="4">Feature</td><td>Name</td><td>no</td></tr>
<tr><td>State</td><td>yes</td></tr>
<tr><td>ExploitedComm</td><td>no/yes</td></tr>
<tr><td>LocationType</td><td>no/yes</td></tr>
<tr><td rowspan="3">Common</td><td>Info</td><td>Accuracy</td><td>n.a.</td></tr>
<tr><td rowspan="2">Feature</td><td>PrivacyLevel</td><td>no/yes</td></tr>
<tr><td>PowerConsumption</td><td>no/yes</td></tr>
<tr><td rowspan="2">Peculiar</td><td>Info</td><td>FixType (GPS)</td><td>n.a.</td></tr>
<tr><td>Feature</td><td>Status (Ekahau)</td><td>no</td></tr>
</table>

4.5 PoSIM Implementation Insights

In this section we present the PoSIM middleware at work in our actual experimental test-bed; the main purpose is to exemplify how it is possible to provide infos and set/get features of three off-the-shelf positioning solutions (GPS, an IEEE 802.11-based positioning system, and a Bluetooth-based one) and a generic positioning system compliant with the JSR-179 Location API for J2ME. We provide some details about the operational mode of each integrated positioning system and compare exposed control capabilities and provided information. Furthermore, we point out how we have actually integrated these positioning systems implementing proper PSW components. Finally, we present an example of development and deployment of an LBS built on top of PoSIM together with some considerations related to the achieved experimental results. Additional information and the downloadable code of the PoSIM prototype, together with the PSWs for the presented positioning systems, are available at the PoSIM Web site [PoSIM].

4.5.1 Integrated Positioning Systems

Several heterogeneous positioning systems are currently widespread. Here we focus our attention on three of them, GPS, Ekahau and BTProximity, because they exemplify positioning system heterogeneity in terms of exploited positioning technique (e.g., triangulation, proximity), provided information (e.g., physical, symbolic location), and positioning delivery mode (e.g., on demand, event-driven). The sub-section provides the few needed implementation insights about these three positioning techniques to understand the implementation decisions described in the following.

GPS is currently the most spread positioning system, exploited in several commercial applications ranging from navigation aid to car tracking. GPS determines node location via triangulation by exploiting knowledge about satellite constellation position and node-satellite constellation distance [McNeff 2002].

Ekahau [Ekahau] is a positioning system for Wi-Fi-based nodes and is based on techniques of scenario analysis and on characteristics of IEEE 802.11 communications, similarly to RADAR [Bahl and Padmanabhan 2000]. Scene analysis techniques include two phases: a preliminary off-line phase and an operational one. In the former phase, the positioning system gets knowledge about AP RSSI in the monitored environment, i.e., it associates physical locations with neighbor AP MAC addresses and corresponding RSSIs. In the latter phase, nodes send RSSI data to the Ekahau Positioning Engine (EPE), the Ekahau component which actually calculates node localization. EPE compares historical and currently observed RSSI data, inferring node current location.

BTProximity [PoSIM] is our original positioning system with user privacy capabilities, based on proximity techniques and Bluetooth communication technology. In particular, BTProximity simply associates one node with the location of the closest reference point, i.e., Bluetooth device, whose distance is inferred by exploiting baseband connection RSSI. Other Bluetooth-based positioning systems are available in the literature [Genco 2005, Anastasi 2003]. Differently from them, BTProximity specifically focuses on privacy management: user privacy is achieved by carefully hiding node presence to reference points, that is not revealing to infrastructure nodes where the node is notwithstanding the node exploits reference points to determine its location. In particular, BTProximity supports the provisioning of three privacy levels: low, medium, and high. Each privacy level corresponds to a different Bluetooth device configuration, as better detailed in the following. In particular, when BTProximity privacy level is

- Low, the Bluetooth node periodically broadcasts a message, as reference points do, by revealing its presence to anyone (the Bluetooth node is in Page/Inquiry Scan mode [Bluetooth]);
- Medium, the Bluetooth node does not broadcast messages but only accept incoming connections (the Bluetooth node is in Page Scan mode). If an external device knows the MAC address of the Bluetooth node, it could try to connect to it by performing a sort of blind connect; if the connection attempt is successful, node location is revealed. Moreover, the Bluetooth node connects to visible reference points to determine RSSI values, by potentially revealing its presence (the Bluetooth protocol requires active baseband connections to determine RSSI);
- High, the Bluetooth node completely hides its presence (stealth mode – the node is in No Scan mode). It neither broadcasts messages nor accepts incoming connections; it can only listen to reference points broadcasting messages. To maximize user privacy, the Bluetooth node does not even connect to reference points. Since without connection RSSI data is not available in Bluetooth, the Bluetooth node cannot understand which is its closest reference point. In this case, BTProximity provides, as current location, the set of the locations of all reference points in radio communication range.

Let us rapidly observe that BTProximity accuracy relevantly depends on required privacy level: the high-privacy level is intrinsically associated with a significantly lower accuracy than low and medium BTProximity privacy levels.

4.5.2 PSW Implementation Insights and Supported Infos/Features

The current PoSIM prototype includes wrappers for all the positioning systems presented in the previous section plus an additional generic PSW suitable for any positioning solution exposing a JSR-179-compliant API. Table 4.2 reports infos and features offered by the implemented PSWs and describes how they transform/represent gathered information to comply with the proposed ontology.

GPS provides physical location information in terms of latitude, longitude, and altitude: no additional transformation actions on determined positioning data are required to be compliant with our default PoSIM ontology. The GPS PSW gathers information from a GPS device communicating through a serial port (possibly via a Bluetooth-based virtual serial port) by exploiting the standard Java Communication API [JavaComm]. This permits to achieve full portability independently of the underlying operating system.

In particular, when the State feature is on, the GPS PSW reads and parses NMEA 0183 sentences to achieve location information from the wrapped GPS positioning system. When that info is valid, i.e., the GPS device has a 2D or 3D fix, the PSState info value is set to on. The privacy level is fixed at the maximum value because the node computes its location in a completely decentralized manner, without any support by neighbors or network servers. Finally, the GPS accuracy is dynamically inferred from the Horizontal Dilution Of Precision (HDOP), a GPS-specific value dependent on the current configuration of the satellite constellation. In particular, our experiments have pointed out a rather linear relationship between HDOP values and accuracy in meters (see Figure 4.10). Therefore, the GPS PSW sets accuracy to 9 when HDOP is close to 0, to 0 when HDOP is greater than 30, and to linearly determined intermediate values otherwise.

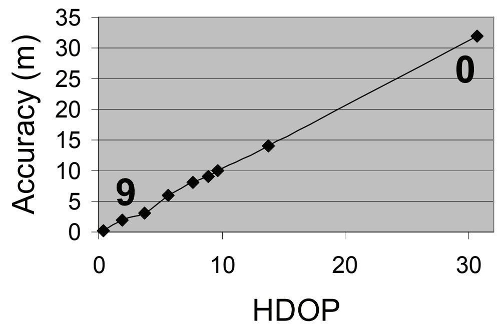

Figure 4.10 Experimental results about the relationship between HDOP and accuracy in GPS.

Ekahau can provide both physical and symbolic location data via event-driven API. However, positioning info is provided in relation to Ekahau internal maps; therefore, the Ekahau PSW must perform actions to transform the “proprietary” Ekahau location info accordingly to the exploited ontology. In particular, the Ekahau PSW is in charge of transforming physical coordinates and logical areas of Ekahau maps into latitude/longitude/altitude and layered location information, respectively, by exploiting additional context data related to maps. To this purpose, the only requirement is that Ekahau administrators orient their Ekahau maps to north and specify their top-left and bottom-right point coordinates, altitude, and possibly higher-layer symbolic location information, e.g., [Italy, Bologna], the country and the city where the map is located. When State is off, the Ekahau PSW stops locally gathering and sending RSSI data to EPE, with the benefit of relevantly limiting power consumption. To this purpose, we do not exploit the proprietary non-controllable Ekahau client but our own original and more flexible Ekahau client implementation with power consumption optimizations [PoSIM]. The Ekahau privacy level is limited: in fact, to gather RSSI values, the IEEE 802.11 node willing to be positioned must turn on its wireless card, thus providing the network infrastructure with a certain degree of knowledge of its location. Moreover, when the EPE server is not local (the location estimation is made by an EPE server not running on the mobile client itself), the location info is necessarily disclosed to the EPE node. Finally, let us note that Ekahau allows accuracy tuning: it is possible to request either an accurate or a latest computed location.

BTProximity only provides symbolic information via event-driven API, directly in the form required by the ontology. The BTProximity PSW accuracy depends on the number of locations provided; when only one location is provided, the accuracy is 8 (due to Bluetooth short range), when BTProximity provides a set of multiple locations corresponding to the visible Bluetooth reference points (e.g., because the privacy level is set to high), the accuracy level is set to 6.

Finally, as already stated, **the JSR-179 PSW implements a generic wrapper to every JSR-179 compliant positioning system**. The JSR-179 PSW provides both physical and symbolic information (when made available by the wrapped positioning solution). To test our implementation, we have developed a JSR-179 PSW encapsulating GPS devices and offering an interface that partially implements the JSR-179 API [PoSIM].

Table 4.2 Features/infos for the 4 positioning systems integrated in the current PoSIM prototype.

Positioning System	Category	Capability	PSW Implementation	Modifi-able
GPS	Info	Location	no required actions	n.a.
		PSState	*off* if invalid fix, *on* if valid fix	n.a.
		Timestamp	time of the last location update	n.a.
		Accuracy	dependent on HDOP	n.a.
		FixType	*no fix*, *2D fix*, *3D fix*	n.a.
	Feature	Name	GPS	no
		State	*on*: reading and parsing NMEA sentences *off*: not reading	yes
		ExploitedComm	serial port name, e.g., *COM2* or *rfcomm*	yes
		LocationType	physical	no
		PrivacyLevel	9 (stealth mode)	no
Ekahau	Info	Location	actions required to transform Ekahau map dependent information in absolute information	n.a.
		PSState	*off*: location information are not available *on*: location information are available	n.a.
		Timestamp	time of the last location update	n.a.
		Accuracy	either 5 (LatestLocation) or 7 (AccurateLocation)	n.a.
	Feature	Name	Ekahau	no
		State	*on*: RSSI sending and location gathering *off*: neither RSSI nor location gathering	yes
		ExploitedComm	IEEE 802.11a/b/g	no
		LocationType	both	yes
		PowerConsumption	dependent to the underlying IEEE 802.11 network interface (7 if always on, 4 if in power saving)	yes
		PrivacyLevel	either 3 (remote EPE) or 6 (local EPE)	no
		Accuracy	either 5 (LatestLocation) or 7 (AccurateLocation)	yes
		Status	detailed state information provided by EPE	no
BTProximity	Info	Location	no required actions	n.a.
		PSState	*off*: positioning deactivated, *on*: elsewhere	n.a.
		Timestamp	time of the last location update	n.a.
		Accuracy	8 if only one location, 6 if more than a location	n.a.
	Feature	Name	BTProximity	no
		State	*off*: positioning deactivated, *on*: elsewhere	yes
		ExploitedComm	Bluetooth device name, e.g., hci0	yes
		LocationType	symbolic	no
		PowerConsumption	2 (Bluetooth imposes limited power consumption)	no
		PrivacyLevel	5 (low), 7 (medium), 9 (high)	yes
JSR-179 (Location API for J2ME)	Info	Location	no required actions	n.a.
		PSState	*on*: state is *AVAILABLE*, *off*: elsewhere	n.a.
		Timestamp	time of the last location update	n.a.
		Accuracy	horizontal accuracy dependent	n.a.
	Feature	Name	JSR179	no
		State	*on*: gather location every second *off*: location gathering deactivated	yes
		ExploitedComm	JSR179	no
		LocationType	both	yes

Figure 4.11 provides a global overview of our integrated positioning systems and related PSWs. An interesting aspect is that most middleware components are completely independent of the underlying operating system. On the contrary, each positioning system actively interacts, to some extent, with the operating system, often in a proprietary and non-portable way. Therefore, to maximize portability, we have designed and implemented a few interoperability middleware components (striped in the figure). GPS, Ekahau, and BTProximity PSWs are widely portable: on the one hand, the Comm API provides a portable access to serial port; on the other hand, the IEEE80211NetInt/BluetoothNetInt component allows to portably access IEEE 802.11/Bluetooth features (we exploit the Comm API implementation provided by the rxtx project since the Sun Comm API does not support MS Windows anymore). However, while BTProximity can fully work on Linux since BlueZ drivers provides RSSI estimation, similar drivers are not yet available for MS Windows (in particular, it is impossible to gather Bluetooth connection RSSI on MS XP/Vista). On MS Windows BTProximity cannot determine which is the closest visible reference point and thus provides every visible reference point as current location. IEEE80211NetInt and BluetoothNetInt functions are provided by the Network Interface Provider (NIP) component; additional details about NIP and how it transparently gathers RSSI values are extensively described in the following chapter.

Another interesting implementation insight is that PSW complexity greatly depends on the characteristics of wrapped positioning systems, in particular the positioning data format and the adopted access method. For instance, the BTProximity PSW can access the underlying positioning system via event-driven API, thus reducing possible overhead due to unnecessary polling; in addition, in this case, gathered information are already compliant with the adopted ontology. On the contrary, the GPS PSW has to command the reading of NMEA 0183 sentences provided by GPS via serial port and to parse these sentences to extract location information, while the Ekahau PSW has to transform map-related location information in absolute coordinates, thus increasing PSW implementation complexity. Further implementation details, out of the scope of this paper, and the PoSIM prototype source code are available at the project Web site [PoSIM].

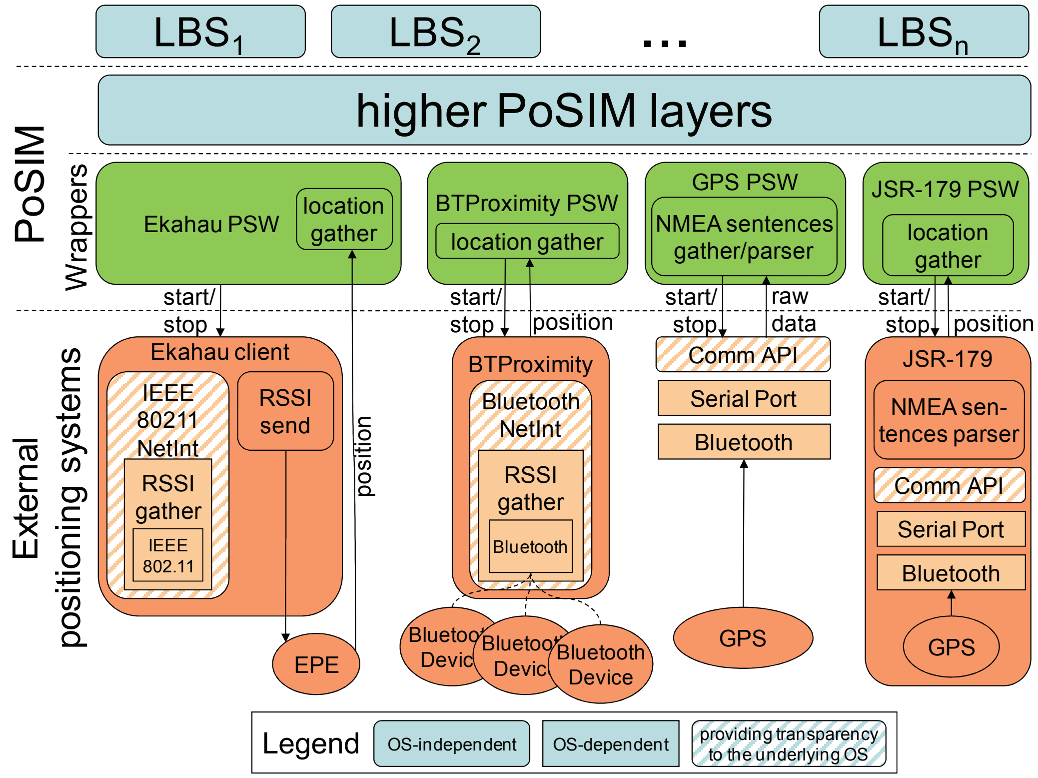

Figure 4.11 The detailed architecture of the implemented PSWs and integrated positioning systems.

4.5.3 An Example of PoSIM-based LBS

To practically show how PoSIM integrates heterogeneous positioning systems and supports rapid LBS prototyping and deployment, this section presents an example of LBS that takes advantage of PoSIM capabilities. In particular, we report about the development and testing of an Advertising service, deployed in a wide shopping mall consisting of several distributed buildings. The **Advertising LBS** aims to offer commercial information whenever a user is in the proximity of predefined locations, such as previously registered shops. Moreover, if the user accepts to disclose her location data, the LBS wants to record user paths for user movement pattern analysis, both inside buildings and in the paths between buildings. To gather the maximum amount of positioning-related data, the LBS needs to simultaneously exploit all the available positioning systems: GPS for outdoor localization, Ekahau for indoor physical and symbolic localization, and BTProximity for indoor symbolic localization.

Delving into finer details, it is possible to summarize the Advertising LBS requirements as follows: i) to simplify LBS working, the collected location information must be represented in a uniform way, ii) to correctly perform user tracking, physical information must be delivered at least every 10 meters, despite the actually exploited positioning system, and iii) a secondary requirement is to improve the robustness of LBS results by exploiting the location information from the most accurate source whenever multiple sources are simultaneously available.

```
<Data>
  <timestamp time="1173974696718" />
  <sources>
    <source name="GPS">
      <info Location = "physical: latitude = 00.00 N, longi-
tude = 00.00 E, altitude = 50.0" />
      <info PSState="off" />
      <info Accuracy="0" />
      <info FixType="No fix" />
      <info Timestamp="..." />
    </source>
    <source name="Ekahau">
      <info Location = "physical: latitude = 44.48 N, longi-
tude = 11.32 E, altitude = 103; symbolic: [(Italy, Bologna,
ShopCentre, TravelAgency)]"/>
      <info PSState="on" />
      <info Accuracy="7" />
      <info Timestamp="..." />
    </source>
    <source name="BTProximity">
      <info Location= "symbolic: [(Italy, Bologna, ShopCen-
tre, TravelAgency), (Italy, Bologna, ShopCenter, CoffeShop)]"
/>
      <info PSState="on" />
      <info Accuracy="6" />
      <info Timestamp="..." />
    </source>
  </sources>
</Data>
```

Figure 4.12 The positioning information document provided by LBS in the Advertising LBS example.

PoSIM dramatically facilitates the design and implementation of such an Advertising LBS. First of all, PoSIM provides uniform location information in compliance with the representation syntax and semantic described in the adopted ontology. For instance, Figure 4.12 reports the information provided by PoSIM inside a building: note that GPS accuracy is minimum because GPS is unsuitable for indoor localization (PSState is off), Ekahau provides both physical and symbolic location data, and BTProximity accuracy is only 6 because in this case it can only provide multiple locations of Bluetooth reference points in visibility.

Since Ekahau provides physical information in terms of latitude, longitude, and altitude, the `distance(...)` triggering event presented in Section 4.2 could be exploited even when the user moves from inside to outside a building, i.e., even when Ekahau becomes unavailable and GPS starts to be the positioning system actually providing the location data (and vice versa). In that way, the second requirement is easily fulfilled.

Finally, it is possible to answer to the third requirement by simply activating the `highAccuracy(8)` isolated policy and the `onBestAccuracy(1)` ordering policy. The former automatically deactivates positioning systems with accuracy lower than 8; the latter, with higher priority, always maintain the positioning system with highest priority switched on. Therefore, consequently to the enforcement of these two policies, the only positioning system available outdoor is GPS, and both Ekahau and BTProximity are deactivated there. When the node to be positioned moves indoor, the GPS accuracy rapidly decreases while Ekahau and BTProximity become available (BTProximity with high privacy level and thus with limited accuracy). When GPS accuracy goes lower than 8, the `highAccuracy` policy would try to deactivate it but the prioritized `onBestAccuracy` policy keeps GPS active because it is still the only positioning system available. Only when GPS accuracy goes lower than 7, i.e., below Ekahau accuracy, PoSIM deactivates GPS and activates Ekahau.

In place of `highAccuracy` and `onBestAccuracy` policies, the Advertising LBS could exploit the `onlyHighAccuracy` filter rule, thus gathering information only related to positioning systems with high accuracy. However, by adopting the two policies above, it is possible to achieve the additional goal of limiting power consumption because policies switch off positioning systems instead of just discarding unnecessary positioning information.

4.6 Privacy Enabler: Effective and Privacy-enabled Location Management

When LBSs will get out from research labs and will involve a wide public of final users, two primary issues will come out as essential: the former is the technical challenge of how to effectively manage the exchange of positioning information (and of its variations), also by considering the high heterogeneity of currently used positioning systems and location information; the latter is a social issue, i.e., how to guarantee the proper level of user privacy given the need to disclose, to some extent, user location information to enable LBSs.

While our PoSIM middleware provides a valuable support to easily and quickly develop and deploy LBSs based on heterogeneous positioning systems, it does not consider user privacy requirements when disclosing location information to LBSs themselves. In fact, LBSs can be deployed on remote untrusted nodes and can potentially harm user privacy, e.g., publishing her location on public Web pages.

In addition to the PoSIM middleware, we have developed the **Privacy Enabler** solution; its objective is to support the **management of the user privacy considering both user and LBS requirements**. In particular it discloses user location at the proper degree of accuracy, to correctly invoke LBSs while not providing the actual user position. Note that the Privacy Enabler is external to the PoSIM middleware itself and can be even regarded as an example of LBS application, e.g., exploiting high-level API provided by the DM component. However, it can be used also as an additional and optional PoSIM component LBS clients can take advantage of.

Delving into finer details, the original contribution of our Privacy Enabler solution is the extension of the PoSIM middleware with functions for **efficient handling of location information with privacy requirements**. Our extended middleware prototype can provide LBSs with location data exposed at the most proper level of precision (location granularity); the main idea is to manage user privacy by disclosing user location only partially, i.e., at a lower accuracy. The definition of proper granularity strictly depends on the applicable privacy and efficiency requirements, dynamically negotiated between clients and LBSs. This is important particularly when LBSs are deployed on remote nodes; once the mobile client has disclosed user location information, there is no guarantee the remote LBS actually exploits this information only to properly provide the required service and then discards it. In fact, the LBS could even spread this information without any capability of the user to prevent or even only monitor it. In this section we specifically consider the case of remote LBSs, since more challenging in relation to user privacy management.

In addition note that Privacy Enabler manages user location privacy in a greatly different manner if compared with positioning systems described in the previous sections. In fact, Privacy Enabler does not aim to change the behavior of underlying positioning systems. Instead it actively downgrades the accuracy of the location information retrieved by positioning systems or the PoSIM Data Manager component. The primary goal is to disclose the location information only partially.

A first element to consider in LBSs with user location privacy requirements is to determine who is in charge of positioning. We claim that privacy-enabled LBSs are simpler to develop and deploy when clients (or trusted positioning servers in client localities) are the only entities fully aware of their location and are responsible for communicating it to LBSs. Note that the spread of client-based positioning systems, such as GPS and BTProximity, is pushed not only by their great scalability, but also by their gained acceptance in user communities. Positioning systems able to compute the user location without any external component aid are inherently considered as more suitable in relation to privacy concerns, and thus more widely adopted. Therefore, we focus our work on localization solutions where clients estimate their positions either in a completely autonomous decentralized way or via local trusted servers close to them, such as in the case of Ekahau.

Another relevant factor to improve location privacy is to disclose positioning information at the proper granularity, i.e., with the minimum precision needed to satisfy the LBS provisioning requirements. To this purpose we have focused on the simple symbolic representation model with variable granularity levels. Table 4.3 exemplifies possible client positions with different granularity: depending on the precision required by an LBS, the useful position information for a mobile client may be either α (granularity=3) or β (granularity=4). In particular, if an LBS requires granularity x, even if the client can obtain its position with granularity y>x, the client should only divulgate its localization with granularity x, i.e., with the minimum possible precision. For instance, locations from 1 to 7 may represent successive positions in a user path with granularity=6, while LBS-required granularity could be lower; that example will be exploited in the experimental result section in the following.

Table 4.3 Our Granularity-differentiated Symbolic Location Model.

Location ID	Granularity	Location information
α	3	Italy, Tuscany, Florence
β	4	Italy, Emilia, Bologna, EngFaculty
l1	6	Italy, Emilia, Bologna, EngFaculty, Lab2, PhDZone
l2	6	Italy, Emilia, Bologna, EngFaculty, Lab2, Office
l3	6	Italy, Emilia, Bologna, EngFaculty, Lab2, StudZone
l4	6	Italy, Emilia, Bologna, EngFaculty, CommLab, BTStation
l5	6	Italy, Emilia, Bologna, EngFaculty, CommLab, Admin
l6	6	Italy, Emilia, Bologna, MathFaculty, Floor1, Room12
l7	6	Italy, Emilia, Bologna, MathFaculty, Floor1, Room5

The proper location granularity should be negotiated, for any client-LBS pair, **depending on both user preferences and LBS requirements**. Our primary solution guideline is to adopt middleware-level proxies, which execute on the fixed network in client proximity, for granularity negotiation and location obfuscation on behalf of their associated clients. Proxies can alleviate resource-limited devices from location management operations and, most important, can enforce location privacy requirements with no impact on client application logic. By focusing on middleware efficiency, let us point out that usual CPU/memory limitations of mobile clients suggest deploying middleware components over the fixed network, possibly in proximity of the served mobile clients, while portable devices should only host thin clients, loaded by need and automatically discarded after service. In addition, by choosing appropriate granularity, they can significantly reduce the network traffic exchanged due to position modifications. For instance, in the case of LBSs with results to update at location changes with granularity=4, our proxies can inform LBSs about the movements of their associated clients only when changing faculty buildings and not when entering new rooms in the same building.

We have identified two different proxy-based architectures for privacy-enhanced efficient management of location data: with proxies only on client side (CProxies) and with both client- and server-sided proxies (CProxies and SProxies). Middleware solutions based on CProxies only are simpler to deploy since there is no need for server-sided support infrastructure; however, the achievable privacy is intrinsically limited by the fact that LBSs could identify clients by tracking associated proxies (the countermeasure is overloading clients by forcing them to continuously change exploited proxy instances). A double level of middleware proxies can achieve greater privacy and anonymity: when using CProxies and SProxies, the middleware can mediate any communication between clients and LBSs; in addition, CProxies/SProxies can be the only entities to know the specific privacy preferences of their associated clients/LBSs.

Figure 4.13 depicts the architecture of our middleware solution, based on two level of proxies (CProxies and SProxies). Each mobile client hosts the execution of a lightweight Mobile Node Stub (MNStub) and is assisted by one CProxy running on the fixed network, in the same network locality of the Wi-Fi access point that currently provides client connectivity. MNStub works to achieve seamless roaming by pre-fetching data when its mobile client is expected to perform a handover, by alleviating problems due to temporary network unavailability. CProxy is a mobile agent that migrates by following mobile client changes of access points, thus maintaining co-locality with the served MNStub (the dotted line in the figure represents wireless communications between MNStub and its CProxy). CProxy co-locality with its associated MNStub notwithstanding client roaming permits to reduce network latency and overhead during service provisioning. CProxy is in charge of client-side privacy maintenance and granularity negotiation. Finally, each deployed LBS interworks with one server-sided SProxy that transparently enhances its LBS with privacy negotiation functionality.

CProxy and SProxy exploit a secure SSL communication channel to negotiate the appropriate location granularity and to exchange the needed position modifications. Note that, in our middleware solution, users should exclusively trust their CProxies and the same applies to LBSs with their SProxies, thus requiring the establishment of only local trust relationships with middleware components.

To better detail how our middleware works, the mobile client informs CProxy about its privacy level requirements, i.e., the maximum granularity at which it agrees on exposing its location. CProxy possibly decreases exposed location granularity in the case that LBS requirements are compatible (lower than privacy-related ones). Then, CProxy invokes service execution to SProxy, which finally contacts the actual LBS component. Let us note that each proxy level can contribute to reduce network traffic to preserve wireless link bandwidth: CProxy does not communicate location changes not relevant for LBS granularity requirements; similarly, SProxy does not notify service result variations with finer granularity than client privacy requirements in the case of publish/subscribe model of interaction.

Our flexible middleware architecture can simply enable alternative solutions for location granularity filtering: for instance, CProxy could forward mobile client location at maximum precision, together with user privacy requirements, to SProxy; SProxy could be the only responsible for granularity reduction, by increasing location update traffic but potentially enlarging the usability of location data if SProxy serves different LBS components with differentiated granularity requirements in its locality. It is also possible to downscale location granularity only to respect LBS desiderata, thus obtaining only a form of user anonymity and not location obfuscation.

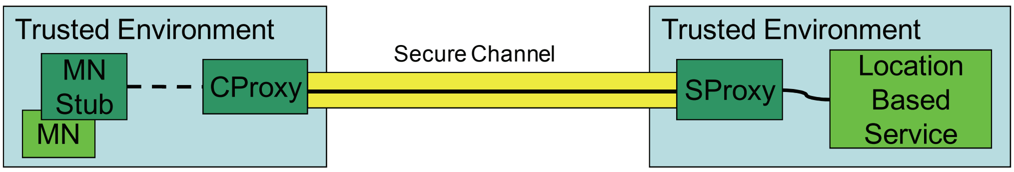

Figure 4.13 The Architecture of our Proxy-based Middleware for Location Management.

Let us finally point out that, to achieve stronger user anonymity, our middleware can be easily extended by implementing either Onion or Mix mechanisms in CProxy [Dingledine et al. 2004, Berthold et al. 2000]. These solutions both specifically focus on providing users with strong anonymity, independently of position and movements, by concentrating on real-time Internet services and on malicious attackers capable of observing any communication link. Another possibility is to associate one CProxy with all mobile clients served by one access point, thus mix-ing the requests of different users to the same LBS but introducing a potential performance bottleneck.

Experimental Results

To quantitatively evaluate the performance of our two-level proxy-based middleware, we have considered the case of a simple LBS that provides clients with the list of all resources available in their locality. Suppose that clients move from l1 to l7 (see Table 4.3) at provision time; the positioning system can estimate user location with granularity 6, the LBS requests granularity 5, and users desire to disclose their position with granularity 4; each location has the same number of resources (10) and resource descriptions have all the same size (3.3KB each).

To identify the isolated overhead contribution due to our middleware, we have decided to consider four possible working modes:

- PrivacyOff – the proposed middleware is not used, i.e., mobile clients request service provisioning directly to LBS;
- Anonymous – the middleware does not perform any location granularity downscaling;
- Server Side Privacy Management (SSPM) – the middleware SProxy performs granularity downscaling and resource filtering;
- Client Side Privacy Management (CSPM) – the same as SSPM but with CProxy in charge of downscaling and filtering.

The experimental results reported in the following are specific for the above scenario. However, similar results can be obtained in any deployment environment where potential positioning granularity is greater than LBS requirements (that condition applies to most LBSs, such as in city/museum guide assistants based on Wi-Fi positioning estimation.

From the deployment point of view, in the experimental testbed CProxy, SProxy, and LBS run on different nodes with different available bandwidths. In particular, the mobile client and CProxy communicate through a wireless link with limited bandwidth of 500Kbit/s; CProxy and SProxy can exploit a 2Mbit/s wired connection to mimic geographic distribution; the bandwidth between SProxy and LBS is 8Mbit/s. We have deployed our middleware components on Pentium4 2.8GHz desktops with 1GB RAM connected to the same 100 Mbit/s LAN; differentiated bandwidths are obtained via emulation.

Since LBS granularity is greater than user privacy level, when the middleware performs location downscaling for privacy requirements, LBS tends to send more service results than needed. For instance, when a client is in l1, LBS provides all the objects in EngFaculty and not only Lab2 objects. To reduce useless traffic and service response time, the middleware tailors LBS results accordingly to actual user location, independently of enforced privacy. Moreover, when a client moves from l1 to l2/l3, or from l4 to l5, or from l6 to l7, the middleware does not propagate new mobile client service requests (request dropping) since it is aware that no location variation of interest for LBS has occurred.

We have identified one synthetic performance indicator, Cumulative Service Time (CST), defined as the sum of all service response times experienced in the current and already visited locations. For instance, CST at l3 is the sum of response times measured in l1, l2, and l3. The CST indicator is relevant to understand middleware performance while used in the typical usage scenario of clients continuously accessing their LBS while moving along a path, where it is sometimes possible to reduce response time and network traffic thanks to request dropping.

Figure 4.14 reports CST for the different middleware working modes (not including the delay for SSL channel instantiation between CProxy and SProxy - about 547ms – which has to be sustained only once at CProxy startup).

In the case of PrivacyOff, the mobile client directly contacts LBS and performs service requests anytime the mobile client changes location, regardless LBS granularity. Therefore, CST exhibits an almost linear growth when increasing location ID. When the working mode is Anonymous, instead, service response time in each location greatly depends on current and already visited locations: the two proxies introduce a nonnegligible delay when the mobile client does its first request to LBS (l1) due to request/response propagation through our middleware components (1438ms instead of 813ms for PrivacyOff). However, successive responses, e.g., the ones from l2 and l3, are prompter (about 200ms in place of more than 700ms) because CProxy can also perform request dropping.

In all cases where location management has the twofold goal of privacy enforcement and traffic reduction, the most interesting middleware working modes are SSPM and CSPM. In both modes, LBS sends more objects than strictly needed because our middleware provides it with downscaled client location. In

SSPM mode, it is the SProxy that performs service tailoring and unfiltered data only overload the LBS-to-SProxy link: CST at l7 is only 140ms higher than in Anonymous mode. In CSPM mode, since CProxy is in charge of service tailoring, unfiltered results also overload the SProxy-to-CProxy communication link, by introducing additional delay. However, actual user location is visible only at the client side in CSPM, thus achieving a stronger and more secure level of privacy.

Finally, let us note that the middleware performance can be further increased in any deployment case where i) the difference between location granularity and LBS requirements is large, thus enabling frequent request dropping at CProxy, and ii) the caching of either client location data or service results makes sense (for instance, result caching at SProxy when it serves multiple clients and at CProxy when it deals with successive requests of the same user).

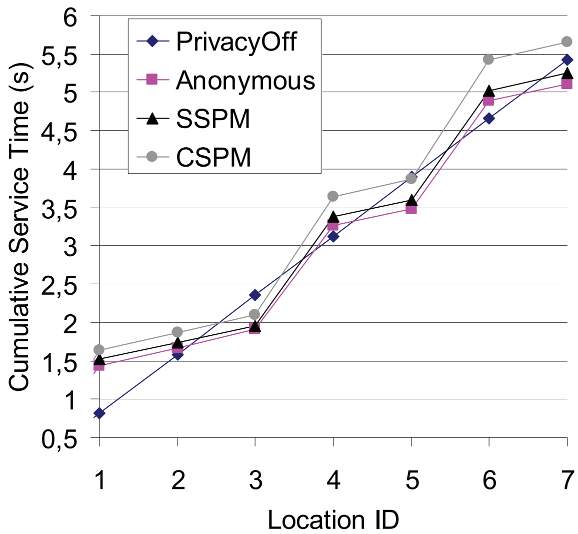

Figure 4.14 CST for different working modes.

4.7 Summary of Contributions and Original Aspects

As Chapter 3 has already shown, several research activities have recently addressed the area of dynamically integrating positioning systems, in particular to fuse location information from different sources [Ranganathan et al. 2004, Spanoudakis et al. 2003]. Most solutions propose transparent approaches that hide applications from positioning complexity, but do not support any application-specific form of control on the positioning techniques currently available at a node. Instead, PoSIM provides a uniform access to both **gather data and control the behavior** of integrated heterogeneous positioning systems.

Only a few proposals have just started to delineate cross-layer supports that provide application-level visibility of low-level details and control features of available positioning techniques [Graumann et al. 2003, Agre et al. 2002]. However, [Graumann et al. 2003] only claims the need for cross-layer middleware solutions to smartly select the most suitable positioning system at runtime. [Agre et al. 2002], instead, supports the control of positioning systems in a hard-coded and not flexible manner. In addition, to achieve the visibility of data and control features of a specific positioning system, [Agre et al. 2002] requires its full static knowledge, thus significantly increasing the LBS development complexity. Instead, PoSIM not only provides full visibility of low-level details, but permits even to **dynamically change the adopted control policy**.

In other words, PoSIM considers the aforementioned contributions and answers similar issues by greatly improving the dynamicity, flexibility, and extendibility of the support for positioning integration and management. To the best of our knowledge, no support solution in the literature addresses the challenge of **cross-layer integrated control of available positioning systems** by considering runtime application-level requirements in a flexible and extensible way and at dynamically differentiated levels of visibility.

In fact, the original translucent approach of the PoSIM middleware permits to access integrated positioning systems **in both a transparent and middleware-mediated way**, respectively fitting simple and smart LBS requirements. PoSIM not only supplies low-level information, but also **permits an active control of integrated positioning systems**, via the proper exploitation and/or definition of policies, events, and filters. In addition to providing a useful integration tool freely available for download and further refinement to the LBS community, the PoSIM project has also demonstrated, via practical examples, how the adoption of our middleware can leverage LBS development by relevantly facilitating both the synergic exploitation of positioning systems and the rapid LBS prototyping/deployment.

In addition, to the best of our knowledge, there are no research activities proposing middleware solutions to simultaneously face location privacy issues and efficient management of positioning data, by exchanging location information with differentiated granularity levels. Some interesting work in the literature already handles some partial aspects related to client-based positioning and user location privacy. [Ward et al. 2004] proposes location obfuscation: LBSs can only access a uniformly downscaled location information (with lower precision and lower geographical granularity) instead of exact client positions. [Titkov et al. 2003] realizes user anonymity through a Mediator Agent, i.e., either a user-controlled or a trusted-third-party mediator that separates mobile terminals from service providers on the fixed network. Differently from the above contributions, our solution performs **location management in a simple and lightweight manner**, by providing a **partial form of anonymity**, suitable and sufficient for most LBSs. However, let us rapidly observe that solutions like [Dingledine et al. 2004, Berthold et al. 2000] are complementary to our proposal and can be integrated with it.

Let us finally note that our novel Privacy Enabler solution supports the privacy-enabled location management decoupling the responsibility of location maintenance/processing from service-side application components. In this manner it also simplifies the design and implementation of LBSs. The most novel contribution of our Privacy Enabler is that **it discloses user location only when actually required and at a proper granularity considering both user and LBS requirements**. Our Privacy Enabler prototype demonstrates that it is possible to achieve feasible performance even without sacrificing portability, by adopting decentralized proxy-based solutions capable of reducing network traffic via proper management of different location granularities.

Chapter 5 – Middleware for Seamless ABS connectivity in Multi-hop Multi-path Scenarios

The widespread availability of multiple wireless interfaces (Wi-Fi, Bluetooth, UMTS, …) and of increasing computing resources at portable devices is pushing towards the novel ABS scenario where there is a wide set of connectivity opportunities, changing at anytime. We have already showed the need of innovative models and original context-aware middlewares to face the challenge of ABS connectivity management over multi-hop, multi-path, heterogeneous wireless networks. In particular, the full exploitation of multi-hop multi-path connectivity opportunities offered by heterogeneous wireless interfaces could enable innovative deployment scenarios where mobile nodes dynamically self-organize to offer/exploit Internet connectivity at best.

In this chapter, we propose a novel middleware that tries to find the optimal balance in several directions by following a general solution of a tradeoff between the extreme ones. We have decided to follow design rules of tradeoff between **local and global management**, tradeoff between **single- and multi-path granularity**, and tradeoff between **static and dynamic responsiveness**. On behalf of running applications, the middleware aims to seamlessly exploit the available connectivity opportunities at best (in terms of bandwidth, economic costs, durability, …), by composing them at runtime depending on context, e.g., application requirements, user preferences, and expected node mobility. In particular, our novel middleware can enable the ABS scenario by **effectively considering a limited set of practical indicators for a coarse-grained estimation of expected reliability/quality of multi-hop paths available at runtime**.

In short, our innovative middleware manages the durability/throughput-aware formation and selection of different multi-hop paths simultaneously, based on practical lightweight indicators on node mobility and wireless network characteristics. In addition, it proactively manages active connections in order to minimize user perceived service interruption whenever a mobile client perform a handover procedure.

Differently from first middleware proposals emerged for the ABS scenario, we have focused our attention to the support of continuous services via multi-hop multi-path heterogeneous connectivity. To that specific purpose, we claim the need for innovative evaluation processes with **the primary goal of maximizing connectivity durability**, e.g., to minimize the number of channel reconfiguration unless quality requirements cannot be met with currently used connectors. **As a secondary but crucial goal**, these evaluation processes should work to **maximize useful throughput** for the served services while minimizing power consumption at mobile clients. In addition, for the sake of performance and scalability, evaluation metrics should be primarily based on **context** data that can be directly gathered at mobile nodes, limiting interaction with the network infrastructure to crucial context information to reduce the imposed monitoring overhead.

About context awareness, let us notice that we do not aim to provide here a general-purpose framework for the gathering and disclosure of high-level context information. Our solution only aims to identify and exploit a limited subset of relevant context information, e.g., dynamically changing degree of node mobility, to effectively address mobility-related multimedia management issues.

Based on the above guidelines, we have designed and implemented the Multi-hop Multi-path Heterogeneous Connectivity (MMHC) middleware, which specifically targets the support of continuous services in ABS scenarios.

In particular, the chapter presents how MMHC models any possibly available connector, how it performs its original evaluation process based also on connector classification, and how it manages connectivity continuity. In fact, there are three primary novelty aspects in MMHC: i) the support to **very heterogeneous type of connectors**, by considering their differentiated characteristics, ii) **the wide set of context information**, at different levels of abstraction, exploited by the evaluation process, and iii) **the dynamic management of connectivity opportunities and connections**, to achieve the most suitable channels and minimize user perceived service interruption when a channel re-configuration is required.

First, we provide a description of the supported connector types and analyze our middleware architecture from a high-level point of view (Section 5.1); note that the provided connector classification reflects our primary guideline of considering user mobility as the most crucial context information. Then, we point out how the proposed MMHC middleware actually supports multi-hop multi-path heterogeneous connectivity, by analyzing its primary components in charge of managing networking opportunities and presenting achieved performance results (Section 5.2). Finally, we describe how the middleware provides the application layer with continuous connectivity by considering the challenging case of a service with strict continuity requirements, e.g., audio/video streaming (Section 5.3).

5.1 MMHC Deployment Scenario and Architecture

MMHC originally supports different types of connectors and classifies them according to the taxonomy depicted in Figure 5.1. MMHC supports infrastructure-based connectors with IEEE 802.11 and GPRS interfaces. In addition, MMHC supports peer-based connectors with IEEE 802.11 and Bluetooth interfaces. Infrastructure connectors are always fixed, i.e., it is assumed they cannot move; peer connectors can be either fixed or mobile.

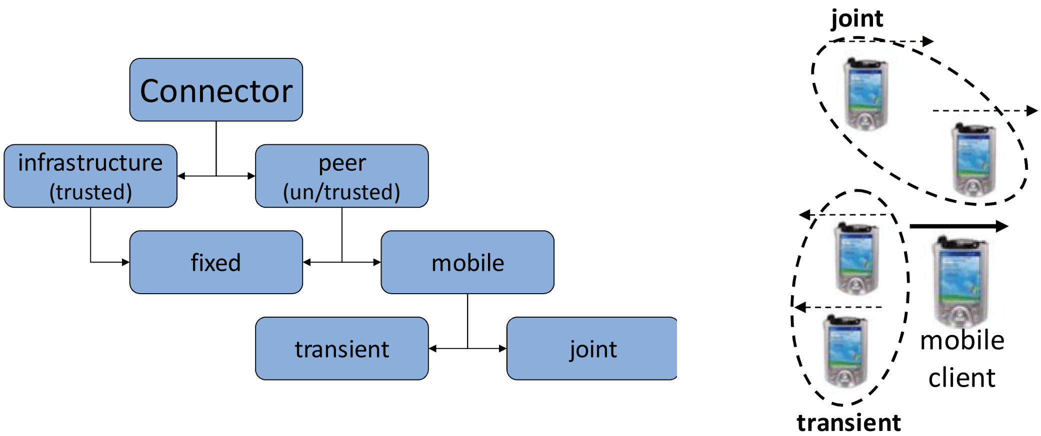

Figure 5.1 Types of connectors supported in MMHC.

MMHC considers crucial aspects that deeply differentiate the runtime behavior of peer and infrastructure connectors. First of all and most important, MMHC distinguishes **fixed and mobile connectors**. Understanding whether a peer connector is fixed/mobile is crucial (and a challenging issue) because it directly impacts on the stability of offered connectivity: mobile peers usually become unavailable with higher probability because more easily they can exit the client radio range. In particular, MMHC considers an innovative and highly dynamic context indicator:

the mutual degree of mobility between a mobile client and the associated mobile peer connector. MMHC classifies mobile connectors as either transient or joint, depending on the fact that, respectively, the connectors move with either different or the same speed (in both module and direction) of the associated mobile client. The transient/joint sub-class obviously depends on mobility behaviors at runtime and its correct dynamic determination/update is a key point for MMHC effectiveness in terms of limited overhead and durable evaluation process results. In fact, MMHC supports the exploitation of both transient and joint mobile connectors, but transient ones usually have higher probability of becoming rapidly unavailable, e.g., because the transient mobile peer is a PDA carried by a user walking on the same sidewalk with opposite direction. On the contrary, joint mobile peers, such as a PDA connector sharing the same train wagon with its client, can probably provide a more suitable connectivity offer with greater durability.

Besides client and connector mobility, MMHC considers even the **trust degree**. Infrastructure connectors are considered always trusted, i.e., MMHC assumes that this type of connectors always try to forward the traffic of associated clients (and are expected to succeed, apart from dependability issues due to traffic congestion) and do not endanger user privacy (no traffic auditing). Security issues are not the primary focus of the MMHC research project and are not addressed in the thesis. Instead, MMHC dynamically determines the trust level of peer connectors depending on connector runtime behavior (based on user preferences, past interaction history, and client location). MMHC evaluates the above classification for the eligible connectors of a given client only at the beginning of its service session. The classification of connectors into trusted/untrusted sub-classes is a rather infrequent decision, expected not to change during a service session.

Starting from above connector considerations, we have designed and implemented the MMHC middleware to support the ABS scenario, by originally and specifically considering client mobility degree as primary context information. Figure 5.2 represents how the MMHC middleware is logically organized. The MMHC layered architecture reflects the major phases in handover procedures already presented in Chapter 3: Context Gathering to collect information about mobile clients and remote connectors, Metric Application to evaluate interfaces/channels/paths suitability, and Continuity Manager to support the seamless handover among connectors. The **Context Gathering** layer consists of the Network Interface Provider (NIP) and the Mobility & Peer Estimator (MPE) components, which are the primary context sources in MMHC. NIP provides a uniform and aggregated access to underlying network interfaces, e.g., by providing the set of available connectors for each interface; MPE determines mobility state for clients and connectors, e.g., whether a client is currently still or mobile and the probability a handover will occur in a short time interval. MPE strictly reflects the importance of the mobility degree in our original solution. The **Metric Application** layer consists of Connector Manager (CoM), Routing Manager (RoM) and Path Application Selector (PAS) that respectively evaluate connectors, channels, and paths. CoM identifies the list of suitable connectors and performs one-hop channels with them depending on context information locally available and on the whole mobile client requirements to perform to; RoM considers both local and remote context information and accordingly changes routing rules between channels provided by CoM in order to achieve multi-hop paths; PAS selects the most suitable path in the RoM-provided list by additionally considering application-specific requirements. The **Continuity Management** layer is mainly composed of the Smart Buffer (SB) that proactively adapts client-side buffer size and trigger reconfiguration/migration of infrastructure-side components (further details in Section 5.3).

Let us note that Figure 5.2 not only points out each MMHC component role, but also delineates the associated abstraction layer: MIP provides information about available connectors; MPE is at a slightly higher abstraction level and exploits NIP output to dynamically evaluate mobile client and connector mobility degree; CoM interacts with connectors to estimate their suitability for realizing reliable and durable channels; RoM provides durable paths based on both local and remote information; PAS evaluates RoM-provided paths to select the best one for each application; SB provides applications with the abstraction of a continuous connectivity.

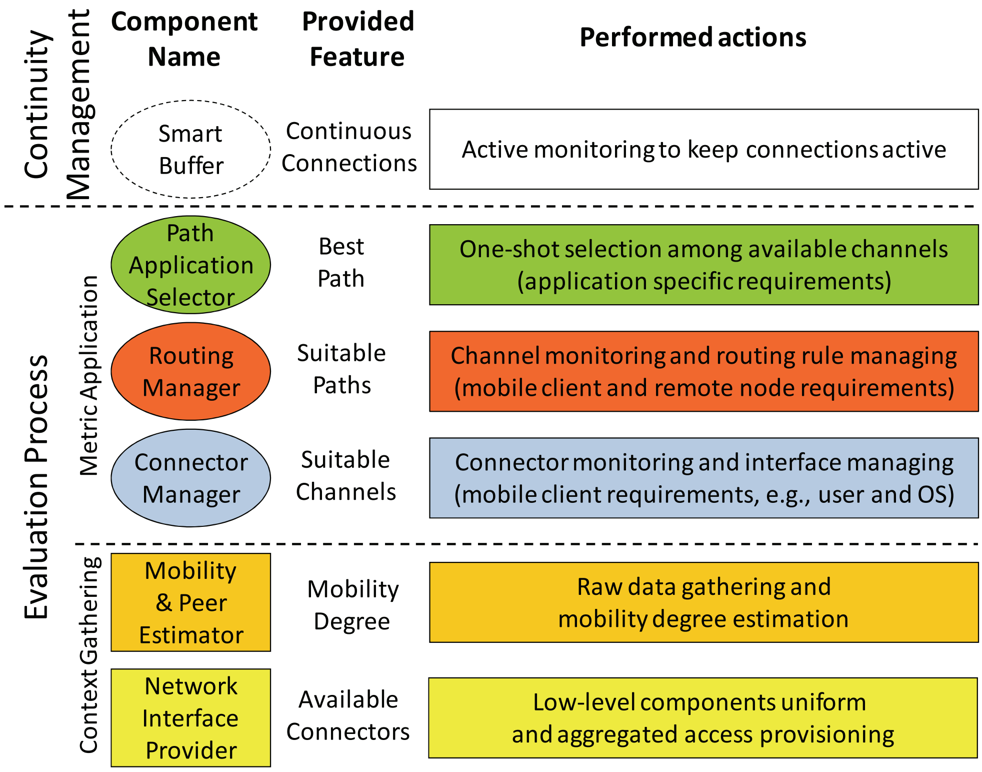

Figure 5.2 MMHC logical organization.

Figure 5.3 depicts how we have implemented the model of Figure 5.2 in the design and implementation of the MMHC prototype architecture. Differently from the purely layered model of Figure 5.2, the MMHC architecture adopts a cross-layer solution for the sake of performance: NIP behaves as input for both MPE and CoM; MPE provides information to both CoM and SB, and is configured by CoM by exploiting a feedback loop; moreover, each MMHC component not only provides its features and information to other middleware components, but also to the application layer. In this manner, MMHC is able to behave both as an autonomous and self-contained system providing the most suitable path and as a support component useful for other support infrastructure, e.g., the SB solution which is actually implemented as an external middleware for continuity management deployed on top of MMHC. In any case, a clear distinction between context gathering, metric application and continuity management is ensured. The rest of the chapter provides details about the primary components of the current MMHC middleware prototype: NIP and MPE for context gathering, CoM, RoM and PAS for metric application, SB for continuity management.

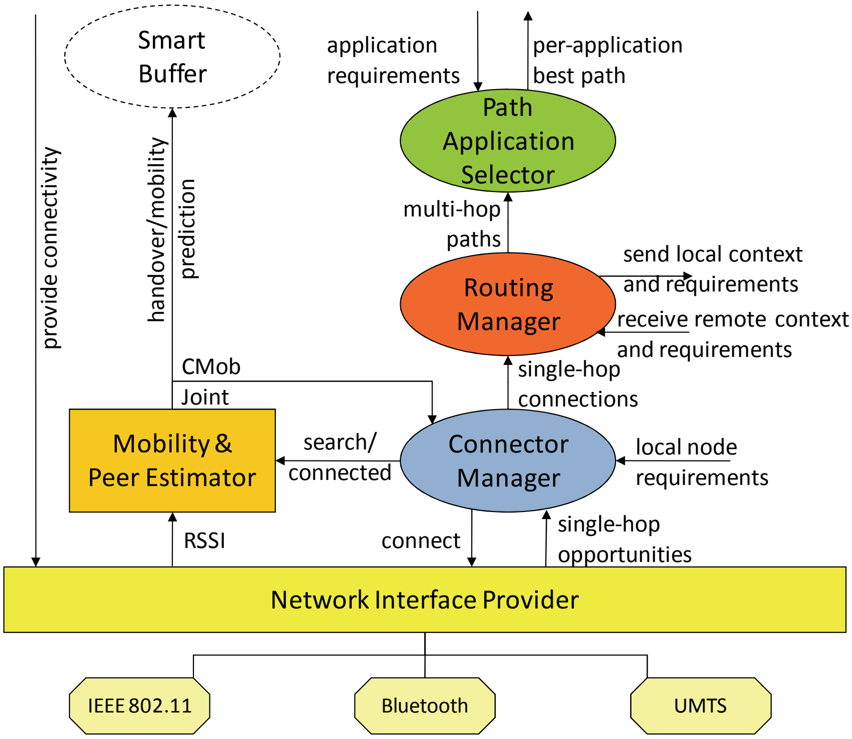

Figure 5.3 MMHC architecture.

5.2 MMHC Evaluation Process

The evaluation process is in charge of gathering context information, evaluating the suitability degree of networking opportunities, and of providing applications with connections they can exploit to access remote resources.

First of all this section shows how it is possible to evaluate networking opportunities based on the two crucial context information we believe a middleware solution for ABS scenario consider, i.e., connectivity durability and throughput (Section 5.2.1). Then, it presents how it is possible to gather these context information via both previous considerations related to multi-hop heterogeneous paths and runtime wireless environment monitoring (Section 5.2.2). Finally, it points out how the context information above are exploited to achieve the most suitable channels and paths (Section 5.2.3).

Let us rapidly note that the section not only provides details related to component implementation, but also performance results achieved deploying and testing our MMHC middleware in simulated environments and actual on-field test-beds.

5.2.1 Adopted Context Information

By looking at the CAMPO area state-of-the-art, it is possible to note that most work proposes elegant but complex models for dynamic selection of network opportunities, without considering practical mobility aspects that can relevantly simplify the management with notable advantages in terms of performance and with limited negative effects on decision optimality.

In fact, based on our on-the-field experience, we claim that two main parameters have a key impact on networking opportunity characteristics: expected durability and throughput, which are specific representatives, respectively, of the general properties of reliability and quality. On the one hand, given that clients and peers are all mobile and may join/leave their networks abruptly, **ABS connectivity durability is far more "fragile" than in traditional single-hop connections** to APs/BSs. As better detailed in the following, our practical experience shows that it is crucial to favor networking opportunities by devices that are relatively slow (not transient) relatively to requesting clients/peers and to consider paths with only very small number of hops. On the other hand, once that durability is potentially ensured, it is reasonable to operate **connectivity management based on coarse-grained estimated throughput**. Throughput has demonstrated to mainly depend, in most practical scenarios, on a few factors that are relatively easy to determine and keep updated, such as number of served clients at any peer and number of path hops (see the following).

Let us rapidly observe that the possibility of exploiting multiple multi-hop multi-path heterogeneous connectivity opportunities has a cost in terms of management complexity and power consumption, e.g., at least the different wireless interfaces at collaborating nodes should be switched on. However, this cost is widely compensated by the major benefit of prompt replacement of paths when they are abruptly lost, which is very frequent in the dynamic environments addressed by our proposal.

5.2.1.1 MMHC Durability

We have experimentally found that it is possible to obtain significant networking management indicators for single-hop path decisions by exploiting only lightweight local monitoring. In particular, we claim that, in first approximation, **single-hop connection durability** depends on **mutual mobility** of connected nodes and **coverage range** of employed wireless technology. These two parameters concisely summarize two main properties affecting reliability in wireless environments: user mobility, as the inclination to either stay close to or move away from nodes offering connectivity, and wireless technology characteristics, e.g., higher durability of medium-range IEEE 802.11 links if compared with short-range Bluetooth ones.

We define mutual mobility as the mobility relationship between a given participating node X and a fixed/mobile connector to X, such as an AP or a collaborating mobile peer. We introduce two indicators: i) **CMob** to measure X's mobility with regard to a fixed connector; ii) **Joint** to evaluate X's tendency to move together with another mobile peer connector (relative stillness). Both indicators are inferred via a simplified technique based on the measurement of RSSI values at X and on their variation in a recent timeframe. We compute CMob and Joint via the linearization in the [0, 1] range of the first harmonic module of the low-pass filtered RSSI sequence, as Section 5.2.2.2 better details.

For each **single-hop** path opportunity, we propose to quantitatively evaluate its **Endurance Estimation** (EE), i.e.:

EE = (1 - CMob) • CR for APs/BSs
EE = Joint • CR for mobile peers

where Coverage Range (CR) is in [0, 1] and, in first approximation, only depends on the exploited wireless technology.

While EE provides a single-hop management information about expected durability, obtained locally without the need of any access to distributed monitoring data, we introduce the Path Mobility (PM) indicator to the purpose of coarse-grained evaluation of multi-hop path durability:

- PM is equal to EE in the case of a single-hop path;
- the PM of a k-hop path is equal to the EE of the k^{th} hop multiplied by the PM of the remaining sub-path starting from the $(k-1)^{th}$ mobile node.

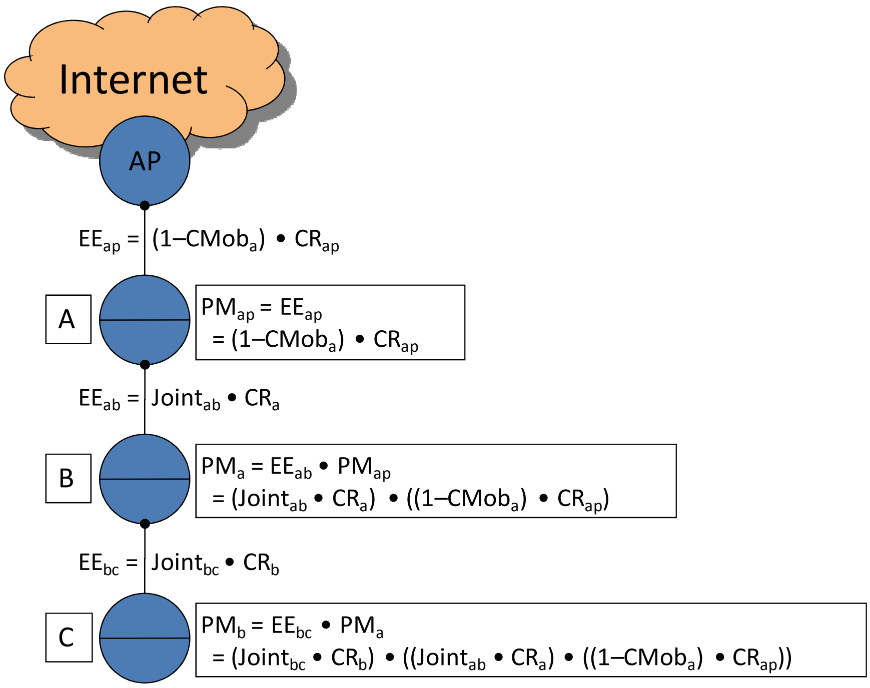

Figure 5.4 Our coarse-grained PM estimation.

Let us observe that PM quickly degrades while increasing path length, to model the desired effect of strongly favoring the selection of short durable paths, as better described in the following.

5.2.1.2 MMHC Throughput

Based on our large campaign of measurements on heterogeneous wireless networks, we have observed that three factors are decisive to determine **path throughput**: i) the **wireless technology** of each single-hop sub-path, ii) the **number of hops** in the path, and iii) the **number of clients/peers** simultaneously served by any path node. Other factors, which have non-negligible effects on path throughput, e.g., node mobility, are not so influential in first approximation. As better detailed in the following, about iii), we have verified that in the challenging case of simultaneous transmit/receive operations by all clients over the same single-hop link up to throughput saturation, competing devices tend to fairly share the total bandwidth.

Just to give some experimental details, to quantitatively evaluate the costs of multi-hop paths, we have tested various ABS environments with different sets of clients and peer connectors. By using iperf on the root peer connector attached to the Internet, we have measured throughput and Round Trip Time (RTT) for single/multi-hop scenarios. Delving into finer details, a single-hop path based on Bluetooth/IEEE 802.11 PROWireless/IEEE 802.11 Orinoco Gold has exhibited average throughput of 68.63/85.22/582.4KB/s and RTT of 31.6/3.9/3.01ms. In a multi-hop heterogeneous scenario (Figure 5.5a), with one hop based on Bluetooth and the other on IEEE 802.11, throughput is 50.75KB/s and RTT is 34.2ms, with negligible variations if the first hop is Bluetooth or IEEE 802.11-based. Note that while RTT is about the sum of each single-hop RTT, overall throughput slightly degrades the performance of the worst hop. That also measures the little additional overhead (which has shown to increase linearly with throughput) imposed by the operating system to forward packets between two heterogeneous interfaces.

Finally, in the case of multiple clients concurrently exploiting the same connector (Figure 5.5b), we have measured good scalability unless the overall bandwidth requested to the connector is below the 75% of its maximum nominal bandwidth. RTT is similar to the single-client single-hop scenario. For instance, for two clients, each one has a throughput of 34.92/40.1/549.12KB/s when exploiting Bluetooth/IEEE 802.11 PROWireless/IEEE 802.11 Orinoco Gold, fairly distributed between the clients (throughput discrepancies between nodes are below 4%).

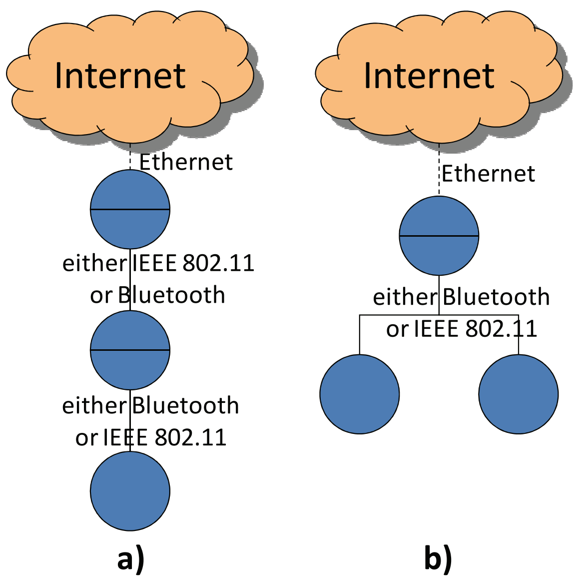

Figure 5.5 Evaluation in different ABS environments: a) serialized connectors in a multi-hop path and b) connector/channel scalability with multiple clients.

These results are a useful feedback on when MMHC should profitably exploit multi-hop connectivity, in particular depending on the number of hops and of already served clients. In particular, we adopt the conservative simplifying assumption that in any case a node can achieve a maximum throughput that is inversely proportional to the number of active nodes on that single-hop link (see Figure 5.6).

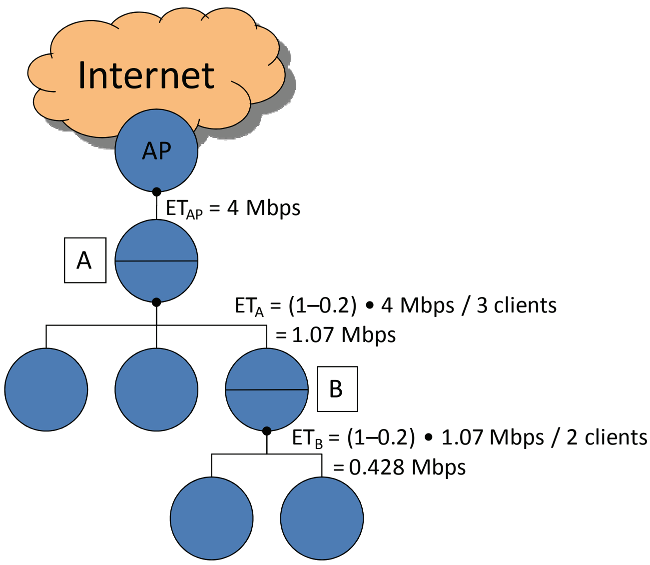

Figure 5.6 Our coarse-grained ET estimation.

Given the above considerations, we propose to adopt a simplified lightweight model to evaluate **Estimated Throughput** (ET):

ET = NB	for APs/BSs
ET = (1 - HD) • MT / #clients	for mobile peers

where Nominal Bandwidth (NB) depends on the exploited wireless technology, e.g., 4 Mbps for IEEE 802.11, Hop Degradation (HD) represents per-hop throughput degradation (we consider an average 20% value) in first approximation independently of the number of local clients, and Maximum Throughput (MT) is the expected maximum throughput toward the traditional Internet, i.e., min {ET of previous single-hop sub-path, NB of the considered single-hop sub-path}. Note that the number of clients is not considered in the case of direct connections to APs/BSs, also given the practical impossibility to portably obtain this information when working with currently deployed AP/BS network equipment. Let us stress again that the proposed procedure for ET estimation is certainly only a rough calculation of actual runtime values, but is extremely simple and lightweight, thus enabling the scarcely intrusive comparison of multi-hop paths (see Section 5.2.3).

Figure 5.7 practically exemplifies how we propose to jointly exploit PM and ET indicators. Both A and B can offer connectivity opportunities that are considered durable (PM=1). Based on ETA and ETB, D selects the two-hop A-AP_1 path, while B chooses the one-hop AP_2 connection for itself and its E/F/G clients. If A moves away, D perceives PMA degradation and switches to B before A disappears, thus limiting user-perceived service degradation. Otherwise, if AP_2 abruptly becomes unavailable, B can re-route its client connections to A. Even if this new path selection certainly limits E/F/G throughput, i.e., from 0.8 to 0.214Mbps, it ensures connectivity maintenance. In short, the spreading of very few and simple management indicators, most of which can be fully computed locally, permits us to reasonably manage and take advantage of ABC opportunities, even if sub-optimally.

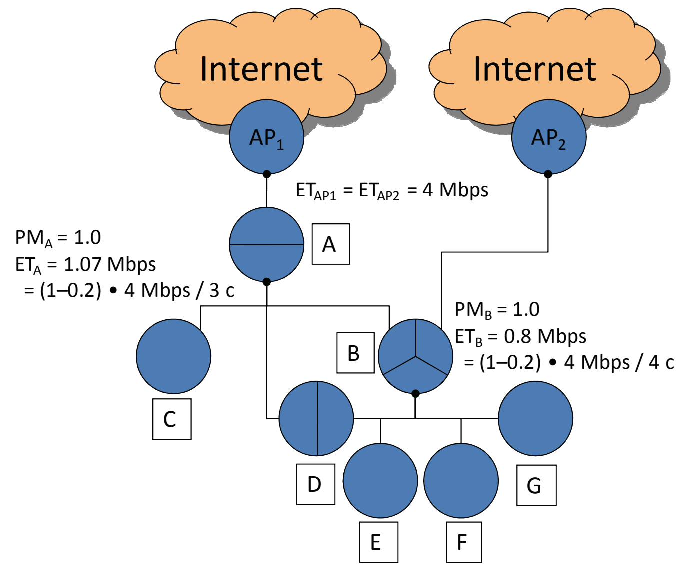

Figure 5.7 Exploiting PM and ET to evaluate MMHC opportunities.

5.2.2 Context Gathering

To correctly estimate each connector and path suitability degree, MMHC has to gather several context data at different levels of abstraction. To that purpose, MMHC requests users and applications to express their requirements related to the whole mobile client. User requirements are assumed not to change too often and may include energy consumption (power saving or maximum performance), maximum affordable cost, and required level of trust. Application requirements are assumed not to change during a service session and may include bandwidth, channel endurance, and other channel related requirements.

Due to the importance we devote to user and connector mobility, we dedicate most space of this sub-section to our novel mechanism able to estimate CMob and Joint required to correctly compute the PM parameter. In addition, we provide experimental results demonstrating the soundness of our solution.

5.2.2.1 Network Interface Provider (NIP) and Mobility & Peer Estimator (MPE)

Network Interface Provider (NIP) is the component in charge of actively interacting with network interfaces. NIP provides upper layers with a transparent access to interface capabilities, by completely hiding low-level details related to underlying interface drivers and operating system. In fact, to simplify interface interaction, NIP offers a uniform API to heterogeneous interfaces while preserving peculiar characteristics an interface could be able to provide, as better detailed in the following.

NIP is structured in two layers: feature and wrapper. At middleware initiation time the feature layer considers the underlying operating system and loads the right wrappers to communicate with interface drivers. In addition, it exposes an API to upper layers to access interfaces without knowledge of low-level and interface-specific implementation details. The wrapper layer is in charge of directly interacting with interface drivers to perform required commands, possibly in an operating system-dependent way. Note that the upper layer is developed once for every interface, while the lower layer once for every exploited operating system. In this manner, NIP facilitates the introduction and exploitation of new interfaces over different operating systems.

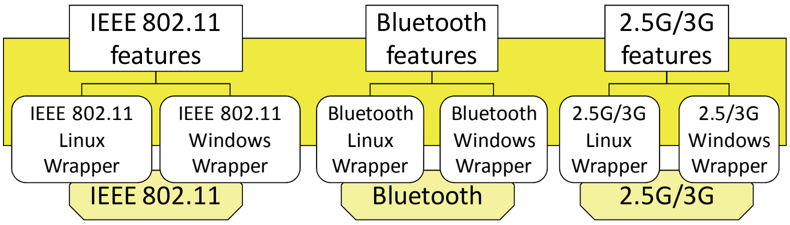

Figure 5.8 Network Interface Provider.

Delving into finer details, the feature component provides a set of capabilities common to any interface:

- **get available single-hop opportunities**, i.e., to obtain the set of 1-hop-distant connectors offering connectivity and the related management information. The function is transparently performed via inquiry procedures for Bluetooth interfaces and scan operations for IEEE 802.11;
- **connect to single-hop devices**, which allows to connect one of the local interfaces to a discovered connector. It is transparently implemented via Personal Area Network (PAN) connections for Bluetooth and associations for IEEE 802.11;
- **provide connectivity**, which allows the node administrator to simply decide whether to behave as a peer connector, offering local connectivity through a given interface, via PAN service for Bluetooth and Independent Basic Service Set (IBSS) for IEEE 802.11.

Not any interface could be able to provide the above features and, in any case, the same feature applied to different interface types could behave in a slightly different manner, depending on the capabilities offered by the underlying wrapper. For instance, peer-to-peer connectivity in Bluetooth could be offered via the Personal Area Network (PAN) service, in IEEE 802.11 by creating a new ad hoc network, while it is not possible for UMTS devices. In addition, some interfaces may provide additional capabilities: for instance, the Bluetooth interface can obtain the set of currently connected remote devices, while IEEE 802.11 can connect to a specific AP (via BSSID identification) and even to a specific target network (via ESSID identification).

The current MMHC prototype supports IEEE 802.11 and Bluetooth interfaces, by including wrappers for both Windows XP/Vista and Linux. The former interface is accessed on Linux mobile clients via the Linux Wireless Extensions, on

Windows XP/Vista mobile clients via the Microsoft Network Driver Interface Specification User-mode I/O (NDISUIO), which is platform-dependent but portable among different wireless interface implementations. For instance, MMHC exploits the NDISUIO function `DeviceIOControl()` to query the `OID_802_11_BSSID_LIST_SCAN` object to retrieve the complete list of currently reachable connectors, either IEEE 802.11 APs or peer nodes in ad hoc configuration. The latter interface is accessed on Linux mobile clients via the standard API provided by the BlueZ protocol stack, on Windows XP/Vista mobile clients via API provided by the Windows Driver Kit and the Software Development Kit tools. For example, MMHC becomes aware of the set of available Bluetooth devices close to a client by invoking `BluetoothFindFirstDevice` and `BluetoothFindNextDevice` functions.

While NIP provides raw information and access to interfaces, **Mobility & Peer Estimator (MPE)** provides context information at a higher abstraction level. It provides a dynamic estimation of CMob, i.e., the client node movement degree, and Joint for each peer connector, i.e., its mobility degree in relation to the mobile client. As already stated, the correct estimation of CMob and Joint parameters is crucial, since they directly affect the PM value. To estimate these values, MPE monitors the execution environment and collects RSSI data about any eligible connector. In addition, based on the monitoring of RSSI values MPE provides handover prediction estimation the SB component exploits to provide continuous connectivity. Since mobility prediction is specifically related to the SB component, we postpone the description of this NIP feature to Section 5.3.

By delving into finer details, for each local wireless interface at client, MMHC determines the list of available connectors and collects RSSI sequences for each connector. Then, for each fixed (mobile) connector CMob (Joint) is set linearly depending on the variability of the RSSI sequence for that connector. To estimate RSSI sequence variability, MPE adopts the processing chain in Figure 5.9a. First of all MPE low-pass filters RSSI fluctuations due to signal noise, in order to identify only RSSI modifications due to actual client node movements. RSSI low-pass filtering is achieved applying the Discrete Fourier Transform (DFT) to 4s-long RSSI sequences (additional details are available in Section 5.3.2) and regenerating the RSSI sequence via the Inverse Discrete Fourier Transform (IDFT) exploiting only the first harmonic, thus discarding high frequency signal components. In particular, when evaluating IEEE 802.11 (Bluetooth) connectors, MPE gathers 4 (1) RSSI values per second, thus applying the DFT to 16 (4) values. We exploit different RSSI sequence lengths since IEEE 802.11 RSSI values show greater noise if compared with Bluetooth ones, thus requiring more aggressive RSSI low-pass filtering. Then, mobility degree indicators are computed via the linearization in the [0, 1] range of the first harmonic module of the low pass filtered RSSI sequence. We have experimentally validated how CMob and Joint depend on RSSI variability and the values used in MMHC are the result of these experimental evaluations. Additional implementation details and performance results are presented in the following.

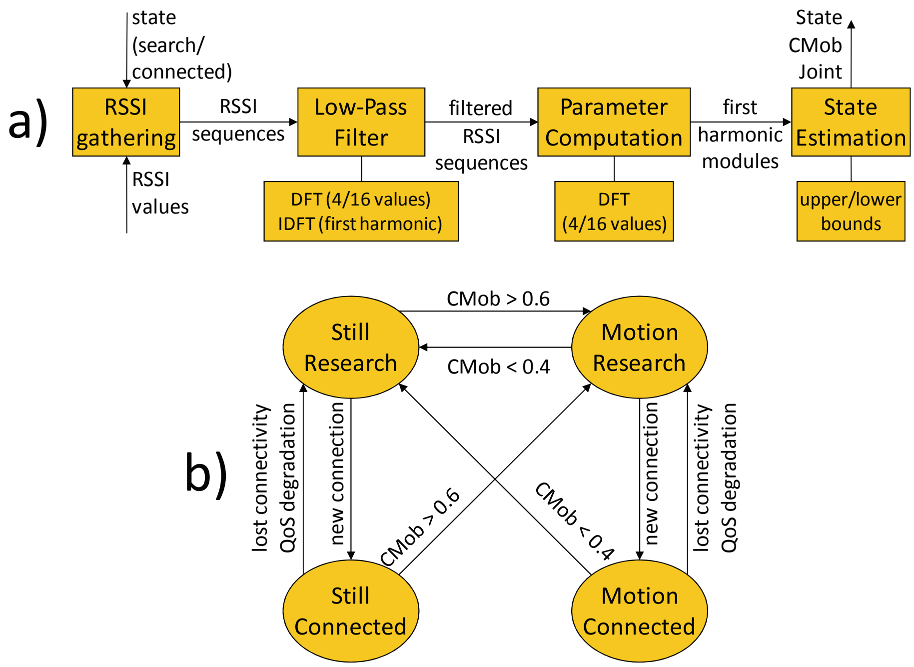

Figure 5.9 Mobility and Peer Estimator: a) processing chain and b) still/motion state diagram.

Let us note that MPE only performs local monitoring at client nodes. In that way, it achieves a twofold benefit. First, MPE exploits only local information that is available despite clients are currently connected to other clients. Then, it does not require any external special-purpose support component, e.g., monitoring components working on the infrastructure side, thus enabling the potentially immediate MMHC adoption in any ABS scenario by only deploying MMHC components at mobile client. However, even the only local monitoring of network interfaces is a power consuming process [Ferro and Potorti 2005]. Therefore, to minimize power consumption, MPE performs "aggressive" context gathering only when required (client in research state), while performing "lazy" monitoring otherwise (client in connected state). In addition, MPE considers even the client mobility state, either still or motion, by performing an aggressive monitoring whenever client state changes because each connector suitability degree may vary dramatically, as better detailed in the following section. To understand whether a client is in still/motion states, MPE exploits CMob monitoring and its time evolution according to the state diagram in Figure 5.9b. MPE switches the client state from still to motion whenever CMob becomes greater than 0.6, while it performs the inverse switch when CMob passes below the 0.4 threshold. The adoption of different thresholds for the two state transitions has been decided to prevent from bouncing effects. In fact, frequent switching between still and motion states would impose repeated perturbations in connector/channel selection, by possibly causing frequent and expensive connector/channel changes and by consequently degrading connection quality.

5.2.2.2 Context Gathering Performance Results

We have performed several experiments to accurately establish effective configurations of MPE and of metric parameters fitting most common deployment scenarios. For instance, only to mention some practical configuration details, our experiments in IEEE 802.11 (Bluetooth) testbed environments suggested us to set CMob (Joint) to 0 (1) when the first harmonic module is <= 0.35 (0.15), to 1 (0) when the module is >= 0.85 (0.60), to a linearly dependent intermediate value otherwise. Additional information about MMHC implementation and the downloadable code of the mobility estimator prototype are available at: http:// lia.deis.unibo.it /Research/MAC/

Here, for the sake of briefness, the main purpose of this section is to point out the robustness of MPE in relation to several different wireless environments. The reported experimental results are mainly focused on the CMob parameter gathered via an IEEE 802.11 simulated environment (simulations permit to achieve several performance results in different deployment environments with easily controllable configurations). However, our experience has demonstrated that very similar performance results may be achieved also in actual IEEE 802.11/Bluetooth environments for both CMob and Joint indicators. In particular, to evaluate the MPE performance, we have defined the following indicators:

*hitRate% = (Correct / Total) * 100*

where Correct is the number of correctly estimated client node mobility states, either still or mobile, and Total is the total amount of sampled states (sample frequency = 1 Hz);

Responsiveness = [$\sum$ (correctly perceived state change time - actual state change time)] / correctly perceived state changes

where Responsiveness models how quickly MPE is able to perceive the state change from still to mobile or vice versa;

longTermHitRate%

the same as hitRate but without considering samples in a 5s-long time window after any mobility state change.

The primary environment characteristics that may affect MPE performance are RSSI noise, strongly dependent on concrete walls disposal and human presence, and client node mobility pattern, e.g., maximum user speed. We have compared MPE performance in a simulated environment with 17 APs deployed in a hexagonal grid, adopting the following parameters: RSSI with a noise standard deviation of 1, 3 or 5 dBs; a waypoint mobility pattern with a speed in the [0.5, 1.5], [1.5, 2.5] or [2.5, 3.5] m/s range.

Table 5.1 MPE performance results.

RSSI Std. Dev. (dB)		1			3			5		
Average Speed (m/s)		1.0	2.0	3.0	1.0	2.0	3.0	1.0	2.0	3.0
Hit Rate (%)		72	73	73	70	73	67	65	61	53
Responsiveness (s)	average	13.5	4.7	4.3	12.8	5.2	5.1	9.6	9.9	9.3
	std. dev.	12.7	1.3	1.9	10.0	3.2	2.9	7.5	6.4	6.0
Long Time Hit Rate (%)		84	99	97	85	96	94	78	74	65

In general, MPE has shown to correctly evaluate client node mobility state; in particular, after the 5-s transition period following still/mobile state change, MPE achieves great performance. As Table 5.1 shows, only imposing very relevant RSSI noise, i.e., with a 5dB standard deviation, the achieved performance starts to decrease because RSSI fluctuations due to signal noise are more frequently evaluated as client movements. Another interesting aspect to underline is that MPE usually behaves better when client node speed is relatively high. In fact, MPE is less effective in recognizing slow movements and, in any case, it requires a non-negligible time interval to recognize mobility state changes. Finally, let us stress that RSSI gathering and CMob/Joint estimation are performed in a completely autonomous and decentralized manner, thus introducing a limited overhead. The reported performance results, coupled with the low overhead imposed, demonstrates the MPE capability to provide mobility-related context information in an effective manner, by actually permitting to compute and exploit channel durability in the evaluation process for mobility-aware always best connectivity.

5.2.3 Metric Application

As already stated, the evaluation process involves multiple steps and considers several entities, i.e., interfaces, connectors, channels, and paths. In particular, Connector Manager (CoM) and Routing Manager (RoM) are the MMHC components that respectively evaluate connectors and channels, thus actively interacting with interfaces and the operating system. In fact, they actually change the client node behavior, e.g., by establishing a channel with a given connector or by changing routing rules from a previous to a novel channel. Instead, Path Application Selector (PAS) simply evaluates RoM-provided paths to estimate which is the most suitable one for a given application. Coupling CoM, RoM and PAS makes possible to provide a flexible, context-aware, and effective evaluation process, by clearly separating connector/channel/path evaluation and system/user/application requirements.

5.2.3.1 Connector Manager, Routing Manager, Path Application Selector

CoM is a crucial component of the MMHC middleware because it directly affects the mobile client channel decisions. In fact, it interacts with the underlying interfaces to change their configuration. Due to the criticality of the actions it performs, CoM cannot be directly set by applications: indeed, applications could be selfish, requiring always as much performance as possible, even if their requirements may affect other applications. For these reasons, CoM provides RoM and applications with a limited set of channel possibilities, i.e., only with the channels suitable for the entire client node with "no risks" for other running applications. While this may decrease the potential capabilities of applications, it ensures the safety of the whole client. In order to correctly estimate whether a connector is suitable for establishing a channel, CoM has to gather and consider many client-related context data, since channel realization may affect the capabilities of the whole mobile client. For instance, preferring Bluetooth connectors could become compulsory in the case of battery shortage, while accessing an untrusted peer connector may affect mobile client security.

In particular, CoM determines the set of single-hop paths to activate based on durability estimation (EE): if several 1-hop-distant devices have EE values compliant with application requirements, the manager prioritizes APs and BSs. In addition, it periodically monitors the activated single-hop connections to check whether they are correctly working; only when one activated path is found disconnected, the manager re-applies the above selection metrics. This reactive approach is motivated by the relatively high overhead imposed by single-hop connection establishment: for instance, remote node discovery, connection, and DHCP configuration require, on average, about 5s for IEEE 802.11 and more than 10s for Bluetooth, as better detailed later.

RoM primarily works to ensure path durability, while throughput is considered only in a second step. It interacts with CoM to get the set of activated single-hop connections. When notified of a single-hop path disruption, it autonomously changes routing rules. In addition, routing rules are updated in an on-demand way anytime a new device becomes available or there is the need for a path renegotiation, e.g., because a path goes below the negotiated thresholds for expected throughput.

At any update, RoM first selects paths with PM equal or greater than 80% of the Required Reliability (RR), i.e., the client-specified preference on desired reliability (RR ranges in the [0,1] interval, with 1 for maximally privileging reliability at the expense of throughput). If at least one path is found, the procedure stops; otherwise, Connector Manager starts examining also paths with PM greater than 50% RR. If no valid path is identified also in this way, in the third phase the manager takes into account any potentially available path. That permits to limit path selection overhead, while taking reasonably consistent decisions.

On the contrary, as shown by the performance results reported in the following, routing management is much less time-consuming than single-hop connection establishment; for this reason, our Routing Manager can be configured to update paths only at connection disruption (minimum intrusion) or also whenever the expected durability/throughput goes out of the allowed, application-specific, variation range (maximum responsiveness).

Finally**, PAS** interacts with the lower layers of our middleware to achieve visibility of the set of available paths. At the beginning of any service session, it works to provide the requesting application/flow with the most suitable path among them depending on application-specific requirements. In other words, PAS operates the path choice based on per-application requirements, while RoM applies per-node requirements to change node routing behaviors.

Note that the monitoring data transferred among participating nodes is limited because restricted only to the events of relevance for the adopted indicators, such as modifications in the number of served clients. That permits to impose a negligible overhead on totally available bandwidth, even in the case of bandwidth-limited Bluetooth links.

Looking at metric application components together, it is possible to do some interesting considerations. Since every channel provided by CoM/RoM is considered suitable for connectivity provisioning, PAS gives the possibility to applications to specify their specific requirements and then accordingly selects the most suitable path. PAS evaluation metric considers the only context information related to the available paths, e.g., path durability and bandwidth, thus relevantly reducing the optimal path selection complexity. Note that PAS scope is rather limited: it cannot either interact with interface or change mobile client routing rules.

In other words, CoM/RoM and PAS behave greatly differently. While the formers interact with interfaces, actively changing their configuration, and operating system, changing routing rules, the latter simply monitors available paths to select the most suitable one for each application separately. PAS does not change the behavior of underlying components; it simply provides each application with the best path considering application specific requirements. Another relevant difference is that while CoM/RoM actively monitor available connectors and determine potential connectors/channels despite application-level connectivity requirements, PAS evaluates provided paths only as consequence of application path requests. Once an application has obtained a path from PAS and started its session, PAS performs neither path monitoring nor connection re-establishment in case of lost path; the application (or the Continuity Manager component on top of PAS) has to explicitly require path re-establishment whenever its path does not fit its requirements anymore.

5.2.3.2 Experimental Evaluation of Cost/Effectiveness of MMHC Metric Application

We have tested our MMHC middleware in different wireless environments, with changing sets of mobile nodes getting/offering connectivity via Bluetooth and IEEE 802.11. The primary goal was to quantitatively evaluate the connection establishment performance and the overhead/delay introduced by our MMHC middleware. The reported results concentrate on channel establishment, path update responsiveness, and path selection; they are average values over hundreds of experiment repetitions over a typical MMHC environment (see Figure 5.10). As already stated, in practically useful self-organizing networks, the number of hops is limited as well as the number of collaborating clients at each topology level; the reported results have been measured by exploiting Linux-based nodes equipped with IEEE 802.11 PROWireless/Orinoco Gold cards and Mopogo Bluetooth 1.2 dongles (the MMHC prototype is also available for MS Windows XP/Vista and includes modules for several interfaces, such as IEEE 802.11 Orinoco Gold cards and all MS-BluetoothAPI-compliant dongles).

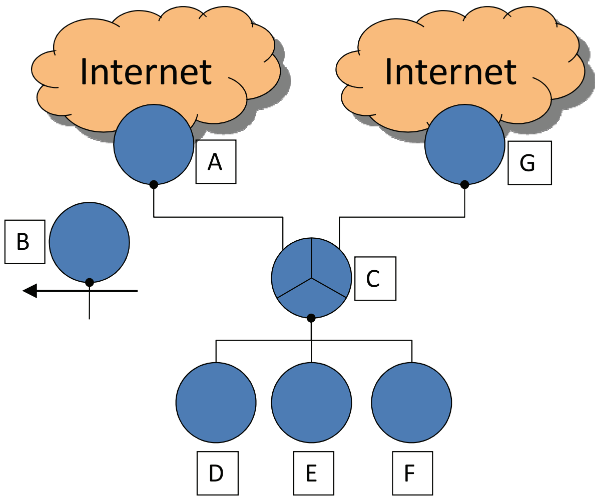

Figure 5.10 Test-bed scenario.

For the sake of simplicity and rapid presentation, let us consider the initial case of three IEEE 802.11 nodes, A (still) and B (moving) that offer connectivity, and C that requires connectivity. At test-bed startup, CoM at C estimates EE_A and EE_B and to consequently select A because of reliability, mainly due to the long time for the needed construction of the time series of RSSI samples. To have a concise evaluation of **CoM** performance when performing new channels, we report as indicator the **time needed to update the set of available channels** when a new connector becomes reachable. In the case of a new Wi-Fi/Bluetooth connector arrival, e.g., node A and C, CoM spends 5.137/22.808s to configure the new channel, due to 3.041/14.370s to discover the connector, 0.039/0.116s to evaluate its suitability, 0.022/3.430s to connect to it via association/PAN connection, and 2.035/3.292s to complete the needed IP configuration via DHCP. The main performance differences between the two interface types have been exhibited for connector discovery and connection: Bluetooth inquiries and PAN connections are slower than IEEE 802.11 scans and associations [Ferro and Potorti 2005]; the longer IEEE 802.11 discovery phase is mainly due to the time needed to set up the ad hoc mode, which is of infrequent usage and not optimized in several Wi-Fi cards. In addition to interface types, the reported performance indicators have demonstrated to highly depend on card model and driver implementations. For instance, in our testbed the IEEE 802.11 ad hoc throughput is much higher for Orinoco Gold than for our PROWireless 3945ABG interfaces (about 6 times) because the latter only support ad hoc transmission at 1MB/s. Similarly, MMHC can halve the Bluetooth inquiry period over MS operating systems at the expense of risking not to sense only a small fraction of connectors as proposed first in [Peterson et al. 2006]; that optimization is impossible with Linux-based BlueZ drivers. In any case, CoM operations for connector evaluation and channel establishment are very time-consuming, especially for Bluetooth; that makes necessary the proactive approach adopted in MMHC, where CoM operates continuously to offer an already determined set of channels to RoM.

Starting from this initial situation, to evaluate middleware responsiveness, consider the case of D, E, and F that join the self-organizing network. The middleware decides to connect them to C, which routes their packets toward A (greater throughput than G). **RoM** at C requires 273ms on average to **establish the new paths** between A and the new clients: 60ms to select the best path and to consequently update routing rules, the remaining part to distribute monitoring data.

About **responsiveness to path disruption**, for instance in the case that A abruptly leaves the network, RoM at C needs 1357ms on average to re-establish a new path from its clients to G. That relatively relevant delay is mainly due to the fact that our CoM does not aggressively monitor all the available paths: to reduce communication overhead, it only pings remote devices with a default (dynamically reconfigurable) period of 1s.

About **responsiveness to the availability of new suitable paths**, if A returns to join the network, the middleware will reconnect C to it again, while E and F will continue to exploit their path to G. On the opposite, D will switch to the newly available path because it requires greater throughput toward C. RoM at C employs 258ms, from path renegotiation request by D to actual path reconfiguration.

Finally, we have evaluated the time needed to **PAS** to re-qualify a path in response to the abrupt disappearance of the associated channel. PAS has shown to require only 0.106/0.228s to re-establish connectivity via another path. Let us stress that the PAS performance is also compatible with seamless soft handover in challenging deployment scenarios, such as for video-on-demand applications, especially if coupled with context-aware pre-fetching techniques [Bellavista et al. 2007].

These results show that the MMHC middleware can effectively manage multi-hop multi-path heterogeneous connectivity opportunities by imposing limited delay, fully compatible with currently available device characteristics and widely within the order of magnitude of usual time intervals that standard IEEE 802.11/Bluetooth requires to simply establish single-hop connections. Finally, let us note that, in response to availability or reliability/quality variations, our MMHC middleware updates the exploited paths by operating local management actions on the limited sub-set of nodes in the locality where variations occur. Localized management operations, which do not propagate with domino effects along the whole paths, favor limited middleware intrusiveness and good scalability while growing the number of hops/nodes.

5.3 Middleware Continuity Management

Service provisioning in ABS scenario must dynamically consider the characteristics of currently served mobile clients, primarily their possible limits on local resources and their high heterogeneity. Limited processing power, memory, and file system make portable wireless devices unsuitable for traditional services designed for fixed networks. These constraints call for both assisting mobile clients in service access and downscaling service contents depending on terminal resource constraints. In addition, as already stated, in ABS scenarios mobile clients exhibit extreme heterogeneity of hardware capabilities, operating systems, installed software, and connectivity technologies. This heterogeneity makes hard to provide all needed service versions with statically tailored contents and calls for on-the-fly adaptation of service contents.

Client resource limits and heterogeneity are particularly crucial when providing continuous services, i.e., applications that distribute time-continuous flows of information to their requesting clients, such as in the case of audio and video streaming [Ramanathan et al. 1999]. Continuous services provisioning in ABS scenario should address several challenging issues, from quality management to runtime personalization of streaming contents. A particularly hard task, especially when associated with the above issues, is to **avoid temporary flow interruptions when clients roam from one connector to another**, also by considering the often strict limits on client memory, which do not allow traditional buffering solutions based on proactive client caching of large chunks of multimedia flows.

We have developed, deployed, and tested an original middleware solution for the provisioning of continuous services to portable devices in ABS scenarios, by locally mediating their access and by dynamically adapting service content to client terminal properties, client location, and runtime resource availability [Stallings 2001, Ramanathan et al. 1999, Saha et al. 2001, Curran and Parr 2003]. Our novel middleware components are dynamically deployed by following client roaming among **wireless localities**, i.e., on the infrastructure-side close to the mobile client, in order to locally assist clients during their service sessions. Client memory limitations suggest having middleware components executing on the fixed network, where and when needed. Instead, mobile devices should only host thin clients, loaded by need and automatically discarded after service and only to support the management of local resources, e.g., performing new channels or providing applications with suitable paths.

Our middleware solution for continuity management is based on the already proposed Secure and Open Mobile Agent (SOMA) [SOMA] proxies and partially deployed on top of our original MMHC middleware, **to support continuous services to mobile clients with strict limits on on-board resources** [Bellavista et al. 2003a, Bellavista and Corradi 2004]. The primary idea is to dynamically deploy **mobile proxies acting on behalf of clients** over the fixed hosts in the network localities that currently offer client connectivity. Mobile proxies hide the complexity of maintaining personalized service sessions (notwithstanding provision-time client roaming) from device clients, which can remain simple and lightweight.

The section focuses on an essential aspect of our middleware: how to avoid interruptions of continuous service provisioning when a client performs and handover from a connector to another at runtime. To achieve this goal, **handover prediction** is crucial. On the one hand, it permits to **migrate mobile proxies in advance** to the wireless localities where mobile clients are going to reconnect, so to have enough time to proactively reorganize user sessions in newly visited localities. On the other hand, service continuity requires maintaining **client-side buffers of proper size** with flow contents to play during the handover process and to reconnect to corresponding nodes, e.g., a Web server providing on-demand audio/video streams, from the new locality. Handover prediction can enable the adaptive management of client buffers, by **increasing buffer size (of the amount**

expectedly needed) only in anticipation of client handovers, thus improving the efficiency of memory utilization, which is essential for portable devices.

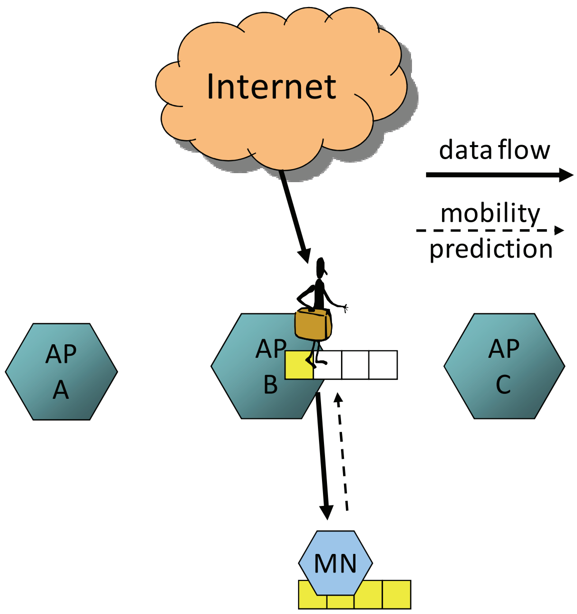

Figure 5.11 Smart Buffer solution: handover prediction triggers buffer resize and mobile-proxy proactive migration.

The rest of the section specifically consider the case of IEEE 802.11 interfaces directly connected to infrastructure connectors and where the firmware triggers the handover procedure with no control capabilities by external components; in addition, the type of handover procedure is hard, i.e., during the handover procedure for a short time interval mobile clients are connected neither to the previous nor to the destination locality. However, the SB solution is suitable for many ABS scenarios with horizontal/vertical handover procedures, e.g., considering a handover from an IEEE 802.11 connector to a Bluetooth one. Furthermore, SB can be effective even in the case of the soft handover supported by the MMHC middleware, i.e., mobile clients access two channels simultaneously while performing a handover. In particular, SB permits to easily ensure seamless handover minimizing service reconfiguration procedures in the new locality, thus maximizing user perceived quality of service. In fact, the proposed adaptive buffering, specifically developed for our mobile proxy-based middleware to avoid streaming interruptions, can help any class of ABS applications that benefit from content prefetching in the client locality. In addition, the predictor runs at the client side, is completely decentralized, and only exploits locally available RSSI monitoring data; RSSI awareness is achieved in a completely portable way over heterogeneous platforms.

In the following we present how our solution performs handover/mobility prediction, by showing that the achieved performance results greatly vary in relation to the adopted low-pass filter to reduce RSSI fluctuation due to signal noise. In addition, we describe performance results of our client- and proxy-side adaptive buffer when adopting the Grey Model low-pass filter, evaluated the most suitable one for handover/mobility prediction purposes.

5.3.1 Handover and Mobility Prediction

The goal of our handover/mobility prediction solution is to provide information about the probability a handover process is going to start and to which locality the involved mobile client is going to connect at. To precisely describe how our handover/mobility prediction mechanism performs, it is first necessary to exactly clarify how communication-level handover works. In fact, the IEEE 802.11 standard does not impose any specific handover strategy: that permits network equipment manufacturers to be free to implement their own handover strategies, as detailed in the following. The different communication-level handover strategies in the market motivate different variants of our handover/mobility prediction mechanism: therefore, we propose and compare two implementations specifically designed for the two most relevant classes of possible handover strategies, i.e., Hard Proactive (HP) and Soft Proactive (SP).

Delving into finer details, cell-based wireless communications can adopt diverse strategies for communication-level handover, which mainly differ in the event used to trigger the handover process. In particular, it is possible to identify two main handover classes: reactive and proactive. **Reactive handover tends to delay handover as much as possible**: handover starts only when wireless clients completely lose their current AP signal. These strategies are effective in minimizing the number of handovers, e.g., by avoiding to trigger a handover procedure when a client approaches a new wireless cell, without losing the origin signal, and immediately returns back to the origin AP. However, reactive handovers tend to be long because they include looking for new APs, choosing one, and asking for re-association only after having lost previous AP signal.

Proactive handover, instead, **tends to trigger handover before the complete loss of origin cell signal**, e.g., when the new cell RSSI overpasses the origin one. In general, these strategies are less effective in reducing the number of useless handovers, but are prompter by performing search operations for new APs before the handover procedure starts. By concentrating on proactive handover, a further classification is possible. On the one hand, **HP** strategies trigger a handover any time the RSSI of a visible AP is greater than the RSSI of the currently associated AP plus an Hysteresis Handover Threshold (HHT); HHT is introduced mainly to prevent heavy bouncing effects. On the other hand, **SP** strategies are "less proactive" in the sense that they trigger handover only if i) the HP condition applies (there is an AP with RSSI greater than current AP RSSI plus HHT), and ii) the current AP RSSI is lower than a Fixed Handover Threshold (FHT).

For instance, the handover strategies implemented by Cisco Aironet 350 and Orinoco Gold Wi-Fi cards follow, respectively, the HP and SP models. More in detail, Cisco Aironet 350 permits to configure its handover with the "Scan for a Better AP" option: if the current AP RSSI is lower than a settable threshold, the Wi-Fi card monitors RSSI data for all visible APs; for sufficiently high threshold values, the Cisco cards be-have according to the HP model. Orinoco Gold cards exactly implements the SP strategy, without giving any possibility to configure the used thresholds.

Our handover/mobility prediction solution is based on a pipelined architecture consisting of two modules. The first one (**Filter**) is in charge of **filtering RSSI sequences to mitigate RSSI fluctuations due to signal noise**. The second module (**Prob**) tries to estimate **the probability a handover happens in the near future (handover prediction)** and **which is the most probable next AP (mobility prediction)** based on RSSI values provided at its input from Filter.

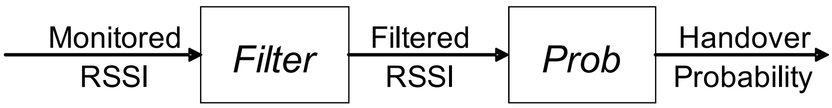

Figure 5.12 The modular Handover Prediction architecture.

The modular architecture of our predictor permits a completely separated implementation and deployment of Filter and Prob, thus simplifying the exploitation and experimentation of different filtering and handover/mobility prediction mechanisms, even dynamically composed at provision time by downloading the needed module code [Bellavista et al. 2003b]. The experimental results in Section 5.3.2 will show the performance of our middleware when the Prob module is fed with either actual RSSI sequences (Filter is the identity function) or filtered RSSI values produced by 4 alternative filters.

In particular, we have implemented two variants of the Prob module, one suitable for communication-level HP handovers and the other for SP ones. We have decided not to work on Prob prototypes for reactive strategies because of two reasons: first, handover prediction is less challenging in the case of reactive handovers than of proactive ones since the triggering of a reactive handover only depends on the RSSI data from one single AP; secondly, reactive communication-level handovers are of minor interest for services with session continuity requirements, given their longer time needed to complete handover.

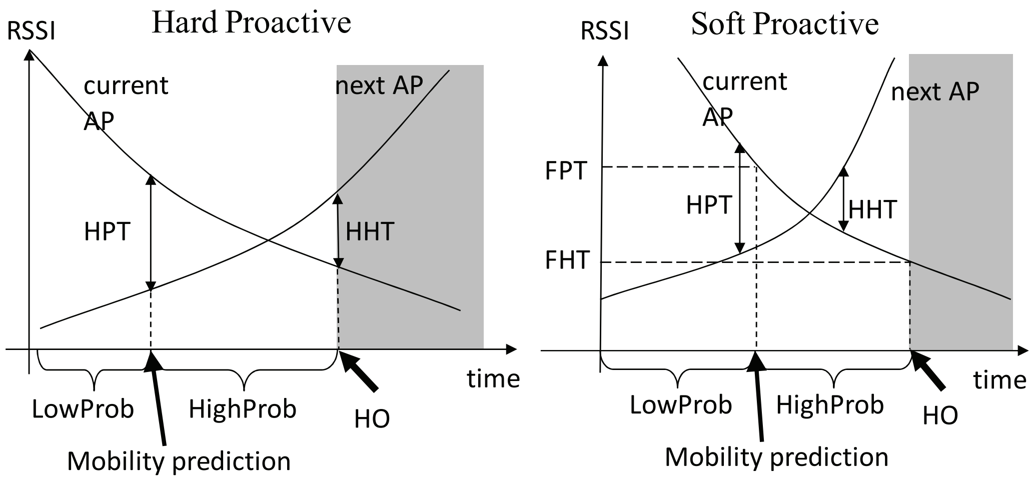

Figure 5.13 The different states of our two Prob variants: HP (left) and SP (right).

The HP-variant of our Prob module is in the state: **LowProb**, if the filtered value for the current AP RSSI is greater than the filtered RSSI values for any visible AP plus a Hysteresis Prediction Threshold (HPT); **HighProb**, otherwise. The SP-variant of the Prob module can assume the following states: LowProb, if the filtered RSSI value for the current AP is greater than either a Fixed Prediction Threshold (FPT) or the filtered RSSI value for any visible AP plus HPT; HighProb, otherwise. Figure 5.13 represents filtered RSSI values for current and next APs, in proximity of a HP (left) and SP (right) handover. A client, moving from the origin AP locality to the destination AP one, is first associated with the origin AP (white background), then with the destination AP (grey background).

Let us rapidly anticipate that the performance of our prediction mechanisms can be quantitatively evaluated in terms of hit rate, efficiency, and stability. Informally speaking, **hit rate** estimates how many actual client handovers are correctly predicted, **efficiency** the capability to predict only client handovers that actually occur, and **stability** the ability to minimize Prob state changes. **Our primary goal is to maximize hit rate** to be able to proactively rearrange the new wireless access localities. In addition, **as a secondary requirement, our prediction mechanism tries to maximize efficiency and stability**. In fact, as better detailed in the following section, both factors affect the overall system performance: depending on the type of provisioned service, handover/mobility predictions can produce service management operations of non-negligible overload, e.g., buffer migration and resource re-binding, and thus it is recommended to reduce useless predictions as much as possible.

5.3.2 RSSI Filtering

RSSI fluctuations due to signal noise significantly affect both stability and efficiency. For instance, in HP communication-level handover, when the RSSI value of the current AP is slightly greater than the sum of another AP RSSI plus HHT, even small RSSI fluctuations can produce several Prob changes, thus relevantly reducing stability. In addition, in those conditions RSSI over/under-estimation may trigger unnecessary predictions, thus lowering efficiency. The section presents 4 different filtering components we have implemented to mitigate RSSI fluctuations: Grey Model, Fourier Transform, Discrete Kalman, and Particle.

Grey Model

We have designed and implemented a first-order Grey Model filtering module that calculates filtered RSSI values on the basis of a finite series of RSSI values monitored in the recent past [Deng 1989]. In particular, given one visible AP and the set of its actual RSSI values measured at the client side $R_0 = \{r_0(1), \ldots, r_0(n)\}$, where $r_0(i)$ is the RSSI value at the discrete time i, it is possible to calculate $R_1 = \{r_1(1), \ldots, r_1(n)\}$, where:

$$r_1(i) = \sum_{j=1}^{i} r_0(j)$$

Then, from the Grey Model discrete differential equation of the first order:

$$\frac{dr_1(i)}{di} + ar_1(i) = u$$

the wireless client can autonomously determine a and u, which are exploited to obtain the predicted RSSI value *pr(i)* at discrete time *i* according to the Grey Model prediction function:

$$pr(i) = \left(r_1(1) - \frac{u}{a} \right) e^{-ai} + \frac{u}{a}$$

Let us observe that filtered RSSI depends on N, the number of actual RSSI values $r_0(i)$ employed in the Grey Model. In principle, greater N, more regular the RSSI filtered values, and slower the filtered RSSI sequence follows the possibly abrupt time evolution of actual RSSI. We have experimentally evaluated the Grey Model performance while varying N. The best tradeoff between RSSI fluctuation mitigation and actual-to-filtered RSSI delay has demonstrated to be for N=15 in most common deployment scenarios. We used that value to obtain the reported experimental results.

Fourier Transform

Our Discrete Fourier Transform (DFT) [Bloomfield 2000] filtering module extract from R_0 a Fourier coefficient set (A_i and B_i) representing the RSSI sequence in the frequency domain in a time window of duration *(R_0 size)*(RSSI sampling period)*, with R_0 as defined in for the Grey Model. The coefficient set is extracted with the usual Fourier equations:

$$A_0 = \tfrac{1}{N} \sum_{n=1}^{N} y(t_n) \qquad B_0 = B_{N/2} = 0 \qquad A_{N/2} = \tfrac{1}{N} \sum_{n=1}^{N} y(t_n) \cos(n\pi)$$

$$A_p = \tfrac{2}{N} \sum_{n=1}^{N} y(t_n) \cos(\tfrac{2\pi pn}{N}) \quad \text{where } p = 1 \ldots N/2 - 1$$

$$B_p = \tfrac{2}{N} \sum_{n=1}^{N} y(t_n) \sin(\tfrac{2\pi pn}{N}) \quad \text{where } p = 1 \ldots N/2 - 1$$

where $t_n = n\Delta t$, $\Delta t = \frac{T}{N}$, $\omega_p t = \frac{2\pi pn}{T}$, $N = R_0\ size$

The Fourier coefficient set is the basis to define an Inverse Discrete Fourier Transform (IDFT) to regenerate the RSSI signal:

$$f(t_n) = \tfrac{1}{2} A_0 + \sum_{p=1}^{M} [A_p \cos(\omega_p t) + B_p \sin(\omega_p t)]$$

When IDFT exploits only a subset of the above series terms, the regenerated RSSI sequence do not exhibit its high frequency components and shows a more regular trend, i.e., IDFT behaves as a low pass filter. We have tested our Fourier Transform filter with several N values and different numbers of addends. We have found a good trade-off between fluctuation mitigation and actual-to-filtered delay with N=4 and M=1.

Discrete Kalman

Our Discrete Kalman filtering module tries to estimate RSSI values by representing the RSSI time evolution as a combination of signal noise (measurement noise) and maximum signal evolving (process noise) [Welch and Bishop 2001]. A linear stochastic equation models the RSSI evolution, with signal/process noise assumed to be independent of each other, white, and with normal probability distribution (standard deviation *Q/R*).

Our filter works by minimizing process noise (w) through a two phase algorithm: first, a predictor performs next RSSI estimation; then, a corrector improves RSSI estimation by exploiting current RSSI measurement. Therefore, an iteration of our Discrete Kalman filtering module processes:

$$\hat{x}^{-}(k) = A\hat{x}(k-1) + w(k-1)$$
$$P^{-}(k) = AP(k-1)A^{T} + Q$$
$$K(k) = P^{-}(k)H^{T}(HP^{-}(k)H^{T} + R)^{-1}$$
$$\hat{x}(k) = \hat{x}^{-}(k) + K(k)(z(k) - H\hat{x}^{-}(k))$$
$$P(k) = (1 - K(k)H)P^{-}(k)$$

where $P_{(k)}$ is the covariance matrix of the state estimate error at step k, with initial value Q, and $K_{(k)}$ is usually indicated as the Kalman Gain in the filtering literature.

In our scenario x and z are RSSI values, the state coincides with the output (A is a 1x1 identity matrix), and the estimation of the next state estimate is equal to the current state (H is a 1x1 identity matrix). After several tests, we have found a good tradeoff between RSSI fluctuation mitigation and filtered-to-actual RSSI delay by setting *Q=1.6* and *R=6*.

Particle Filtering

Like Discrete Kalman, our Particle filtering module tries to estimate RSSI by minimizing measurement and process noise, but without imposing a linear equation modeling and, more important in our deployment scenario, without imposing normal distribution for signal noise [Merwe et al. 2000]. The main idea at its basis is to have an algorithm that computes, at each step, several possible filtered RSSI values for each measured RSSI; then, the filter associates each candidate value with a weight and chooses the most promising values among them when a new measured RSSI is available; finally, it perturbs candidate values, according to the rules shown below, thus obtaining a new filtered RSSI (the average value of the most promising candidates).

To better and practically understand how Particle Filter works, let us show a rapid example of algorithm iteration with 10 particles, which represents 10 possible filtered RSSI values (Figure 5.14):

1. starting step - there are 10 possible filtered RSSI values (light circles), all with the same weight;
2. importance weight step - by exploiting state estimate probability (black curve) obtained from RSSI measurement, the algorithm assigns a weight at each filtered RSSI value (dark circles);
3. re-sampling step - heavy RSSI are spread in different RSSI values, all with the same weight (light circles), while light RSSI values are discarded;
4. sampling/prediction step - filtered RSSI are randomly perturbed (light circles).

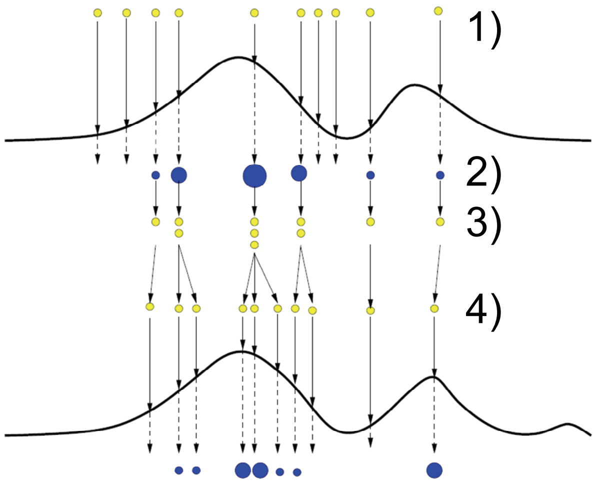

Figure 5.14 An example of Particle iteration.

The number of particles strongly influences the Particle filter performance; in general the greater is the particle number, the better the filtered RSSI follows the actual RSSI sequence. Differently from the previous 3 filters, Particle has a non-negligible computational load, which deeply depends on the particle number. For in-stance, by exploiting an Intel Pentium4 2.80GHz, 1024 MB of RAM machine, one iteration of our Particle filter takes about 132ms with 250 particles and 521ms with 500 particles; in a practical deployment scenario, where multiple APs are simultaneously visible from wireless clients, that time should be multiplied by the number of visible APs, and the computation occurs at any sampling interval. For the sake of completeness, in the following we will report also Particle performance, when used with 250 particles and with the same Q and R are as in Discrete Kalman; however, current client-side resource limitations discourage the exploitation of the Particle filtering module.

Comparison of RSSI Filtering Modules

Just to give a preliminary rough comparison of the behavior of proposed filters, Figure 5.15 reports 60 RSSI samples, either actually measured or filtered according to one of the 4 filtering modules. In particular, the figure points out how much filters are able to mitigate RSSI fluctuations and how fast filtered RSSI sequences follow the actual ones in the case of rapid RSSI evolving, e.g., samples 5 and 45.

Figure 5.15 reports a typical graph of RSSI strong fluctuations (filter = Identity). Compared to actual RSSI, the output of the Grey Model has definitely less fluctuations, but tends to overestimate and to amplify RSSI growth in the case of rapid increasing, for instance when wireless clients are very close to their APs (see samples 10 and 50). Fourier and Kalman have similar behavior: both tend to mitigate RSSI fluctuations quite well and accurately follow the actual RSSI sequence, without overestimating RSSI changes. Particle filter mitigates RSSI fluctuations very well, but sometimes introduces a non-negligible delay between actual and filtered values (see samples 8 and 47).

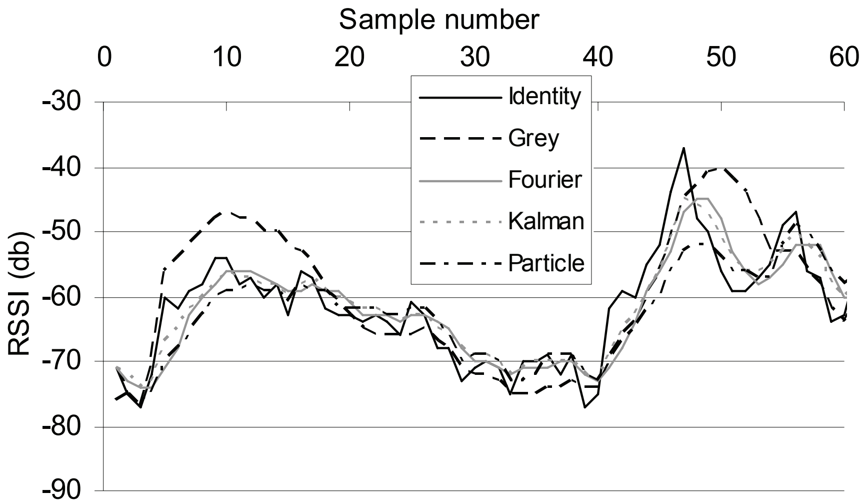

Figure 5.15 Actual and filtered RSSI.

Delving into finer details, in the following we report experimental results about the different hit rate, efficiency, and stability performance achieved when feeding our Prob module either with actual RSSI data or filtered RSSI (exploiting each time one of the 4 proposed filters). Note that the main purpose is to compare performance when adopting different filters in the same deployment scenario. In the following sections we will present other results specifically related adaptive buffering and proactive migration of our Smart Buffer solution when exploiting our handover/mobility prediction mechanisms. Performance results related to handover/mobility prediction hit rate will be similar, even if achieved in slightly different scenarios.

As already stated, the primary goal of RSSI filtering in our proposal is to mitigate RSSI fluctuations due to signal noise in order to primarily improve hit rate, with simultaneous acceptable values for efficiency and stability. However, better a filter mitigates RSSI fluctuations and longer is filtered-to-actual RSSI delay; in the following, for each filter we have adopted the parameter tuning previously described, which achieves a suitable tradeoff between introduced delay and filtering performance.

In particular, in the case of handover prediction we have defined the following performance indicators:

- $hit\ rate = \left(\frac{HPpre}{HPopt} \right) * 100$;

where HPpre is the total time elapsed in HighProb state in the t-long interval before handover and HPopt is the time an optimal predictor should stay in HighProb state (exactly t seconds before each handover). We have chosen t=4s because such a time interval is largely sufficient to perform the needed service management operations in the localities going to be visited [Bellavista et al. 2003b];

- $efficiency = \left(\frac{HPopt}{HPtot} \right) * 100$;

where HPtot is the total time elapsed in HighProb state;

- $stability = \left(\frac{PC}{PCopt} \right) * 100$;

where PC is the number of Prob state changes of our predictor and PCopt the optimal number of Prob state changes.

On the contrary, in the case of mobility prediction, we are interested in evaluating:

- $hit\ rate = \left(\frac{CP}{NH} \right) * 100$

where CP is the number of correctly predicted handovers and NH is the total number of performed handovers;

- $efficiency = \left(\frac{CP}{NP} \right) * 100$

where NP is the total number of triggered predictions.

Note that we do not propose a stability indicator for mobility prediction since our proactive middleware for service management automatically inhibits further mobility predictions for a configurable time interval after a triggered prediction.

Obviously, efficiency and hit rate are strongly correlated: on the one hand, **very good values for hit rate can be achieved with poor efficiency**; on the other hand, it is possible to obtain **very good efficiency by simply delaying as much as possible handover predictions**, with the risk of missing handovers (too low hit rate). Moreover, it is necessary to maintain enough high stability not to continuously perturb the service infrastructure with useless and expensive management operations.

We have measured the five indicators above in a challenging simulated environment, with a large number of Wi-Fi clients roaming among a large number of wireless APs (17 APs regularly deployed in a hexagonal cell topology); RSSI fluctuation has a 3dB standard deviation, FPT=72dB, FHT=80dB, HPT=2dB, HHT=6dB. Mobile clients follow movement trajectories according to the usual Random Waypoint model: speed is in the range [0.6m/s, 1.5m/s] and "thinking time" between 0s and 10s [Hyytiä and Virtamo 2006].

We have compared the behavior of the 4 proposed filtering modules within 6 scenarios (about 200 handovers performed in each one), differentiated for AP to AP distance (20m, 30m or 40m) and type of communication-level handover (HP or SP). For the sake of briefness, we report the results for the most challenging deployment environment with greatest AP density (AP distance=20m); results about the other scenarios are rapidly commented in the following and extensively described at http://lia.deis.unibo.it/ Research/ SOMA/ MobilityPrediction

Figures 5.16 and 5.17 reports average values of performance indicators for handover and mobility prediction, respectively. In general, the adoption of the proposed **filtering techniques significantly improves stability**, with relevant benefits from the point of view of system overhead due to useless Prob state changes. Efficiency also increases when exploiting filtered RSSI, except than in the case of Grey Model: in fact, Grey Model filtering tends to amplify the growth/decreasing trends of actual RSSI values, as observed in the previous section; that produces handover/mobility predictions with a large advance time but also with limited efficiency. Fourier, Kalman, and Particle filters, instead, tend to delay a little more Prob state changes, thus increasing efficiency. Let us observe that the hit rate performance indicator slightly decreases when adopting filtering techniques, except than in the Grey Model case. That effect is tied to what observed before: a slightly greater delay in handover prediction tends to weakly worsen hit rate, but with the relevant advantage of a significantly greater stability.

Similarly to handover prediction, the adoption of **filtering techniques slightly lowers mobility prediction hit rate but**, most important, **increases its efficiency**. Note that the reported efficiency is quite low, but it can be significantly improved with a better tuning of the Prob module configuration parameters e.g., HPT and FPT. However, we specifically concentrate on pointing out the independent contribution on prediction performance of the 4 proposed filters and, for that reason, we have decided to exploit generic Prob settings here.

By taking a look at the whole set of reported performance results, it is clear that no filtering technique always outperforms the others. Filtering exploitation has demonstrated to be crucial for increasing stability; the decision on which filter to exploit depends on the specific goals of the provisioned mobile service. On the one hand, **if it is crucial to achieve very high hit rates, the Grey Model should be considered**. On the other hand, **if prediction costs must be contained (high efficiency), Fourier and Kalman are optimal candidates**. Finally, Particle filter should not be considered. In fact, if compared with other filtering techniques, the Particle filter not only imposes great computational cost (as already detailed), but it also provides neither high hit rate nor high stability.

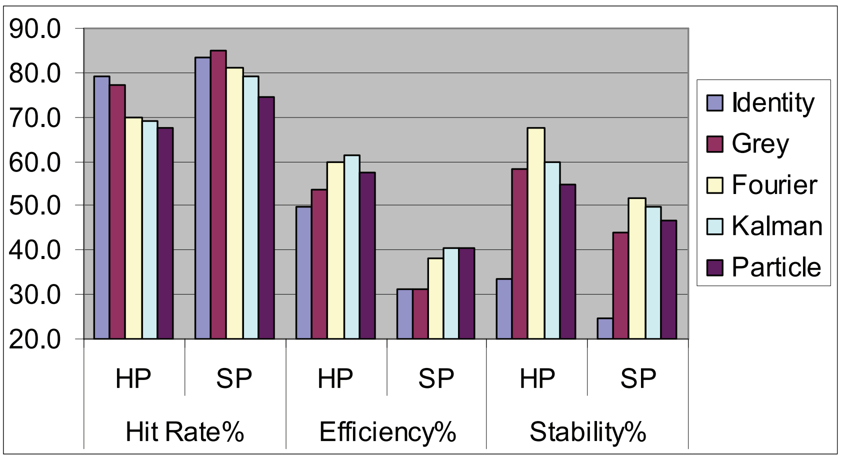

Figure 5.16 Handover prediction performance.

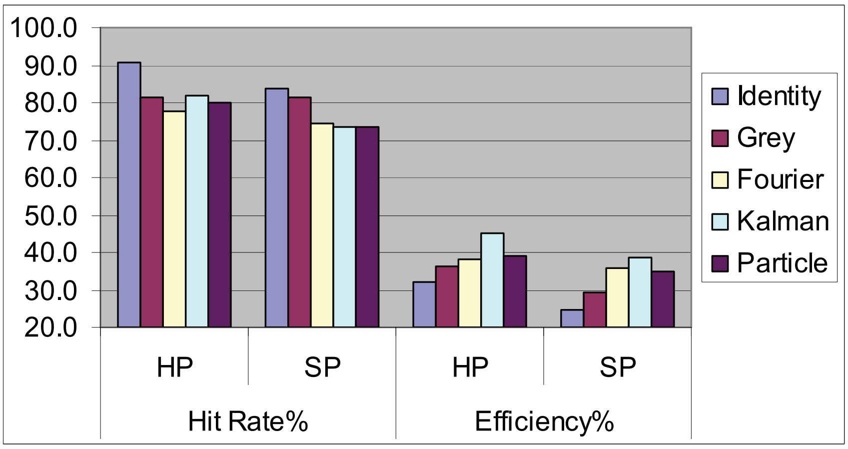

Figure 5.17 Mobility prediction performance.

Let us stress that, thanks to the modular architecture of our middleware, it is possible to **choose the exploited filtering module at provisioning time, by adapting middleware behavior to the current service context**. For instance, let us consider the case that our middleware exploits handover prediction to support the pre-fetching of data chunks before handover, in order to continuously provide service flows also during AP changes with temporary disconnections [Bellavista et al. 2003b]. If the crucial point is to avoid temporary service interruptions, the best choice is Grey Model filtering; otherwise, if the priority is to minimize useless exploitation of client-side buffers, either Fourier or Kalman should be chosen. Finally, as already stated, Particle should be excluded from possible choices because of its high client-side computational load.

In addition to the proposed dense deployment scenario, we have evaluated our filtering modules in environments with greater AP-to-AP distance (30m/40m). The primary difference is a uniform increase in hit rate: when clients perform their handovers, APs are more distant and RSSI values exhibit a slower time evolution, thus facilitating handover/mobility prediction. That confirms the suitability of presenting only the most challenging deployment scenario: the trend of performance indicators is the same, with no tight dependence on AP density.

In the following we will specifically consider only the Grey Model, primarily to take advantage of its better hit rate. However, the same client- and proxy-side solutions can be exploited even with the Kalman and Fourier filters, if there is the need to push the tradeoff toward better efficiency and stability.

5.3.3 Client-side Smart Buffer

The goal of our handover prediction-based client-side buffer management is to have **client buffers of the maximum size and full exactly when re-associations to the destination APs occur**. Wrong handover predictions produce incorrect dimensioning of client-side buffers; correct but late handover predictions cause correctly-sized buffers that are not fulfilled with the needed pre-fetched streaming data.

Just to give a rough idea of the magnitude order of the advance time needed in handover prediction, let us briefly consider the example of a client receiving a multimedia stream played at 1000Kbps constant bit-rate and a handover procedure taking 1.5s to complete. That time interval includes the time for communication-level handover and the time to locally reconnect to the migrated companion proxy, and largely overestimates the actual time measured in [Bellavista and Corradi 2004]. In this case, the client-side buffer size must be at least 187.5KB. If the client available bandwidth is 1500Kbps on average, the buffer fills with a speed of 500Kbps on average, by becoming full (from an empty state) in about 3s. Therefore, in the worst case, our handover prediction should be capable of anticipating the actual handover of 3s, to trigger buffer pre-fetching in time.

Besides the Filter and Prob modules, we have developed a Dim one. The Dim module determines the correct buffer size to enforce, depending on handover probability. By delving into finer details about the already implemented predictor modules available in our middleware, Prob can assume three different states:

- LowProb, if handover is considered highly improbable in the near future;
- HighProb, if handover is considered almost certain in the near future;
- MedProb, otherwise.

Dim exploits the state delivered by Prob to dynamically modify the size of associated client-side buffers (note for client-side adaptive buffering we exploit a three-degree Prob module). In the current implementation, when in the HighProb state, Dim sets the buffer size at the maximum for that multimedia flow (flow bit-rate * 1.5s); when in LowProb, Dim sets the size at the minimum (maximum/10); and when in MedProb, it sets the size at (maximum+minimum)/2. We are currently evaluating more complex processing functions for Prob and Dim modules (e.g., with finer granularity for the discrete states of handover probability and buffer size) that could improve our middleware performance; first results encourage to exploit simple and lightweight module functions, which can achieve the needed performance results with a limited computational overhead.

To quantitatively evaluate the effectiveness of the proposed modular handover predictor and of its application in our adaptive buffer management infrastructure, we have identified some performance indicators and measured them both in a simulated environment, with a large number of Wi-Fi clients roaming among a large number of wireless AP localities, and in our campus deployment scenario, where four laptops move among the different coverage areas of six APs. Two laptops are Linux-based, while the other two host Microsoft Windows.NET; they alternatively exploit Cisco Aironet 350 (HP handover) and Orinoco Gold (SP handover) IEEE802.11 cards. In addition, we have compared the performance of our HP/SP Prob modules when connected to either our GM(1,1) Filter function or an identity Filter function that provides output values identical to its input: the goal was of understanding the isolated contribution of the GM(1,1) Filter function to the overall performance of our adaptive buffer management infrastructure.

In particular, we have considered the following performance indicators:

- *Average Buffer Size (AvBS)* = $\frac{1}{T}\int_0^T BS(t)dt$

where BS(t) is the time-varying buffer size. In other words, AvBS is the time-weighted average of the buffer size;

- *Average Buffer Duration (ABD)* = $\frac{1}{T}\int_0^T BD(t)dt$

where BD(t) is the time-varying validity of a chosen buffer size. In other words, ABD is the average time interval between two successive operations of buffer re-sizing;

- *Successful Handover ($SH_{\%}$)* = $\left(1-\frac{DH}{NH}\right)*100$

where DH is the number of actual client handovers and NH is the number of handovers predicted by the proposed HP/SP predictors.

In general, the goal of an optimal buffer management solution is to contemporarily achieve minimum values for AvBS and sufficiently large ABD values, with maximum $SH_{\%}$. Obviously, AvBS and $SH_{\%}$ are strongly correlated: on the one hand, very good values for $SH_{\%}$ can be easily achieved with large AvBS values;

on the other hand, it is possible to obtain very low AvBS values by simply delaying as much as possible the buffer size enlargement, but with the risk of streaming interruptions (too low $SH_{\%}$ value). Moreover, it is necessary to maintain sufficiently large ABD values not to continuously perform useless and expensive buffer re-size operations.

We have measured the three indicators above in a challenging simulated environment where 17 APs are regularly placed in a 62m x 84m area and RSSI fluctuation has a 3dB standard deviation. Wireless clients follow trajectories with a randomly variable speed and with a randomly variable direction (with a Gaussian component for the standard deviation of Π/6). The speed is between 0.6m/s and 1.5m/s to mimic the behavior of walking mobile users; FST = 66dB; FIT = 70dB; FHT = 80dB; HST = 10dB; HIT = 6dB; HHT = 6dB; note that in this case we exploit Fixed/Hysteresis Superior/Inferior Threshold (FST, FIT, HST, HIT) instead of only FPT and HPT to get a three-degree Prob module. On the average, each wireless client has the visibility of ten APs at the same time, which represents a worst case scenario significantly more complex than the actually deployed Wi-Fi networks (where no more than five APs are usually visible at any time and from any client position).

Table 5.2 reports the average results for the three performance indicators over a large set of simulations, each one with about 500 handovers. For the video streaming exploited in all the experiments, the buffer size required to avoid interruptions in the case of static fixed dimensioning is 200KB. The most important result is that any proposed Prob module, when provided with either GM-filtered RSSI values or actual RSSI values, significantly reduces AvBS (between 27.5% and 33.5%), thus relevantly improving the client memory utilization. In addition, Prob modules fed with GM-filtered RSSI values largely outperform the cases with actual RSSI values, especially with regard to the ABD performance indicator. In fact, even if AvBS has demonstrated to maintain good values in all cases, directly monitored non-filtered RSSI (with its more abrupt fluctuations) tends to trigger a higher number of useless handover predictions and, consequently, more useless modifications in the enforced buffer size.

Table 5.2 Performance indicators for HP and SP predictors.
In the case of static fixed buffer: AvBS=200KB, $SH_{\%}$=100, and ABD=∞.

Handover Strategy	**Filter Function**	**AvBS (KB)**	**$SH_{\%}$**	**ABD (s)**
HP	Identity	140	92.1	2.80
	GM(1,1)	133	92.8	5.20
SP	Identity	145	91.6	2.79
	GM(1,1)	138	97.5	5.66

Figure 5.18 points out the correlation between GM-filtered RSSI and buffer size. In particular, it depicts the time evolution of buffer size (dotted line) depending on both GM-filtered RSSI of the currently associated AP (grey line) and the greatest RRSI among the non-associated visible APs (black line). Let us stress that when the currently associated AP RSSI is significantly greater than RSSI from other APs, our buffer management infrastructure maintains buffer size at its minimum (20KB in the example); when the RSSI of the currently associated AP, instead, is similar to the RSSI of another AP, buffer size increases at its maximum (200KB); otherwise, our infrastructure works to manage a medium-sized buffer (110KB).

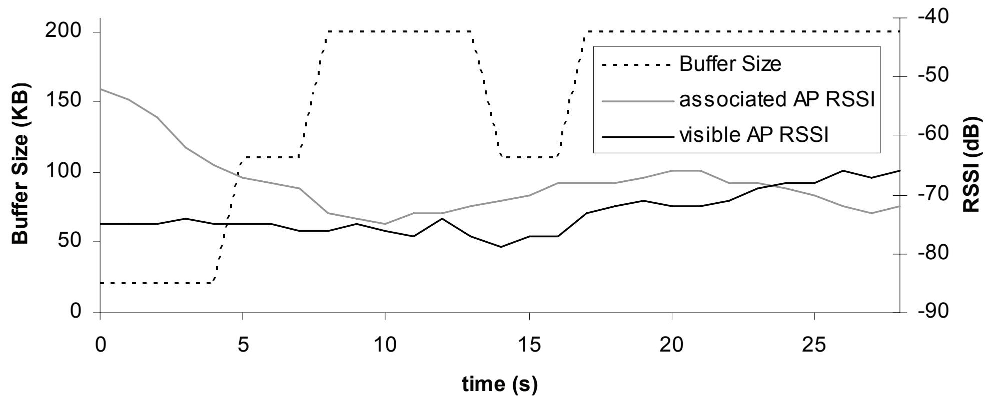

Figure 5.18 Buffer size variations depending on time evolution of GM-filtered RSSI values for the currently associated AP and for another AP in visibility.

In addition to simulations, we have evaluated the performance of HP/SP Prob modules also by using a service prototype, built on top of our middleware, and by moving four client laptops among the campus localities during streaming provisioning. Even if the number of considered in-the-field handovers is largely lower than the simulated one (thus, less relevant from the statistical point of view), in-the-field performance results confirm the simulation-based ones. In particular, the prototype-based AvBS, $SH_{\%}$, and ABD results have demonstrated to be better than simulation-based ones, on the average, also due to the lower number of considered APs, and the consequently simpler handover prediction. However, we have experienced a significant degradation of prototype-based performance indicators in the case of extreme RSSI fluctuations, e.g., when a client follows a trajectory in strict proximity of relevant obstacles, such as the reinforced concrete walls of our campus buildings.

5.3.4 MA-based Smart Buffer

The ultimate goal of our prediction-based infrastructure-side buffer management is to proactively **migrate a proxy to the next client access locality before the actual client handover**; pre-fetched data in proxy **buffers should grow only immediately before starting migration**, so to minimize buffer/bandwidth consumption, as better detailed in the following.

Similarly to the previous section, we point out a simple example to quantitatively estimate the proxy buffer size needed in the addressed multimedia scenario. Let us consider the simple case of a client receiving a multimedia stream played at 1.0Mbps constant bitrate, a client-to-proxy bandwidth of 1.5Mbps, and resource rebinding operations after proxy migration taking 2s (rebinding interval includes the time for server reconnection and for client-specific service personalization). After handover, the proxy buffer already available at the new access locality should be at least 2s*1.0Mbps=250KB. Let us note that if handover prediction is too anticipated, migrated proxy buffers become obsolete and useless. Therefore, it is crucial to migrate proxies only when needed and to overestimate buffer size with regards to the minimum 250KB. In the following, we will consider a maximum buffer size of 800KB, corresponding to 6.4s of pre-fetched streaming content consumed at 1.0Mbps. In the case the average useful bandwidth between wired hosts is 6Mbps (successive wireless access localities are close), the movement of a full buffer proxy takes about 1.5s, approximately the same time interval needed for completing communication-level handover in most common Wi-Fi equipment [Velayos and Karlsson 2003].

Our adaptive buffering imposes buffer size to be usually low (200 KB) to save memory at the proxy host and to avoid useless network overhead; in fact, when clients do not change their APs, the buffering goal is only to smooth possible server-to-proxy bandwidth fluctuations. When the predictor notifies a proxy that its associated client is going to change its wireless cell, the proxy sets buffer size to maximum (800 KB), waits for buffer fulfillment, and then commands the migration of its clone, with the fulfilled buffer, to the predicted location. If the client disassociates from the origin AP before buffer is full, the proxy immediately sends its clone to the predicted location with the already buffered data. After clone migration, the proxy in the origin locality sets buffer size again to minimum and continue serving its client until it leaves the cell. If client entrance in the predicted cell occurs too late with regards to clone migration, part of the migrated buffer becomes obsolete. For this reason, in the case of client not arrived yet, the middleware automatically re-sends an updated buffer to an already predicted location after a time interval equal to buffer duration – buffer fulfillment – proxy migration + communication handover (6.4-1.92-1.5+1.5=4.48s in the above scenario).

To thoroughly and quantitatively evaluate the effectiveness of our proactive buffer management solution, we have identified some performance indicators and measured them in the same simulated and test-bed environments presented in the previous section. In particular, besides the already presented AvBS and $SH_{\%}$, we have considered the following performance indicators:

- *Useful Buffered Data after handover (UBD)*, the available useful streaming data buffered at the proxy clone when the client associates with the new wireless cell after an handover;
- *Waiting for Service after handover (WfS)*, the average time between client handover completion and the start of proxy-to-client data streaming in the new wireless cell.

In general, the goal of an optimal buffer management solution is, at the same time, to minimize AvBS and WfS and to maximize $SH_{\%}$, by maintaining a sufficiently high value for UBD.

We have measured the four indicators in a challenging simulated environment where 17 APs are regularly placed in a 62m x 84m area and RSSI fluctuation has a 3db standard deviation. Wireless clients follow trajectories with a randomly variable speed and with a randomly variable direction (with a Gaussian component for the standard deviation of Π/6). The speed is between 0.2m/s and 1.2m/s to mimic the behavior of walking mobile users; FPT=72db; FHT=80db; HPT=3db; HHT=6db. On the average, each wireless client has the visibility of 10 APs at the same time, which represents a worst case scenario significantly more complex than actually deployed Wi-Fi networks (where no more than 5 APs are usually visible at any time and from any client position).

Table 5.3 reports the average results for the five performance indicators over several simulations, each simulation with about 500 handovers. The most important result is that **both HP and SP predictors significantly reduce AvBS** (55% and 50%) if compared with the case of a statically dimensioned non-adaptive buffer (AvBS=800KB). This relevantly improves the memory utilization at the proxy host and reduces the network traffic due to useless pre-fetching. In addition, both predictors achieve a good value for $SH_{\%}$, thus pointing out the satisfying performance of the GM-based handover prediction.

Client streaming players overcome cell handover with no streaming interruptions if the proxy buffer has enough useful data (not obsolete because already sent to clients by origin proxies) to fill the time interval between the end of the client-side buffer and the completion of proxy-based session re-establishment in destination cells. Since in the worst case the rebinding process lasts 2s and client-sided buffers run out during communication-level handover, UBD should be greater than 1Mbps*2s=250KB. **UBD has demonstrated to be much greater than that threshold** with both the proposed predictors.

Finally, the table reports experimental results for WfS in two different conditions, with and without successful prediction. In the case of correct handover prediction, a proxy with useful pre-fetched data is ready to start streaming provisioning to its client just at the completion of communication-level handover; only the time to locally re-establish the client-to-proxy connection is to be waited (about 0.2s). On the contrary, in the case of unsuccessful prediction (wrongly predicted cell or insufficient pre-fetched data at the proxy due to anticipated migration/late buffer fulfillment) a client has to discover the proxy unavailability/unsuitability, to request a new proxy/buffer, to wait for proxy/data movement, and finally to wait for service re-binding (more than 4s).

Table 5.3 Adaptive buffering performance results when using either the HP predictor or the SP one.

Predictor	AvBS (KB)	$SH_{\%}$	WfS (s)		UBD (KB)
HP	360	83.3	0.17	4.64	499.2
SP	400	88.0	0.24	4.06	540.8

The code of the handover prediction prototype, additional details about prototype implementation, and further simulation/prototype-based experimental results are available at http://lia.deis.unibo.it/Research/SOMA/SmartBuffer/

5.4 Summary of Contributions

Our research work shows the suitability of novel middleware that performs interface/connector/channel evaluation and dynamic management, by considering not only traditional monitoring parameters but also more expressive context metadata related to each client and running application. MMHC demonstrates the feasibility of the approach, with effective performance and limited costs, thanks to proper tradeoffs between responsiveness, visibility, and introduced overhead, facilitated by the clean separation into physical/logical layers.

Several proposals have recently investigated some specific partial aspects of the above ABS scenario characterized by multi-hop multi-path heterogeneous connectivity. For instance, [Faccin et al. 2006] points out the primary technical aspects of WLAN-based multi-hop networks, while [Le and Hossain 2007] aims to extend cellular network capabilities via relay stations, with the main goal of increasing cellular coverage. [Pack et al. 2007] and [Lam and Liew 2007], instead, specifically address the issue of managing client mobility among heterogeneous multi-hop networks. These contributions were crucial for the full understanding of both the theory and the main characteristics of multi-hop networks. However, they did not focus on realistic, feasible, and practical solutions to guide the design and implementation of prototypes for seamless, mobility-aware, and self-organizing networks. [Conti and Giordano 2007a] and [Conti and Giordano 2007b] provide a relevant contribution by identifying major drawbacks and weaknesses of theoretical work in the literature; however, they do not propose practical solutions for these weaknesses. By looking at the state-of-the-art with a wider and more general perspective, it is possible to note that **most work proposes elegant but complex models for ABS**, without considering practical mobility aspects that can relevantly simplify MMHC management with notable advantages in terms of performance and with limited negative effects on decision optimality.

We claim that only novel mobility-aware middleware can seamlessly enable the ABS scenario, by effectively considering a limited set of practical indicators for a coarse-grained estimation of expected reliability/quality of multi-hop paths available at runtime. To that purpose, we have developed our innovative MMHC middleware that manages the **durability/throughput-aware formation and selection of different multi-hop paths simultaneously**, based on practical lightweight indicators on node mobility and wireless network characteristics. The reported results show that our middleware, notwithstanding its simplifying estimation assumptions and its application-layer approach, can effectively take **mobility-aware connectivity management decisions with limited overhead**.

The exploitation of mobile middleware proxies that work over the fixed network on behalf of their resource-constrained clients is demonstrating its suitability and effectiveness in the ABS scenario, especially when associated with handover prediction. Handover prediction can enable the proactive performing of service/middleware management operations to maintain session continuity in the provisioning of personalized services, independently of runtime client roaming. In particular, handover prediction can help in realizing **novel adaptive buffering solutions that optimize client- and proxy-side buffer size and pre-fetching depending on the expected handover probability**. The work of design, implementation, and experimental evaluation of our solution prototype has shown that our prediction-based client- and proxy-sided adaptive buffering can preserve streaming continuity with limited requirements on wireless device memory capabilities. In addition, our buffering solution, specifically developed for mobile proxy-based middleware for multimedia streaming, has a general applicability to any class of ABS services that can potentially benefit from service content pre-fetching close to the client terminal access localities.

Chapter 6 – Conclusive Remarks

The spread of powerful mobile clients equipped with multiple context sources and heterogeneous communication interfaces pushes for novel solutions providing users with the capability to access local and remote resources everywhere. In particular, we propose the novel and challenging ABS scenario characterized by mobile clients able to exploit multiple and heterogeneous context sources and networking opportunities simultaneously.

The heterogeneity deriving from the plethora of context information and wireless technologies risks to slow down the development and deployment of applications actually taking full advantage of all available ABS opportunities. For this reason we propose to adopt a middleware-based solution that autonomously monitors and controls underlying low-level components in a context-aware way. Such a middleware should be based on a translucent access to low-level components, thus providing both a direct but homogeneous access to components and a middleware-mediated control of their behavior. In addition, it should fully exploit all capabilities local communication interfaces provide, i.e., not only getting but also providing connectivity in a peer-to-peer fashion. Finally, it should not hide user mobility but consider it as crucial context information to properly control mobile client behavior.

After an in-depth analysis of the state-of-the-art in the wide area of context-aware autonomic management of preferred network opportunity, we have developed a novel CAMPO model proposing a common terminology to simplify the description of novel solutions and proposed a novel taxonomy able to clearly and precisely position already proposed solutions. The analysis of CAMPO solutions, and specifically of the ones related to the ABS scenario, has suggested us to focus our research on three points: a) context-aware evaluation process, to correctly estimate the suitability degree of available networking opportunities, b) deployment scenarios characterized by both infrastructure and ad hoc connectivity, and c) the development of open and decentralized solutions supporting service continuity.

We have proposed the PoSIM middleware and the MMHC solution to facilitate the management of heterogeneous context sources and communication interfaces and to support the easy development and deployment of novel ABS applications.

PoSIM provides the context-aware integration of positioning systems; MMHC supports the context-aware management of local communication interfaces and the provisioning of continuous services without interruption during horizontal/vertical handovers.

PoSIM is based on the novel design rules of differentiated access to low-level details and differentiated control of low-level components. On the one hand, PoSIM provides a homogeneous access to low-level components, dynamically retrieving low-level details even if not known at middleware development/deployment time and providing the capability to directly control integrated positioning system behavior from the application layer. On the other hand, PoSIM supports the mediated control of positioning systems by providing the capability to control positioning systems by only activating pre-defined policies.

Our MMHC solution follows the originally proposed design rules of tradeoff between local and global management, between single- and multi-path granularity, and between static and dynamic responsiveness. MMHC exploits novel context information related to user mobility to self-organize networks based on multi-hop multi-path heterogeneous connectivity. On the one hand, it exploits mobile client and connector mobility degree to dynamically evaluate available networking opportunities and provide applications with most suitable paths in relation to their requirements. On the other hand, it exploits user mobility prediction to trigger client- and infrastructure-side proactive buffer-size adaptation and component migration/reconfiguration.

Experiments based on both simulated and test-bed environments show that the MMHC middleware manages multi-hop multi-path heterogeneous connectivity in an effective way. MMHC introduces only an overhead greatly lower than time intervals imposed by most spread wireless technologies, i.e., IEEE 802.11 and Bluetooth, to establish single-hop connections. Furthermore, the proposed solution for service continuity can not only perform handover prediction in a lightweight, scalable, and completely decentralized manner, but also adapt its behavior at provisioning time depending on service/system requirements, thus ensuring minimal intrusiveness and good stability. Achieved experimental results demonstrate our solution effectively supports service continuity minimizing user perceived service interruption during mobile client handover procedures.

The encouraging results obtained by the design and implementation of our middleware prototypes are further stimulating our on-going research activities. In particular, we are currently evaluating the MMHC performance over a wide deployment scenario with dozens of Wi-Fi/Bluetooth infrastructure-based and peer connectors, to validate our middleware capability to support continuous services. In addition, we are extending the current MMHC prototype to include the support to additional interfaces such as UMTS and Wi-Max. Finally, we want to enhance information dissemination over self-organizing networks to support not only the access to Internet, but also to provide the offer, discovery and invocation of services provided in a peer-to-peer way. This implies not only the dissemination of information to the most suitable sub-set of network nodes, thus achieving a trade-off among full information dissemination and update costs, but also the capability to reconfigure active connections when a peer that offers currently accessed services moves from a (sub-)network to another.

Publications

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Mobility Prediction for Mobile Agent-based Service Continuity in the Wireless Internet*. MATA International Workshop on Mobility Aware Technologies and Applications, Florianòpolis, Brasil, October 2004.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Adaptive Buffering based on Handoff Prediction for Wireless Internet Continuous Services*. The 2005 International Conference on High Performance Computing and Communications (HPCC-05), Sorrento, Italy, September 2005.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Efficiently Managing Location Information with Privacy Requirements in Wi-Fi Networks: a Middleware Approach*. 2nd International Symposium of Wireless Communication Systems 2005 (ISWCS 2005), Siena, Italy, September 2005.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Mobile Proxies for Proactive Buffering in Wireless Internet Multimedia Streaming*. SIUMI International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet, Ohio, USA, June 2005.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Coupling Transparency and Visibility: a Translucent Middleware Approach for Positioning System Integration and Management (PoSIM)*. 3rd International Symposium of Wireless Communication Systems 2006 (ISWCS 2006), Valencia, Spain, September 2006.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Enhancing JSR-179 for Positioning System Integration and Management*. 1st Workshop on Distributed Agent-based Retrieval Tools (DART'06), Pula-Cagliari, Sardinia, Italy, June 2006.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Evaluating Filtering Strategies for Decentralized Handover Prediction in the Wireless Internet*. 11th IEEE Symposium on Computers and Communications (ISCC'06), Pula-Cagliari, Sardinia, Italy, June 2006.

- Paolo Mistroni, Carlo Giannelli, Paolo Bellavista, Letizia Ghirardello: *LIFE.net over Web: an Advanced Monitoring Protocol for UPS Systems*. 29th International Telecommunications Energy Conference (Intelec 2007), Rome, Italy, September 2007.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Mobility-Aware Connectivity for Seamless Multimedia Delivery in the Heterogeneous Wireless Internet*. 2nd Workshop on multiMedia Applications over Wireless Networks (MediaWiN 2007), Aveiro, Portugal, July 2007 (Invited Paper).

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *A Layered Infrastructure for Mobility-Aware Best Connectivity in the Heterogeneous Wireless Internet*. 1st International Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2008), Innsbruck, Austria, February 2008.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *The PoSIM Middleware for Translucent and Context-aware Integrated Management of Heterogeneous Positioning Systems*. To be published in Computer Communications, special issue on Advanced Location-Based Services, Elsevier, Spring 2008.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Resource and Service Discovery*. To be published as invited chapter in the book "Mobile Agents in Networking and Distributed Computing", Jiannong Cao, Sajal K. Das (eds), Wiley Series in Agent Technology, April 2008.

- Paolo Bellavista, Antonio Corradi, Carlo Giannelli: *Mobility-aware Connectivity Management in Always Best Served Wireless Scenarios*. To be published as invited paper in the special issue of ACM/MONET, Winter 2008.

Bibliography

3GPP 2002. Feasibility Study on 3GPP System to WLAN Interworking. Tech. rep. 3GPP TR 22.934 v1.2.0.

Adamopoulou, E., Demestichas, K., Koutsorodi, A., Theologou, M. 2005. Intelligent Access Network Selection in Heterogeneous Networks - Simulation Results. In Proceedings of the 2nd International Symposium on Wireless Communication Systems, Siena, Italy, September 2005, 279- 283.

Agre, J., Akenyemi, D., L. Ji, Masuoka, R., Thakkar, P. 2002, A Layered Architecture for Location-based Services in Wireless Ad Hoc Networks. IEEE Aerospace Conf., Big Sky, Montana, USA, Mar. 2002.

Ahmavaara, K., Haverinen, H., Pichna, R. 2003. Interworking architecture between 3GPP and WLAN systems. IEEE Communications Magazine 41, 11, 74- 81.

Ahmed, T., Kyamakya, K., Ludwig, M. 2006. Architecture of a Context-Aware Vertical Handover Decision Model and Its Performance Analysis for GPRS - WiFi Handover. In Proceedings of the 11th IEEE Symposium on Computers and Communications, Pula, Italy, June 2006, 795- 801.

Akyildiz, I.F., Mcnair, J., Ho, J.S.M., Uzunalioglu, H., Wenye Wang 1999. Mobility management in next-generation wireless systems. Proceedings of the IEEE 87, 8, 1347-1384.

Akyildiz, I.F., Jiang Xie, Mohanty, S. 2004. A survey of mobility management in next-generation all-IP-based wireless systems. IEEE Wireless Communications 11, 4, 16- 28.

Akyildiz, I.F., Mohanty, S., Jiang Xie 2005. A ubiquitous mobile communication architecture for next-generation heterogeneous wireless systems. IEEE Communications Magazine 43, 6, S29- S36.

Anastasi, G., Bandelloni, R., Conti, M., Delmastro, F., Gregori, E., Mainetto, G. 2003. Experimenting an indoor bluetooth-based positioning service. Int. Conf. on Distributed Computing Systems Workshops (ICDCSW'03), Providence, Rhode Island, USA, May 2003.

Bahl, P., Padmanabhan, V.N., 2000. RADAR: an in-building RF-based user location and tracking system. 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000), Vol. 2, pp. 775-784, Mar. 2000.

Balasubramaniam, S., Indulska, J. 2004. Vertical Handover Supporting Pervasive Computing in Future Wireless Networks. Computer Communications (Elsevier) 27, 8, 708-719.

Banerjee, N., Wei Wu, Das, S.K. 2003. Mobility support in wireless Internet. IEEE Wireless Communications 10, 5, 54- 61.

Bari, F., Leung, V.C.M. 2007. Automated network selection in a heterogeneous wireless network environment. IEEE Network 21, 1, 34-40.

Bellavista, P., Corradi, A., Stefanelli, C. 2003a. Application-level QoS Control and Adaptation for Video on Demand. IEEE Internet Computing, Vol. 7, No. 6, Nov.-Dec. 2003.

Bellavista, P., Corradi, A., Montanari, R., Stefanelli, C. 2003b. Context-aware Middleware for Resource Management in the Wireless Internet. IEEE Trans. on Software Engineering, Vol. 29, No. 12, Dec. 2003.

Bellavista, P., Corradi, A., 2004. A QoS Management Middleware based on Mobility Prediction for Multimedia Service Continuity in the Wireless Internet. IEEE Int. Symp. on Computers and Communications (ISCC), July 2004.

Bellavista, P., Corradi, A., Giannelli, C. 2005a. Mobile proxies for proactive buffering in wireless Internet multimedia streaming. In Proocedings of 25th IEEE International Conference on Distributed Computing Systems Workshops, Columbus, OH, June 2005, 297-304.

Bellavista, P., Corradi, A., Giannelli, C. 2005b. Adaptive Buffering based on Handoff Prediction for Wireless Internet Continuous Services. In High Performance Computing and Communications (Sorrento, Italy), Dongarra, J., Yang, L.T., Rana, O.F., Di Martino, B., Eds. Lecture Notes in Computer Science, vol. 3726. Springer-Verlag, Berlin, Germany, 1021- 1032.

Bellavista, P., Corradi, A., Montanari, R., Toninelli, A. 2006. Context-aware semantic discovery for next generation mobile systems. IEEE Communications Magazine 44, 9, 62-71.

Bellavista, P., Corradi, A., Foschini, L. 2007. Context-Aware Handoff Middleware for Transparent Service Continuity in Wireless Networks, Elsevier Pervasive and Mobile Computing Journal, Vol. 3, No. 4, Aug. 2007, pp. 439-466.

Bernaschi, M., Cacace, F., Iannello, G., Za, S., Pescape, A. 2005. Seamless internetworking of WLANs and cellular networks: architecture and performance issues in a Mobile IPv6 scenario. IEEE Wireless Communications 12, 3, 73-80.

Berthold, O., Federrath, H., Kohntopp, M. 2000. Project ”Anonymity and Unobservability in the Internet”. Int. Workshop on Freedom and Privacy by Design, Canada, Apr. 2000.

Bloomfield, P. 2000. Fourier Analysis of Time Series: An Introduction. John Wiley & Sons, 2nd edition, 2000.

Bluetooth Special Interest Group, "Bluetooth Specification Version 2.0 + EDR", www.bluetooth.org, Nov. 2004.

Buddhikot, M.M., Chandranmenon, G., Seungjae Han, Yui-Wah Lee, Miller, S., Salgarelli, L. 2003. Design and implementation of a WLAN/cdma2000 interworking architecture. IEEE Communications Magazine 41, 11, 90-100.

Cavalcanti, D., Agrawal, D., Cordeiro, C., Bin Xie, Kumar, A. 2005. Issues in integrating cellular networks WLANs, AND MANETs: a futuristic heterogeneous wireless network. IEEE Wireless Communications 12, 3, 30-41.

Chebrolu, K., Rao, R.R. 2006. Bandwidth aggregation for real-time applications in heterogeneous wireless networks. IEEE Transactions on Mobile Computing 5, 4, 388-403.

Chen, Y., Chen, X.Y., Rao, F.Y., Yu, X.L., Li, Y., Liu D. 2004. LORE: An infrastructure to support location-aware services. IBM Journal of Research & Development, Vol. 48, No 5/6, Sept./Nov. 2004.

Chen, G., Kotz, D. 2000. A survey of context-aware mobile computing research. Dartmouth College Technical Report TR2000-381, http://www.cs.dartmouth.edu/reports/

Cheng Wei Lee, Li Ming Chen, Meng Chang Chen, Yeali Sunny Sun 2005. A framework of handoffs in wireless overlay networks based on mobile IPv6. IEEE Journal on Selected Areas in Communications 23, 11, 2118-2128.

Chuanxiong Guo, Zihua Guo, Qian Zhang, Wenwu Zhu 2004. A Seamless and Proactive End-to-End Mobility Solution for Roaming Across Heterogeneous Wireless Networks. IEEE Journal on selected areas in Communications 22, 5, 834-848.

Chunyan Fu, Khendek, F., Glitho, R. 2006. Signaling for multimedia conferencing in 4G: the case of integrated 3G/MANETs. IEEE Communications Magazine 44, 8, 90- 99.

Conti, M., Giordano, S. 2007a. Multihop Ad Hoc Networking: the Theory. IEEE Communications Magazine, vol.45, no.4, pp.78-86, Apr. 2007.

Conti, M., Giordano, S. 2007b. Multihop Ad Hoc Networking: the Reality. IEEE Communications Magazine, vol.45, no.4, pp.88-95, Apr. 2007.

Coulouris, G., Naguib, H., Samugalingam, K. 2002. FLAME: An Open Framework for Location-Aware Systems. Int. Conf. on Ubiquitous Computing, Goteborg, Sweden, Sept. Oct. 2002.

Curran, K., Parr, G. 2003. A Middleware Architecture for Streaming Media over IP Networks to Mobile Devices. IEEE Int. Conf. Wireless Communications and Networking (WCNC), Mar. 2003.

Deng, J.L. 1989. Introduction to Grey Theory. The Journal of Grey System, Vol. 1, No. 1, 1989.

Di Flora, C., Ficco, M., Russo, S., Vecchio, V. 2005. Indoor and outdoor location based services for portable wireless devices. Int. Conf. on Distributed Computing Systems Workshops (SIUMI), Columbus, Ohio, USA, Jun. 2005.

Dingledine, R., Mathewson, N., Syverson, P. 2004. Tor: The Second-Generation Onion Router. 13th USENIX Security Symposium, Aug. 2004.

Ekahau Positioning Engine, www.ekahau.com

ETSI 2001. Requirements and Architectures for Interworking between HIPERLAN/3 and 3rd Generation Cellular Systems. Tech. rep. ETSI TR 101 957.

Faccin, S.M., Wijting, C., Kenckt, J., Damle, A. 2006. Mesh WLAN Networks: Concept and System Design. IEEE Wireless Communications, vol.13, no.2, pp. 10-17, Apr. 2006.

Ferreira, L., Serrador, A., Correia, L.M. 2005. Simultaneous Use in Mobile Communications, In Proceedings of the 61st IEEE Vehicular Technology Conference, Stockholm, Sweden, May-June 2005, 1468-1472.

Ferro, E., Potorti, F. 2005. Bluetooth and Wi-Fi Wireless Protocols: a Survey and a Comparison. IEEE Wireless Communications 12, 1, 12-26.

Fodor, G., Eriksson, A., Tuoriniemi, A. 2003. Providing quality of service in always best connected networks. IEEE Communications Magazine 41, 7, 154-163.

Forgy, C.L. 1982. Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem. Artificial Intelligence, Vol. 19, No. 1, Sept. 1982, pp. 17-37.

Frattasi, S., Fathi, H., Gimmler, A., Fitzek, F.H.P., Prasad, R. 2006. Designing socially robust 4G wireless services. IEEE Technology and Society Magazine 25, 2, 51-64.

Gazis, V., Alonistioti, N., Merakos, L. 2005. Toward a generic "always best connected" capability in integrated WLAN/UMTS cellular mobile networks (and beyond). IEEE Wireless Communications 12, 3, 20-29.

Genco, A. 2005. Three Step Bluetooth Positioning. International Workshop on Location- and Context-Awareness (LoCA 2005), Munich, Germany, May 2005.

Ghini, V., Salomoni, P., Pau, G. 2005. Always-best-served music distribution for nomadic users over heterogeneous networks. IEEE Communications Magazine 43, 5, 69-74.

Graumann, D., Lara, W., Hightower, J., Borriello, G. 2003. Real-world Implementation of the Location Stack: The Universal Location Framework. IEEE Work. on Mobile Computing Systems and Applications, Monterey, CA, USA, Oct. 2003.

Gustafsson, E., Jonsson, A. 2003. Always Best Connected. IEEE Wireless Communications 10, 1, 49-55.

Hasswa, A., Nasser, N., Hossanein, H. 2005. Generic vertical handoff decision function for heterogeneous wireless. In Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks, Dubai, United Arab Emirates UAE, March 2005, 239-243.

Hightower, J., Borriello, G. 2001a. Location systems for ubiquitous computing. IEEE Computer, Vol. 34, No. 8.

Hightower, J., Borriello, G. 2001b. Location sensing techniques. UW CSE Technical Report.

Hightower, J., Brumitt, B., Borriello, G. 2002. The Location Stack: A Layered Model for Location in Ubiquitous Computing. IEEE Work. on Mobile Computing Systems and Applications, Callicoon, NY, USA, Jun. 2002.

Hongyang Bing, Chen He, Lingge Jiang 2003. Performance analysis of vertical handover in a UMTS-WLAN integrated network. In Proceedings of the 14th IEEE Personal, Indoor and Mobile Radio Communications, Beijing, China, September 2003, 187-191.

Hou, J., O'brien, D.C. 2006. Vertical handover-decision-making algorithm using fuzzy logic for the integrated Radio-and-OW system. IEEE Transactions on Wireless Communications 5, 1, 176- 185.

Hosokawa, Y., Takahashi, N., Taga, H. 2004. A System Architecture for Seamless Navigation. Int. Conf. on Distributed Computing Systems Workshops (MDC), Tokyo, Japan, Mar. 2004.

Hui Luo, Zhimei Jiang, Byoung-Jo Kim, Shankaranarayanan, N.K., Henry, P. 2003. Integrating wireless LAN and cellular data for the enterprise. IEEE Internet Computing 7, 2, 25-33.

Hung-Yu Wei, Gitlin, R.D. 2004. Two-hop-relay architecture for next-generation WWAN/WLAN integration. IEEE Wireless Communications 11, 2, 24-30.

Hyytiä, E., Virtamo, J. 2006. Random Waypoint Model in Cellular Networks. Wireless Networks, 2006.

IEEE 802.11e 2005. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. IEEE Std 802.11e-2005.

Inoue, M., Mahmud, K., Murakami, H., Hasegawa, M., Morikawa, H. 2004. Novel out-of-band signaling for seamless interworking between heterogeneous networks. IEEE Wireless Communications 11, 2, 56-63.

JavaComm. Java Communications 3.0 API, http://java.sun.com/products/ javacomm/

Jess, http://herzberg.ca.sandia.gov/jess/

Johnson, D.B., Perkins, C. 2001. Route optimization in mobile IP. Internet draft draft-ietf-mobileip-optim-11.

JSR-179 Location API for J2ME, http://www.jcp.org/ aboutJava/ communityprocess/final/jsr179/index.html

Jun-Zhao Sun, Riekki, J., Sauvola, J., Jurmu, M. 2004. Towards Connectivity Management Adaptability: Context Awareness in Policy Representation and End-to-end Evaluation Algorithm. In Proceedings of the 3rd international conference on mobile and ubiquitous multimedia, College Park, MD, October 2004, 85-92.

Jun-Zhao Sun, Riekki, J., Jurmu, M., Sauvola, J. 2005. Adaptive connectivity management middleware for heterogeneous wireless networks. IEEE Wireless Communications 12, 6, 18-25.

Jyh-Cheng Chen, Hong-Wei Lin 2005. A gateway approach to mobility integration of GPRS and wireless LANs. IEEE Wireless Communications 12, 2, 86-95.

Kappler, C., Poyhonen, P., Johnsson, M., Schmid, S. 2007. Dynamic network composition for beyond 3G networks: a 3GPP viewpoint. IEEE Network 21, 1, 47-52.

Karetsos, G.T., Kyriazakos, S.A., Groustiotis, E., Di Giandomenico, F., Mura, I. 2005. A hierarchical radio resource management framework for integrating WLANs in cellular networking environments. IEEE Wireless Communications 12, 6, 11-17.

Kibria, M.R., Jamalipour, A. 2007. On designing issues of the next generation mobile network. IEEE Network 21, 1, 6-13.

Koien, G.M., Haslestad, T. 2003. Security aspects of 3G-WLAN interworking. IEEE Communications Magazine 41, 11, 82-88.

Kristiansson, J, Parnes, P. 2006. An Application-Layer Approach to Seamless Mobile Multimedia Communication. IEEE eTransactions on Network and Service Management 3, 1.

Lam, P.P., Liew, S.C. 2007. Nested Network Mobility on the Multihop Cellular Network. IEEE Communications Magazine, vol.45, no.9, pp.100-104, Sept. 2007.

Lampropoulos, G., Passas, N., Merakos, L., Kaloxylos, A. 2005. Handover Management Architectures in Integrated WLAN/Cellular Networks. IEEE Communications Surveys & Tutorials 7, 4, 30-44.

Le, L., Hossain, E. 2007. Multihop Cellular Networks: Potential Gains, Research Challenges, and a Resource Allocation Framework. IEEE Communications Magazine, vol.45, no.9, pp.66-73, Sept. 2007.

Lera, A., Molinaro, A. Polito, S., Ruggeri, G. 2005. End-to-end QoS provisioning in 4G with mobile hotspots. IEEE Network 19, 5, 26-34.

Li Ma, Fei Yu, Leung, V.C.M., Randhawa, T. 2004. A new method to support UMTS/WLAN vertical handover using SCTP. IEEE Wireless Communications 11, 4, 44-51.

Luan Huang, Yajian Liu, Thilakawardana, S., Tafazolli, R. 2006. Network-centric user assignment in the next generation mobile networks. IEEE Communications Letters 10, 12, 822-824.

Luo, H., Ramjee, R., Sinha, P., Li, L.E., Lu S. 2003. UCAN: a unified cellular and ad-hoc network architecture. In Proceedings of the 9th annual international conference on Mobile computing and networking, San Diego, CA, September 2003, 353-367.

Marques, V., Costa, X.P., Aguiar, R.L., Liebsch, M., Duarte, A.M.O. 2005. Evaluation of a mobile IPv6-based architecture supporting user mobility QoS and

AAAC in heterogeneous networks. IEEE Journal on Selected Areas in Communications 23, 11, 2138-2151.

McNair, J., Fang Zhu 2004. Vertical Handoffs in Fourth-Generation Multinetwork Environments. IEEE Wireless Communications 11, 3, 8-15.

McNeef, J.G. 2002. The global positioning system. IEEE Transactions on Microwave Theory and Techniques, Vol.50, Iss.3, Mar 2002, pp. 645-652.

van der Merwe, R., Doucet, A., de Freitas, J.F.G., Wan, E. 2000. The Unscented Particle Filter. Advances in Neural Information Processing Systems (NIPS13), Dec. 2000.

Minghui Shi, Xuemin Shen, Mark, J.W. 2004. IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks. IEEE Wireless Communications 11, 4, 66-75.

Minji Nam, Nakjung Choi, Yongho Seok, Yanghee Choi 2004. WISE: energy-efficient interface selection on vertical handoff between 3G networks and WLANs. In Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Barcelona, Spain, September 2004, 692-698.

Mohanty, S., Akyildiz, I.F. 2006. A Cross-Layer (Layer 2 + 3) Handoff Management Protocol for Next-Generation Wireless Systems. IEEE Transactions on Mobile Computing 5, 10, 1347-1360.

Nasser, N., Hasswa, A., Hassanein, H. 2006. Handoffs in fourth generation heterogeneous networks. IEEE Communications Magazine 44, 10, 96-103.

Niyato, D., Hossain, E. 2005. Call admission control for QoS provisioning in 4G wireless networks: issues and approaches. IEEE Network 19, 5, 5-11.

Nord, J., Synnes, K., Parnes, P. 2002. An Architecture for Location Aware Applications. 35th Hawaii Int. Conf. on System Sciences, Hawaii, USA, Jan. 2002.

Nursimloo, D.S., Chan, H.A. 2005. Integrating fast mobile IPv6 and SIP in 4G network for real-time mobility. In Proceedings of the 13th IEEE International Conference on Networks, Kuala Lumpur, Malaysia, November 2005, 917-922.

Ormond, O., Perry, P., Murphy, J. 2005. Network selection decision in wireless heterogeneous networks. In Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Berlin, Germany, September 2005, 2680-2684.

Pack, S., Shen, X., Mark, J.W., Pan, P.J. 2007. Mobility Management in Mobile Hotspots with Heterogeneous Multihop Wireless Links. IEEE Communications Magazine, vol.45, no.9, pp.106-112, Sept. 2007.

Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Makela, J.P., Pichna, R., Vallstron, J. 2000. Handoff in Hybrid Mobile Data Networks. IEEE Personal Communications 7, 2, 34-47.

Perkins, C. 2002. IP Mobility Support for IPv4. Internet proposed standard RFC3344.

Peterson, B.S., Baldwin, R.O., Kharoufeh, J.P. 2006. Bluetooth Inquiry Time Characterization and Selection. IEEE Transactions on Mobile Computing, Vol. 5, No. 9, Sep. 2006, pp. 1173-1187.

Politis, C., Chew, K.A., Akhtar, N., Georgiades, M., Tafazolli, R., Dagiuklas, T. 2004. Hybrid multilayer mobility management with AAA context transfer capabilities for all-IP networks. IEEE Wireless Communications 11, 4, 76-88.

PoSIM Web site, http://lia.deis.unibo.it/Research/PoSIM

Qi Wang, Ali Abu-Rgheff, M. 2006. Mobility management architectures based on joint mobile IP and SIP protocols. IEEE Wireless Communications 13, 6, 68-76.

Qian Zhang, Chuanxiong Guo, Zihua Guo, Wenwu Zhu 2003. Efficient mobility management for vertical handoff between WWAN and WLAN. IEEE Communications Magazine 41, 11, 102-108.

Qingyang Song, Jamalipour, A. 2005. Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques. IEEE Wireless Communications 12, 3, 42-48.

Ramamurthy, B. Feng, H., Datta, D. , Heritage, J.P., Murkjerjee, B. 1999. Transparent vs. Opaque vs. Translucent, Wavelength-Routed Optical Networks. OFC 1999 Tech. Dig., Washington, DC, Mar. 1999, vol. 1, pp. 59–61.

Ramanathan, P., Sivalingam, K.M., Agrawal, P., Kishore, S. 1999. Dynamic Resource Allocation Schemes during Handoff for Mobile Multimedia Wireless Networks. IEEE Journal on Selected Areas in Communications, Vol. 17, No. 7, July 1999.

Ranganathan, A., Al-Muhtadi, J., Chetan, S., Campbell, R., Mickunas, M.D. 2004. MiddleWhere: A Middleware for Location Awareness in Ubiquitous Computing Applications, ACM/IFIP/USENIX Int. Conf. on Middleware, Oct. 2004, Toronto, Ontario, Canada.

Reinbold, P., Bonaventure, O. 2003. IP micro-mobility protocols. IEEE Communications Surveys & Tutorials 5, 1, 40-57.

Riegel, M., Tuexen, M. 2006. Mobile SCTP. Internet draft draft-riegel-tuexen-mobile-sctp-07.

Roberts, M.L., Temple, M.A., Mills, R.F., Raines, R.A. 2006. Evolution of the Air Interface of Cellular Communications Systems Toward 4G Realization. IEEE Communications Surveys & Tutorials 8, 1, 2-23.

Saha, S., Jamtgaard, M., Villasenor, J. 2001. Bringing the Wireless Internet to Mobile Devices. IEEE Computer, Vol. 34, No. 6, June 2001.

Saha, D., Mukherjee, A., Misra, I.S., Chakraborty, M., Subhash, N. 2004. Mobility support in IP: a survey of related protocols. IEEE Network 18, 6, 34-40.

Salkintzis, A.K., Fors, C., Pazhyannur, R. 2002. WLAN-GPRS integration for next-generation mobile data networks. IEEE Wireless Communications 9, 5, 112-124.

Sarikaya, B. 2006. Home agent placement and IP address management for integrating WLANs with cellular networks. IEEE Wireless Communications 13, 6, 77-86.

Seung-Seok Kang, Matt W. Mutka 2005. A mobile peer-to-peer approach for multimedia content sharing using 3G/WLAN dual mode channels. Wireless Communications and Mobile Computing (Wiley) 5, 6, 633-645.

Shen, G., Tucker, R.S. 2007. Translucent optical networks: the way forward. IEEE Communications Magazine, Vol.45, Iss.2, pp. 48-54.

Shenoy, N., Montalvo, R. 2005. A framework for seamless roaming across cellular and wireless local area networks. IEEE Wireless Communications 12, 3, 50-57.

SOMA - http://lia.deis.unibo.it/Research/SOMA/

Snoeren, A. C., Balakrishnan, H. 2000. An End-to-End Approach to Host Mobility. In Proceedings of the 6th annual international conference on Mobile computing and networking, Boston, MA, August 2000, 155-166.

Spanoudakis, M., Batistakis, A., Priggouris, I., Ioannidis, A., Hadjiefthymiades, S., Merakos, L. 2003. Extensible Platform for Location Based Services Provisioning. Int. Conf. Web Information Systems Engineering Workshops, Rome, Italy, Dec. 2003.

Stallings, W. 2001. Wireless Communications and Networks, Pearson Education.

Stavroulaki, V., Buljore, S., Roux, P., Melin, E. 2006. Equipment management issues in B3G, end-to-end reconfigurable systems. IEEE Wireless Communications 13, 3, 24-32.

Stemm, M., Katz, R.H. 1998. Vertical handoffs in wireless overlay networks. Mobile Networks and Applications (Baltzer) 3, 335-350.

Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., Conrad, P. 2004. Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration. Internet draft draft-ietf-tsvwg-addip-sctp-09.

Titkov, L., Poslad, S., Juan Jim Tan 2003. Privacy Conscious Brokering in Personalised Location-Aware Applications. Int. Conf. Autonomous Agents and Multi Agent Systems (AAMAS), Australia, July 2003.

Velayos, H., Karlsson, G. 2003. Techniques to Reduce IEEE 802.11b MAC Layer Handover Time. Technical Report TRITAIMIT-LCN R 03:02, KTH, Stockholm, Sweden, Apr. 2003.

Vidales, P., Baliosian, J., Serrat, J., Mapp, G., Stajano, F., Hopper, A. 2005. Autonomic system for mobility support in 4G networks. IEEE Journal on Selected Areas in Communications 23, 12, 2288-2304.

Ward, Q.B., Ottrey, C., Cutting, D., Kummerfeld, R. 2004. BlueStar, a Privacy Centric Loca-tion Aware System. IEEE Int. Symp. Position, Location and Navigation, USA, Apr. 2004.

Wei Song, Hai Jiang, Weihua Zhuang, Xuemin Shen 2005. Resource management for QoS support in cellular/WLAN interworking. IEEE Network 19, 5, 12-18.

Wei Song, Weihua Zhuang, Yu Cheng 2007. Load balancing for cellular/WLAN integrated networks. IEEE Network 21, 1, 27-33.

Wei Wu, Banerjee, N., Basu, K., Das, S.K. 2005. SIP-based vertical handoff between WWANs and WLANs. IEEE Wireless Communications 12, 3, 66-72.

Wei Zhuang, Yung-Sze Gan, Kok-Jeng Loh, Kee-Chaing Chua 2003. Policy-based QoS-management architecture in an integrated UMTS and WLAN environment. IEEE Communications Magazine 41, 11, 118-125.

Welch, G., Bishop, G. 2001. An Introduction to the Kalman Filter. SIGGRAPH 2001, Aug. 2001.

Xing, B., Nalini Venkatasubramanian 2005. Multi-constraint dynamic access selection in always best connected networks. In Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, San Diego, CA, July 2005, 56-64.

Yi Pan, Meejeong Lee, Jaime Bae Kim, Suda, T. 2004. An end-to-end multipath smooth handoff scheme for stream media. IEEE Journal on Selected Areas in Communications 22, 4, 653-663.

Ylitalo, J., Jokikyyny, T., Kauppinen, T., Tuominen, A.J., Laine, J. 2003. Dynamic network interface selection in multihomed mobile hosts. In Proceedings of the 36th Annual Hawaii International Conference on System Sciences, Hawaii, HI, Jan. 2003.

Acknowledgments

First of all I would like to thank my advisors Maurelio Boari, Antonio Corradi and Paolo Bellavista not only for their attempts to teach me how to be a researcher, but also for their patience and moral support I have really appreciated on so many occasions.

Then, I would like to thank all the LIA's people and in particular Alessandra/o since they have shared with me the rather hard period spent at LAB2!

I would like to thank even my family, since they always had great faith in me and my capabilities: thank you Mum, Dad, Elisa, and Claudina.

But the most important person I would like to thank is certainly Frà, not only for her patience waiting for me at lunch/dinner, but also since she has always remembered to me there is something else besides the research world!