Integrated Support for Handoff Management and Context Awareness in Heterogeneous Wireless Networks

Paolo Bellavista[2], Marcello Cinque[1], Domenico Cotroneo[1], Luca Foschini[2]

(1) Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio, 21 - 80125 Napoli - Italy
{macinque, cotroneo}@unina.it

(2) Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna
Viale Risorgimento, 2 - 40136 Bologna – Italy
{pbellavista, lfoschini}@deis.unibo.it

ABSTRACT
The overwhelming success of mobile devices and wireless communications is stressing the need for the development of mobility-aware services. Device mobility requires services adapting their behavior to sudden context changes and being aware of handoffs, which introduce unpredictable delays and intermittent discontinuities. Heterogeneity of wireless technologies (Wi-Fi, Bluetooth, 3G) complicates the situation, since a different treatment of context-awareness and handoffs is required for each solution. This paper presents a middleware architecture designed to ease mobility-aware service development. The architecture hides technology-specific mechanisms and offers a set of facilities for context awareness and handoff management. The architecture prototype works with Bluetooth and Wi-Fi, which today represent two of the most widespread wireless technologies. In addition, the paper discusses motivations and design details in the challenging context of mobile multimedia streaming applications.

Categories and Subject Descriptors
C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design – *distributed networks, wireless communication, network communications.*

General Terms
Management, Design, Experimentation.

Keywords
Mobility awareness, handoff, heterogeneity, middleware.

1. INTRODUCTION
The widespread use of mobile terminals, today equipped with multiple wireless communication interfaces, is paving the way to the all-the-time everywhere connectivity view of pervasive computing. Moreover, the diffusion of wireless technologies is identifying new scenarios of service provisioning where mobile users are willing to have ubiquitous and continuous access to both traditional and novel context-aware Internet services while they move in smart spaces, such as smart museums or airports.

In particular, novel context-aware services should be aware of all the possibilities offered by each wireless technology and their respective limitations. The most adopted wireless technologies today, such as IEEE 802.11 (Wi-Fi), Bluetooth (BT), and 3G cellular, have different characteristics in terms of bandwidth, coverage, and per-byte transmission cost. Furthermore, services should be able to localize users, e.g., to obtain their current Access Point (AP) to the Internet, in order to provide location-dependent contents, and to follow their movements as they move in a smart space. Finally, due to wireless medium volatile nature and device mobility, wireless technologies can introduce non-negligible sudden variations of network conditions in terms of delay, bandwidth, and coverage status. Those unpredictable changes require application services being able to monitor network conditions, e.g., signal strength, of all available wireless interfaces, in order to adapt service behavior to current availability and load of the wireless infrastructure. For these reasons, services are required to be context-aware, in the sense that they i) should be conscious of the possibilities offered by each wireless technology, ii) should know user position and movements, and iii) should have easy access to low-level wireless network conditions.

In this paper, the focus is on delay-sensitive services, e.g., multimedia streaming, that will spread more and more as a central application class for smart environments. This class of services has very strict requirements in terms of network delay, jitter, and packet losses. Thus, delay-sensitive services are highly affected by handoffs (see Section 3), which often occur in heterogeneous wireless environments, introducing unpredictable delays and intermittent discontinuities. Handoffs occur either because client devices change their points of access to the fixed network, e.g., when users move from one wireless cell to a different one, or because the used wireless technology changes, e.g., from BT to Wi-Fi. To meet their application requirements, there is the need for those services to monitor and possibly manage the handoff process directly. For instance, when a multimedia streaming service detects a significant delay increase, it should be able to switch to a better access point. That points out the need for a tight relationship between context awareness and handoff management in heterogeneous wireless networks.

MPAC '05, November 28- December 2, 2005 Grenoble, France.

One of the paper core claims is the relevance of middleware infrastructures in pervasive computing scenarios, mainly to provide services with necessary context information and to let services take directly part in the handoff management process. We call mobility-aware service an application service which is context-aware and participates directly in handoff management. However, technological differences complicate the development of mobility-aware services. In fact, each wireless technology has its own set of parameters, mechanisms, and Application Programming Interfaces (APIs) to let the application level gather context information and manage handoff. Hence, while traditional fixed-network programming requires developers knowing only one standard API, i.e., the Berkeley sockets, advanced wireless network programming forces them to know all the peculiarities of each different wireless technology and its relative APIs. Therefore, another major paper claim is the need for mobility-aware middleware infrastructures to provide mobility-aware service developers with a uniform set of APIs that abstract technological peculiarities.

To achieve these goals, the paper proposes a middleware architecture consisting of two layers. The mechanism layer hides technology-related programming aspects and provides a uniform API to access and control context information and handoffs, irrespective of underlying wireless technologies. Exploiting the underlying mechanisms, the architecture provides mobility-aware service developers with advanced and technology-independent facilities at a higher layer, i.e., the facility layer. Both common and domain-specific facilities are taken into account. Common facilities offer basic primitives for communication and technology-independent retrieval of context information. Services can monitor the context and adapt their behavior accordingly. Moreover, common facilities support service development with high-level handoff management: services are allowed to simply and directly control the handoff process, for instance they can switch to other wireless technologies when service requirements are not met. Domain-specific facilities are instead offered to ease the development of services with well defined characteristics depending on the specific application domain. In this case, the middleware can directly manage context information monitoring, adaptation, and handoffs, offering simpler ready-to-use abstractions. In particular, the paper considers and discusses the case of a multimedia streaming domain-specific facility.

The proposed architecture has been implemented for BT and Wi-Fi wireless technologies, since they are nowadays among the most adopted connectivity solutions. In addition, they present several handoff-related differences which help to point out the generality of the approach. Furthermore, to justify motivation and design choices, the paper focuses on the specific application scenario of a museum smart guide service.

2. APPLICATION SCENARIO

Let us introduce a realistic application scenario to better analyze the issues and motivations for the integrated usage of different wireless technologies.

Luke, Paul, and Maggie go to visit Florence and their tour operator provides them with a smart audio guide application that they can install on their devices. The smart audio guide application adaptively exploits all device-available technologies: Luke has a BT and 3G enabled smart-phone, Paul a Wi-Fi enabled PDA, and Maggie a Wi-Fi and BT enabled PDA. When moving outdoor, the application exploits position information collected by the outdoor 3G infrastructure to localize users and to stream short audio clips describing museums in their vicinity; Luke listens to those clips through his smart-phone.

When the three friends pass by the art museum, they decide to stop and visit it. When moving indoor, the guide content changes as people roam in the museum depending on the wireless infrastructures present inside the building and on the wireless connectivity available on visitor devices. Figure 1 shows BT and Wi-Fi coverage of a museum area where the three guys are moving: each stand is covered by a BT cell and each room is covered by a Wi-Fi cell; finally, the interior of the museum is not covered by 3G. The application exploits BT to localize Luke and Maggie, and automatically changes the audio guide content depending on the stand, i.e., the BT cell, they are close to. Paul, instead, who can exploit only Wi-Fi, is localized with a coarser room grain and when he enters a new room the application sends him a map of the room. The map contains a link for each stand in the room and Paul has to click the link to listen to the corresponding audio description.

Imagine that Maggie is visiting the stand B.1 and receiving the related guide via BT. Suddenly, a large group of tourists provided with their audio guide on BT smart-phones crowds stand B.1 and the quality of Maggie's audio stream begins to degrade due to the BT infrastructure heavy load. Consequently, the smart guide application should automatically test the load conditions of the available Wi-Fi infrastructure and possibly command handoff from BT to Wi-Fi.

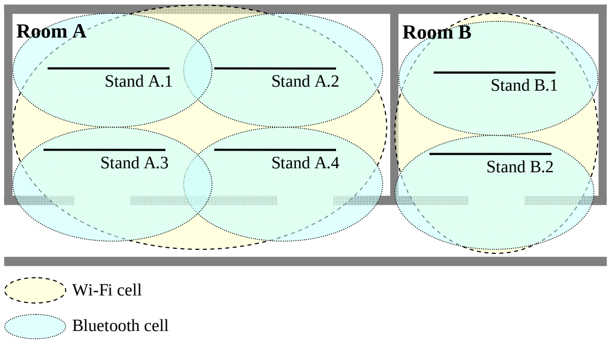

Figure 1. Virtual guide application scenario.

There are several lessons to learn from the above application scenario. The integrated usage of different communication technologies can help a service provider, e.g., the museum, to serve a wider set of users equipped with either BT, Wi-Fi or 3G. Moreover, the determination of the device physical location can benefit from the different location granularity offered by diverse wireless technologies. As pointed out in the scenario, the BT coverage range, i.e., the cell size, is lower than the coverage range of Wi-Fi, hence the coarse grained Wi-Fi coverage can be improved by using BT. Moreover, the integrated usage of various technologies enables the development of services that can fulfill specific application requirements otherwise difficult or impossible to reach with only one communication solution. For instance, the smart guide application can exploit different technologies to balance network load and achieve a better Quality of Service (QoS) level by automatically switching from one technology to one another when a problem occurs with a given connectivity solution, e.g., if a cell is overloaded or a link breaks.

Nonetheless, there are several challenging issues that arise when considering the integration of different wireless technologies. First of all, it is necessary to provide a mobility-aware support that provides service developers with application-level and application-relevant context information, e.g., location information, QoS information, such as communication delay, and technology-dependent information, such as per-byte communication cost. Moreover, the mobility-aware support has to offer a set of interfaces/facilities to let applications easily monitor parameters of the available wireless networks and to dynamically choose the communication technology that best fits their requirements. Finally, the support has to guarantee service continuity while switching from one technology to another one, in particular to soft-realtime applications such as audio streaming, as better detailed in the following.

3. MOBILITY-AWARE SUPPORT: ISSUES AND DESIGN GUIDELINES

This section first introduces the context information that mobility-aware supports have to expose, along with the relative technological issues and design guidelines. Then it gives definitions about horizontal and vertical handoffs, by discussing problems and solution guidelines for mobility-aware horizontal/vertical handoff management.

3.1 Mobility-Aware Context Information

Mobility-aware services need to monitor executing context parameters to flexibly adapt their behavior to changing conditions. Many of those parameters, e.g. datalink/network, are highly dependent from the adopted wireless technology and from runtime conditions of communication channels. In particular, each wireless technology can be characterized by several static parameters, e.g., per-byte transmission cost, battery cost, and localization resolution. For instance, the 3G cellular technology has an AP-based low localization resolution and its transmission cost is fairly high; conversely, BT localization resolution is high and the transmission cost is low (since BT exploits unlicensed spectrum for transmission). Middleware support for mobility-aware applications has also to consider the following dynamic parameters: inter-arrival packet delay, bandwidth, received signal strength, connection availability (connected, not connected or handoff), and device location. The middleware should extract those static and dynamic parameters by lower system layers and combine them together to provide mobility-aware services with context data at the most suitable level of abstraction. For instance, the middleware should provide a location information including the currently used AP, its coverage area size, and possibly a label that logically identifies that area.

It is worth noting that monitoring the above parameters is not trivial and requires a deep knowledge of technology-related aspects and specific software libraries. For instance, the Wi-Fi received signal strength can be obtained through the `iwconfig` command in Linux [1], while the same value for BT in Linux is gathered trough a specific primitive of the BlueZ stack [2]. Therefore, there is a strong need for middleware infrastructures capable to hide all those details and to offer a simple uniform interface to access parameter values at runtime.

In addition, depending on their application requirements, mobility-aware services could benefit from either pull or push interaction paradigms to obtain context information from the middleware support. In other terms, the middleware interface should provide calls either to poll parameter values or to asynchronously notify the application level about changes in parameters of interest.

3.2 Horizontal and Vertical Handoff

Horizontal handoff is a handoff process that occurs within one homogeneous wireless infrastructure, e.g., when a Wi-Fi mobile device working in infrastructure mode moves between two Wi-Fi cells by changing the AP it is connected to. We distinguish two types of data-link horizontal handoff: i) hard handoff, i.e., the destination cell AP takes over from the origin cell AP in a relay mode, minimizing signaling overhead but increasing latency and packet losses; and ii) soft handoff that activates the new data path to the destination AP before client disconnection from the origin cell [3].

Vertical handoff, instead, involves different wireless infrastructures. For instance, vertical handoff occurs when a device hosting different network interfaces is operating in an area served by various APs using different types of wireless technologies, and it decides to perform a handoff from one wireless infrastructure to one another, e.g., to switch from a Wi-Fi network to a cellular one. Moreover, [4] defines two types of vertical handoffs depending on the cell size imposed by the wireless connectivity solution: i) upward vertical handoff is a handoff to a wireless network with larger cell size, e.g., from BT to Wi-Fi or from Wi-Fi to 3G, and ii) downward handoff is a handoff to an infrastructure composed by smaller cells, e.g., from 3G to Wi-Fi or from Wi-Fi to BT.

Finally, handoff strategies can be classified in reactive and proactive. Reactive strategies look for other available APs only after the current AP signal is lost, by minimizing power consumption, at the cost of higher handoff durations; proactive strategies continuously monitor signal strength conditions and start communication-level handoff before losing current AP signal, at the cost of higher battery consumption.

3.3 Mobility-Aware Handoff Management

Handoff management is a key aspect in the development of supports for mobility-aware applications. In particular, as introduced by the application scenario, only careful technology-aware handoff management can avoid service interruptions, during both horizontal and vertical handoffs.

Horizontal handoffs are often already managed by the wireless technology at the data-link layer. For instance, the 3G infrastructure automatically performs soft horizontal handoff, whereas Wi-Fi adopts a hard horizontal handoff strategy. Nonetheless, some wireless technology, such as BT, does not automatically manage horizontal handoff. All those differences have to be hidden by the middleware infrastructure, which should offer the same view for all the technologies, even by transparently managing horizontal handoffs when required. However, mobility-aware services should be aware of the characteristics of horizontal handoff. For instance, in the case of Wi-Fi, mobility-aware services need visibility of handoff processes to recover the data that might be lost due to hard handoff.

The situation is further complicated if we consider different wireless technologies and the possibility to accomplish vertical handoffs. First, it is necessary to know which wireless interfaces are available and which of them are on. Then, different application requirements could require different vertical handoff management strategies, possibly depending on the previously discussed static and dynamic context information available for each technology. For instance, a smart guide application could privilege precise localization, e.g., by switching to BT whenever possible, or it could favor available bandwidth by switching to Wi-Fi-based communication whenever available.

4. ARCHITECTURE DESIGN

The technical differences discussed in the previous section, along with the need to adopt different technologies in different situations, clarify the difficulties of mobility-aware service development. Therefore, we claim the need for a middleware architecture to ease mobility-aware services development via a uniform and simple API. We designed an architecture capable of handling heterogeneous handoff procedures by exploiting all the possibilities offered by underlying communication technologies. Technological aspects are hidden to the application level and the middleware provides service developers with uniform and easy-to-use interfaces. Moreover, the middleware is able to provide a wide visibility of underlying events, such as horizontal/vertical handoff, user location change, and network failure events.

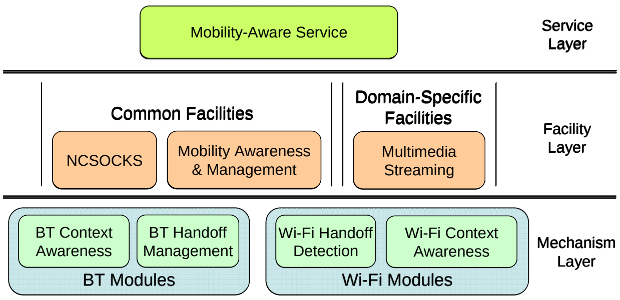

Figure 2. Middleware layered architecture.

More in detail, the proposed middleware architecture (see Figure 2) consists of two layers: the *Mechanisms Layer* and the *Facilities Layer*. Mechanism layer modules depend on the specific technology and encapsulate all the logic/knowledge necessary to control/monitor that particular wireless technology at data-link/network layers. The facility layer instead, similarly to CORBA [5], offers to the application level a set of common facilities, such as the NCSOCKS and the Mobility Awareness & Management presented in the following, to develop general purpose applications without specific needs, and a set of domain specific facilities such as the Multimedia Streaming. It is worth noting that services can use the Mobility Awareness & Management facility to gather information about the current channel status and to perform vertical handoffs, as detailed in Section 4.2.1. Thus, a service can undertake vertical handoffs once it verifies that its requirements are not met. That highlights the tight coupling between context-awareness and handoff management.

The different components of the proposed architecture interwork to perform their tasks. For instance, the Multimedia Streaming facility uses the Mobility Awareness & Management to check the channel status and to manage the handoff process. Therefore, mobility-aware services, e.g., the audio guide, can handle streams via the Multimedia Streaming facility without the need of explicit context monitoring and mobility management. Further details about the relationship among the components of the proposed architecture will be given in the following subsections.

We have implemented the three facilities and the mechanism modules for BT and Wi-Fi. Nevertheless, the implemented prototype of the proposed architecture is modular, flexible, and easily extensible: new technologies can be integrated as new mechanisms, while other domain specific facilities can be added.

4.1 Mechanisms Layer

The Mechanisms layer includes one module for each supported wireless technology. Each module is composed of two entities: one to gather context information and another one to monitor and manage (when possible) the handoff process. Each entity encapsulates all the logic necessary to program the specific technology. Vertical handoff, involving two or more different wireless technologies, cannot be managed by mechanism modules and is handled at the facility layer.

4.1.1 BT Handoff Management

BT [6] is a short-range wireless technology operating in the 2.4 GHz ISM band. BT devices form piconets where at most 7 active slave devices can communicate each other under the coordination of a master device.

Despite its widespread use, the Bluetooth Special Interest Group has neither defined nor standardized handoff management mechanisms. For this reason, a middleware approach is necessary to transparently provide a handoff management solution for horizontal handoff in BT. To this extent, the BT Handoff Management module has to execute horizontal handoff by exploiting the BT primitives offered to create/destroy BT connections and search for APs (via inquiry/scan procedures and service discovery searches).

The detection and search phases are performed adopting the Last Second Soft Handoff (LSSH) scheme, introduced in [7]. The handoff detection is based on the Receiver Signal Strength Indicator (RSSI) (the current RSSI value is obtained from the BT Context Awareness mechanism, see 4.1.3). With a proper tuning of its parameters, this scheme allows each AP to cover a well defined zone. This characteristic is desired in all scenarios where it is important to keep track of the current area a device is located in, as in the scenario of Section 2.

Most of the proposed handoff schemes for BT initiates the search phase once the RSSI falls below a certain threshold [8, 9]. As opposite, the LSSH scheme filters the RSSI signal using a α-count function [10]. This helps to carefully discriminate transient signal degradations, due to electromagnetic interferences or shadowing, from permanent ones.

During the search phase, multiple connections are established in order to implement the soft handoff scheme. To reduce the number of monitored APs, a topology-based solution is adopted that allows a mobile device to choose the next AP to use only among AP neighbors. For example, see Figure 1, a device that is leaving stand B.1 will attempt to connect only with the AP serving stand B.2. Further details about the LSSH scheme are available in [7].

4.1.2 Wi-Fi Handoff Management

Wi-Fi is the de-facto standard for wireless LAN deployment [11]. The IEEE 802.11 standard specification currently adopts hard handoff, thus potentially causing relevant packet losses in particular to delay-sensitive services, e.g., by introducing flow gaps in an audio streaming application (see Section 4.2.3). Differently from BT, the IEEE 802.11 standard specifies the mechanisms to implement the handoff procedure, i.e., handoff detection, target AP search, and AP re-association [12]. However, the standard leaves unspecified the mechanism combination and durations. Consequently, Wi-Fi card and AP models highly influence the overall handoff duration, which can vary from some hundreds of ms to even 2s, as shown in [12]. Moreover, the standard does not define any API to directly control the handoff process. This motivates the middleware approach to hide vendor-specific differences and to provide a uniform interface at least to monitor the underlying horizontal handoff.

Various enhancements to reduce handoff disconnection time have been recently proposed both from academia and IEEE 802.11 standardization committees: those proposals aim to reduce handoff duration by modifying the MAC layer protocol [12]. Nonetheless, those lower layer approaches are still not available in most diffused APs and would require firmware upgrade or re-deployment of all old Wi-Fi equipment. On the contrary, our application-level middleware aims to assist mobility-aware applications by adopting a cross layer approach. The primary idea is to monitor RSSI values of all APs visible at the client node to predict handoff occurrence and, thus, to accomplish at the middleware layer all those operations necessary to possibly mask Wi-Fi handoff occurrence to the application level. The Wi-Fi Handoff Management module encapsulates our original prediction solution called Received Signal Strength Indication-Grey Model (RSSI-GM). RSSI-GM handoff prediction exploits client-side RSSI monitoring for any IEEE 802.11 AP in visibility. Given a set of reachable AP and their actual RSSI values measured at the client, our RSSI-GM technique can estimate future RSSI values by using the GM prediction model [13]. Once obtained the predicted RSSI values for the current AP and the maximum value among all APs in visibility, the proposed architecture exploits a threshold-based technique to estimate both handoff probability and duration; the prediction model is adaptively configured for the specific Wi-Fi client card model in use [14].

RSSI-GM solution is completely local and lightweight: client hosts autonomously estimate future RSSI values by simply maintaining a finite series of previous RSSI data; RSSI prediction processing imposes a very limited overhead; and RSSI-GM execution does not impose any additional data exchange with the Wi-Fi infrastructure. The RSSI-GM solution is deployed and installed on client hosts as a lightweight stub that interacts with the Wi-Fi Context Awareness mechanism to gather RSSI values. Additional information and performance results about RSSI-GM are out of the scope of the paper and available in [14, 15].

4.1.3 BT and Wi-Fi Context Awareness

BT and Wi-Fi context awareness mechanisms provide facilities with information about current context. By following the design guidelines introduced in Section 3.2, these mechanisms provide a uniform API to either poll the context status or register to receive context change notifications. The following context information is made available: a) connection status in terms of availability (e.g. connected, not connected, handoff), RSSI, bandwidth, and delay; b) location status, i.e., the current AP being used (in terms of its logical AP name, its IP address, its MAC address, its technology, and the local IP address); and c) static parameters, i.e., communication cost, battery consumption, and location granularity indicator.

4.2 Facilities Layer

On top of the mechanism layer, we have designed and implemented the facilities layer consisting of two common facilities, i.e., the Mobility Awareness & Management and the NCSOCKS modules, and one domain-specific facility, i.e., the Multimedia Streaming module. Through the facilities layer, mobility-aware services can either directly manage the vertical handoff process, via Mobility Awareness & Management, or delegate it to the middleware, via Multimedia Streaming.

4.2.1 Mobility Awareness & Management

The Mobility Awareness & Management facility provides mobility-aware services and domain-specific facilities with current context information, together with the possibility to manage vertical handoffs. This facility consists of three main entities: *ConnectionMonitor* to get the current connection status and static parameters, *LocationMonitor* to get the current location of client devices, and *ConnectionBroker* to activate/deactivate available wireless network interfaces.

ConnectionMonitor and *LocationMonitor* exploit the underlying context-awareness mechanisms and make the context information available to services according to a uniform representation format. Similarly to context awareness mechanisms, services can either poll the current connection/location status, or register for notification about context changes. The difference is that technological differences are hidden by *ConnectionMonitor* and *LocationMonitor*.

ConnectionBroker allows mobility-aware services to switch on/off wireless network interfaces. Our support offers this possibility because the triggering of a vertical handoff may depend on specific service requirements and services should have the possibility to detect the handoff need by exploiting context awareness. For example, if the current perceived bandwidth is low, a mobility-aware file transfer service can switch to another available wireless technology with the hope to achieve better bandwidth. In addition to services, also domain-specific facilities can exploit Mobility Awareness & Management to perform vertical handoffs. For example, the Multimedia Streaming facility (see 4.2.3) performs vertical handoffs (via *ConnectionBroker*) as soon as perceived packet delay (monitored via *ConnectionMonitor*) is low. Therefore, a streaming service, such as the audio guide, can use Multimedia Streaming without caring about vertical handoffs.

4.2.2 NCSOCKS

Nomadic Computing SOCKetS (NCSOCKS) [7] are a common facility which provides service developers with an object-oriented communication API specifically suited for general purpose mobility-aware applications. They can be thought as a uniform, basic communication support that abstracts the actual communication channel being used. NCSOCKS are typically employed to realize communication in a transparent way from the point of view of vertical handoff. For instance, a messaging service may use NCSOCKS to receive/send messages independently of possible vertical handoffs during a session.

The NCSOCKS communication abstraction is similar to the one of connection-less Java Sockets. We preferred to avoid connection-oriented communication due to its known unsuitability to wireless scenarios. For example, in the case of TCP, if the wireless link breaks, TCP decrease the window and increase the retransmission timeout, leading to low throughput. Moreover, several connection-oriented solutions for wireless environments have been recently proposed in the literature [16].

Differently from the Java implementation, NCSOCKS data sending/receiving primitives take into account vertical handoffs. When a NCSOCKS socket is allocated, it creates an OS native socket on the currently active wireless network interface. The *send*/*receive* call detects whether a vertical handoff has occurred since the last utilization of the channel. If it has occurred, the old native socket is invalid, and a new socket has to be created. The information about the new interface to use (for instance, its IP address) is retrieved using *LocationMonitor*. To reduce packet losses during handoff, NCSOCKS is also in charge of simple buffering operations.

4.2.3 Multimedia Streaming

Due to their isochronous nature, multimedia services, such as video on demand and live streaming applications, are more affected by either horizontal or vertical handoffs than other classes of services. In fact, multimedia services usually adopt connection-less protocols such as UDP and manage data re-transmissions directly at the application level to react to high jitter and packet losses. Let us note that traditional multimedia applications developed for the fixed Internet assume that only sporadic packet losses occur, i.e., only during network congestion/failure, and do not consider the possibility to change the communication channel during streaming provisioning. In particular, frequent and potentially long packet losses can cause client side re-buffering, which results in possible playback discontinuities, while connectivity change imposes service re-start. Let us note that in current wireless-enabled environments horizontal and vertical handoffs frequently occur by causing relevant packet losses and complicating streaming provisioning.

The Multimedia Streaming facility aims to facilitate multimedia service development and overcomes the mentioned problems by granting lossless horizontal handoffs and directly managing vertical handoffs. This facility interacts with an articulated proxy-based middleware infrastructure extensively presented in [15]. The basic idea is to introduce middleware proxies located at wired network edges to locally serve mobile wireless clients. Middleware proxies are able to split the typical client-server direct interaction and enable handoff management at intermediate nodes by employing an original two-level buffering technique that stores incoming data flows at both client and proxy nodes. Moreover, middleware proxies receive and store incoming flows during horizontal and vertical handoff to avoid frame losses and server-to-client re-transmissions and can promptly fill up client buffers at client reconnection after handoff completion. The introduction of second-level buffers permits to achieve seamless streaming and introduces very limited overhead: Multimedia Streaming enlarges proxy buffers only when needed, i.e., for the time needed to complete the handoff management procedure, while, far from handoffs, proxy buffers contain only a little data chunk. For additional details about proxy-based two-level buffering, please refer to [15].

Multimedia Streaming adapts to horizontal handoffs by exploiting the mechanism layer modules. For instance, when running over Wi-Fi, the facility gathers Wi-Fi handoff probability from *Wi-Fi Context Awareness* and uses that information to trigger horizontal handoff management; the facility acts similarly in the case of BT. For vertical handoff, instead, Multimedia Streaming uses facility layer modules: it monitors connectivity status through *ConnectionMonitor*; it triggers vertical handoff management when perceives signal degradation; and it uses *ConnectionBroker* to switch from one connectivity type to the other. Let us note that vertical handoff requires the re-binding of client/proxy-side middleware components; however, thanks to our two buffering levels, the facility can preserve streaming continuity also in this case. Finally, Multimedia Streaming can possibly downgrade multimedia content when necessary, e.g., if the service switches from a powerful Wi-Fi link to a BT one.

5. SMART AUDIO GUIDE CASE STUDY

Vertical handoff management represents one of the most complex aspects of mobility-aware service development by involving different technologies and possibly requiring service adaptation. This section highlights the vertical handoff procedure (and related adaptation) in the challenging case of our museum smart audio guide service. The service has to be adaptive in spite of location and connectivity changes. Location adaptation means that the provided audio streams and texts depend on the current location of mobile client devices. Moreover, the service must tailor the audio content by down/up-scaling the delivered multimedia flow depending on current network conditions, i.e., connection type and perceived network delays.

The service development is eased by the presence of middleware facilities. To achieve location adaptation, the audio guide application can register a callback to the *LocationMonitor* facility. Let us suppose that Luke is standing close to the stand A.1 (see Figure 1). He is receiving the audio guide flow for that stand via his BT enabled smart-phone. Being the content boring, he moves towards stand A.2. As soon as *LocationMonitor* detects a location change, the callback is invoked and i) the service refreshes the text contents on the application GUI by requesting the text via NCSOCKS, and ii) the service asks Luke whether the current audio stream should be interrupted in favor of the one about the new location. If Luke decides to keep listening to the A.1 clip, the Multimedia Streaming facility transparently accomplishes audio stream horizontal handoff, by preserving stream continuity. Otherwise, a new stream request for the A.2 audio clip is issued via Multimedia Streaming.

Let us focus on the challenging issue of maintaining audio continuity in the case of vertical handoff to show the Multimedia Streaming insights about distributed resource management and content adaptation. In the following, we show the Multimedia Streaming facility at work by referring to the last part of our application scenario: Maggie is visiting the stand B.1 and is receiving the B.1 audio clip via BT when a big group of tourists, provided with their audio guide on BT enabled smart-phones, crowds stand B.1.

From the deployment point of view, Multimedia Streaming consists of two main entities, as depicted in Figure 3: i) a middleware proxy deployed on the wired network, and ii) a client stub deployed on the mobile device. Proxy components are installed in those networks that provide connectivity to mobile clients to locally serve mobile clients and to distribute handoff management load to network edges [15]. In normal executing conditions, i.e., far from vertical handoffs, proxies receive audio streams from legacy multimedia servers and forward them towards their client stubs, by only buffering a small amount of audio packets to smooth network jitter. In addition to this, client stubs register to the Mobility Awareness & Management module to receive communication delay updates.

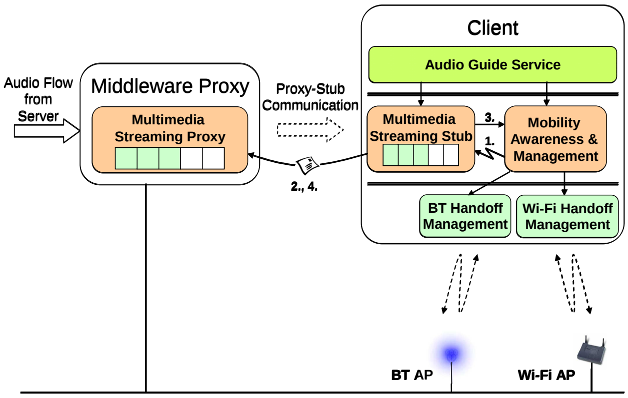

Figure 3. Multimedia Streaming at work in the audio guide service.

When the group of tourists approaches to stand B.1, their arrival causes several new connections to the BT AP, thus exceeding the maximum of 7 active slave devices per time. Consequently, the AP (the BT piconet master) starts to distribute the limited time slots to slaves that continuously switch from active (communication) to parked (wait for a free time slot); this degrades the packet delay and the overall audio quality. Maggie's multimedia streaming stub is notified by *ConnectionMonitor* about that significant delay degradation (step 1 in Figure 3). The client stub reaction is twofold: on the one hand, the stub asks the proxy to reduce audio stream bandwidth, i.e., the Multimedia Streaming facility aims to maintain streaming continuity and comprehensibility, by preferring a QoS degradation; on the other hand, the stub starts an upward vertical handoff.

The upward vertical handoff procedure begins when the client notifies the proxy about the imminent handoff (step 2). Triggered by that message, the proxy begins to enlarge its buffer to host those packets that could be potentially lost during handoff. In the meanwhile, the stub begins to accomplish handoff operations (step 3): it contacts Mobility Awareness & Management (and in particular *ConnectionBroker*) to obtain a Wi-Fi connection; then, it executes necessary rebinding operations; and, finally, it asks Mobility Management to disconnect BT, thus guaranteeing soft handoff. The average time to create the Wi-Fi connection is highly depending on the Wi-Fi card model and may vary from 0.5s to 2s, which includes the AP search and IP configuration. During this time, the client side buffer is consumed by the audio guide service, and slowly filled using the overloaded BT connection. On the wired side the proxy stores the packets produced by the audio server in its buffer. As soon as the new Wi-Fi connection is ready, the client stub requests the proxy to flush all the packets stored during handoff (step 4). Having Wi-Fi a better bandwidth than BT, the client side buffer should be quickly filled, letting the application seamlessly provide Maggie with her audio guide flow. Moreover, given the smaller communication delay, the streaming quality can be up-scaled with Maggie's satisfaction.

The above described vertical handoff procedure focuses on upward handoff. Our middleware infrastructure also manages downward handoff similarly to upward handoff, with a few variations. In particular, there are two main differences: i) the search and connection time (including IP configuration) for BT is longer than for Wi-Fi and may approximately range from 1s to 5s, depending on how many devices are present, and ii) the lower BT bandwidth causes a slower refilling of the client side buffer. Therefore, downward handoffs may stronger affect service continuity than upward handoffs.

We are currently in the process of experimentally evaluating audio discontinuities introduced by upward and downward handoffs. We are also exploring the implementation details of the specific entity to do down/up-scale that will be integrated in the audio component. We are exploiting our past experiences on the field to achieve this goal. In particular, two prototypes have been realized over the past two years: one for BT, which implements the LSSH scheme (see section 4.1.1) along with NCSOCKS[1] (see 4.2.2), and the other for Wi-Fi, implementing the handoff detection (see 4.1.2) and the Multimedia Streaming facility[2] (see 4.2.3).

6. RELATED WORK

The growing interest in service provisioning over heterogeneous wireless networks has recently stimulated several research activities. In [17], QoS and mobility aware solutions are proposed for a heterogeneous network, including satellite and terrestrial access networks (2.5G, 3G, and Wi-Fi) connected to an IP core network. Handoff strategies are based on fuzzy logic: vertical handoff is initiated depending on current signal strength and user QoS parameters. The goal is to ensure QoS by performing admission control and by preserving the established QoS during handoff. Nevertheless, that solution prefers an application-transparent approach: there is no support for service on-line adaptation to varying network conditions. On the contrary, we rather believe that the support of dynamic service management and adaptation is crucial to facilitate mobility-aware service development.

---

[1] http://www.mobilab.unina.it/NCSOCKSdwnd.htm

[2] http://www.lia.deis.unibo.it/Research/MUM

An Adaptive Transport Layer (ATL) is proposed in [18] for the Next-Generation Wireless Internet, which is characterized by the presence of heterogeneous wireless infrastructures, ranging from WLAN to satellite networks. In this work, the focus is on providing a new adaptive transport protocol (TCP-ATL) for reliable data transport and a new adaptive rate control protocol (RCP-ATL) for multimedia delivery. According to the requested service type, i.e., reliable data or multimedia, ATL selects the appropriate protocol. However, vertical handoff issues are masked by the architecture also in that approach, by limiting its suitability for mobility-aware services.

Gerla et. al propose in [19] a smart decision algorithm for vertical handoff in heterogeneous wireless networks. The algorithm exploits user-defined quality parameters to tailor the handoff procedure. Only the vertical handoff problem is taken under consideration, whereas no facilities are offered to guide and simplify the service development process.

7. CONCLUSIONS AND FUTURE WORK

The paper has presented a middleware architecture to ease the development of mobility-aware services in heterogeneous wireless networks. Developers of general-purpose services can exploit our common facilities to implement basic communication, adaptation, and vertical handoff management. In addition, our support provides ready-to-use domain-specific facilities to quickly develop applications with specific domain-dependent requirements. A first prototype of the proposed architecture has been implemented for BT and Wi-Fi, with emphasis on the multimedia streaming facility. Nevertheless, the approach is flexible and extensible: using the experience of BT and Wi-Fi integration, new technologies such as 3G cellular, can be integrated as new mechanism modules. Moreover, other domain-specific facilities can be added: we are currently implementing a location-aware group communication facility to enable multicast communication to devices sharing the same location.

We have already collected first performance measures, in particular to evaluate the challenging aspect of the overhead imposed by our middleware-level management of vertical handoff. Preliminary results are encouraging: our future work will be primarily devoted to extensively test our prototype over a large set of testbed conditions in order to better assess the possibility to eliminate/smooth service discontinuities due to upward and downward handoffs.

8. ACKNOWLEDGMENTS

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

9. REFERENCES

[1] Debian, Tools for Linux Wireless Extensions, http://packages.debian.org/stable/net/wireless-tools.html

[2] BlueZ, the Official Linux Bluetooth Protocol Stack, http://www.bluez.org

[3] D. Saha et al. Mobility Support in IP: a Survey of Related Protocols. *IEEE Network*, Vol. 18, No. 6 (2004)

[4] M. Stemm, R.H. Katz. Vertical Handoff in Wireless Overlay Networks. *Kluwer Mobile Networks and Applications*, Vol. 3, No.4 (1999)

[5] Object Management Group. Common Object Request Broker Architecture (CORBA), www.omg.org (2001)

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

[7] M. Cinque, D. Cotroneo, and S. Russo. Achieving All the Time, Everywhere Access in Next-Generation Mobile Networks. *ACM SIGMOBILE Mobile Computing and Communication Review*, Vol. 9, No. 2 (2005)

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

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

[10] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. *IEEE Trans. on Computers*, Vol. 49, No. 3 (2000)

[11] IEEE 802.11 Standard: Wireless LAN MAC and PHY Specifications (1999)

[12] H. Velayos, G. Karlsson. Techniques to Reduce IEEE 802.11b Handoff Time. *Proc. of IEEE Int. Conf. On Communications (ICC’04)* (2004)

[13] J.L. Deng. Introduction to Grey Theory, *The Journal of Grey System*, Vol. 1, No. 1 (1989)

[14] Paolo Bellavista, Antonio Corradi, Carlo Giannelli: Adaptive Buffering based on Handoff Prediction for Wireless Internet Continuous Services. *To be published in Proc. of the 2005 International Conference on High Performance Computing and Communications (HPCC’05)* (2005)

[15] P. Bellavista, A. Corradi, L. Foschini. Application-level Middleware to Proactively Manage Handoff in Wireless Internet Multimedia. *To be published in Proc. Of the IEEE Int. Conf. On Management of Multimedia Networks and Services (MMNS’05)* (2005)

[16] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang. TCP Westwood: Bandwidth estimation for enhanced transport over wireless links. *Proc. of the ACM International Conference on Mobile Computing and Networking (MobiCom’01)* (2001)

[17] G. Bianchi, N. Blefari-Melazzi, M. Holzbock, Y. Fun Hu, A. Jahn, and Ray E. Sheriff. Design and Validation of QoS Aware Mobile Internet Access Procedures for Heterogeneous Networks. *Mobile Networks and Applications, Special Issues on Mobility of Systems, Users, Data and Computing*, Vol. 8, No. 1 (2003)

[18] O.B Akan and I.F. Akyildiz. ATL: An Adaptive Transport Layer Suite for Next-Generation Wireless Internet. *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 5 (2004)

[19] L-J. Chen, T. Sun, B. Chen, V. Rajendran**,** and M. Gerla. A Smart Decision Model for Vertical Handoff. *Proc. Of the 4th International Workshop on Wireless Internet and Reconfigurability (ANWIRE’04)* (2004)