Agent-based Middleware to access Multimedia Services in a GRID Environment

D. Bruneo, A. Zaia, A. Puliafito
Universita’ di Messina, Dipartimento di Matematica
Contrada Papardo, Salita Sperone
98166 Messina, Italy
e-mail: {dbruno,azaia,apulia}@ingegneria.unime.it
Abstract

The increasing development of wireless devices and the diffusion of mobile computing require big efforts in order to satisfy the users’ needs. The management of advanced services such as the multimedia streaming involves several operations like QoS management, location and context awareness, tailoring, load balancing, power saving and so on. This paper presents the very complex and challenging issue to design a comprehensive middleware to allow mobile wireless user to access advanced services available in the wired part of the network. The overall middleware is distributed on the areas involved in wireless service provisioning (Wireless Network, Wired Network, Access Points) and has been implemented by using the mobile agent paradigm. In order to improve the management of wired resources the use of the Grid Computing paradigm has been investigated. A testbed has been developed that is used as an experimental environment to provide mobile users the access to multimedia streaming flows while moving and at a given level of QoS.

Keywords: Mobile Computing, Location and Context Awareness, QoS Management, Adaptive Services, Mobile Agents, Grid Computing

I. INTRODUCTION

During the last few years, the increasing development in network technologies has enabled to extend the types of services available through the Internet. In fact, if once only services such as e-mail or file exchange were available, now it is possible to view hypertext pages and to receive the streaming of multimedia flows. Services such as Video On Demand are today available for any Internet user, thanks to the considerable amount of bandwidth available until one’s home [30][13][23].

At the same time, portable devices such as notebooks, laptops, personal digital assistants (PDAs), etc. have experienced a wide diffusion. The diffusion of the wireless technology has allowed such devices to be connected anywhere and at any time [31][29].

Mobile computing is diffusing very quickly. Today users wish to access varied services transparently and effectively, while working or travelling. In order to enable mobile devices to access the resources and the services available through the Internet, some difficulties need to be solved. Such difficulties are related both to the limited resources of wireless devices and channels, and to the current inadequate services, which have been conceived for being used in wired environments. First of all, the limited bandwidth available reduces
the Quality of Service (QoS), mainly in the case of multimedia applications. Furthermore, the low reliability of the wireless channels makes the transmission less effective, and leads to unstable connections. The limited hardware resources (CPU, RAM, display, power, etc.) of mobile devices affect the connection time, the wireless transmission and the use of advanced applications. Finally, the user’s mobility causes some problems, such as the management of connections, the resources allocation, the QoS maintenance, and the address management.

If on the one hand solutions to improve the final devices (that are becoming more and more powerful) and the available bandwidth (by studying new standards) are been proposed, on the other more effort is needed in order to manage the resources present in the wired area as effectively as possible, and to create a middleware that can compensate for the weaknesses of the devices [8].

This paper presents a middleware for the management of advanced services in wireless environments. Such middleware covers all the components involved in wireless service provisioning (Wireless Network, Wired Network, Access Points). In particular, an architecture that allows to solve the main issues for the streaming of multimedia flows in case of user’s mobility has been created. In order to create a strong interaction between the architecture modules, the mobile agent paradigm has been adopted. Until a few years ago, the main research activity about mobile agents was limited to investigating new techniques for the creation of the mobile code, as well as to designing more effective platforms. Conversely, the researchers’ attention has recently been focused on finding new fields of application for agents, and on using this technology for solving several problems. In fact, several examples have been provided for the use of mobile agents in the solution of complex problems such as network management, simulation and network security [26] [32] [18], or in the creation of advanced services such as information retrieval and e-commerce [20] [5]. The big success of mobile agents is due to their ability to combine the typical features of software agents (autonomy, delegation, etc.) with the opportunity of "migrating" by moving from a position to another [33] [11]. On the one hand, this feature allows to decentralize the operations. On the other, an interaction is possible with the environment where the agent is. This way, the scalability of the system can be increased, and some context and location aware mechanisms can be used.
To improve the management of the wired resources, the paper proposes an architecture based on the Grid Computing paradigm able to carry out tasks as distributed transcoding of multimedia format, data dissemination, load balancing and discovery of service.

The implementation of this middleware and its integration with the current multimedia clients and servers has raised the need to manage the data streaming at a lower level. In fact, some functionalities provided by the middleware are related to the possibility to switch between different multimedia streaming servers at provision time and to the change of the device used or of the device address. To solve this, a RTSP/RTCP/RTP manager called Virtual Server and integrated in the middleware has been created. The Virtual Server works between the clients and the servers, and allows to solve the issues regarding the instability of connections and the change of client and server during the streaming. Such issues are not adequately supported by existing software products because the current servers and clients need to restart from the setup phase when the timeout has expired. They do not allow to send the streaming from the point it was timed out. Furthermore, switching from a device to another, or moving from the server where the streaming is being done, is not allowed. Solving these issues is essential in the management of the services provided by the middleware.

This paper is organized as follows: section II provides an overview of the issues related to the use of multimedia services in mobile environments. Section III presents the reference scenario and the architecture of the middleware proposed focusing on the innovative aspects introduced. Section IV describes the agent community that implements the middleware architecture, analyzing the agent to agent and the agent to system interactions. Section V shows the experimental results. Finally, section VI presents the conclusions.

II. RELATED WORK

The development of smaller and powerful devices, as well as the considerable investments made on wireless technologies, have allowed the extensive diffusion of mobile computing. Users equipped with laptops, PDAs, and lightweight devices, connected through multiple technologies (802.11 [19], GPRS [15], Bluetooth [1]), can access the resources present in the fixed network [17].
In such a heterogeneous and dynamic environment, the management of advanced services (such as multimedia applications) causes many issues that need to be solved, in order to access the services with an acceptable QoS [10] and an adequate level of transparency.

This section provides an overview of the state-of-the-art about the issues related to the use of multimedia services in mobile environments, in order to introduce and justify the solutions proposed in the following sections.

Let us consider the case of a user equipped with a PDA, who wishes to receive some information about the movies available in the network. He/she will select the movie to watch, according to some video clips that will be presented to him/her. In order to meet the user’s requirements, the system should be able to: know the user’s actual position, search for the services requested and select the ones more appropriate for him/her; know the user’s habits, so to select the results to be proposed (for instance, the movie genre); establish and rate the QoS level to be used; format the information to be presented according to the hardware and software features of the device used; manage the user’s mobility, by allocating the resources in advance for using the service (for instance, the bandwidth needed for the streaming) and managing the changes of address; manage the load of the wired network, so to optimize the resources by changing the server where the streaming has to be done from. Furthermore, the user might change the device for watching the video (for instance, while going back home, he/she might decide to switch to his/her PC). In this case, the system should be able to configure the parameters again, in order to resume the view from the point it was stopped, and to adapt it to the new device. Some operations of clip transcoding, reallocation of resources, and QoS management will therefore be necessary.

The previous example outlines the many issues involved in the management of multimedia services in mobile environments.

First of all, the Discovery of Services has a primary role, because it allows the user to access the list of services available. The techniques currently used include the use of distributed directory services, which are more scalable than the centralized approach [22]. In order to filter the services available, the user’s profile will need to be managed. Initially, only the services corresponding to his/her preferences and habits should be provided.

Multimedia services, and the streaming of multimedia flows in particular, need very strict
QoS requirements. In fact, a variation in the quality of the transmission may degrade the application requested and make it inaccessible. For instance, while viewing a video, the bit rate cannot be reduced below a specific threshold. Otherwise, the video itself cannot be viewed properly. The QoS management is therefore very important. The major problem is the limited bandwidth available in the wireless channels. This involves the investigation of techniques of bandwidth reservation, in order to anticipate the user’s handoff and reallocate the resources in the new Access Point (AP) that will be visited by the user [7][24].

The system will need to be able to recognize the features of the device, in order to adapt the service requested to the type of terminal used. In fact, if the same video is viewed on a notebook with a 16.8 million color display or on a PDA with a 256 color display, different resolutions will be needed. Considering the policies of bandwidth saving required in wireless environments, the need for transcoding the same video in several formats is evident, in order to tailor it according to the client device. For instance, an MPEG-2 video can be encoded in MPEG-4 when switching from a wired to a wireless station. In order to enable applications to adapt to such context evolutions, a tailoring phase through parameter re-configuration at provision time is necessary [3][9]. Information can be distributed on the wired network. This technique allows to access the resources more quickly and effectively, and to have more copies of the same video, even in different formats.

The system cannot disregard the user’s actual position (Location Awareness), since it needs to propose the services and the resources available near the user’s position. This is necessary because of the strict limitations in terms of QoS, and because the distance (latency) between the multimedia server and the mobile client is decisive. Location aware middleware, able to provide services according to the user position and to manage the user movements, have been developed in scenarios such as e-health [28], e-learning [16] and cultural heritage [21].

The system will need to make load balancing operations. In order to do the streaming, the system will need to select the less loaded servers, as well as those with a lower latency than the Access Point where the user is. The flows of the same video will be associated, and the multimedia files will be moved, if necessary.

Finally, all these operations must be done considering the limited resources of mobile
devices, and power consumption in particular. Some power saving techniques need to be used, for reducing the wireless transmission (download and upload) [2].

If some of the discussed issues need the study of solutions at a lower level (in [6] a resource reservation mechanism based on the Cellular IP routing protocol is proposed) other address towards a middleware approach able to provide solutions at application level [34][8][4]. In fact, these solutions appear more flexible and therefore able to adapt to such heterogenous environments.

One of the most interesting technology in the development of such middleware is the Mobile Agent Technology (MAT) [33][12]. The use of the MAT provides several benefits. In fact, mobile agents can follow the user during his/her movements and act as intermediaries in the wired network, thanks to their mobility and autonomy. The ability to follow the user or (in some cases) to anticipate his/her movements, is essential in the QoS management. The agent anticipates the user, and performs all the resource allocation operations needed. This way, the operations become totally transparent to the user. The Location Aware is made possible by the agent’s ability to follow the user through his/her movements. In fact, the agent can always locate the user, and sends such information to the wired components involved in this process. Agents are also important in the Discovery of Services, in the management of connections, and in the possibility of connecting again with a different device. Furthermore, mobile agents are essential for solving the issues related to the wireless transmission (with regard to signal discontinuity, in particular) and, as a consequence, to power saving. In fact, the user will have the opportunity of delegating some operations and reducing his/her network traffic, which is very expensive in terms of power.

This paper presents an agent-based middleware which allows to combine mobile computing with complex applications such as multimedia streaming. The development of such middleware calls for new solutions able to provide QoS to mobile users. These solutions are related to the integration of new technologies such as Grid Computing [14] and to the design of proxy systems that interact with the streaming by using the RTSP/RTCP/RTP protocols. This middleware will be able to manage the user’s mobility effectively, and to assure the use of the services with the desired QoS. Location and Context Aware techniques have been used in order to manage the resources of the wired area, provide the user with
the local services, and adapt these services to the different mobile devices used.

III. MIDDLEWARE ARCHITECTURE

The proposed architecture includes all the software components involved in providing multimedia services to mobile users. In Fig. 1 the reference scenario is shown, it consists of the following areas:

- Wireless Area
- Access Area
- Wired Area

Fig. 1. The reference scenario is composed by three areas: wireless, access and wired.

The Wireless Area is the coverage area of an Access Point, where one or more mobile devices can be found.

The Access area is the contact area between the Wireless Area and the Wired Area. It consists of the Access Points (APs), which allow the mobile devices to access the services.
available in the Wired Area. Each AP is supplied with one or more PCs that allow the accomplishment of the requested tasks. In the remainder of the paper we logically refer to the APs meaning the PCs close to them.

The Wired Area is the core network infrastructure.

The management of the issues related to the provisioning of multimedia services in wireless environments calls for a middleware solution that covers all the areas present in the scenario. In such a way, in fact, an interaction between the wired and the wireless components of the system is created and a high level of transparency to the users can be achieved. Moreover, the distribution of this middleware over the three described areas will allow the accomplishment of expensive tasks wherever the more convenient resources will be available. In this the system will never overload mobile devices, responding to the power saving issues.

In the Wired Area, all the tasks requiring a high computational load will be carried out. Such tasks are related to the multimedia data management: storage, tailoring, streaming and load balancing. A powerful management of these tasks is mandatory in order to make effective the QoS strategies provided by the other areas of the scenario and to improve the service provisioning. Let us consider, for example, the tailoring of the multimedia format to the real feature of the client device. It can be easily observed that by sending high resolution multimedia data to a smart device the system will experience a high percentage of wasted resources that affects the other users present in the same cell, making useless any resource reservation policy.

In order to carry out the tasks related to the Wired Area the Grid paradigm has been used. As defined in [14] the objective of the Grid technology is to support the sharing and the coordinated use of several resources in dynamic, distributed Virtual Organizations (VOs). In contexts of distributed supercomputing and Internet-based parallel computing, the Grid paradigm can provide the infrastructure for collaborative engineering, data exploration and provision of advanced value added services. In the scenario under exam, a VO shares its resources to allow mobile users to access multimedia contents suitable for their device, optimizing wireless network traffic and balancing the computing load. Organizations will be composed by several Grid Nodes. As depicted in Fig. 2, three Organizations (O1, O2,
and O3), each representing a provider, share their resources (Grid Nodes) in terms of computational and storage capacity in a VO. Some Inter-Organization backbones connect the Os through a high speed and secure link.

In this scenario, two multimedia-related activities can be identified.

1) Multimedia Upload and Storage
2) Multimedia Streaming

In the first activity, operators in each organization make a multimedia object available to the users of the whole VO by uploading the file in one or more Grid Node. In the Multimedia Streaming activity the movie, which is requested by the user, is adapted according to the characteristics of the device. If necessary, the movie is moved to a neighbor Grid Node, in order to improve the performances, and then is streamed to the user’s device. During the adaptation phase some chapter of the requested content might be moved in specific set of Grid Nodes, according to the effective load of each node. Therefore the movie will be decomposed in several chunks scattered on the Wired Area. In this case the Streaming activity will take place in a distributed way by starting the streaming of chunk 1 from server S1; then, at the end of this chunk, the system requests the next chunk to another server etc.,
as shown in Fig. 3. However, other middleware components, present in the Access Area, allow users to watch their selected movie in a transparent way. The assumption of the Grid paradigm in the Wired Area will carry out the data format transcoding (based on users’ requirements) in an effective way by reducing the initial time and by improving the access to the wired resources.

A similar mechanism takes place during user’s handoff. If a less loaded server is identified, the system will be able to move the movie on the new server and to switch the streaming from the point at which it was arrived. In this way the middleware responds to the load balancing requirements of the wired components of the system. A streaming system based on the Grid paradigm has been proposed in [35].

In the Access Area the tasks related to the resource reservation, the user management and the multimedia flow management are present. The resource reservation is performed by admission control techniques that restrict the access to the wireless environment. In fact, the bandwidth available in the wireless interface quickly decreases according to the increased number of users present in the cell [25]. This is due to the frequent collisions occurring while data are being sent. In order to guarantee the service provisioning during the users’
movements, the proposed middleware implements techniques able to reserve in advance the resources in the new AP where the user is going to. The advanced reservation strategies need to know the users’ position and to manage the bandwidth of each cell by leaving a portion of it for the users coming from neighbor cells. QoS levels are also been created in order to allow the bandwidth re-configuration according to users’ needs and resource availability. The middleware, at provision time, can automatically adapt the bandwidth assigned to a user, between a fixed range, in order to accept new users from neighbor cells, decreasing the call-blocking probability [7].

Two solutions are possible for the mobility management. The first one is based on positioning techniques, such as Global Positioning System (GPS). The second one is related to the use of triangulation algorithms able to understand the real position of the user according to signal power analysis. The task of the Access Area is the management of data related to users’ position and the interaction with the resource reservation and load balancing mechanisms. The user’s management inside the Access Area will be accomplished by an intermediation phase between the user and the wired components of the system. In particular, the Access Area manages the Context and Location Aware mechanisms by activating the tailoring process in the Wired Area and by choosing the services on the base of the user position.

As already mentioned, the transcoding and load balancing operations require the switching between several streaming servers in order to react to the changes of the network conditions. The effective integration of the switching operations with the current multimedia players and servers has raised the need to manage the multimedia streaming at a lower level. In fact, the disconnections and the new re-connections will cause the session closing and the need to restart from the setup phase. In order to deal with these issues a RTSP/RTCP/RTP manager called Virtual Server (VS) has been designed.

The VS is an entity that works between a multimedia player and one or more servers. Its task is to manage, handle, and readdress the data/control flow transparently for both the player and the servers. The VS architecture, shown in Fig. 4, is constituted of the Server Side that manages the interaction with the streaming Player, the Client Side that manages the interaction with the streaming Server, and the Socket Manager that manages the TCP and
UDP sockets involved in the streaming. The Server Side and the Client Side have a specular structure. They are composed by the components related to the management of streaming protocols (RTSP, RTCP, RTP); these components are able to decouple the streaming Player from the streaming Server by receiving, processing and forwarding the protocol packets. Such operations will allow the disconnection and re-connection tasks; in fact, servers and players will continue to interact only with the VS. Moreover, the Client Side and the Server Side are provided of buffers able to store the incoming packets. The size of the reception buffers of the VS must be assessed according to the bandwidth required by the stream and to the network congestion. The selection of the buffer affects the fluidity of the stream. The RTP buffer of the Server Side is not implemented because the RTP flow will be exclusively in the server - player direction. The Socket Manager stores the status (IP address, port) of the connections established by the VS in order to carry out the packets management.

In figures 5 and 6 the VS tasks related to the management of the RTSP protocol during the connection phase and the streaming switching phase are shown.

In order to allow the handoff management, the VS will be located in part on the client device, and in part in the Access Area. The part of VS located on the device will be called Device Virtual Server (DVS), while the part located on the Access Area will be called Infrastructure Virtual Server (IVS).
Thanks to such VS chain, the Player located in the mobile device has to establish a connection only with the DVS located in the same device; in this way the issues related to the address change during the handoffs can be addressed. The DVS implementation takes care of the device needs by reducing the resource consumption thanks to an ad-hoc implementation and to the reduction of the buffer capacity.

Furthermore, the change of device must also be considered in a situation of mobility. In fact, once the user has reached a place where a device with better multimedia features is available, he/she may decide to switch his/her mobile device and to use, for example, a set-top box. In this case, the concept of VS comes into play when the DVS is disabled. During the closing phase, the DVS stores the information about the suspended stream on a specific account. This account is related to the user, and is located in the wired area.

Once the set-top box is switched on, the DVS of the new device loads the information about the suspended streams. This way, the user can view or listen to the stream again, by using the new player.

When a connection is started, or a device is changed, or the network conditions change, the tailoring of the stream may be a problem. In this case, the VS system can act on the
Fig. 6. RTSP protocol management carried out by the VS during the server switching phase in the distributed streaming management.

flow. It can change the flow according to some parameters such as the sound quality, as well as the video size and quality. These changes can be done by using the Grid facilities provided by the Wired Area, in order to reduce the waiting time for the user. This operation is made transparently for the player that will use the most appropriate multimedia stream for the device where it is installed.

Furthermore, the VS allows to reduce the effects due to the signal loss, frequent in wireless environments. Buffering the data in the Access Area allows on the one hand to avoid the presence, on the mobile device, of large buffer (that cause the waste of the resources) and on the other to reduce the latency due to the data re-transmission from the server (in fact the data will be already available on the AP). In multimedia streaming for mobile devices the VS introduces flexibility and adaptability, with no need to require the design of multimedia players and of the server streaming. Since the VS is based on the standard protocols for this type of communication, any system (Player/Server) adopting RTSP/RTCP/RTP can be used.

The Wireless Area hosts the middleware components related to mobile devices. Due to the limited resources of such devices, these middleware components are reduced only
to an interface with the system. Besides the DVS described before, this Area hosts: the mobility management modules (only in the case of GPS systems), the user and device profile management modules, the graphical interface for the user - device interactions and a communication mechanism able to exchange data with the other middleware components present in the scenario.

Fig. 7 shows the internal architecture of the nodes belonging to the areas covered by the proposed middleware. Each node of the wireless area hosts a Device Manager module which may directly access primitives of the operating system (in low cost devices) or JAVA services (Standard Edition / Micro Edition), if more advanced devices are available. The Device Manager deals with information related to the user and the device itself, exchanges such information with the other nodes belonging to the Access Area and provides a multimedia proxy service in order to deliver transparent streaming services to the player.

Nodes belonging to the Access Area are equipped with a JAVA-based agent platform. The modules depicted in Fig. 7 (QoS Manager, Context and Location Aware Manager, User Position Manager and RTSP/RTCP/RTP Manager) are developed as software agents being executed in such agent platform. The QoS Manager manages the QoS policies active in each node, negotiating and adaptating QoS levels according to users’ desires. The Context and Location Aware Manager takes care to adapt multimedia contents to the client device and to deliver services depending on the user location. The User Position Manager reacts
to user’s movements, activating all the necessary procedures to ensure service continuity. Finally the RTSP/RTCP/RTP Manager manages the streaming operations.

Each Grid Node needs the Globus Toolkit that provides services for managing and monitoring distributed resources, such as the Globus Resource Allocation Manager (GRAM), and for the data transfer, such as the GridFTP. JAVA-based Mobile Agents interact with the Globus Toolkit through the Commodity Grid Kits (COG). The Grid Node also needs a Streaming Server module that supports the RTSP protocol, some Transcoding Tools to manipulate multimedia contents, and a Mobile Agents Platform, to manage and interact with the agents scattered in the middleware.

IV. AGENT COMMUNITY

Table I shows the agents present in the middleware and indicates their specific role, if they are mobile and which are the agents they are supposed to interact with.

<table>
<thead>
<tr>
<th>Name</th>
<th>Acronym</th>
<th>Area</th>
<th>Mobility</th>
<th>Interactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Agent</td>
<td>DA</td>
<td>Wireless Area</td>
<td>No</td>
<td>AF, UA, VSA, Media Player</td>
</tr>
<tr>
<td>Agent Factory</td>
<td>AF</td>
<td>Access Area</td>
<td>No</td>
<td>DA, UA</td>
</tr>
<tr>
<td>User Agent</td>
<td>UA</td>
<td>Access Area</td>
<td>Yes</td>
<td>DA, QA, GLA, VSA</td>
</tr>
<tr>
<td>QoS Agent</td>
<td>QA</td>
<td>Access Area</td>
<td>No</td>
<td>UA, QA</td>
</tr>
<tr>
<td>GeoLocation Agent</td>
<td>GLA</td>
<td>Access Area</td>
<td>No</td>
<td>UA, GLA</td>
</tr>
<tr>
<td>Virtual Server Agent</td>
<td>VSA</td>
<td>Access Area</td>
<td>Yes</td>
<td>UA, DA, GA</td>
</tr>
<tr>
<td>Streaming Agent</td>
<td>SA</td>
<td>Wired Area</td>
<td>Yes</td>
<td>VSA, GA, Streaming Server</td>
</tr>
<tr>
<td>Grid Agent</td>
<td>GA</td>
<td>Wired Area</td>
<td>No</td>
<td>SA, Grid components</td>
</tr>
</tbody>
</table>

TABLE I

AGENT COMMUNITY AND SPECIFIC ROLES

The purpose of the Device Agent (DA) is to manage all the communications between the user’s portable device and the Access Area. The task of the DA is to contact the AF listening on the corresponding Access Area’s node, requesting to enable an UA. From now on, the communication between the DA and the UA starts. Two types of messages are exchanged in the communication between the DA and the UA. The first type is a background one,
and consists of messages about the user’s position in the wireless network. The second one consists of any request for services sent by the user to the system.

The Agent Factory (AF) is a static agent, and is the user’s access point to the system. Once the AF is contacted by the DA, the AF starts the logging operations, and creates an UA. The AF has the task to manage the communications about the initialization of this procedure (confirmation messages, information about the resources, etc.). Furthermore, the AF manages the phase of the user’s disconnection. In fact, in this case the AF suspends the user’s UA, and saves its status in an opportune repository. This operation allows to save an image for the progress of the multimedia flow, as well as resuming it, if the user enables it again. This mechanism is also valid if the user changes his/her device between an interruption and a resuming. The AF checks for the presence of a suspended UA during the initialization, should the user log in again.

The User Agent (UA) is the intermediary of the user in the network. In fact, the UA is strictly linked to its owner, and follows the user from an AP to another. It performs all the task assigned to it. Once it is created or enabled, it waits for a connection from the DA, and establishes a direct communication with the DA through a socket. One of the main operations of the UA is related to the management of connections according to the position. In fact, the UA cooperates with the GLA, and can start any procedure for preparing the movement, anticipating the user’s handoff. These procedures consist of contacting the remote QA, in order to obtain the required bandwidth. As soon as the bandwidth is obtained, the UA can clone itself in the new AP taking its status and waiting for the user’s handoff. In order to carry out the binding operation, the UA will communicate the identity of the new UA to the DA. Furthermore, the UA has the task to activate the tailoring operations and to manage the multimedia streaming, by communicating with the VSA.

The GeoLocation Agent (GLA) is the agent that manages the information about the position of the mobile devices. This agent periodically monitors the signal level for each mobile device perceived by its AP. In an optimal design criterion for the wireless networks, some overlapping of the influence areas for each AP is expected. This allows to obtain a realistic position of the mobile devices, as well as some information about the direction of their movement. If a device has probably moved towards an area covered by a new AP,
the GLA provides the UA with the address of the AP where the user is moving to. If the device is equipped with a GPS system, it periodically will communicate its position. In this case the task of the GLA will be related to the management of such position data and to the discovery of the new AP reached by the user.

The Quality of Service Agent (QA) manages the QoS on a specific AP. This action is done through the reservation and/or negotiation of resources (in terms of bandwidth) on each AP involved in the multimedia flow. The QA carries out the admission control mechanisms dealing with the QoS in the Access Area i.e., with the wireless resources. The QoS requested in the Wired Area is not involved. When a user has to move to another cell, the new QA is contacted by the UA and a new negotiation phase starts.

The Virtual Server Agent (VSA) has the task to manage, handle, and readdress (if necessary) the data/control flow (RTSP, RTCP and RTP) transparently for both the player and the servers. Its main task is to make the distributed streaming transparent to the user by communicating with the SAs in order to know the features of the video chunk (chunk number, duration). During users’ handoff, the VSA migrates on the new AP together with its status, i.e., the point where the streaming was arrived and the coordinates of the next video chunks.

The Streaming Agent (SA) manages the streaming operations in the Wired Area by activating the streaming server in each Grid Node. It is downloaded on each Grid Node involved in the distributed streaming phase together with the corresponding video chunk.

The Grid Agent (GA) manages the Grid resources of the VO: it starts the load balancing policies, carry out the Multimedia Upload and Storage activities, activates the data transcoding and manages the dissemination of the video chunks to the Grid Node downloading the corresponding SAs.

Here are the steps that are followed in a typical streaming session. The user uses the DA to request for the streaming of a video, selecting among a list updated in real time by the UA. The UA manages such list by updating the available movies chosen according to the user’s preference and position. These informations will be available through a distributed directory systems (such as LDAP), located in the Wired Area. The streaming request is forwarded by the DA to the UA that contacts the GA resident on the Grid Node where
the movie is available. The UA communicates to the GA the mobile device profile and the GA responds by indicating the bandwidth needed to the streaming. At this point the UA starts the negotiation phase with the QA in order to obtain the necessary QoS level. Once the required bandwidth is obtained, the GA will carry out the tailoring phase (if the movie is not already available in the right format) by distributing the chunks and by sending the SAs to the corresponding Grid Nodes. The UA, informed about the SAs’ coordinates, will create the VSA and will send an ACK to the DA. The DA forwards the streaming request to the user’s favorite multimedia player with a request like rtsp://localhost:554/myvideo.mpg. The request for the multimedia player is therefore done locally. At this point the streaming starts. The VSA will manages the distributed streaming phase by switching the streaming between the Grid Node involved. If the user makes a handoff during a streaming, the UA, contacted by the GLA, clones itself to the new AP together with the VSA. The new VSA is ready to resume the streaming to the DA. If some agents are not involved any longer in the communication, they can be destroyed after a time-out, and the reserved resources can be released [34]. If a new server must be selected (variations in the network conditions, in the load of the server involved, or in the user’s position), the multimedia flow is readdressed by the VSA. This operation is totally transparent for the media player (and for the user, too), since the user can still watch the stream from rtsp://localhost:554/mystream.mpg. The management of the server switching will be done entirely by the VS (DVS + IVS). As a consequence, a connection is closed, and another is opened starting from the point where the stream is timed out.

The proposed middleware consists of a community of JAVA mobile agents. In order to create the mobile agents of the community, the MAP platform [27] has been used which provides all the tools useful for mobility and communication, and for cloning the agents. Furthermore, the MAP provides the mechanisms needed to guarantee an adequate level of security in communications, as well as the opportunity of managing several access policies for system resources. Each agent is therefore implemented by a JAVA class consisting of a core (running the tasks of the agent) and of some methods that can respond to the events (incoming messages, pausing the agent, migration, etc.). One of the most important features of mobile agents is cooperation. In fact, this enables a community of agents to exchange
some information, by sharing their knowledge. In a system like the one presented in this paper, the interaction among the agents plays an important role. In fact, agents must be able to exchange information about the user, the device used, the video requested, the fixed level of QoS etc. Two types of messages exist in the MAP: synchronized and non-synchronized messages. The synchronized messages cause the sender to stop running, until a reply is received from the receiver. Conversely, the non-synchronized messages do not stop any activity of the sender. Both types of messages have been used for creating the middleware. Figures 8 and 9 outline the interactions among the agents involved in a new user’s login procedure and in the handoff operations. The dashed curves refer to the non-synchronized messages, while the ones with a continuous line refer to the synchronized messages.

![Figure 8](image.png)  
**Fig. 8.** Interactions among the agents when a user accesses the system.

The agents present in the system can be divided into two categories: mobile and static agents. A third type of agents present in the system includes the agents in the mobile device (DA). It is needed to outline that the installation of a MAP on the mobile device is not required. In fact, the DA is first of all a static agent. Consequently, no mobility mechanisms are required. Furthermore, the DA is the only agent needed on the device. Finally, the system needs to not increase the load for mobile devices, whose resources and energy are
The communication between the AP and the mobile devices takes place through sockets or encrypted communications. This way, the DA can be implemented with any language compatible with the available hardware, with no need to be bound to JAVA language. Furthermore, the DA acts as a graphic user interface, and shows the list of the available movies. This way, a movie can be selected, and the player can be automatically enabled. A screenshot of the DA is shown in Fig. 10.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In order to test the actual operation of the agent middleware and its integration with the concept of Virtual Server, a testbed in our campus has been created. This testbed reflects the scenario analyzed in Fig. 1.

The Wired Area consists of a 100Mb/s LAN. Some Streaming Servers are present in it. Each streaming server, based on a Pentium IV architecture with Red Hat Linux (Kernel 2.4.18-3) OS and 512MB Ram, has been equipped with RealNetworks Helix Universal Server Basic. The Wired Area also includes the distributed directory services, as well as a DB server used as Agent Repository.

The Access Area is implemented by using several Linux-based APs. Several types of antennas (directional and omnidirectional) have been installed on each AP, in order to
optimize the coverage with a limited number of APs. More than one antenna has been necessary on some APs, in order to increase the bandwidth available in the busiest areas. Each AP is connected to the Wired Area through Ethernet Protocol, and provides the DHCP service for mobile devices. The Access Points used for setting up the experimental environment adopt a Pentium MMX @ 166 MHz architecture, 128 MB of RAM and 64 MB DOM (Disk On Module) of mass storage, and are equipped with an embedded version of Linux based on the kernel 2.4, and with JRE 1.2.2.014.

The Wireless Area extends on the whole campus area, covering both the premises and the surrounding area. Several Clients can move within this Area. They consist of wireless devices equipped with Multimedia Players. The clients considered are laptops based on a Pentium III architecture, and Mobile Handhelds. Each laptop is equipped with the following Players:

- QuickTime Player 5.0.2 (Windows).
• RealNetworks RealOne Player + Envivio Plugin (Windows).

The Handhelds used are Compaq iPAQ 3870 equipped with Windows PocketPC 2002 OS, Wireless card 802.11b, InterObject SMIL Player, and RealOne Real Player. In order to create the agent community, the MAP platform [27] has been installed on each AP by using Sun’s JAVA Virtual Machine version 1.2.2.014. The size of the agents created in the middleware varies between 2 and 9 KB, excluding buffers and/or internal variables. The small size makes the migration of these agents easy and effective. The Device Agent has been created for each mobile device, by using the language C++. This allowed to improve the effectiveness of devices with limited features. In particular, the embedded tools of Visual Studio 6.0 have been used for the handhelds using Windows PocketPC 2002 as operating system. In order to allow the Device Agent to communicate with the other agents present in the architecture, a communication protocol has been established, based on sockets.

Several movies of different resolution and duration have been made available in the system, to test the basic functionalities and the performance offered. The middleware has shown to be able to work with several types of coding such as MPEG4, Real Media, MP3, MPEG1, and SWF.

In order to test the policies used in the middleware, some viewing sessions have been executed, by accessing the services through different devices. These sessions have highlighted that the system was able to adapt the multimedia flow to the features of the device used and to the user’s actual position in the Wireless Area, thanks to the exchange of information among agents. This did not dramatically affect the times for accessing the service, and has caused delays for a few seconds.

In Fig. 11 the service accessing time is plotted varying the number of active users¹ present in the cell from 0 to 5. Two different situations are shown with and without the use of the VS. It can be observed that the use of the VS introduces a delay in the service accessing time of about 0.6 seconds more than the value observed without the use of VS and that this value remains approximately constant with the increase of the number of users. This difference causes a low delay between the user request and the beginning of the streaming.

¹The term active user refers to a user that is doing a streaming.
that does not affect the service provisioning. Besides, the system is not affected by the number of users present in the same cell, and reaches a high level of scalability.

![Service accessing time with and without VS vs. number of active users.](image)

Fig. 11. Service accessing time with and without VS vs. number of active users.

These sessions have been characterized by a high level of user mobility - who switch among the APs many times (handoff), with temporary loss of signal. The middleware has proven to be effective in managing such situations. In fact, it does not significantly affect the users perception during handoffs, and resumes the viewing from the point it was stopped, if the signal is lost.

In Fig. 12 the re-arrange time after a handoff is shown. This time is related to the signal loss and reflects the time needed for the middleware to carry out the agents’ migration and the streaming management. This measure has been realized using two adjacent cells. A user starts a streaming session, and after, makes several hard handoffs\(^2\) between the adjacent cells. The re-arrange time is calculated varying the number of active users in the cells (the x-axis refers to the number of users present in each cell) and under three different strategies:

- reactive - the UA anticipates the user’s handoff but waits fot the user before starting the streaming management tasks;
- proactive - the UA anticipates the user’s handoff, starts the preliminary streaming management tasks and waits the user before sending the PLAY RTSP message;

\(^2\)The term hard handoff refers to a handoff between different subnets with consequent change of IP address.
• proactive with buffer - the UA anticipates the user’s handoff, starts the streaming management tasks and bufferizes the data until the user actually makes the handoff.

It can be observed that the time related to the signal loss decreases remarkably when switching from the reactive strategy to the proactive; on the contrary, the difference between the two proactive strategies (with and without buffer) is negligible and does not justify the use of buffers due to their high ”cost”. Another aspect that can be easily observed is that the re-arrange time increases with the increasing of the users due to the number of agents and VSs present in both the source and the destination APs. Besides, it is important to notice that the number of users accepted by the streaming server is double regarding the case of Fig. 11, as the number of cells is double. However, analyzing the values of the re-arrange time (that is the time of signal loss) it is important to notice that the streaming re-starts from the point it was timed out and that the same hard handoff will cause, without the use of the proposed middleware, the unconditional suspension of the streaming session.

Fig. 12. Re-arrange time after a handoff vs. number of users under three different strategies.

The perception results obtained are supported by some measurements about the VSA (which is considered one of the most important elements of the architecture), and which confirm its effectiveness and reliability.

Such measurements regard the delays introduced by VSA with reference to both RTSP control packets and RTP data packets, and the mean CPU load for the PC where the VSA is executed.
Table II shows the results observed in the CPU load, due to the stream processing at low and high bit rate (128kbps and 350kbps respectively).

It easy to observe that each stream influences the CPU load around 2% to 3%. The evolution of the CPU load when time varies is shown in figures 13(a), 13(b) and 13(c): the average values that can be observed are the ones of Table II.

Another group of experiments aimed to evaluate the overhead introduced from RTSP control packets. The results of the trials concerning the RTSP packets are shown in figures
### Table II

<table>
<thead>
<tr>
<th>Environment</th>
<th>MeanCPU Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Stream with 128kbps</td>
<td>2.24035%</td>
</tr>
<tr>
<td>Single Stream with 350kbps</td>
<td>3.41535%</td>
</tr>
<tr>
<td>3 Streams with various bit rates</td>
<td>8.68554%</td>
</tr>
</tbody>
</table>

**Mean CPU Load under different working conditions**

14(a) and 14(b), assuming 1 video stream and 5 simultaneous video streams, respectively at 150kbps and 128kbps. Fig. 14(a) shows that the VSA is able to process more than 70% of RTSP packets with a delay shorter than 20ms. With 5 video streams active at the same time, Fig. 14(b) shows that the VSA can process RTSP packets with a delay shorter than 20ms in about 70% of the cases. This confirms that the performance of the Virtual Server scales quite well when the number of simultaneous user sessions increases.

Figures 15(a) and 15(b) show the delays introduced by the VSA on the flow of RTP packets, assuming a single user with a low bit rate session. Fig. 15(a) shows the processing time for each single packet. It can be observed that most of the packets (roughly 95%) experience a very short delay of 1.3 ms (the value almost coincides with the x-axis). The lower part of Fig. 15(b) shows a histogram of RTP packets with a given processing time; the upper plot is the cumulative distribution function of the processing times, i.e., the probability that a RTP packet experiences a processing delay shorter than or equal to a given value. It can be noticed that there is a probability greater than 60% that a RTP packet is processed in less than 400µs.

In order to test the fluidity of the stream, the processing bit rate of the VSA has been measured. The results are shown in Fig. 16 where it can be observed as about the 6% of RTP packets is processed by the VSA at a bit rate lower than the bit rate of low resolution video (128kbps). This result proves that most of the RTP packets (94%) are processed more quickly than what is required by the player, and that the VSA does not become an obstacle in communication, and leaves the streaming characteristics unchanged.
To conclude, the testing phase has demonstrated that the QoS perceived by a user is practically the same under the different conditions considered. This is mainly due to the low delays introduced by the middleware, which is indeed an interesting strategy to be adopted.

VI. CONCLUSIONS

Recently, wireless technology has become mature enough to satisfy the users’ needs for mobility and information anywhere and at any time. Mobile computing is diffusing very quickly and efforts are being done to allow mobile users to access advanced services such as multimedia streaming. In order to manage the challenging problems related to the management of such new environments, this paper has presented an agent-based middleware that allows to solve the main issues for the streaming of multimedia flows, notwithstanding the device used and the user’s moves. In order to achieve this purpose with an adequate level of transparency, and considering the heterogeneity of the devices and of the networks, mobile agents have been assumed as the underlying technology, thus exploiting their interesting features of mobility, asynchronicity and autonomy. The Grid Paradigm has been used in order to improve the management of the wired resources involved in multimedia services.
provisioning, in particular to carry out tasks as distributed format transcoding, data dissemination and load balancing. Thanks to this middleware, while roaming from a cell to another, a mobile user is constantly connected to the most convenient resource, which is dynamically identified in order to guarantee adequate levels of QoS. Some mechanisms of Context and Location Awareness have been discussed. The implementation of this middleware and its integration with the current multimedia clients and servers has raised the need
to manage the data streaming at a lower level. Hence, a RTSP/RTCP/RTP manager that acts as an intermediary level between the client and the server has been designed and implemented. Some experimental trials have been conducted in order to test the validity of the proposed solution and its applicability in a real scenario. The middleware described is being implemented in an experimental environment, developed at the Engineering Faculty of the University of Messina under the Campus One project, which aims to guarantee Faculty members and students wireless access to multimedia data, while roaming inside the whole University Campus.

REFERENCES


