A Demonstrator of an IP-based Access Network for Broadband Multimedia Services

Roger Salgado[a], Rui Valadas[b], Susana Sargento[c], Ricardo Cadime[d], Jorge Gonçalves[e]

[a] Institute of Telecommunications, Portugal
[b, c] University of Aveiro / Institute of Telecommunications, Portugal
[d, e] Portugal Telecom Inovação / Institute of Telecommunications, Portugal

Abstract

This paper describes a demonstrator of an IP-based access network for broadband multimedia services, which was designed to be low cost and easily manageable. In order to support broadband multimedia services, the network architecture incorporates a number of newly introduced technologies: SIP (Session Initiation Protocol) for session initiation, RSVP (resource ReSerVation Protocol) for resource reservation, and COPS (Common Open Police Service) for QoS policy management and AAA (Authentication, Authorization and Accounting). Some modules of the access network elements were implemented using the VOCAL (Vovida Open Communication Application Library) system, developed by Vovida. We report both functional and performance results obtained with the demonstrator, and traffic management experiments related with the policing, packet scheduling, resource reservation and admission control mechanisms.

Keywords: Access Network, QoS, Multimedia, SIP, RSVP.

1. Introduction

The exponential growth of the Internet is pushing for a migration towards IP-based access networks capable of supporting multiple services with different QoS requirements. Presently, Internet access is mainly dominated by dial-up connections. In the incumbent traditional telecommunications operator, a more advanced solution that combines the use of ATM (Asynchronous Transfer Mode) and *x*DSL (*x* Digital Subscriber Line) technologies supported on copper or copper and fiber, is also being deployed. These solutions provide only for limited resource sharing in the access network, since service separation is achieved through reserved bandwidth circuits. Moreover, since ATM is a connection-oriented technology, it complicates the management of access networks, due to the need to establish and manage virtual circuits. This problem is aggravated if on-demand resources are required (i.e., when using VB5.2). It is expected that using IP as the base technology in access networks, the ideal plug-and-play scenario, where the management actions of the access network operator are kept to a minimum, will be achieved easily. However, migration towards IP-based access networks has to be done smoothly, reusing as much as possible current technologies. Also, a major factor to be taken into account in the design of access networks is the cost of the network elements and of the signaling functions.

The design of a technological solution for next generation access networks is deeply constrained by cost factors, because of the large number of network elements that need to be replaced or newly deployed. Traditionally, access network technologies have provided for very limited resource sharing. For example, in dial-up networks users are completely isolated from each other and, in ADSL networks, the broadband and voice services are separated at the physical layer. The lack of resource sharing can be mainly attributed to the cost of signaling. Resource sharing at the access network can help to reduce the cost of bandwidth but requires more functionality and, therefore, more complex and costly equipment. In future access networks, the critical balance between cost and functionality has to be carefully considered.

In [1] we proposed an architecture for IP-based access networks with QoS support, targeted for the integrated support of broadband multimedia services, and designed to be low-cost and easily manageable. This proposal included the detailed definition of the network elements and its functions, and the technologies supported in the access network. In this paper we describe the implementation of a demonstrator for the proposed IP-based access network architecture. We also report both functional and performance results obtained with the demonstrator, and traffic management experiments related with the policing, packet scheduling, resource reservation and admission control mechanisms. The results validate the correct operation of the access network, and its integrated support of multimedia services with differentiated QoS. Some modules of the access network elements were implemented using the VOCAL (Vovida Open Communication Application Library) system, developed by Vovida [2].

The remainder of this paper is organized as follows. Section 2 presents the proposed access network architecture and section 3 describes the operation of the VOCAL software modules that were used in the implementation of some access network elements. Section 4 presents the developed demonstrator and

section 5 reports the experimental results. Finally, the conclusions are presented in section 6.

2. Access network architecture and technologies

Figure 1 shows the architecture of an envisaged IP-based access network, which was designed having the cost factor in mind, while still providing for the integrated support of broadband multimedia services.

The main elements of the access network architecture are the NTs (Network Terminations), the IP-MUXs (IP Multiplexers) and the BAS (Broadband Access Server). The use of IP multiplexers, instead of IP routers, helps simplifying a number of functions (e.g. routing and addressing) and to reduce the cost of the access network. However, one consequence is that there is no network layer redundancy (the access network is a logical tree at the IP layer). We do believe that this is not a requirement for access networks, and that these networks can rely on physical layer redundancy, such as the one provided by SDH.

In order to support broadband multimedia services, the access network has to incorporate a number of newly introduced technologies: SIP (Session Initiation Protocol) [3] for session initiation, RSVP (resource ReSerVation Protocol) [4] for resource reservation, and COPS (Common Open Police Service) [5] and/or DIAMETER [6] for QoS policy management and AAA (Authentication, Authorization and Accounting).

The recent introduction of an RSVP extension to support flow aggregation, allows the use of RSVP in mixed IntServ / DiffServ environments, which provides a flexible framework for establishing trade-offs between cost and performance in the access network.

The signaling protocol used to establish, modify and terminate multimedia sessions is SIP. The main elements of the SIP architecture are user agents (UA), registrars, and proxy and redirect servers. The user agents are the end-points of sessions; registrars allow for user registration; proxy servers are application-layer routers that forward SIP messages; and redirect servers return alternative locations of user agents or servers. SIP supports some functions that are important in access networks. For example, it supports user registration, which allows for pre-call terminal mobility, and it can be synchronized with the resource reservation and the AAA process. In SIP users are identified by URLs, which are similar to e-mail addresses (e.g. roger@av.it.pt). In each session, the URL is resolved to an IP address by using the SIP proxy server and DNS lookups. SIP uses a description format, called SDP (Session Description Protocol) [11], to allow each party to declare its receiving capabilities and the characteristics of the media streams it wants to receive.

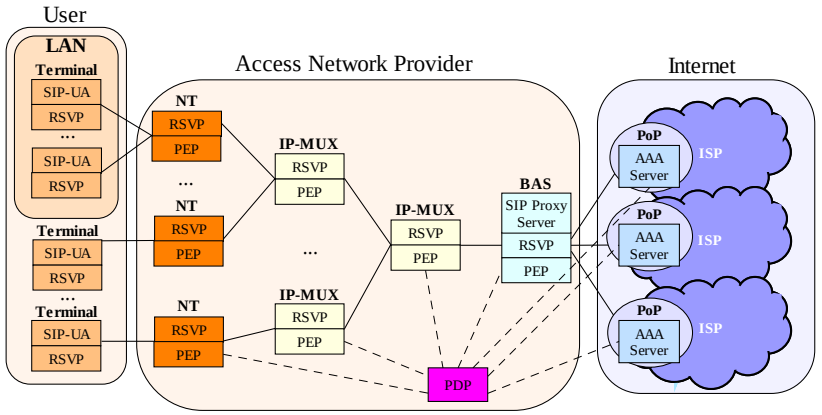

Figure 1 –Access network architecture.

SIP includes mechanisms, called preconditions, for coordinating the session signaling and the establishment of end-to-end resource reservations or security tunnels. A precondition is a condition that must be verified by one or more users. The progress of the signaling process can be made dependent on the success of this condition. There are two options for the interaction between session signaling and resource reservation, resulting in so-called QoS assured or QoS enabled calls [12]. In QoS assured calls, the call may only start after resource reservation has completed successfully. In QoS enabled calls, the call setup and resource reservation are decoupled and may proceed concurrently. In case of resource reservation failure, the caller can be notified and given the option of continuing the call with best effort service only.

A multi-service network with QoS support places additional requirements in the AAA functionality. These functions need now to be performed on a user/service/QoS level basis. For example, a service may be authorized only for some users, with specific QoS requirements, between authorized points, and only at agreed-upon times [13]. The current trend is to include the AAA functions under the scope of a QoS policy management framework. The IETF framework [14] defines two main architectural elements for policy control: the PEP (Policy Enforcement Point) and the PDP (Policy Decision Point). The PEP is the element that enforces the policy decisions, and the PDP makes decisions based on the policies it retrieves from policy repositories, AAA servers and other entities. The policy repository is a remote database such as a directory service or a network file system. The PEP is a component of a network node

and the PDP is a remote entity that may reside at a policy server. Usually there is one PDP in a network domain and several PEPs. COPS is the suggested protocol to exchange information between the PDP and the PEP.

In the proposed access network architecture, all access network elements are RSVP-aware and include a Policy Enforcement Point. In addition, a SIP proxy server is collocated with the BAS. The BAS has access to a Policy Decision Point, which contacts the AAA servers at the ISPs and the Policy Repository. Packet scheduling is required at all network elements to discriminate the various types of traffic. We consider a hierarchy with strict priority scheduling at the first level, and Weighted Fair Queuing at a second level, to handle the user traffic with QoS requirements.

3. Vovida Open Communication Application Library

The software of the SIP user agents and BAS was taken from the Vovida Open Communication Application Library (VOCAL) software project. The VOCAL implementation supports multimedia services through RTP (Real-time Transport Protocol) [8]. The VOCAL system is an open-source project formed to aid the VoIP adoption by the market. VOCAL provides tools and the base software required to build new VoIP applications, services and features. VOCAL has some advantages compared to other implementations, such as its versatility, stability, and cost.

Beyond the software to implement the user agents, VOCAL also provides the following modules:

- Redirect Server (RS) - provides SIP registration and call routing services.
- Marshal Server (MS) - acts as a trusted boundary for calls entering or leaving a network; it provides authentication and collects billing information for the CDR (Call Detail Record Server) server.
- Provisioning Server (PS) - provides, configures and manages subscribers and servers from a GUI (Graphic User Interface).
- Feature Server - provides CPL (Call Processing Language) based scripts that run basic telephony features.
- Policy Server - uses COPS to make policy decisions to enable or disable QoS across network routers. Uses OSP (Open Settlements Protocol) to negotiate with clearing houses [7] for inter-network settlement.
- CDR Server - collects billing information from MS and interfaces with billing systems using the RADIUS (Remote Authentication Dial In User Service) accounting protocol [9].

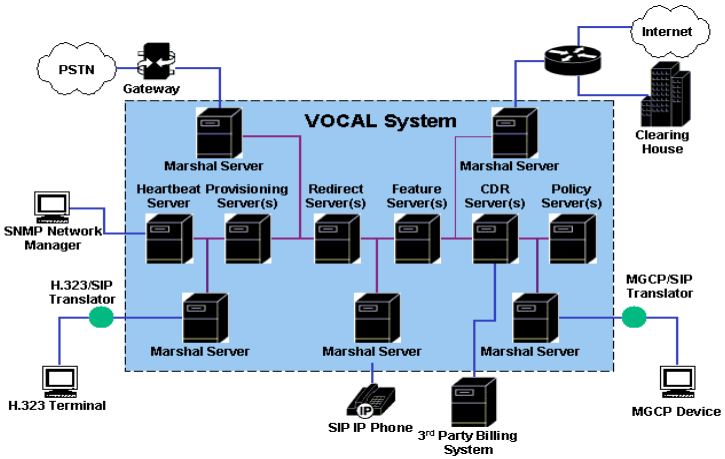

Figure 2– Vocal system.

- Heartbeat Server - monitors heartbeats sent by other servers and forwards this information to an SNMP (Simple Network Management Protocol) network manager.
- H.323 Translator - acts as a Gatekeeper to control H.323 endpoints; talks SIP to the rest of the network for routing and features.

The overall Vocal system is illustrated in Figure 2. All these functionalities will be implemented in the BAS. In this section we will use the VOCAL notation when referring to the SIP servers.

When a UA connects to the network, it must be registered in the domain. For this purpose, the UA sends a SIP REGISTER message to the MS. If the MS does not have the UA's IP address record in its database, the MS must get this information from the PS. The MS generates a Lookup message and sends it to the PS, and this one answers with a Retrieval message containing the UA's record. Upon receiving this message, the MS creates an IP address UA record and adds it to the database. It is possible to raise the system security during all this process using special registration messages. Those messages must incorporate the UA credentials. Considering that the UA does not send its credentials in the SIP REGISTER message, the MS will answer with the SIP 401 Unauthorized message indicating that credentials are required for registration. At this point, the UA generates a second SIP REGISTER message with its credentials and sends this new message to the MS. The MS will

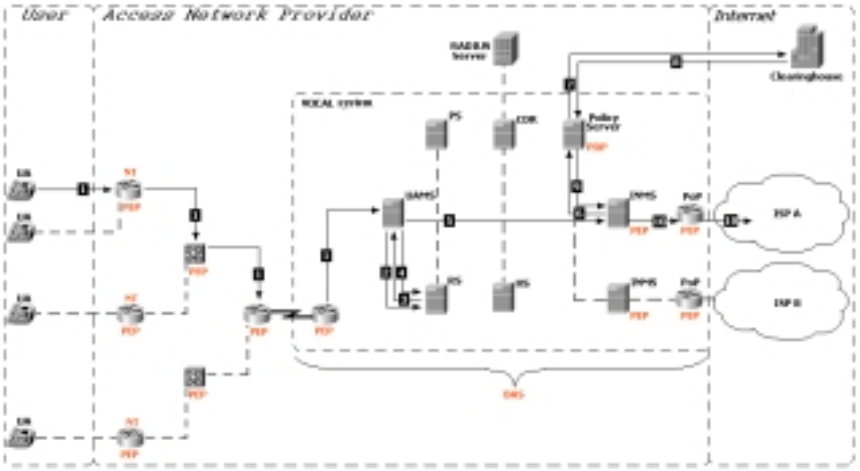

Figure 3– Establishing a call, sender side.

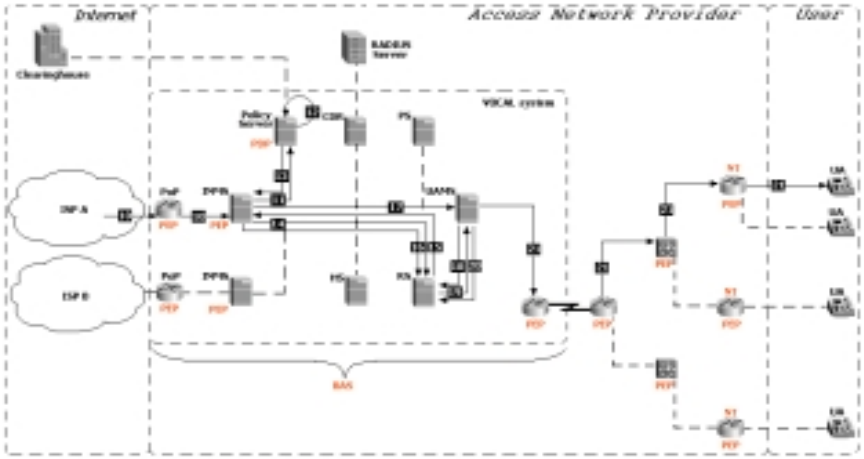

Figure 4– Establishing a call, receiver side.

authenticate and redirect this SIP REGISTER message to the RS. At this point the RS answers with the SIP 200 OK message in order to notify the UA. Finally, this message is redirected to the UA by the MS.

After UA registration, the process of call establishment may start. Figure 3 illustrates the call establishment process at the sender side. Consider the example of a bi-directional conference call. To initiate a call the UA sends a SIP INVITE message (message 1). The MS captures this message, authenticates the UA and redirects the message to the RS (message 2). The RS then answers with a SIP 302 Moved Temporarily message (message 3) that has additional forwarding information. When the MS receives this message it sends an ACK message (message 4) to the RS notifying that the SIP 302 Moved Temporarily message has been received. From the received information, the MS can now redirect the SIP INVITE message to its destination (message 5). This process can easily be extended to work with several ISPs, making use of an external clearinghouse and the COPS protocol.

When the UA wants to establish a call whose destination is located outside the system, the MS must forward the SIP INVITE message to the respective Internet gateway (message 5). The UA can choose the ISP at the UA registration process. If the UA registers itself as, for example, *roger@av.it.pt*, the domain *av.it.pt* defines which ISP the UA wants to connect. It may also be necessary to send a subscription message to the ISP to confirm the choice made.

The VOCAL system supports this kind of messages natively. The VOCAL gateway, or more precisely, the Inter-Network Marshal Server (INMS), receives the SIP INVITE message, generates an authorization request using COPS protocol and sends it to the Policy Server (message 6). Upon receiving this message, the Policy Server creates an authorization request using the OSP protocol and redirects it to the Clearinghouse (message 7). The Clearinghouse verifies the route, by confirming that the dialed digits are correct, and answers with an OSP Authorization message plus a token (message 8). The authorization token is signed with a Clearinghouse certificate and a private key. It is used for Clearinghouse validation and authentication purposes only. After this process, the Policy Server generates a COPS decision, incorporates the received Clearinghouse token in the decision, and sends this message to the INMS (message 9). The INMS now adds the received token to the SIP INVITE message (message 10) and sends it to the corresponding ISP.

In the destination side, illustrated in Figure 4, the SIP INVITE message containing this token is forwarded to the INMS. Upon receiving this message, the INMS generates a COPS request (message 11), includes the token in the message, and sends it to the Policy Server for verification. The Policy Server verifies the received token through its OSP client. The criteria used in the verification may be the source, the destination and the Clearinghouse name. After verification, the Policy Server generates a COPS decision and sends it to

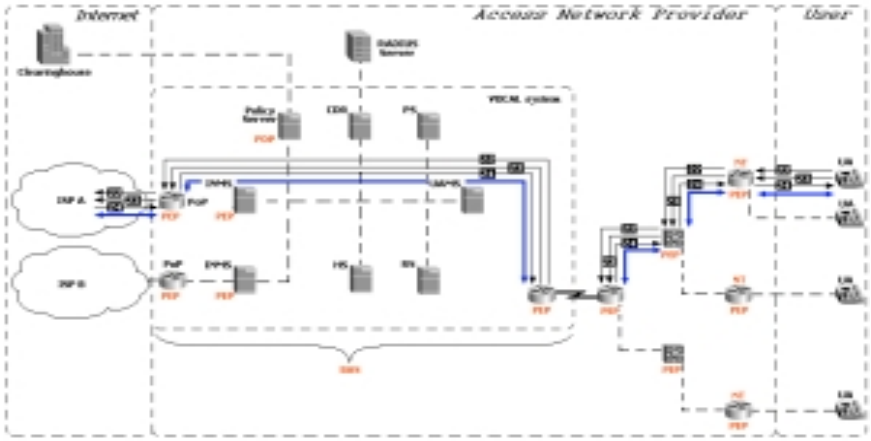

Figure 5– Path establishment, receiver side.

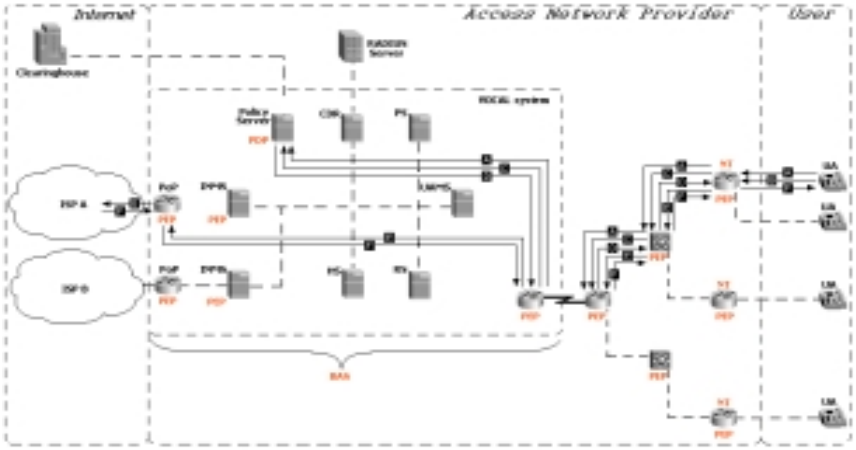

Figure 6– Resource reservation, receiver side.

the INMS (message 13), which removes the token from the SIP INVITE message header and redirects this new message to the RS (message 14). The RS then answers the request with the SIP 302 Moved Temporarily message (message 15), which contains additional forwarding information. When the INMS receives this message, it sends back to the RS (message 16) a confirmation message, redirects the SIP INVITE message to the MS (message 17) and, finally, this last server redirects the received message to the destination UA through the RS (messages 18, 19, 20 and 21).

Figure 5 illustrates the path establishment in the destination side. Upon receiving the SIP INVITE message, alerting starts in the destination UA, and the sender UA is notified through a SIP 180 Ringing message (message 22). Note that in Figure 5 the return messages do not traverse the SIP proxies, since both sender and receiver already know the other party's IP address. In SIP it is possible to force all messages to traverse the proxies, e.g., to collect the data necessary to the *CDR* server.

Upon answering the call, the destination UA sends back to the sender a SIP 200 OK message (message 23), and the sender answers with a SIP ACK (message 24). At this point the audio path is established. In Figure 5, for simplicity, we do not show the exchanged messages in the sender side.

Since the implemented access network supports QoS differentiation, resources need to be reserved during call establishment. The VOCAL system only supports the QoS enabled call scenario. For the purpose of resource reservation, both sender and receiver, considering a bi-directional call, send PATH and RESV messages in the call establishment phase. If the resources cannot be allocated for the call, the call may continue with the Best Effort service. This process is illustrated in Figure 6. In our example, in order to reserve resources, the sender and receiver generate a COPS message asking the Policy Server to establish a call with QoS (message A). Following, the sender and the destination UA send to its LAN router a PATH message in order to establish a reservation state (message B). The router then generates a COPS-RSVP request and sends it to Policy Server (message C), which answers with a COPS decision (message D). If the request is accepted, the router forwards the RSVP-PATH message (message E) to the UA in the other side. This UA answers with a RSVP-RESV message (message F) and reserves the required bandwidth. Once more, for simplicity, Figure 6 only shows the exchanged messages in the destination side.

The developed demonstrator allows the establishment of calls with our without QoS. It may also provide the authentication, authorization and accounting services for the calls. As it can be seen in the previous figures, it is possible to connect the demonstrator to an external Radius server that allows external billing systems to collect call records information. Those records are created and maintained by the CDR server that collects information from the proxies while calls are active. This process requires that both call parties belong

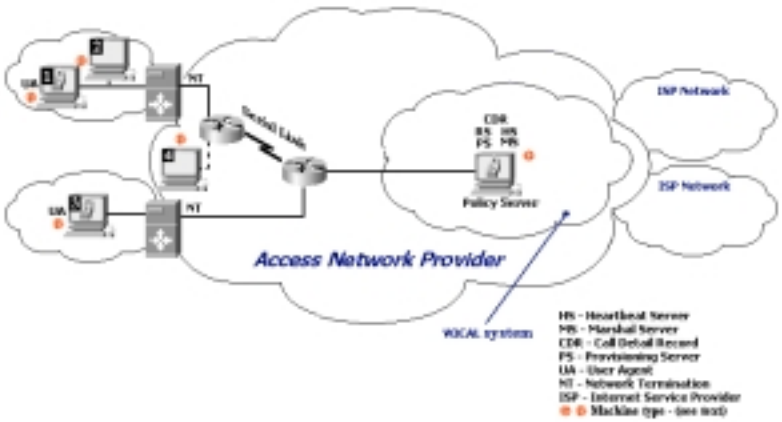

Figure 7– Testbed scenario.

to the same network domain. External calls are authenticated, authorized and accounted by the Clearinghouse that collects information from the gateways using the Policy Server.

4. Demonstrator

The demonstrator is illustrated in Figure 7. The NTs are based on Linux routers, running RedHat Linux 7.1 operating system, enhanced with RSVP daemon ISI RSVP Rel4.2a4. These routers are configured with static routes. In order to handle the user traffic with QoS requirements, they are also configured to use strict priority scheduling discipline in a first level, and CBQ (Class-Based Queuing) discipline in a second level, which is implemented with the TC (Traffic Control) [10] configuration tool.

The RSVP distribution is an RSVP API (Applications Programming Interface) that includes one RSVP signaling daemon named 'rsvpd' and several RSVP management tools. Both terminals and routers have an RSVP daemon running on the system, which can be controlled by any application via an RSVP API. There are APIs both for Windows and Unix based systems. In UNIX systems, raw IP or UDP sockets are used between the RSVP daemon and its neighborhoods. The RSVP daemon used requires a specific CBQ traffic class handler.

The IP-MUXs are simulated using CISCO 3620, running IOS version 12.2-5 operating system. Those routers are configured to use strict priority in a first level and CBQ in the second level. They also support RSVP. The connection

Figure 8– Traffic classes: strict priority and CBQ.

between the IP-MUXs is done through a DTE/DCE serial cable, allowing the configuration of the serial link capacity.

The BAS is implemented using the VOCAL system. The BAS includes all the servers described in section 3. The UAs are also implemented using the VOCAL system. These elements include support for RSVP signaling.

All PCs have secure shell (SSH) support. The BAS is implemented in a high performance PC because it includes all the VOCAL servers. It runs on a PC Pentium at 1.7 GHz with 512 Mbytes of RAM. This PC was previously installed and configured with the following software modules:

- Apache server;
- JDK ≥1.3 for Linux platform;
- Netscape Web Browser version ≥ 4.6;
- Java Plug-in ≥ 1.3.

Linux DNS (Domain Name System) configuration is kept locally into */etc/hosts* configuration file. Future practical implementations will require a dynamic DNS server and a DHCP server.

We already referred that, to handle the user traffic with QoS requirements, all routers need to be configured with a hierarchy of scheduling disciplines, with strict priority scheduling at the first level, and CBQ at a second level. Figure 8 illustrates the scheduling disciplines applied to the traffic classes. The strict priority scheduler differentiates between the best effort traffic, the reserved traffic and the signaling traffic. The lowest priority is assigned to the best effort traffic, and the highest is assigned to the signaling traffic. There is also one class, the isolation class, with no specific priority that includes the remaining link bandwidth. Each priority class is further configured with the CBQ discipline to differentiate between the traffic from different users or different applications that belong to the same priority class.

5. Experimental Results

In order to validate the demonstrator we performed several qualitative and quantitative tests. The qualitative tests assess the global system operation including the support of differentiated QoS. In the quantitative tests we measured, under various conditions, some performance metrics of interest, such as the throughput, the delay, the jitter, and the packet loss.

Both types of tests consisted on the establishment of best effort calls and calls with QoS requirements. We analyzed qualitatively and quantitatively the impact of the QoS calls on the best effort ones and vice-versa. In both experimental tests, there is background traffic (with best effort service) from the PC4 to the PC2, and traffic with QoS requirements (with reserved resources) from the UA3 to the UA1. Since the destination of both traffics is located in the same LAN, if the bandwidth available for both best effort and reserved traffic is not sufficient, the best effort one will be degraded.

In the qualitative tests the background traffic consists of an FTP session from the PC4 to the PC2, and the traffic with QoS consists of an audio transmission (CD music) from the UA3 to the UA1. In the quantitative tests both background and QoS traffic are generated with the traffic generator Mgen [15] that includes support for RSVP signaling.

5.1. Qualitative tests

In these tests, we first established an audio call with QoS. Then, when the audio traffic was being transmitted between UA3 and UA1, we established an FTP session between PC4 and PC2, and transferred a large file, which completely congested the serial link.

Before establishing a call, each UA must registers in the domain. Figure 9 shows the messages sent during UA registration. We observe that the UA begins the registration process by sending a REGISTER message to the proxy server (centaurus.av.it.pt) in port 5060, which is the default port for SIP communication. The proxy answers with a trying notification and after registration is completed, it sends an OK message to the UA.

After registration, UA3 starts the call establishment process. Figure 10 shows the messages exchanged during the call establishment and termination

No.	Time	Protocol	Info
1	0.0000	SIP	Request: REGISTER sip:centaurus.av.it.pt:5060
2	0.0000	SIP	Status: 100 Trying
3	0.0100	SIP	Status: 200 OK

Figure 9–SIP messages sequence during UA registration.

No.	Time	Protocol	Info
1	0.000000	SIP/SDP	Request: INVITE sip:1111@centaurus.av.it.pt:5060, with session description
2	0.010000	SIP	Status: 100 Trying
3	0.030000	SIP/SDP	Request: INVITE sip:1111@100.1.1.225:5060, with session description
4	0.040000	SIP/SDP	Status: 183 Session Progress, with session description
5	0.040000	RSVP	PATH Message
6	0.040000	SIP/SDP	Status: 183 Session Progress, with session description
7	1.450000	SIP/SDP	Status: 200 OK, with session description
8	1.470000	SIP/SDP	Status: 200 OK, with session description
9	1.530000	SIP	Request: ACK sip:1111@100.4.4.129:5060;maddr=100.4.4.129
10	1.540000	SIP	Request: ACK sip:1111@100.1.1.225:5060
11	1.560000	RSVP	RESV Message
12	1.650000	RSVP	PATH Message
13	1.650000	RSVP	RESV Message
14	30.050000	RSVP	PATH Message
15	31.650000	RSVP	RESV Message
16	31.650000	RSVP	RESV Message
17	31.760000	RSVP	PATH Message
18	37.680000	SIP	Request: BYE sip:1111@100.4.4.129:5060;maddr=100.4.4.129
19	37.680000	SIP	Request: BYE sip:1111@100.1.1.225:5060
20	37.690000	RSVP	RESV TEAR Message
21	37.690000	SIP	Status: 200 OK
22	37.690000	RSVP	PATH TEAR Message
23	37.690000	RSVP	RESV TEAR Message
24	37.700000	SIP	Status: 200 OK
25	37.830000	RSVP	PATH TEAR Message

Figure 10–SIP messages exchanged during a call.

phases. UA3 sends a SIP INVITE message. The proxy receives this message, sends back to UA3 a trying notification and forwards this message (via the redirect server) to the destination UA (UA1). UA1 notifies the sender that the call is in progress through the 183 Session Progress message. Concurrently with this process, the resource reservation for this call is also performed, since this call has QoS requirements. After call acceptance, UA1 sends back to UA3 a 200 OK message notifying that it accepts the call. The UA3 acknowledges the reception of this message with an ACK message. Later, to terminate the call, one of the UAs sends to the other a BYE message. The reservations are eliminated through PATH TEAR and RESV TEAR messages and the BYE message is acknowledged with a 200 OK message.

Figure 11 shows the body of the SIP INVITE message. The first line indicates the type of message (INVITE). The caller identification and the current version of SIP are also included in this line. The ‘via’ field contains the IP address of UA3, since this message only traversed this node. The ‘From’ field contains the address of UA3, and the ‘To’ field contains the URL address of UA1. The ‘Call-ID’ field is a globally unique identifier (space and time) for the call. ‘CSeq’ is the sequence number of this message, which is 1 in the present case. The field ‘Content-Type’ gives the type of media description. The set of fields ‘From’, ‘To’ and ‘Call-ID’ are designated by Call Leg. The Call

```
Session Initiation Protocol
    Request-Line: INVITE sip:2222@centaurus.av.it.pt:5060 SIP/2.0
  Message Header
      Via: SIP/2.0/UDP 100.1.1.225:5060
      From: UserAgent-1<sip:1111@100.1.1.225:5060;user=phone>
      To: 2222<sip:2222@intrepid.av.it.pt:5060>
      Call-ID: 0019c570986e8908f4589a41af860640@100.1.1.225
      CSeq: 1 INVITE
      Subject: VovidaINVITE
      Contact: <sip:1111@100.1.1.225:5060;user=phone>
      Content-Type: application/sdp
      Content-Length: 220
Session Description Protocol
    Session Description, version (v): 0
    Owner/Creator, Session Id (o): - 326290408 326290408 IN IP4 100.1.1.225
    Session Name (s): VOVIDA Session
    Connection Information (c): IN IP4 100.1.1.225
    Time Description, active time (t): 3236950778 0
    Media Description, name and address (m): audio 10000 RTP/AVP 0 100
    Media Attribute (a): rtpmap:0 PCMU/8000
    Media Attribute (a): rtpmap:100 telephone-event/8000
    Media Attribute (a): ptime:20
    Media Attribute (a): fmtp:100 0-11
```

Figure 11–Register SIP message containing SDP.

Leg is unique throughout the SIP call. The 'Contact' field contains the IP address of the UA that may be used for direct contact with the UA.

SIP uses a description format, called SDP, to allow each party to declare its receiving capabilities and the characteristics of the media streams each party wants to receive. The SDP part of the SIP message is also shown in this example. It contains information about the SDP version, owner details, subject name and details, start and stop times, and details about the media. The start and stop times are irrelevant in the case of bi-directional calls, but they are required when a UA wants to announce a call to other UAs in a predefined time. The media details contain the type of media (audio/video), the port assigned to the media, and the various codec codes that the UA can support. The codes are given in the AVP (Audio Video Profile). The IETF has recommended that the codes 0 to 5 are mandatory and all SIP end points must support them for the sake of interoperability. The SIP transactions are carried in a default port (5060) or in any other random port configured by the UA. Similarly the media can use any unique port configured by the UA. In our example, we have chosen the port 10000.

After call establishment, the transmission of the audio traffic from UA3 to UA1 is initiated. During this test, we were able to listen the music at UA1 with the same quality as in UA3. In the mean time, we started the FTP session that congested the link. We realized that there were no changes in the quality of the music. The FTP session only used the available link capacity. We can conclude that resources were indeed reserved for the audio call, whose quality was not degraded by the FTP traffic. Then, we performed the same test, but with the audio call configured as best effort service, i.e., no resource reservation. In this case, when the FTP session started the music became almost imperceptible.

5.2. Quantitative tests

In these tests, we first established a call with best effort service (with 3 Mb/sec) from PC4 to PC2. As can be seen from Figure 8, the bandwidth available for the best effort class is 1.1 Mb/sec. Therefore, the call with best effort service will use some bandwidth of the reserved class. Approximately 30 seconds after the start of the best effort call, we established a QoS enabled call from UA3 to UA1. This call will use resources of the reserved traffic class. Therefore, some of the bandwidth that was available in this class for the best effort call will now be assigned to this new call. After 60 seconds, the QoS enabled call is terminated, and releases the bandwidth to the best effort call. For each time period, before the establishment of the QoS enabled call, while this call is active and after its termination, we measure the throughput of each call, the mean and maximum delay, the mean and maximum jitter and the packet loss ratio.

We performed several experiments, where the bandwidth of the QoS enabled call was varied from 0.5 Mb/sec to 2 Mb/sec with steps of 0.5 Mb/sec. Figure 12 to Figure 15 show the throughput evolution over time. The results correspond to averages taken over a total of 5 runs.

We observe in the figures that the throughput of the best effort call is equal to the required bandwidth when there is no reserved traffic (i.e. when the QoS enabled call is not active). In this period, the best effort call is using 1.9 Mb/sec

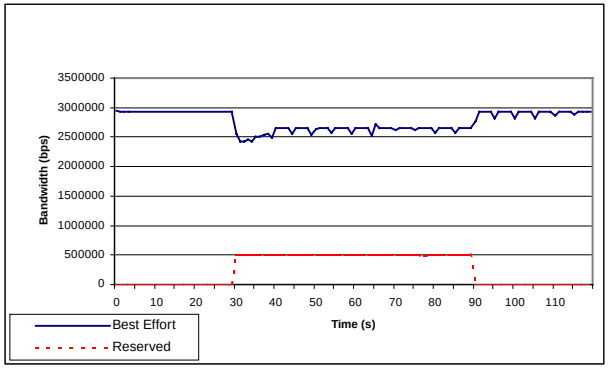

Figure 12– Throughput versus time (QoS enabled call - 0.5 Mb/sec).

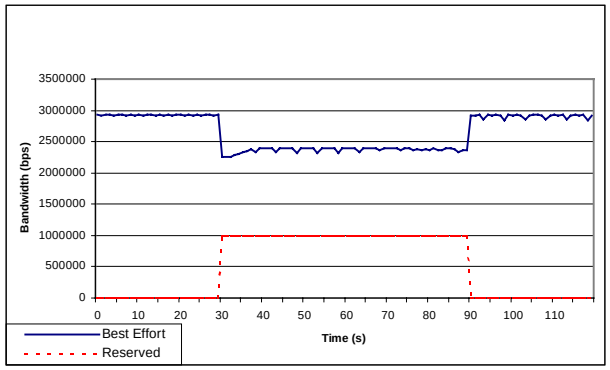

Figure 13– Throughput versus time (QoS enabled call - 1 Mb/sec).

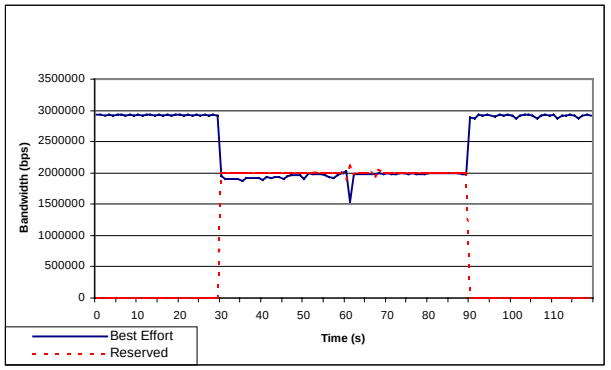

Figure 15– Throughput versus time (QoS enabled call - 2 Mb/sec).

that belongs to the reserved class. When the QoS enabled call starts, the required resources are allocated to this call, and the bandwidth of the best effort call is reduced by approximately the same amount. We can also notice that the QoS enabled calls always occupy the required bandwidth. This fact shows that the resources are actually reserved to those calls and that they do not depend on the amount of traffic in the other classes.

All our experiments present a noise that is most visible in the throughput curves of the best effort traffic. This noise is due to the process of generation of the log files that store the information about each transmitted packet.

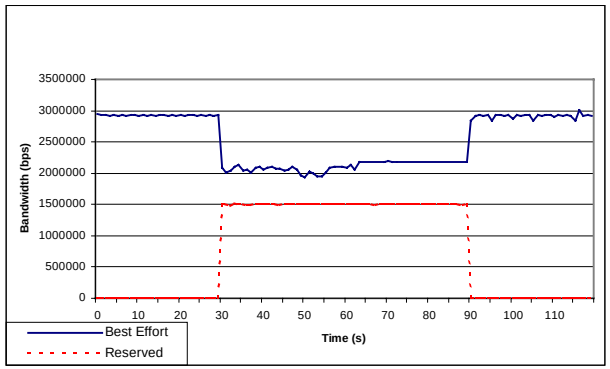

Figure 14– Throughput versus time (QoS enabled call - 1.5 Mb/sec).

Figure 16 shows the mean delay of the best effort traffic, as a function of the reserved bandwidth, measured in three distinct time periods: before the establishment of the QoS enabled call, denoted by 'Before'; when the QoS enabled call is active, denoted by 'During'; and after the termination of the QoS enabled call, denoted by 'After'. It also shows the mean delay of the traffic belonging to the QoS enabled call. We observe that the mean delay of the best effort traffic is always higher than the one of the reserved traffic. Before the activation of the QoS enabled call, the mean delay of the best effort traffic remains almost constant. When the QoS enabled call is activated, the mean delay of the best effort traffic increases. During this period, the mean delay increases with the bandwidth of the QoS enabled call, because less resources can be used by the best effort traffic leading to an increase in the mean occupancy of the buffers. After the termination of the QoS enabled call, the mean delay of the best effort traffic returns to the value of the initial period.

Figure 17 depicts the mean jitter as a function of the reserved bandwidth, for the same time intervals as in the delay case. We observe that the mean jitter of the reserved traffic and the one of the best effort traffic in the first and third time periods are very similar. The mean jitter of the reserved traffic is not affected by the existence of the best effort traffic, since the resources are reserved during the overall period the call is active. The mean jitter of the best effort traffic in the second time period is higher because the occupancy level of the buffers experiences larger variations during this period.

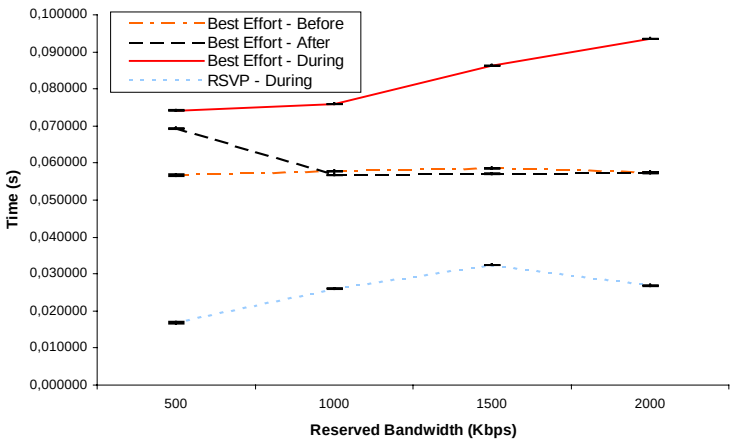

Figure 16 – Mean delay of the best effort (before, during and after the establishment the QoS enabled call) and reserved bandwidth traffics.

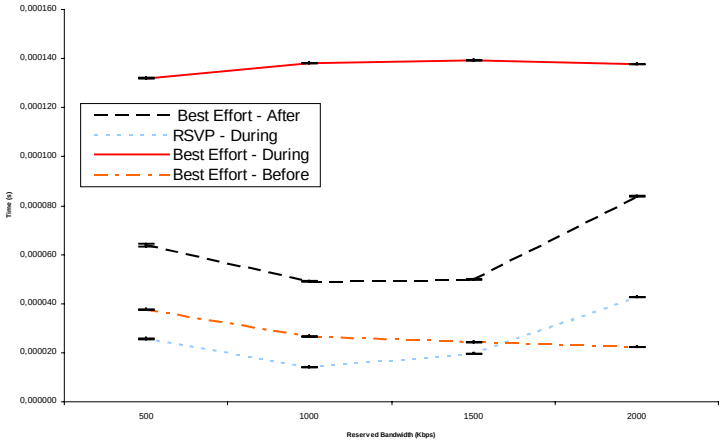

Figure 17 – Mean jitter of the best effort (before, during and after the establishment the QoS enabled call) and reserved bandwidth traffics.

Figure 18 depicts the packet loss ratio of the best effort traffic in the three time periods. We do not show the packet loss ratio of the reserved traffic because it is near 0% for all reserved bandwidths. Its maximum value is 0,000056%. The packet loss of the best effort traffic is similar, on the order of 2.5%, in the first and third time periods. When the QoS enabled call is active, the loss rate of the best effort traffic increases to values between 13% (when the reserved bandwidth is 0.5 Mb/sec) and 35% (when the reserved bandwidth is 2 Mb/sec). As the reserved bandwidth increases, the available bandwidth for the best effort traffic decreases, and the loss ratio increases. On the other side, the reserved traffic gets all the resources requested and its packet loss rate remains almost zero.

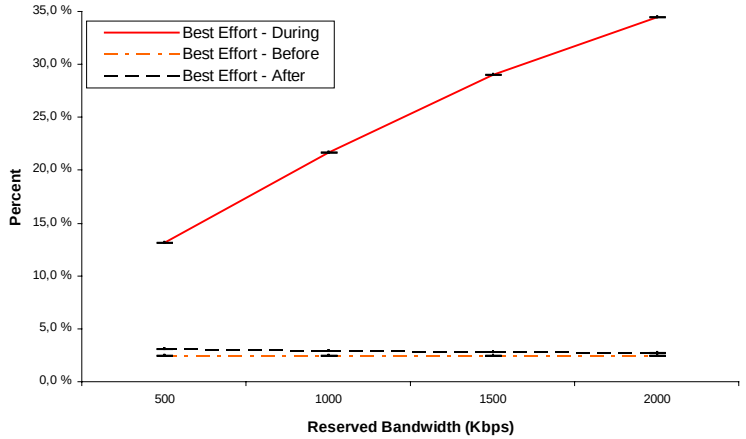

Figure 18 – Packet loss ratio of the best effort (before, during and after the establishment the QoS enabled call).

6. Conclusions

We have described a demonstrator of an IP-based access network for broadband multimedia services, which was designed to be low cost and easily manageable. In order to support broadband multimedia services, the architecture incorporates a number of newly introduced technologies: SIP (Session Initiation Protocol) for session initiation, RSVP (resource ReSerVation Protocol) for resource reservation, and COPS (Common Open Police Service) and DIAMETER for

QoS policy management and AAA (Authentication, Authorization and Accounting). Some modules of the access network elements were implemented using the VOCAL (Vovida Open Communication Application Library) system, developed by Vovida. We reported both functional and performance results obtained with the demonstrator, and traffic management experiments related with the policing, packet scheduling, resource reservation and admission control mechanisms.

7. References

[1] Susana Sargento et al., "IP Access Networks with QoS Support", In Proceedings of SPIE ITCOM 2001 – "Technologies, Protocols, and Services for Next-Generation Internet", Denver (Colorado, USA), August 2001.

[2] VOCAL – Technology Overview", http://www.vovida.org/document/Vocal_Technology_Overview.pdf; "VOCAL – Installation Guide", http://www.vovida.org/document/Vocal_Installation_Guide.pdf; "VOCAL – System Architecture", http://www.vovida.org/document/Training/2_VOCAL_Architecture.pdf; "VOCAL – System Requirements and Scalability", http://www.vovida.org/document/Training/3_System_Requirements_and_Scaling.pdf.

[3] M. Handley et al., "SIP: Session Initiation Protocol", IETF RFC 2543, March 1999.

[4] R. Branden et al., "Resource reSerVation Protocol (RSVP) – Functional Specification", IETF RFC 2205, September 1997.

[5] D. Durhan et al., "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[6] P. Calhoun et al., "Diameter Base Protocol", Internet Draft draft-ietf-aaa-diameter-10.txt, April 2002.

[7] http://www.cisco.com/univercd/cc/td/doc/product/access/sc/rel7/soln/wv_rel1/wvov/.

[8] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "A Transport Protocol for Real-Time Applications. Audio-Video Transport Working Group", RFC 1889, Janeiro de 1996.

[9] C. Rigney, A. Rubens, W. Simpson, S. Willens, "Remote Authentication Dial In User Service", RFC 2138, Abril de 1997.

[10] http://www.sparre.dk/pub/linux/tc/ http://qos.ittc.ukans.edu/howto/.

[11] M. Handley and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[12] W. Marshall et al., "Integration of Resource Management and SIP", Internet Draft draft-ietf-sip-manyfolks-resource-07.txt, April 2002.

[13] QoSForum, "Introduction to QoS Policies", White Paper, 1998.

[14] R. Yavatkar, "A Framework for Policy-Admission Control", RFC 2753, January 2000.

[15] http://manimac.itd.nrl.navy.mil/MGEN/