Internet Service Architectures and ATM – The ELISA Approach

B. F. Koch, C. Prehofer
Berthold.Koch@icn.siemens.de,
Christian.Prehofer@icn.siemens.de,
Siemens AG, Germany

Summary
This chapter reports on design, implementation and experimentation of a network architecture that supports Quality of Service (QoS) for Internet applications. The new architecture supports the Integrated Services as well as the Differentiated Services approaches in a smoothly integrated way and uses the capabilities of an underlying ATM network to realize QoS. The enhancements to the existing network infrastructure are deliberately limited to the integration of a single new type of network element called the Edge Device. The chapter presents the network architecture, the implementation of the components as well as on business models and scalability analysis.

Introduction
The Internet gives a fascinating vision of a future global and user-friendly information infrastructure. Unfortunately, current Internet technology is technically not suitable for reliable multimedia services, since it does not provide adjustable Quality of Service (QoS). The Internet relies on a protocol layer (IP) which enables interoperation of various network technologies, but uses the least common denominator of the capabilities of underlying networks (best-effort quality).

On the one hand, several concepts for QoS on the IP-level have been proposed, in particular the IETF Integrated Services (IntServ) and Differentiated Services (DiffServ) architectures. On the other hand, there are already commercially available networks (of competitive technology, e.g. ATM) where high potential for QoS support exists but is not exploited in the Internet. Therefore, an interaction and integration of QoS Internet and ATM becomes necessary. This chapter describes a new architecture that tightly integrates the mentioned approaches in order to provide QoS. This approach is based on a careful analysis of the main functional building blocks for QoS and shows that rather different-looking concepts can be supported by a single homogeneous and elegant architecture. The approach described defines an open QoS architecture that provides flexibility to incorporate new concepts beyond those already supported. For example, a QoS request (as it is defined by IntServ) may be mapped directly to an individual ATM connection or may be aggregated with other requests in a “controlled” DiffServ traffic class (Expedite Forwarding).

The main objective of the ELISA project is to offer a practical approach for the provisioning
of Integrated Services (with the help of RSVP) and Differentiated Services architectures based on Internet IP technology. Moreover, the ELISA project brings together such an advanced IP service approach with the capabilities offered by an ATM-core network (e.g. guaranteed QoS through ATM switched connections).

In order to achieve that goal the ELISA architecture focuses on an Edge Device, which acts as an edge router as well as an Integrated Services gateway. The AC310 ELISA consortium is formed by public network operators (Deutsche Telekom AG, Telefónica I+D), equipment manufacturers (Siemens AG and Siemens Telefonyár Kft), as well as research institutes and universities (GMD Fokus, CoRiTeL, Dresden University of Technology, National Technical University of Athens, and Technical University of Budapest). It will provide a trans-European testbed to demonstrate the feasibility of the architectural approach.

**The ELISA Architecture**

One of the main goals of the ELISA architecture is to enable operators of wide-area ATM networks to offer the advanced features of ATM technology to end-users, who are using IP-based applications. The architecture assumes as the standard case that end-users are not connected directly to the ATM wide-area network but are connected through the infrastructure of an Internet Service Provider or an intranet, which may use, for instance, various LAN technologies, ISDN dial-up access or xDSL access networks. The ELISA architecture is flexible enough to be applied to configurations where the core network is not ATM based, but this is out of focus of the prototype currently being developed.

In order to bring QoS to IP-based applications, it is necessary that QoS is somehow accessible in the Internet world, i.e. applications or end systems can define their specific QoS requirements. There are two main approaches to QoS, which are currently discussed in the Internet community:

- The Integrated Services (IntServ) architecture [1] is based on individual resource reservations issued by the applications using reservation protocols, in particular the Resource Reservation Protocol (RSVP) [2]. The routers in the core network have to reserve resources for the individual flows.
- The Differentiated Services (DiffServ) architecture [3, 4] takes a much simpler approach and simply assumes that IP packets are marked as belonging to one out of a number of different traffic classes. Network elements treat these with appropriate Per Hop Behaviors (PHB) [2]. Core routers simply have to follow the general rules for the respective traffic class/PHB and do not deal with individual QoS requests.

Both approaches have their advantages and disadvantages. The main problem with the Integrated Services architecture is that it does not scale well to large wide-area networks, since in this case reservation processing on central core routers becomes a bottleneck. The problem with the Differentiated Services approach is that it can give true service guarantees only if it is combined with rigorous policing of the access to the high-quality service classes.
And a problem with both approaches is how to interwork with already available ATM infrastructure. ELISA solves these weak points.

Several aspects related to the interworking of IntServ and DiffServ approaches are considered in [5]. The definition and implementation of the ELISA architecture has progressed in parallel with the work reported in this IETF draft. The ELISA architecture focuses on one important network element, which is particularly relevant for QoS. This network element is called Edge Device (ED) within ELISA, but it may also be called an extended access router or a QoS gateway. The Edge Device prototype developed in ELISA connects an access network to a wide-area core network. The interconnection of EDs realizes an overlay QoS network for the IP-based applications. ELISA targets only end-systems that are connected through an Edge Device. The access network is typically based on LAN technologies or on POTS, ISDN or xDSL.

Figure 29: ELISA reference architecture

The core network can be a plain ATM network, which is the optimal network base for ELISA. But the ELISA Edge Device also works in more general configurations where the core network is an arbitrary DiffServ-based IP network, which possibly resides upon an ATM layer. The ELISA network architecture does not require any changes or upgrades to any of the sub-networks involved.

Figure 29 shows the reference network architecture of ELISA. Details are shown only for a pair of end-systems, but the general assumption is to have many end-systems connected to each Edge Device and many Edge Devices to the core network.

The end-systems connected to the access network use only IP-based protocols and can
specify their QoS requirements in two different ways: either they issue reservation requests using RSVP/IntServ or by subscription to a specific DiffServ traffic class. The Edge Device deals with both kinds of requirements.

Reservations are analyzed, policed and mapped onto appropriate core network traffic classes (Figure 30). These core network traffic classes can be carried over a dedicated ATM Switched Virtual Channel (SVC) to another Edge Device, or over a portion of a pre-established ATM Permanent Virtual Channel (PVC) using a specific DiffServ class. Note that the end-system just uses the reservation protocol without any knowledge of the mechanism chosen by the Edge Device. Note further that the reservation processing is done only in the Edge Device and not in the core network. The flexible mapping into ATM VCs distinguishes the Edge Device from other proposals for the IntServ/DiffServ integration known to the authors.

In summary, the following network capabilities will be implemented in the Edge Devices:

- IntServ, where resources are explicitly requested. Resources will be provided by a dedicated SVC or by aggregation into an existing link.
- DiffServ, where resources are provided according to the marking of the packets. The following classes are available: Best Effort (BE), Priority (P), High Priority (HP), Expedited Forwarding (EF).

DiffServ traffic is analyzed and policed in the Edge Device. The Edge Device is able to carry out DiffServ marking for end-systems that do not support DiffServ marking themselves. In this case, the traffic class is derived from the IP address or other selectors (e.g. TCP port number) based on information, which is kept on the ED (Edge Device) in a configurable way. The ED is able to do policing based on volume limits for traffic in a specific DiffServ class, and

Figure 30: ELISA Edge Device – an approach for integrating IntServ/DiffServ over ATM
therefore enables QoS guarantees for reservations following the DiffServ approach. As an additional function, the ED is able to collect information for usage-based charging for QoS usage, for explicit reservations as well as for DiffServ class usage.

The advantages of the ELISA architecture are that it flexibly combines the Integrated Services and Differentiated Service model by mapping reservation requests onto DiffServ classes where possible and that it removes some of the scalability problems for explicit QoS reservations in wide-area networks. Of course, this raises the question how the scalability of the ELISA architecture itself is ensured. Aspects related to the scalability of an architecture combining IntServ and DiffServ are considered later.

In the ELISA architecture, two different kinds of interfaces must be considered. Fig. 29 shows these two interfaces, which will be referred to as Access Interface and Edge Interface. The Access Interface allows the host to interact with the network and in particular to be connected to the Edge Device. The Edge Interface allows the Edge Device to access the core network. This distinction is consistent with the choice to partition the network in an access domain and a core domain.

Let us briefly analyze the protocol stacks used at both interfaces to transport IP packets (user plane) and the control mechanism needed to signal QoS requirements (control plane). On the Access Interface, the IP packets are carried over LAN (Ethernet) or ISDN (using PPP). At the control plane level, the RSVP protocol is the mechanism which is used by the applications to signal their QoS requirements on the Access Interface, according to the Internet Integrated Service model. The ELISA RSVP protocol messages are fully compliant with the relevant RFCs issued by the IETF [2]. In the ELISA architecture, the RSVP can be used within the access network domain to allocate resources up to the Edge Device. RSVP is used in the ED to allocate resources and to map the user requests onto the mechanisms used in the core network.

The Edge Interface is based on ATM under the hypothesis that a network operator provides ATM semi-permanent and switched connection. It is a UNI interface; for the ELISA project a common subset of ATM Forum 3.1 and ITU-T2931 is used. IP packets are encapsulated on ATM according to RFC 1483.

The control plane of the Edge Interface is based on two concepts. The first concept is that RSVP is used with “end-to-end” significance through the core network domain. This means that RSVP messages are interpreted only in the access network domain and in the Edge Device, but not in the core network domain. The second concept is that the RSVP reservations can be mapped either onto DiffServ classes or directly onto a dedicated ATM connection that will be set-up on demand.

**Edge Device Software Design and Implementation**

In order to accommodate the wide range of tasks of the ELISA Edge Device concerning several management issues (services, customers & contracts) as well as network functionalities (routing, queuing, signaling, etc.) the Edge Device has a distributed, non-
monolithic software architecture containing several functional blocks/packages for various assignments. An internal package view of the Edge Device overall structure is depicted in Figure 31. Note that the arrows between the packages do not mean data flows but indicate usage dependencies.

**Figure 31: Edge Device package structure**

The User Plane contains the functionality for transmission of traffic generated by user applications. The IP stack is responsible for packet routing, and the ATM related User Plane protocols (AAL5) for converting IP packets into ATM cells. The additional Packet Handler (PH) extends the IP stack to QoS-based forwarding.

The Control Plane contains the functionality for establishing and removing data paths through both networks, the access one and the core one. The RSVP module encapsulates a standard RSVP daemon that handles all reservations requests. The ATM control plane, which in the particular implementation is UNI 3.1, provides the means for accessing the ATM core network. Flow Admission Control (FAC) has two main tasks; policy control and admission control. To decide on the latter, QoS & Bandwidth Management (QBM) is in charge of the network resources of the ED by utilizing the advanced features of PH. The Address Resolution (AR) resolves a destination IP address to the E.164 ATM address of the corresponding egress ED. Finally, the Usage Metering & Accounting (UMA) records all accountable events in the system for a later charging.

In addition to these packages, the Service Management (SM) package is responsible to maintain all registered users and their profiles and gives the system administrator the means for configuring and administrating the ED. It keeps all user and network information in a relational database by using the Persistence Layer (PL) as accessing mechanism.
The ED has a generic and open structure so that several QoS architectures (and not just IntServ, DiffServ and ATM) can be supported. It is also designed to be scaleable for further extensions and tasks. The OMG Unified Modeling Language (UML) [9] is used as common base to allow a precise communication between many developers from many different companies, organizations and countries involved in the analysis and design process.

**Details of Implementation**

Each package is realized as a single component and runs in its own process. For the inter-component communication, the Common Object Request Broker Architecture (CORBA) of the OMG is used. Hence components which are used by other components provide an interface definition specified in the Interface Description Language (IDL). The use of these standards allows a clear and well-defined task sharing between the functional blocks on the one hand, and it improves the scalability, maintainability, reusability, and platform independence of the ED on the other hand. However, the use of CORBA may impair the performance of distributed software systems. For this reason, two design decisions were taken:

- CORBA requests are not involved in the IP packet forwarding process.
- Each CORBA server component implements only a fixed set of CORBA server objects which are initialized (and resolved by the CORBA client) at start-up. Therefore, during the run time of the ED, no time-consuming creation and resolving of CORBA objects is needed.

These CORBA server objects forward each request concerning the server component. They may act as a factory and a controller of controlled ‘private’ objects within the server blocks, as shown in Figure 4. (These standard design patterns can be found in [12].) That means, such a class is responsible to create, modify, and remove controlled objects, which may represent special resources of the server package, e.g. network interfaces. The client does not know these objects but only knows their handles, for instance. Moreover, each controller is declared as a singleton class to indicate that it has one single instance and that this object is globally accessible (see also [12]).

The open source tool Arachne is used as the CORBA platform that provides an interoperable, stable and fast object request broker (ORB). An additional Process Manager (PM) handles the initialization of each basic ED component. While these components are implemented in C/C++, the Service Management is enhanced by a remote GUI written in Java. Using the same Sybase database as shared communication and information mechanism this GUI is able to run as an applet in a Java enabled web-browser for remote, platform-independent administration and configuration.

The Edge Device prototype itself runs on the platform of a Sun workstation with ATM and LAN interface cards using Solaris as operation system. Wherever adequate, publicly available software has been used and adapted. In particular, the RSVP implementation is based on the ISI implementation as adapted and freely distributed by Sun Microsystems.
Applications using QoS

There are a number of attractive applications that decide on the success of QoS enhancements of the Internet. For current Internet users, the most attractive ones are traditional Internet applications like WWW and FTP. The QoS provision for WWW and FTP applications can be achieved with DiffServ: The user registers and pays for getting better service, and IP packets are marked either at the end system or at the Edge Device in order to actually benefit from higher priority. Besides some administrative interfaces, no application development is required in this case.

As soon as a QoS infrastructure exists, specific applications may become popular which support resource reservation. Currently, there are no commercial applications yet which support reservation, however the protocol RSVP seems to have the support of major software developers. For demonstration purposes, the project ELISA has implemented a set of applications that support QoS, for example a video conferencing environment based on the MBone tools and running on Unix (Linux) workstations, FTP and WWW applications enhanced with RSVP. A list of the ELISA sample applications and of the mapping into QoS mechanism is reported in the table below.

A more detailed description of the ELISA user application and of the QoS tools in the terminal can be found in [12].

The architecture described above supports various degrees of sophistication for the way how applications make use of QoS. This flexibility is crucial for a successful introduction of QoS into the market, since the introduction of QoS in the network is not linked to upgrades of the user applications.
### Scalability

With the term “scalability” the implications of large networks for the network elements is meant; in particular, the system requirements of a single node should not grow linearly with the number of network elements. For our architecture, there are several issues to be discussed.

The main feature of RSVP is the management of single flows. Once the reservation has been established, all the routers in the path have to recognize the packets belonging to a reserved flow and provide suitable handling. These actions can easily become an unacceptable processing burden when hundreds of thousands of different flows must be handled, for example in a gigabit core router, resulting in a bottleneck. Similarly, the need to exchange and store “per flow” information is another heavy burden for core routers. These scalability concerns have led the Internet community to investigate simpler solutions for the support of QoS.

In the Differentiated Services model [3], the basic idea is to support a set of traffic classes providing a different service to each one. User flows are only controlled at the edge of the network and then aggregated into a small set of traffic classes: this approach relieves the scalability impairments on the core router in the classical RSVP approach and overcomes the scalability issue.

Since IntServ and DiffServ focus on reservations and scalable service differentiation, respectively, it is advantageous to combine both for an overall solution: IntServ can be used in the access network and DiffServ in the core network, as it is supported in the ELISA architecture.

---

### Applications and service mapping

<table>
<thead>
<tr>
<th>Applications</th>
<th>Service components</th>
<th>Service profile</th>
<th>ELISA mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multimedia conference.</td>
<td>Video Audio</td>
<td>Regular IP video conferencing</td>
<td>BE, P, HP</td>
</tr>
<tr>
<td>Video telephony</td>
<td></td>
<td>2 party ISDN like video conference</td>
<td>EF, Dedicated SVC</td>
</tr>
<tr>
<td>Premium Multimedia Conferencing</td>
<td></td>
<td>Regular video conference with QoS</td>
<td>EF, Dedicated SVC</td>
</tr>
<tr>
<td>File Transfer</td>
<td>Data</td>
<td>FTP</td>
<td>BE</td>
</tr>
<tr>
<td>On demand retrieval</td>
<td></td>
<td>Enhanced FTP /Guaranteed bandwidth</td>
<td>EF, Dedicated SVC</td>
</tr>
<tr>
<td>WWW navigation</td>
<td>Video, Audio Data</td>
<td>WWW</td>
<td>BE</td>
</tr>
<tr>
<td>Premium WWW navigation</td>
<td></td>
<td>Enhanced WWW</td>
<td>EF, Dedicated SVC</td>
</tr>
</tbody>
</table>
Restricting RSVP to the access also makes the architecture relatively future-proof, since it can be easily adapted for different mechanisms to signal QoS requests (as they may for instance appear in future object communication middleware). Such an upgrade would only involve parts of the Edge Device but would not affect the core network.

Imagine now that the IP routers in the core network are interconnected by means of an underlying ATM layer, in a small-scale network scenario direct ATM connections among all the EDs can be assumed. This configuration (an example is shown in Figure 33) has been deployed in the ELISA trial. In a large-scale network a fully meshed topology with direct links between all EDs is not feasible. Therefore, DiffServ-enabled routers (DS-R) are used in the core network. In such a scenario, good scalability is achieved since every “per flow” operation is confined to the EDs and does not impact the core of the network, where only highly aggregated traffic flows are dealt with. The underlying network infrastructure (for interconnecting DS-Rs and for connecting EDs) should be ATM in the ideal case, but non-ATM portions can be handled. Two issues must be considered when dealing with this target architecture:

- The first one is typical for the DiffServ approach: how to enforce QoS reservations in a network composed of a set of routers. Obviously the intermediate routers must be involved in providing the QoS between two EDs and since intermediate core routers do not interpret RSVP messages, these routers are not updated on bandwidth requests. There is a wide range of possible solutions, with a different trade-off between complexity and efficiency. The simplest approach is to keep local reservation/admission control in the EDs and to follow a static approach. In this case a sort of advance pre-allocation of bandwidth is needed, which typically results in a loss of efficiency. More efficient solutions imply a dynamic exchange of information, which can involve Edge Devices and the intermediate routers. This new control mechanism could be a peer-to-peer mechanism or ad-hoc devices (often called bandwidth brokers) can be introduced, with the purpose of controlling resource allocation in a DiffServ network.

- The second issue is typical of the wide-area IP over ATM architectures. Scalable mechanisms for translating IP addresses into ATM addresses are needed. These mechanisms (e.g. NHRP) are under study within IETF, but their interaction with RSVP is still unclear. A suitable straightforward extension to RSVP to support IP/ATM address resolution is described [10]. If there is an ATM path between two EDs these mechanisms allow the establishment of direct QoS ATM shortcuts.
BUSINESS SCENARIOS

In this section two possible business scenarios based on the network architecture developed in ELISA are discussed. Depending on who owns which part of the network, different revenue models are considered. In addition, each player can optimize the network usage in order to optimize his part of the network structure and to optimize revenues. There are two main scenarios, the public access and the corporate access scenario, which are discussed below.

The public access scenario is shown in Figure 34. Users connect to Edge Devices owned by ISPs, e.g. by ISDN, xDSL or any other access technology. The ISPs connect their Edge Devices via an ATM network, which is owned by an ATM core network provider. Clearly, a possible special case is that both ISP and ATM operator coincide.

In this scenario, we have at least two business relations. The end-user will be charged by the ISP for the individual user services or by flat rate models, which may include access to some services. In addition, it is possible that the service is initiated and paid by a content provider, who in turn charges the user, e.g. by credit card. Another option is that the ISP charges for application level services, which provide QoS.

In turn, the ISP has to lease the ATM connections from the ATM operator. In the case of dedicated ATM connections for one user QoS request, there is a direct mapping of the cost for the ATM connection to the user service. In the other cases, traffic is aggregated in service classes. The ISP has to lease a number of statically provisioned ATM connections (fixed costs) which are supported by on-demand switched VCs (variable costs). Although one user request
may trigger an extra VC, there is no direct relation between the ATM connection and the user service.

From a technical point of view, bandwidth management can take place on the IP-level (DiffServ scheduling in the ISP Edge Device) and on the ATM layer via ATM VCs. With the former, DiffServ classes can borrow unused bandwidth of other DiffServ classes in order to utilize the available link layer bandwidth. On the other hand, we can map DiffServ classes to different ATM VCs providing different traffic capabilities. In this case, the ATM layer has to assure that bandwidth is not wasted. However, in the first case, the ISP can use extra bandwidth for best effort and other traffic, while in the other case, the ATM operator may use the extra bandwidth. Hence the ELISA Edge Device design provides flexible mechanisms to utilize bandwidth (and in turn service charges) for both ISP and ATM operator.

**Figure 34:** Public access business scenario for the ELISA architecture

In the corporate scenario in Figure 35, the Edge Devices are operated by a cooperation inside its local area network. In this case, the users are typically connected via LAN infrastructure. The CPE equipment and the Edge Devices are administered by the same corporate network operator. (In case the Edge Device is leased from the another operator, the model is similar to the one above.) For the core network, it is convenient for the cooperation to lease ATM connections, which serve as a virtual private network (VPN). In this case, other issues like security and service availability can play a more important role than in the above-mentioned case.
TRIAL

The final target of ELISA is to evaluate the DiffServ and IntServ features mapped onto ATM SVC connections as well as the charging capabilities of the ELISA platform related with the provision of services. For this purpose, an Edge Device and Customer Premises Equipment are being implemented.

The feasibility of the ELISA approach will be tested and demonstrated in an international testbed. The testbed will involve Edge Devices located in four core sites (Munich, Darmstadt, Berlin, Budapest) and end systems deployed in several remote sites in different European countries (Spain, Italy, Greece), as it is shown in Figure 36.

The remote sites and local terminals will be connected to Edge Devices using some selected access technologies.

The experiments focus on:

- the functional verification of interworking functions between IntServ, DiffServ and ATM, signaling, admission control, traffic policing and traffic shaping as well as charging modules,
- performance measures related to the traffic, affecting Quality of Service, and
- experiments assessing end service aspects and taking into account both, the user perception and the service provider perception.
The functional testing will demonstrate the correct interworking of the components of the ELISA architecture, which have been implemented and deployed. The traffic and QoS testing aims at obtaining, at measuring time, some performance parameters (throughput, packet delay, packet loss, QoS session setup-time, etc...) of different flows. For example, a comparison of the performance of “IntServ” flows and “Best effort” flows under different network load will be done. The goal is to show that ELISA provides an efficient QoS framework for IP flows. Quantitative measurements of the relevant parameters will be performed.

Finally, the testing of the end-user applications will demonstrate the functionality of the applications deployed in the ELISA project and furthermore a quantitative analysis of the performance of what the end-users perceive will be given at this level.

**Conclusion and Outlook**

We have introduced an innovative approach that integrates the major trends for QoS (IntServ, DiffServ, ATM) into a single and scaleable architecture. Such a “convergent network”, bringing together the best of all these concepts, is technically feasible and economically viable. Moreover, a generic and elegant design concept for an Edge Device has been presented which enables the practical usage of the proposed architecture in well-defined migration steps, and which is currently being realized.
For a future QoS-aware Internet approach, it can be expected that more sophisticated mechanisms are used for QoS control. The concept of “bandwidth brokers” forms the starting point for a QoS architecture that is based on DiffServ principles in the core network but adapts its resource allocations automatically to user demand. This leads to an overlay network of QoS administration agents that interact autonomously and intelligently, in order to optimize the network configuration under all possible conditions.

References

Contact-information:
I.S. Venieris, G. Mamais, G. Politis
venieris@cs.ntua.gr, gmamais@telecom.ntua.gr, gpol@telecom.ntua.gr, NTUA, Greece
G. Eichler
eichler@tzd.telekom.de, T-Nova GmbH, Germany
M. A. Barba
mabr@tid.es, Telefonica I+D, España
R. Cocca, S. Salsano,
cocca@coritel.it, salsano@coritel.it, CoRiTeL, Italy
H. Hußmann, F. Fünfstück
Heinrich.Hußmann@inf.tu-dresden.de,
Falk.Fuenfstueck@inf.tu-dresden.de, TU Dresden, Germany
T. Van Do
Do@hit.bme.hu, TU Budapest, Hungary