

Supplying legacy applications with QoS *

Anne Thomas

Software engineering group, Department of computer sciences
Technische Universität Dresden, Germany
E-mail: Anne.Thomas@inf.tu-dresden.de

Gerald Eichler

T-Systems Nova,
Technologiezentrum, Darmstadt, Germany
E-mail: Gerald.Eichler@t-systems.com

Abstract: In this article we consider end-to-end interdomain Quality of Service support for non QoS-aware applications. We present a description of QoS requests resp. offers 1- at network level: the so called network service profile, and 2- at application resp. end-user level: the so called application profile. These profiles support an end-to-end and interdomain interworking scenario for legacy non and QoS aware applications. The network (composed of more than one domain) is QoS enabled and the different domains offer guaranteed network services. With the proposed mechanisms (combination of converters and description syntax at network and application level) it is possible on the one hand to offer user-friendly QoS, and on the other hand to provide a negotiation and description syntax for QoS.

1. INTRODUCTION

End-to-end inter-domain Quality of Service (QoS) is an important research field both at a technical level and business level where many actors are involved. Considering the business and political [2] [3] side, many questions are still opened and different scenarios are conceivable. Many solutions are possible, mainly depending [1] on the one hand on the consumer strategy, and on the other hand on the providers' (Internet Service, Content, Access, Application...) business strategy. Research activities in this domain are still ongoing and solutions under discussion.

At the present time there exists no public QoS-enabled network and QoS aware applications are seldom since few QoS APIs (like the RSVP API RAPI and Winsock2 from Microsoft) are available for application developers. This fact implies that most of applications susceptible of being used in a future public QoS enabled network will be non-QoS-aware legacy applications. In this article we address QoS provisioning for legacy applications and propose a method based on a description syntax for QoS at end-user, network and application level. We focus on a QoS request-offer scenario from the end-user using a non QoS aware Internet application through a QoS enabled network in a multiple carrier scenario at application and network level. To support this we firstly present a QoS description syntax at network, application and end-user level, then we discuss which specification of a generic one or an implementation

dependent one is most appropriated. Last we present ongoing related works.

2. APPROACH

It is possible to consider the problem from a top-down (from the end-user to the network) point of view or from a bottom-up (from the network to the end-user) one. As depicted in Fig. 1 QoS is offered at domain level as well as toward the end-user via e.g. an ISP.

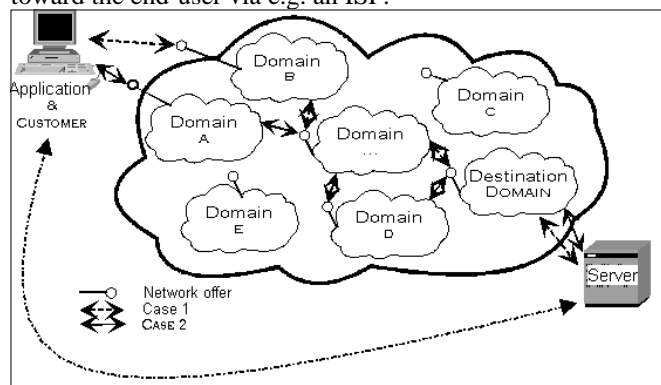


Figure 1: End-to-End scenario.

Customer: residential consumer; Access Network j : Provides access service to customer and manages local access facilities; Domain i : Provides transport to the access ISP. May be vertically integrated or separate firms; Server: Application or content server.

What in the first case corresponds to an offer - the so called *QoS offer* or *network service* corresponds in the second one to a request - the so called *QoS request*. The expression of the QoS request resp. network service depends on the involved entities and their characteristics (e.g. an end-user does not request for QoS in the same manner as an ISP does). The network (composed of more than one domain) (Fig. 1) is QoS enabled and the different domains offer guaranteed network services. Network services like in [4] are designed and implemented in domains to differentiate the treatment of data flows e.g. support different types of traffic flows (strongly related to application types), prioritise certain flows over others, etc... Network operators offer them to ISPs or content providers, they again offer them to end-users. A network

* This work was performed in the framework of IST Project AQUILA (Adaptive Re-resource Control of QoS Using an IP-based Layered Architecture - IST-1999-10077) funded in part by the EU. The author wish to express her gratitude to the other members of the AQUILA Consortium for valuable discussions.

service is offered at inter-domain level as well as border-domain.

In our scenario the end-user uses non QoS aware Internet applications over a QoS enabled network. The application can neither request nor offer QoS. An issue consists in supplying these applications with QoS and in providing mechanisms enabling QoS support by keeping unchanged the legacy applications. One solution consists in offering QoS at end-user level in parallel Fig. 2 to application execution via a so called „QoS portal“. The QoS portal enables the identification of the application, the presentation of the QoS related to the application in use and the selection of the quality level. The aim of the portal is to provide information about the application for the underlying network and vice-versa.

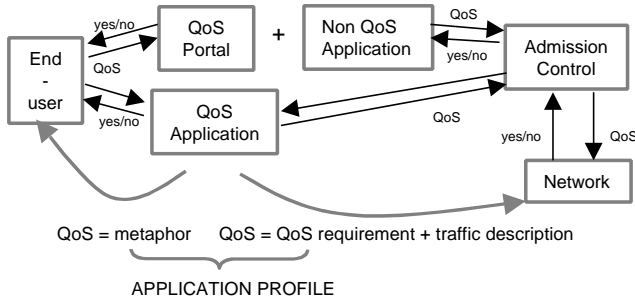


Figure 2 - AQUILA approach.

Application: legacy QoS non aware Internet application; runs in parallel to a QoS portal that enables the identification (via manual selection) of the application and the concerned traffic flows (control like data plane), presents the QoS offer in appropriateness to the application; Application Profiles: repository of application's information; QoS enabled network.

Therefore we need a description syntax for describing 1- the applications at the different levels: end-user, application (control- and data-plane), and network 2- the QoS offer at the different levels end-user, application (control- and data-plane), and network resp. Inter-domain, and mapping mechanisms between the different levels.

3. DESCRIPTION SYNTAX

In this chapter we present a description of QoS requests (resp. offers) at network level: the so called *network service profile*, and at application resp. end-user level: the so called *application profile* by using the eXtensible Markup Language (XML) [8] enhanced by the Data Type Definition (DTD) [7] language. The description syntax is based on the application structure depicted in Fig. 3. The description of the application and QoS at the different levels takes place by means of a DTD syntax that rules the description of the different following artefacts. At 1) Network level: Description of the QoS expectations and requirements of the application,

description of the produced traffic, and description of the implementation dependent QoS request. At 2) Application level (control plane level): Description of the protocol used, port used. At 3) Application level (data plane level): Description of the implementation issues of the different service components, and description of the different configuration options(e.g. audio, video...). At 4) End-user level: Building the metaphors, presenting the possible QoS levels.

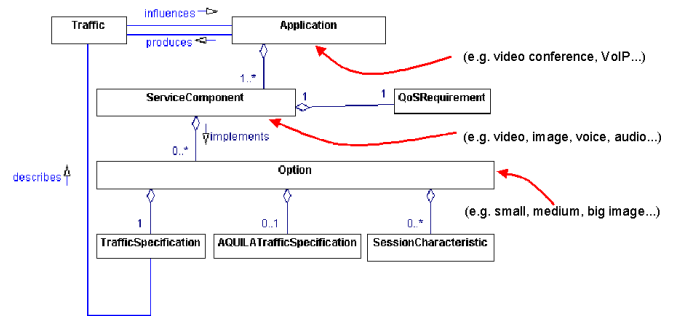


Figure 3 - UML diagram analyzing an application. Application: legacy QoS non aware Internet application produces traffic at the data plane level; is composed of Service Components that have general QoS requirements; they offer many quality options; Each option has a behaviour.

Toward the network the application profile syntax (based on the network service profile) offers the possibility to either implement QoS aware applications or complement non QoS aware (but QoS sensitive) legacy applications in order to make them QoS aware. In this special case an intermediate entity serving as a mediator (the so called converter) between the network and the application is required.

Considering the overall scenario it is obvious that the QoS request resp. offer has to be translated at the different levels: between the end-user representation and the network representation, and between the generic representation and the concrete network implementation. These mappings are the tasks of converters situated at the different levels.

3.1 At Network level

The description syntax at network level enables on the one hand the description of the application QoS request toward the network and on the other hand the QoS offer of the network toward the application or the network.

3.1.1 Generic description

At network level, with the description language network service profile it is possible to specify and abstract the underlying technical network services of a network independently from the concrete implementation (e.g. DiffServ, etc.). This general description is in a first step the basis for the QoS agreement between two domains at border router level. At the access point of the network, this

specification is part of the QoS requirement/request of the application resp. end-user and the ISP QoS offer. A weighting system and a subjective (with wordings) rating enable a better definition of the requirements.

The DTD syntax of the traffic specification Fig. 3 is as follows:

```
<!ELEMENT TrafficSpecification (type+, duration,
  adaptivity, burstiness, packetSize, bitRate,
  flow)>
<!ELEMENT type EMPTY>
<!ATTLIST type
  type (realTime ... | stream | elastic) stream">
...

```

The parameter „type“ describes the type of the traffic, „duration“ the living time of the „traffic“, „adaptivity“ the adaptivity of the traffic to the capacity of the connection (application level QoS adaptation), „burstiness“ the burstiness of the traffic, „packetSize“ the size of the packet, „bitRate“ the bit rate, and the „flow“ the greediness of the flow.

The DTD syntax (extract) of the QoS requirements Fig. 3 of a service component is as follow:

```
<!ELEMENT QoSRequirement (maxDelay, maxJitter,
  maxLoss, bw, ordering)>
<!ELEMENT maxDelay (#PCDATA)>
<!ATTLIST maxDelay
  unit CDATA #FIXED „ms“
  requirement (Low |... High) „medium“
  weight (0 | 1 | 2...| 9 | 10) „5“>
...

```

The parameter „maxDelay“ corresponds to the one way latency, „maxJitter“ to the delay variation, the „maxLoss“ to the packet loss probability, the „bwGuarantee“ to the percentage of bandwidth that is guaranteed, the „ordering“ to the packet ordering.

Below a detail of a concrete application profile of NetMeeting for the video service component:

```
<QoSRequirement>
<maxDelay requirement=„high“ weight=„1“
  unit=„ms“>1200</maxDelay>
<maxJitter weight=„3“ unit=„ms“
  requirement=„low“>120</maxJitter>
<maxLoss weight=„5“ unit=„percent“
  requirement=„medium“>10</maxLoss>
<bw weight=„8“ unit=„percent“
  requirement=„high“>-1</bw>
<ordering weight=„8“ requirement=„true“ />
</QoSRequirement>

```

3.1.2 Implementation dependent description

The following DTD syntax relies on the specification of the [4] QoS request.

```
<!ELEMENT AQUILASpecification (serviceID,
  BSP, BSS, minPU, maxPS, PR, SR)>

```

The „serviceID“ parameter corresponds to the name of the AQUILA network service, „BSP“ to the bucket size for PR, „BSS“ to the bucket size for SR, „minPU“ to the minimum policed unit, „maxPS“ to the maximum (allowed) packet size, „PR“ to the peak rate, „SR“ to the sustainable rate (bit/s).

Detail for the video-conference tool NetMeeting corresponding to a „video very low quality“ scenario:

```
<AQUILASpecification>
<serviceID value=„PVBR“ />
<BSP unit=„bytes“>2000</BSP>

```

```
<BSS unit=„bytes“>5120</BSS>
<minPU unit=„bytes“>60</minPU>
<maxPS unit=„bytes“>1500</maxPS>
<PR unit=„bit/s“>160000</PR>
<SR unit=„bit/s“>75000</SR>
</AQUILASpecification>

```

3.1.3 Discussion

At inter-domain level the generic network service profile enables the negotiation between two domains and the selection of the next domain providing most appropriate network service. For example a „Domain a“ (see Fig. 1) selects on the basis of the general description of its request and the general description of the offers the most appropriate domain fulfilling the commitments to the consumer e.g. „Domain f“. This implies that each domain possesses a mapping entity enabling the translation of the offer resp. the request specified with the network service profile into the proprietary technical representation of the offer resp. request. If we consider a network constituted of n different QoS domains, having a generic description implies having 2n mapping possibilities. Not having a generic description implies n² mapping possibilities.

A mapping with a generic description makes in this manner only sense when the amount of domains offering different QoS is greater than two. As a matter of fact the technology status is such that it still has to be proven that the generic description syntax works with other QoS implementations than [5]. Moreover the IETF [6] activities of the working groups show that network operators etc. are dealing with much more concrete and basic problems and are not so advanced to discuss cross-implementation issues.

As a conclusion it can be pointed out that at a small scale it does not make a big difference having a generic description or not the amount of possible mappings is relatively restricted. But with no generic description the danger is that this amount grows quadratically.

3.2 At Application level

At application level our idea consists in concretely describing applications both in the user space and the network space (see Fig. 1) as well as at control and data-plane level see Fig. 2).

3.2.1 Data plane level

The information described at data-plane level corresponds to the identification of the different flows produced by the application and the corresponding transport protocol. The following DTD syntax enables the listing of the different service components with the corresponding transport protocol.

```
<!ELEMENT Implementation (ServiceComponent,
  TransportProtocol)>
<!ELEMENT ServiceComponent (name, option*)>
<!ATTLIST ServiceComponent
  file CDATA #REQUIRED >
<!ELEMENT option (#PCDATA)>

```

```

<!ELEMENT name (#PCDATA)>
<!ELEMENT TransportProtocol EMPTY>
<!ATTLIST TransportProtocol
  name (TCP | UDP) „TCP” >

```

As example a detail for the NetMeeting tool:

```

<Implementation>
  <ServiceComponent file=„Video.xml”>
    <name>Video</name>
  </ServiceComponent>
  <TransportProtocol name=„TCP”/>
</Implementation>
<Implementation>
  <ServiceComponent file=„Speech.xml”>
    <name>Speech</name>...

```

3.2.2 Control plane level

The information described at control-plane level corresponds to the characterisation of the control protocols, the port numbers, etc. as showed in the following DTD extract.

```

<!ELEMENT protocol
  (lowerPortNo?,upperPortNo?,isControlPort?)>
<!ATTLIST protocol
  name (RTP | ... | H324) „H323”>
<!ELEMENT lowerPortNo (#PCDATA)>
<!ATTLIST lowerPortNo
  value (fixed | configurable) „fixed”>
<!ELEMENT upperPortNo (#PCDATA)>

```

An example for the NetMeeting video-conferencing tool.

```

<protocol name=„H323”>
  <isControlPort value=„true”>1720</isControlPort>
</protocol>
<protocol name=„RTP”>
  <isControlPort value=„false”></is...

```

3.3 At End-user level

Toward the end-user the description is based on the assumption that an end-user cannot express QoS in terms of high complex technical parameters like e.g. RSVP, UMTS, etc. ones. The end-user can express QoS via metaphors corresponding to the human senses: sight, hearing, and on the perception of time-related behaviour. User-friendly descriptions for QoS correspond to a universal apprehension of applications and ideally make reference to well-known similar services from everyday life like: TV, video recorder, hi-fi, or phone etc. The application profile provides a syntax enabling the user friendly description of the QoS sensitive artefacts of applications the so called session characteristic. With this syntax it is possible to describe the application artefacts for the end-user.

The application profile defines a syntax that enables applications to present the QoS offer to the end-user via metaphors and to request for QoS independently from a network implementation.

```

<!ELEMENT SessionCharacteristic (name,
  SemanticalGroup*)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT semanticalGroup description,
  qualifier*>
<!ATTLIST semanticalGroup
  type (Technical|UserFriendly) „UserFriendly”
  language (en |...| gr) „en”>
<!ELEMENT qualifier (#PCDATA)>

```

The name of the service component is e.g. picture size, a semantical group for the service component could be a user friendly description, in English qualified with qualifiers: adjective, terms describing the service component e.g. medium, big ... A detail for the NetMeeting video service component:

```

<SessionCharacteristic>
  <name>Picture size</name>
  <semanticalGroup type=„UserFriendly”
    language=„en”>
    <description>picture size</description>
    <qualifier>very small</qualifier>
  </semanticalGroup>
  <semanticalGroup type=„UserFriendly”
    language=„fr”>
    <description>taille de limage</description>
    <qualifier>tres petite</qualifier>...
</SessionCharacteristic>

```

3.4 Implementation

The presented approach has been implemented within the AQUILA project [5] and successfully deployed. We analysed applications like NetMeeeting from Microsoft, RealSystems from RealNetworks, etc. and produced the corresponding application profiles.

At the beginning of application run-time the end-user tell the AQUILA system (via the AQUILA QoS portal) in a first step which application he uses. Then he choose in the portal (retrieving in a user friendly way QoS information concerning the running application - by using the application and network service profiles) the quality levels he wants. After that the selection is mapped and send to the network.

4. RELATED WORK

The main addressed topic, in connection with the above presented work, is related to QoS description or specification languages, presented in the following. These languages enable the specification of QoS for new applications but does not focus on the support of legacy applications and apart from [10] does not address end-user QoS. The CQML language [9] appears to be the most developed language nevertheless it does not support a standardised description language like XML and its implementation tools (parser, etc.).

The ODP-based QuO [11] (Quality of Service for CORBA Objects) framework provides quality of service (QoS) at the CORBA layer and extends the CORBA functional Interface Description Language (IDL) with the QoS Description Language (QDL).

QDL is an aspect oriented programming language for describing the QoS aspects such as QoS contracts, the adaptive behaviour of objects and delegates, and the configuration of QuO applications.

The MAQS [12] (Management for Adaptive QoS-enabled Services) project includes QIDL an aspect oriented programming language extension of the IDL that supports the specification of QoS interfaces.

The Quality Assurance Language (QuAL) [15] in QoSME [14] enables the specification of how to allocate, monitor, analyse, and adapt to delivered QoS. Applications can express in QuAL their QoS needs and how to handle potential violations.

The QML (QoS Modelling Language) [13] is a general-purpose language for defining QoS properties. QML has three main abstraction mechanisms for QoS specification: contract type, contract and profile.

The Component Quality Modelling Language (CQML) [9] is a language for specifying QoS. The QoS a component provides can be specified independently of how the support is to be implemented and without affecting the specification of its functional properties.

HQML [10] is an XML-based Hierarchical QoS Markup Language, to enhance distributed multimedia applications on the World Wide Web with QoS capability.

5. CONCLUSION

With the concept of network services profiles we demonstrated that at network level an implementation independent inter-domain inter-working scenario is conceivable even recommendable, if more than two different QoS domains interconnect. The working scheme is as follow: at inter-domain level the „Domain a“ maps its request in a generic request that can be compared to the different generic offers provided by the other „Domains i“. This mapping is as well possible at application level between an application and the QoS offers.

With the concept of application profiles we showed that at application level it is possible to supply at least legacy non QoS aware applications with QoS. The solution is based on the assumption that we have a QoS enable network offering network services and non QoS aware Internet applications running on a host. The working scheme is as follows: applications run stand alone at the host in parallel to a so called QoS web portal and protocol gateways. The task of the web portal is to enable the identification of the running application (via manual selection by the end-user), to present the QoS offer in appropriateness with the running application (using the application profiles), and to request for QoS on behalf of the application toward the network. The task of the protocol gateways is to identify control plane information of the running application in order to know which flows are to be supported with QoS. The task of the application profiles is to describe information about application's QoS profile (what requirement does an application have), and so constitute a repository of concrete application profiles.

As a conclusion we want to point out the high efficiency of this solution. With the proposed mechanisms (combination of converters and (generic) description syntax at network and application level) it is possible in an inter-domain inter-working scenario on the one hand to offer user-friendly QoS,

and on the other hand to provide a uniform negotiation and generic description syntax of QoS to support legacy applications.

REFERENCES

- [1] Hatch et al., Selected scenarios and requirements for end-to-end IP QoS management, Deliverable 2, EURESCOM Project P1008, January 2001.
- [2] O'Donnell, Broadband Architectures, ISP Business Plans and Open Access, Paper presented at the 28th Annual Telecommunications Policy Research Conference, Alexandria, Virginia, September 2000.
- [3] Lehr, McKnight, A Broadband Access Market Framework: Towards Consumer Service Level Agreements, Working Paper, Paper prepared for 28th Annual Telecommunications Policy Research Conference, Alexandria, VA, September 2000.
- [4] Winter et al., Final system specification, AQUILA deliverable D1203, 2002. See [5].
- [5] AQUILA Adaptive Resource Control for QoS Using an IP-based Layered Architecture, <http://www-st.inf.tu-dresden.de/aquila/>, 2000-2002.
- [6] IETF Internet Engineering Task Force, <http://www.ietf.org>.
- [7] DTD Document Type Definition <http://www.w3.org/XML/1998/06/xmlspec-report-v20.htm>
- [8] XML Extensible Mark-up Language <http://www.w3.org/XML/>
- [9] Aagedal, Quality of Service Support in Development of Distributed Systems, PhD. thesis, Department of Informatics, University of Oslo, March 2001. <http://www.ifi.uio.no/janoa/papers/thesis.pdf>
- [10] Gu, Nahrstedt, Yuan, Wichadakul and Xu, An XML-based Quality of Service Enabling Language for the Web, Department of Computer Science University of Illinois at Urbana-Champaign, Urbana, UIUCDCS-R-2001-2212, April 2001.
- [11] QuO homepage <http://quo.bbn.com/>
- [12] Becker, Geihs, Generic QoS-Support for CORBA, in Proceedings of 5th IEEE Symposium on Computers and Communications (ISCC'2000) Antibes/France 2000.
- [13] Frølund, Koistinen, QML: A Language for Quality of Service Specification, Software Technology Laboratory, Hewlett-Packard Company, Report: HPL-98-10, 1998. <http://www.hpl.hp.com/techreports/98/HPL-98-159.pdf>
- [14] Soares Florissi, QoSME: QoS Management Environment, PhD thesis, Columbia University, 1996. <http://www.cs.columbia.edu/dcc/publications/thesis/pgsf/>
- [15] QuAL homepage <http://www.cs.columbia.edu/pgsf/qual.html>.
- [16] Salsano et al., Specification of traffic handling for the second trial, AQUILA deliverable D1302, 2001. See [5].