End-to-End QoS Specification Issues in the Converged All-IP Wired and Wireless Environment

Sotiris I. Maniatis, Eugenia G. Nikolouzou, and Iakovos S. Venieris
National Technical University of Athens

ABSTRACT

The emerging next-generation networking environment presents an IP-based core interconnecting many wireless radio access networks, providing ubiquitous access to end users through a vast variety of wireless devices. Although the IP protocol will be the common denominator, the new environment brings together many different interconnecting domains, each following different QoS models, complicating the overall end-to-end QoS process. This article discusses the need to standardize an end-to-end QoS protocol. It does not, however, focus on the signaling mechanism, since there is currently a relevant ongoing activity in IETF. Instead, it concentrates on the formulation of the QoS information describing the QoS requirements of the session to be established. It presents the Generic Service Specification Framework that not only enables the QoS requirements of a specific session to be captured (like a generic QoS template), but also the QoS classes of each IP domain can be described according to it. Through the systematic specification of a domain's QoS classes, an intelligent automatic mapping algorithm can be applied during an end-to-end QoS request, in order to select the most appropriate service class in each domain, as well as to extract the required traffic-related parameters to perform traffic control operations, such as admission control, policing, and scheduling.

INTRODUCTION

There is a strong consensus today that IP will be the foundation of next-generation networking. Since IP already dominates in the wired world, the research community is targeted toward the deployment of all-IP mobile networks, which not only rely on IP for packet transferring, but also exploit IP-based protocols to perform fundamental operations, like mobility, quality of service (QoS), and media control signaling, among others. However, direct exploitation of IP mechanisms and protocols is not straightforward in a mobile environment, since the latter exposes certain characteristics that impose additional requirements.

QoS is still a topic that attains a lot of attention in both the wired and wireless worlds. Regarding the former, the maturity of the integrated and differentiated services frameworks has led to a solidification of the research effort. The main Internet Engineering Task Force (IETF) activity in the field is the definition of an end-to-end QoS signaling protocol within the context of the Next Steps in Signaling charter. On the other hand, in the wireless world there is still substantial research activity, primarily stemming from its intrinsic characteristics (i.e., the involvement of the radio interface and the implications imposed by mobility).

It is evident, however, that the future networking environment will be strongly characterized mainly by the heterogeneity of networks, especially regarding the network access part, although having IP as the common denominator. We can envisage (Fig. 1) that the end-to-end path will first traverse one access network that may be a high-speed wired segment (e.g., DSL), a wireless LAN (e.g., 802.11), a wireless WAN (e.g., UMTS), or even a satellite one. These first-hop segments will probably be supported by an IP-based core network, which will at least supply end devices with IP connectivity, and also include a gateway to the Internet backbone.

In light of the above, even if every part of the envisaged end-to-end path has its own QoS mechanisms, end-to-end QoS provision is not accomplished easily. The shortage of a standardized end-to-end IP-based protocol for establishing QoS, the heterogeneity in QoS models[1] that may apply in different domains of the end-to-end path, and the duality of QoS to be achieved (i.e., in both layer 2 and layer 3) contribute to this distressing situation. Standardization seems to be the only approach to ensure end-to-end QoS provision.

Taking for granted that different networks or domains are free to follow any QoS model, the focus is turned to the enabling factor for establishing an end-to-end QoS path that consists of different provisioning mechanisms in each

---

[1] *The term QoS model is used here to denote a QoS framework or architecture, like DiffServ, IntServ, or the UMTS QoS reference model. It refers to both signaling and the service-level semantics.*

0163-6804/04/$20.00 © 2004 IEEE

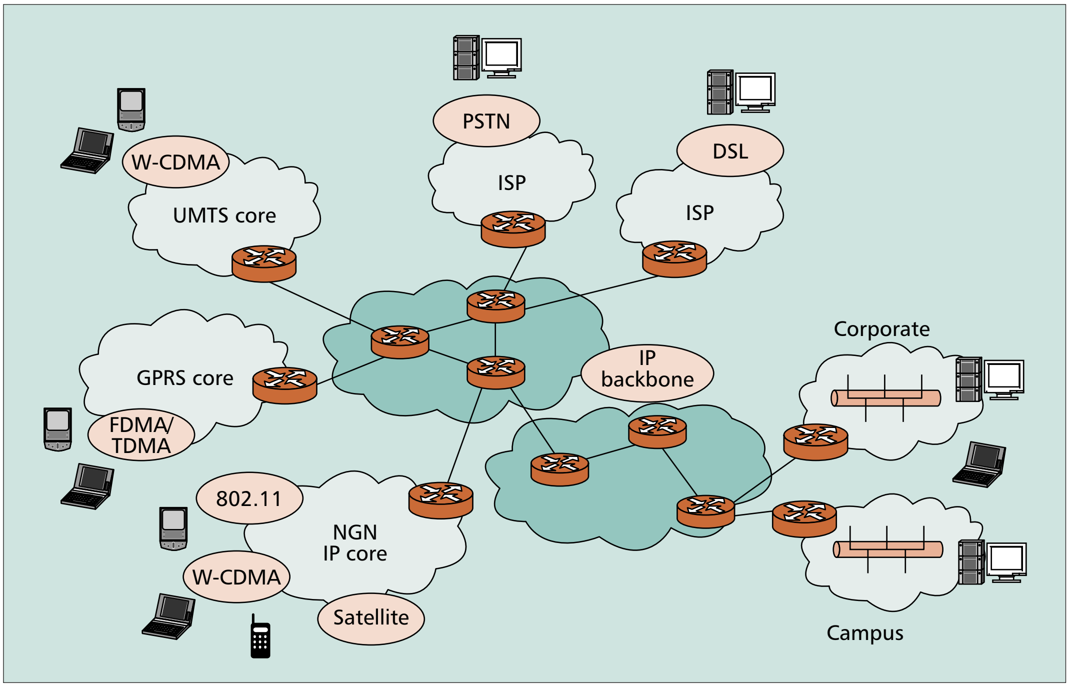

Figure 1. *The future all-IP networking environment will have IP-based domains as the common infrastructure interconnecting various and very different access technologies.*

> *When defining end-to-end QoS resource reservation and management mechanisms, it is imperative that certain requirements are taken into account since the applicability of such solution will span several administrative boundaries.*

domain that must, however, expose similar packet forwarding treatment. In order to accomplish this, we need an end-to-end QoS signaling protocol that can be interpreted in each domain, as well as a universal way to describe the end-to-end QoS requirements. Since wireless access networks have come on the scene, the end-to-end QoS requirements must be able to be translated not only to IP-based (layer 3) service-level semantics for each QoS model, but also to those in layer 2.

The NSIS charter [1] is already advancing toward definition of the NSIS Transport Layer Protocol (NTLP) that is able to serve virtually all end-to-end signaling requirements in the next-generation Internet, also taking into account the special needs of the mobile world. Moreover, as an example of use, they intend to further define the NSIS QoS Signaling Protocol (NSLP), which is based on NTLP, for achieving many aspects of the signaling process.

However, this article does not focus on the signaling protocol. Instead, it mainly focuses on the information this protocol carries for the establishment of the end-to-end QoS. We claim that only through a standardized QoS specification template, the individual networks or domains will be able to understand the QoS requirements. However, we go a step further by defining the Generic Service Specification (GSS) framework, which enables the unhindered exchange and negotiation of QoS information, as well as the mapping process to the service-level semantics of each QoS model.

The article is structured as follows. The article first discusses the requirements of an end-to-end QoS signaling protocol and very briefly presents NSIS. Subsequently, it presents the generic service specification framework in terms of both the schema and the mapping algorithm. Finally, the conclusions and our plans for future work are presented.

QoS SIGNALING PROTOCOLS

When defining end-to-end QoS resource reservation and management mechanisms, it is imperative that certain requirements are taken into account since the applicability of such a solution will span several administrative boundaries, each with its own policies, resource control architectures, and traffic engineering mechanisms. For that reason, a *common language* (signaling protocol) is needed for communicating end to end the QoS requirements of user traffic, while at the same time respecting the individualities of the autonomous operation of each traversed domain.

QoS signaling capabilities are needed to extend the provisioning of QoS in IP-based networks from a static model toward a dynamic one. They will provide the means for communicating the QoS requirements to network entities, and therefore establish the desired QoS level to the end-to-end path. Their role is to carry the information specified in the corresponding QoS framework, which can be understood and interpreted in a uniform way by all networks constituting the end-to-end path. To enable standardization of the carried information, and therefore an interoperable solution across different autonomous networks, it is indispensable to separate the signaling protocol from the carried information. In this way, control information can be carried by any signaling protocol, while being understood by any autonomous system.

When speaking of end-to-end QoS signaling solutions, though, where large transit networks are traversed by a great amount of traffic, scalability and performance issues should be taken into careful consideration. The number of signaling messages exchanged for the establishment and maintenance of reservations, reservation control, and packet forwarding state should be kept at a minimum to allow signaling protocols to scale with the ever-increasing traffic load. Although this coarse granularity comes at the expense of providing hard QoS guarantees to traffic flows, it allows for scalability and efficiency. QoS signaling mechanisms improve the trade-off between quality of guarantee and efficiency of the network, and help to provide differentiated delivery services for individual flows or aggregates, network provisioning, and admission control.

A crucial step in defining an optimal signaling solution is to fix the requirements that such a solution must retain, in terms of framework and protocol that will inter-work with it.

The triggering of the signaling protocols can be performed in different ways; for example, by the user applications, other signaling instances, network management actions, or some network elements. This diversity implies that the signaling can be initiated and terminated in different parts of the network: hosts, edge nodes, or interior nodes. This can be attributed to the current network configuration, client and application capabilities, as well as the access technology used. In some cases, it is desirable to initiate the QoS signaling not from the end user, but from some other element (e.g., a signaling proxy). The need for the introduction of a proxy is apparent in case the end user does not support a specific QoS signaling protocol, or when there are limitations imposed by the capabilities of the end device, security, physical connection between host and network, and customer accounting actions (authentication, authorization).

Another crucial issue in QoS signaling in mobile networks is the interaction between the signaling and mobility protocols. A totally independent operation could lead to ambiguities and even interoperability problems. Loose integration would define the interactions between the two protocols (e.g., how the mobility protocol triggers the transfer of signaling messages), while tighter integration would consider a single protocol carrying both mobility and network state information. Nevertheless, as also discussed by the NSIS working group (WG), tight integration would impede independent development of the signaling protocol and free cooperation with any mobility protocol.

EXISTING PROTOCOLS

Existing QoS signaling solutions seem insufficient in terms of interdomain signaling, scalability, performance, mobility support, and interworking with policy and security mechanisms. At first, Stream Protocol v. 2 (ST2) [2] was designed. Later, Resource Reservation Protocol (RSVP) [3] was developed and standardized, originally designed to establish and maintain resource reservations for end-to-end real-time sessions over the Internet based on the integrated services architecture. However, due to a number of reasons, RSVP and its extensions, even when used for QoS resource reservation, do not meet the requirements of Internet signaling, and are difficult to be deployed in the global Internet. However, despite the lack of its deployment in the Internet, it becomes increasingly important to be able to signal QoS, particularly in access network such as Universal Mobile Telecommunications System (UMTS). To tackle some problems of RSVP, other protocols like YESSIR [4], Boomerang [5], and BGRP [6] were developed, but have not been widely adopted and accepted in the Internet community.

THE NSIS APPROACH

A crucial step in defining an optimal signaling solution is to fix the requirements such a solution must retain, in terms of frameworks and protocols that will interwork with it. NSIS recently produced an RFC [7] that specifies such requirements and describes some usage scenarios where the requirements could be applied. The NSIS WG does not aim to design a new signaling protocol; rather it tries to look into the already existing ones in order to find out the optimal solution to fulfill the requirements.

The main idea of NSIS is to have a layered model, where the lower layer, called the *signaling transport* layer (NTLP), offers generic signaling functionalities, while the upper one, the *signaling application* layer (NSLP), is application-specific. This subdivision caters for the necessity to provide a general model, applicable in any part of the end-to-end path and valid to signal other than just QoS, even though NSIS is mainly focused on QoS. The main function of the NTLP is to forward signaling messages independent of the underlying network locating the appropriate next hop to address the signaling messages and identifying to which flow the messages belong. The NSLP uses the services offered by the NTLP to try to serve the main concern: QoS provision.

THE GENERIC SERVICE SPECIFICATION FRAMEWORK

End-to-end QoS provisioning implies that some kind of traffic control and resource management must exist in each domain that constitutes the end-to-end path. Each QoS model defines its own mechanisms and parameters for traffic control and resource management, although usually at different granularities. It is common, however, for a QoS model to be based on the notion of a class. According to [8], a service class is “the definition of semantics and parameters of a specific type of QoS.” Network administrators use traffic control and resource management techniques to materialize the QoS provisioning within their domain authority, based on the classification of the traffic to a set of service classes.

The most notable instances of existing set of service classes are those of IntServ [9], UMTS [10], and the International Telecommunication Union — Telecommunication Standardization Sector (ITU-T) [11]. In each framework a number of service classes is defined into which applications with similar QoS requirements are categorized. Each service class defines a specific combination of bounds on performance metrics and offers a specific forwarding behavior to its packets. For example, in IntServ there are three service classes (guaranteed, controlled load, and best effort) with different QoS characteristics.

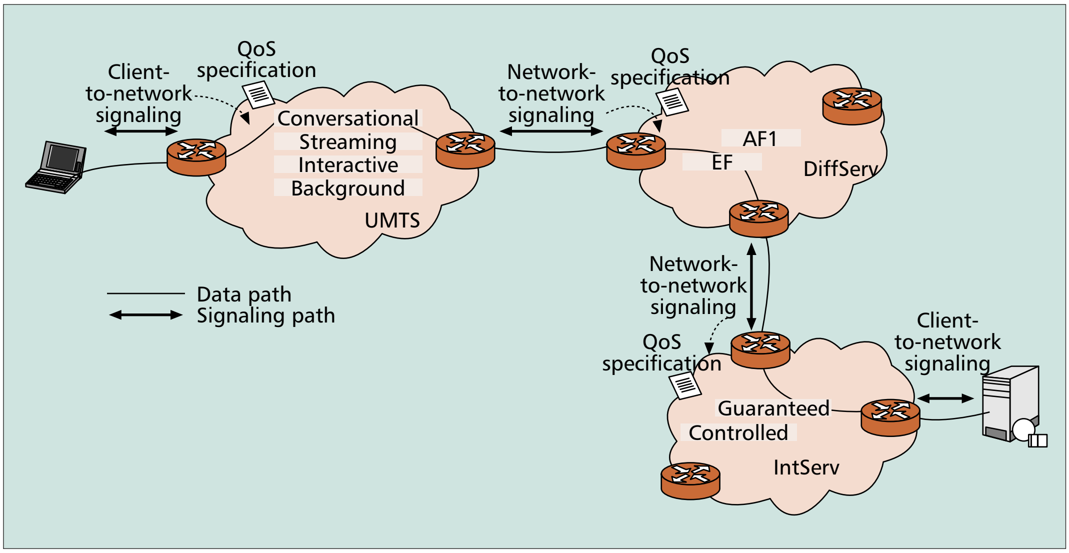

Figure 2. *End-to-end QoS establishment requires selection of appropriate service in each domain and translation of requirements into internal service control elements.*

The GSS schema must be able to describe the QoS objectives of the session to be established as well as to capture services of ultimately any QoS model in accordance to its semantics. In other words, the GSS schema must define all parameters, as well as values and options for them.

The Third-Generation Partnership Project (3GPP) defines four service classes for UMTS (conversational, streaming, interactive, and background [10]), which try to reflect the needs of the corresponding major types of applications. The same observation can be made for the six classes defined recently by ITU-T in Recommendation Y.1541 [11]. On the other hand, the DiffServ paradigm does not delineate a specific set of services, but only provides a rough outline of how services in DiffServ could be defined. For example, the AQUILA project has elaborated one such set of services [12, 13].

In order to facilitate the establishment of the end-to-end QoS, a signaling message carrying the QoS requirements should be interpreted in each different domain, where an appropriate module should translate the QoS requirement into a concrete QoS treatment, based on its own set of service classes, also performing the necessary traffic control functions, like admission control and configuring the resource management mechanisms such as packet scheduling. In light of the above, the QoS requirement must be expressed in terms of a universally accepted format or template, which will also facilitate the automatic translation of the requirements into internal service control elements or QoS parameters (Fig. 2).

The Generic Service Specification (GSS) framework is a step in this direction. It offers a generic service class specification schema that enables network administrators to describe the service classes of their domain according to this schema. Supposing that an end-to-end QoS protocol transfers the QoS objectives of the session to be established, expressed also in this schema, an intelligent mapping algorithm at the network's edges can then be deployed to perform the on-demand correspondence of the session specification to a concrete service class when crossing a domain in the end-to-end path. The generic service specification schema and intelligent mapping algorithm constitute the GSS framework.

GENERIC SERVICE SPECIFICATION SCHEMA

The GSS schema must be able to describe the QoS objectives of the session to be established as well as capture services of ultimately any QoS model in accordance with its semantics. In other words, the GSS schema must define all parameters, as well as values and options for them. The parameters describe different characteristics of the service behavior, while options may take qualitative or quantitative values, or both. Moreover, the GSS schema includes parameters to facilitate the logic of the mapping algorithm to complete.

The parameters cover all aspects, qualitative and quantitative, of the desired service. The goal is twofold: first, the appropriate service class matching the profile described in the requested QoS specification must be selected. Second, the appropriate traffic parameters must be extracted from the requested QoS specification. Looking into the literature, in [8] the authors point out that intrinsic QoS in IP networks is expressed based on at least the following set of parameters: bit rate, delay, jitter, and packet loss. These parameters are related to the nature of the service class and its commitments to the basic IP packet transfer performance metrics, independent of the individual characteristics of the flows served by the service class. Taking that into account, the GSS schema defines the *QoS specification* that consists of the following parameters: maximum transfer delay, maximum delay variation (jitter), maximum two-way delay,[2] packet loss,[3] and bandwidth/throughput guarantee. The values of these parameters can be either numerical or descriptive. For example, one can declare that the maximum transfer delay for a service class is 250 ms. As for descriptive values, the GSS framework determines six discrete values that depict the relative performance of the service class for each parameter. These values are *very low*, *low*, *medium*, *high*, *very high*, and *unspecified*. The quality relativity of the service classes results from direct comparison among the values. The correspondence between the

---

[2] *Probably valuable to capture the responsive nature of an application or a service class.*

[3] *Independent of its origin (congestion or transmission error).*

The bit rate of a flow can be either constant or variable. In case of a variable bit rate, one can further specify the mean, maximum and minimum rates. It is measured in bits/second, but it can also take one of the descriptive values very low, low, medium, high, very high, and unspecified.

numerical and descriptive values is very significant but is left open for further research.

Moreover, the GSS schema defines the *traffic specification* that consists of parameters related mainly to pure traffic issues, like admission control, resource management, and traffic policing. These parameters are *bit rate*, *packet size*, *burstiness*, *greediness*, *adaptivity*, and *duration*.

The *bit rate* of a flow can be either constant or variable. In case of a variable bit rate, one can further specify the mean, maximum, and minimum rates. It is measured in bits per second, but it can also have a descriptive value of *very low*, *low*, *medium*, *high*, *very high*, or *unspecified*. Although bit rate is a traffic-related parameter, it can influence the mapping to a service class. For example, low-delay flows usually presume a low bit rate. However, this parameter is primarily used to perform traffic control operations, like setting a token bucket rate or performing admission control.

The *packet size* can be either variable or constant. For a variable packet size, one can further specify the mean, maximum, and minimum size. These parameters are measured in bytes, while there are also descriptive option values like *very small*, *small*, *medium*, *big*, *very big*, and *unspecified*. The packet size can be very important for some service classes, like those that require low transfer delay, which usually entails very small packet sizes. For example, IP telephony applications usually produce packets of constant size less than 100 bytes. Mixing flows with large packet sizes in a service class intended for IP telephony applications can result in unpredictable and therefore unacceptable delays.

Burstiness reveals a lot of information regarding the bandwidth requirements of the flow. A bursty flow produces traffic with extremely variable bit rate and can severely deteriorate constant bit rate low-delay flows (e.g., voice over IP). Flows or sessions that have bursty behavior must not be mixed with service classes intended for very low-delay flows. This parameter is not only significant for the selection of the appropriate service class, but also for other traffic-oriented operations, like the determination of a token bucket depth. A bursty flow usually requires a larger bucket depth than that of a non-bursty one. The options for this parameter are only descriptive: *very low*, *low*, *medium*, *high*, *very high*, and *unspecified*.

A flow that exposes *greediness* tries to flood the connection with as many packets as possible, for example, a file transfer protocol (FTP) flow. They are usually TCP-based flows. If congestion is detected by the TCP protocol, the flow lowers its bandwidth consumption until it reaches a stable point. Such flows do not match with service classes that try to ensure strict guarantees for the fundamental QoS parameters, like transfer delay and jitter. The options for greediness are *yes*, *no*, and *unspecified*.

An adaptive flow is able to adapt its bandwidth consumption according to the available bandwidth of the line. *Adaptivity* is usually a characteristic of the application that produces the flow, and it is an important factor, because in that case the corresponding resource reservation in the network may have to be modified to reflect the new conditions. Therefore, if a flow is not adaptive, the selection of the service class and determination of its traffic parameters can be very precise so that the end-to-end correspondence is also very effective. The options for adaptivity are *yes*, *no*, and *unspecified*.

Duration is another traffic-related parameter that also reveals some information about the nature of a service. The options for this parameter are *short living*, *long living*, and *unspecified*. Short-living flows are usually related to near-real-time transactions (e.g., a banking application), and require error protection and some loose constraints for two-way delay.

In essence, the QoS specification parameters try to capture the nature of the service, while the traffic specification parameters constitute the traffic-related template for the session. The parameters of the QoS specification reveal the quality relativity among the services. In other words, they present which service is "better" than another. For example, a service with lower delay, jitter, or packet loss, or greater bandwidth guarantee is considered to be a service offering better quality to its flows. For this reason, these parameters are primarily used for the selection of the proper service when a QoS request has come. However, as already discussed, the traffic specification parameters may also provide some important indications about the nature of the incoming traffic, and therefore can also be used in the process of selecting the appropriate service.

INTELLIGENT MAPPING ALGORITHM

In order to facilitate the automatic selection of the appropriate service, upon the arrival of a QoS request, the GSS framework introduces an intelligent mapping algorithm. The mapping algorithm presupposes that a network administrator initially uses the GSS framework to produce an in-depth specification of the service classes of his/her domain. Subsequently, in an end-to-end scenario, a QoS request signaling message will carry the QoS requirements of the session to be established, described based on the GSS schema. Then the mapping algorithm takes as input the session's specification as well as the service classes' specifications, and finds out the best possible correspondence. The outcome of the algorithm is the selection of the most appropriate service class. Subsequently, the traffic parameters that are found in the session's specification are used for operations like admission control and traffic engineering (scheduling, policing, etc.).

The process of the mapping algorithm is in principle very simple. It compares one by one the parameters and selects the service class that is *closer* to the requested QoS specification. In order to define how close a characteristic is to another, we use the concept of quality relativity introduced earlier. It is evident that if the characteristics have the same value, there is a perfect match. However, if the values of the characteristics are different, there is still a chance to select this service class. Imagine, for example, a session that requires a medium transfer delay. This session can be accommodated not only by a service class with medium transfer delay, but also by a

```
<NetworkService servId=”1” servName=”Conversational”>
  <QoSSpecification>
    <maxDelay unit=”ms” value=”veryLow” weight=”10”>100</maxDelay>
    <maxJitter value=”veryLow” weight=”10”/>
    <twowayDelay value=”unspecified” weight=”0”/>
    <maxLoss unit=” percent” value=”low” weight=”5”>0.0001</maxLoss>
    <bwGuarantee value=”unspecified” weight=”0”/>
  </QoSSpecification>
  <TrafficSpecification>
    <packetSize qualifier=”unspecified” weight=”0”>
      <averagePacketSize qualifier=”unspecified” weight=”0”/>
      <maxPacketSize qualifier=”veryBig” unit=”bytes” weight=”2”>1500</maxPacketSize>
      <minimumPolicedUnit qualifier=”unspecified” weight=”0”/>
    </packetSize>
    <bitRate qualifier=”unspecified” weight=”0” variability=”constant”>
      <peakRate qualifier=”high” unit=”bit/s” weight=”2”>2048</peakRate>
      <averageRate qualifier=”unspecified” weight=”0”/>
      <guaranteedRate qualifier=”high” unit=”bit/s” weight=”2”>2048</guaranteedRate>
    </bitRate>
    <burstiness value=”veryLow” weight=”5”/>
    <greediness value=”unspecified” weight=”0”/>
    <adaptivity value=”no” weight=”2”/>
    <duration value=”unspecified” weight=”0”/>
  </TrafficSpecification>
</NetworkService>
```

Table 1. *UMTS conversational class described according to the GSS schema.*

The selection of the language for the GSS schema is important. We used XML for our purposes. Its main advantage is that it offers a standardized text structure, which represents not only the data but also the meta-data (information about the data).

low-transfer-delay one. This, of course, comes at the expense of the other already admitted flows in the low-transfer-delay service class, which may deteriorate, because of other — possibly distressing — characteristics of the medium delay session, such as high burstiness or high bit rate. To be more concrete, a video streaming flow (with medium delay requirements) can be perfectly accommodated by a very low-delay service class, but if this class is intended only for VoIP calls, these may be severely affected, because of the highly probable large packets or very high bit rate of the streaming service. It is apparent from this discussion that it is the choice of the network administrator which algorithm to deploy, subsidizing either only characteristics with an accurate match or also those with some quality relativity. In the latter case, the algorithm should take into account characteristics that reveal some quality relativity, counting them with less weight than those of a perfect match.

The algorithm's operation depends on another parameter defined within the GSS schema. Each parameter of the QoS and traffic specifications is associated with a *weight*. When a service is described in terms of the GSS schema, the weight reveals the importance of each parameter. For example, when describing a service mainly intended for low-delay real-time traffic (e.g., the conversational class of UMTS), the maximum transfer delay and maximum delay variation must have great weight. Moreover, parameters like bit rate (low), packet size (small), and burstiness (low) may also be given relatively large weight, since they are significant in this case. The GSS schema defines 11 options for the weight, from 0 (not relevant) to 10 (most important). Therefore, the algorithm runs through the comparing specifications and adds or subtracts the weight of each parameter to a total value expressing the *degree of correspondence*. If it finds a perfect match, it adds the whole weight of the characteristic; otherwise, if there is still some kind of relativity, it may add a portion of it. On the other hand, if the comparing parameters do not match, it subtracts the whole weight (or a portion of it).

The resulting degrees of correspondence can have positive or negative values. A positive value essentially denotes that the mapping can be adequate for the requesting session; a negative value means the opposite. It is safer to choose the service class with the greatest positive degree of correspondence, although other decisions are not out of scope, further depending on the admission policy deployed. If the mapping algorithm results in all negative degrees of correspondence, the algorithm can be configured to either still select the service class with the greatest degree of correspondence, or select best effort service, in order not to possibly burden the already admitted flows in that service class.

GSS SCHEMA LANGUAGE

The selection of the language for the GSS schema is important. We used the Extensible Markup Language (XML) [14] for our purposes. Its main advantage is that it offers a standardized text structure, which represents not only the data but also the meta-data (information about the data). The latter is embedded in a schema called the Data Type Definition (DTD). The validity of the data can be checked in accordance with the DTD. Last but not least, an XML description is easily followed by a human. In Table 1, the conversational class of UMTS is captured according to the GSS schema.

CONCLUSIONS

This article discussed the process of establishing end-to-end QoS, and focused on the specification of QoS at the service level. It presented the generic service class specification framework

The intention is to try to examine in detail all the important QoS models and their associated service classes, in order to describe the latter in terms of the GSS schema, trying to reveal all shortcomings and deficiencies of our approach.

that contributes to the end-to-end on-demand establishment of QoS-assured services. It provides a schema for describing the service classes of an IP network domain and the QoS objectives of an IP session. The definition of the schema assists in the selection of the appropriate service class in each domain of the end-to-end path that corresponds to the profile of the QoS objectives, based on an intelligent mapping algorithm. Moreover, it facilitates the extraction of the traffic-related parameters for performing traffic control operations.

As far as our future plans are concerned, the intention is to try to examine in detail all the important QoS models and their associated service classes in order to describe the latter in terms of the GSS schema, trying to reveal all shortcomings and deficiencies of our approach. Moreover, we consider that the issue of translating the QoS parameters from the IP layer semantics to layer 2 ones is very crucial to achieve the required end-to-end quality, especially in the light of the next-generation architectures that assume IP as the common network layer, accessible by various complementary wireless and wired access networks. Focusing on wireless access networks, QoS provisioning in the radio access segment is rather significant, since the scarce radio resources, the limited offered bandwidth, and the discontinuations due to link breaks result in a complicated and demanding environment, at least in comparison to traditional wired access networks.

ACKNOWLEDGMENT

This work was performed in the framework of IST Project AQUILA (Adaptive Resource Control of QoS Using an IP-Based Layered Architecture — IST-1999-10077) funded in part by the EU. The authors wish to express their gratitude to the other members of the consortium for valuable discussions.

REFERENCES

[1] Next Steps in Signalling (NSIS) charter, http://www.ietf.org/html.charters/nsis-charter.html

[2] L. Delgrossi and L. Berger, "Internet Stream Protocol Version 2 (ST2), Protocol Specification — Version ST2+," RFC1819, Aug. 1995.

[3] R. Braden *et al.*, "Resource Reservation Protocol (RSVP) — Version 1 Functional Specification," RFC 2205, Sept. 1997.

[4] P. Pan, and H. Schulzrinne, "YESSIR: A Simple Reservation Mechanism for the Internet," *Comp. Commun. Rev.*, vol. 29, no. 2, Apr. 1999.

[5] G. Feher, K. Nemeth, and M. Maliosz, "Boomerang: A Simple Protocol for Resource Reservation in IP Networks," *IEEE Real-Time Tech. and Apps. Symp.*, Vancouver, Canada, June 1999.

[6] P. Pan, E, Hahne, and H. Schulzrinne, "BGRP: A Tree-Based Aggregation Protocol for Interdomain Reservations," *J. Commun. and Networks*, vol. 2, no. 2, June 2000, pp. 157–67.

[7] H. Chaskar, "Requirements of a Quality of Service Solution for Mobile IP," RFC 3583, Sept. 2003.

[8] J. Gozdecki *et al.*, "Quality of Service Terminology in IP Networks," *IEEE Commun. Mag.*, Mar. 2003, pp. 153–59.

[9] J. Wroclawski, "Specification of the Controlled-Load Network Element Service," RFC 2211, Sept. 1997; S. Shenker *et al.* "Specification of Guaranteed Quality of Service," RFC 2212, Sept. 1997.

[10] 3GPP TS23.207, "End-to-End Quality of Service Concept and Architecture," Rel. 5, June 2003.

[11] N. Seitz, "ITU-T QoS Standards for IP-Based Networks," *IEEE Commun. Mag.*, June 2003, pp. 82-89.

[12] E. Nikolouzou *et al.*, "Network Services Definition and Deployment in a Differentiated Services Architecture," *ICC 2002*, New York, NY, Apr. 2002.

[13] S. Maniatis , pp. 82-89, "QoS Issues in the Converged 3G Wireless and Wired Networks," *IEEE Commun. Mag.*, no. 8, Aug 200, pp. 44–53.

[14] Extensible Markup Language (XML), http://www.w3.org/XML

[15] W. Pattara-Atikom *et al.*, "Distributed Mechanisms for Quality of Service in Wireless LANs," *IEEE Wireless Commun.*, June 2003, pp. 26–34.

BIOGRAPHIES

SOTIRIS I. MANIATIS (sotos@telecom.ntua.gr) received a Dipl.-Ing degree in computer engineering and informatics from the University of Patras, Greece, in 1996, an M.Sc. degree in information systems engineering from UMIST, United Kingdom, in 1997, and a Ph.D. degree from the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in 2002. He is currently a senior research associate in the ICBNet laboratory of NTUA. His primary research interests are mobile computing, mobility, QoS and resource management, and 3G and b3G mobile networks. He has received an academic excellence award from the National Scholarship Foundation and is a member of the Technical Chamber of Greece.

EUGENIA G. NIKOLOUZOU [M] (enik@telecom.ntua.gr) received her Dipl.-Ing and Ph.D. degrees from the School of Electrical and Computer Engineering of NTUA, Greece, in 1999 and 2003, respectively. She is currently a senior research associate in the ICBNet laboratory of NTUA. Her research interests include QoS in IP networks, traffic engineering, resource management, and performance analysis. She has participated in various European Union and national projects. She is a member of the Technical Chamber of Greece.

IAKOVOS S. VENIERIS [M] (ivenieri@cc.ece.ntua.gr) received a Dipl. -Ing. degree from the University of Patras, Greece, in 1988, and a Ph.D. degree from NTUA in 1990, all in electrical and computer engineering. During 1991–1992 he was with the National Defense Research Center, Athens, Greece, performing research in the area of telecommunication networks for military applications. From 1992 to 1994 he was a research associate in the Telecommunications Laboratory of NTUA. In 1994 he became an assistant professor in the Electrical and Computer Engineering Department of NTUA, where he is now a full professor. His research interests are in the fields of broadband communications, Internet, mobile networks, Intelligent Networks, internetworking, signaling, service creation and control, distributed processing, agents technology, and performance evaluation. He has over 100 publications in the above areas. He has received several national and international awards for academic achievement. He has been exposed to standardization body work and has contributed to ETSI and ITU-T. He has participated in several European Union and national projects dealing with B-ISDN protocols, mobile networks, intelligent networks, ATM switching and access techniques, intelligent softwar'e, and Internet technologies. He is an associate editor of *IEEE Communication Letters* and a member of the editorial board of *Computer Communications* (Elsevier), and has been a guest editor for *IEEE Communications Magazine*. He is a reviewer for several journals and has been member of the Technical Program Committee and session chairman of several international conferences. He is the editor and co-author of two international books, *Intelligent Broadband Networks* (Wiley 1998) and *Object Oriented Software Technologies in Telecommunications* (Wiley 2000). He is a member of the Technical Chamber of Greece.