UMTS Terminal Equipment For All-IP Based Communications

M. Ricardo[1,2], J. Dias[3,2], G. Carneiro[2], J. Ruela[1,2]

1. Faculdade de Engenharia da Universidade do Porto, Rua Dr. Roberto Frias, 4200, Porto, Portugal
2. INESC Porto, Praça da República, 93 R/C, 4200 Porto, Portugal
3. Instituto Superior de Engenharia do Porto, Rua Dr. Bernardino de Almeida, 4200 Porto, Portugal

{mricardo, jdias, gjc, jruela}@inescporto.pt

ABSTRACT

The paper describes the architecture of an UMTS terminal equipment that provides Quality of Service (QoS) management functions for emerging 3G services in all-IP based communication scenarios. The mapping of IP and UMTS QoS parameters, the reservation procedures and the main IP traffic control mechanisms (scheduling and shaping) are discussed. The activation of a service is exemplified.

I. INTRODUCTION

This paper presents some results of the work carried out in the European IST project ARROWS (Advanced Radio Resource Management for Wireless Services) [1]. This project aims at providing advanced Radio Resource Management (RRM) and Quality of Service (QoS) management solutions for the support of integrated services within the context of Universal Terrestrial Radio Access (UTRA). The project addresses packet access, asymmetrical traffic and multimedia services, all based on IP. The main objectives of ARROWS are: 1) to simulate and validate advanced RRM algorithms for an efficient use of the radio resources at UTRA; 2) to provide QoS bearer services for packet switched flows at the UTRA through the use of QoS management procedures; 3) to demonstrate the benefits of the developed algorithms and procedures by means of an IP based multimedia testbed.

The work described in this paper is related to the third objective. The testbed consists of several functional blocks: 1) an all-IP based UMTS terminal; 2) an UTRAN emulator, implementing the UMTS radio interface and the relevant RNC (Radio Network Control) functions; 3) a gateway implementing functions traditionally assigned to SGSN (Serving GPRS Support Node) and GGSN (Gateway GPRS Support Node); 4) a backbone IP network and its associated routers; 5) a server.

In the remaining of the paper only the all-IP UMTS terminal, which supports multimedia applications and is implemented in a LINUX based PC, is addressed. Its architecture, the selected applications, as well as the interface with the UMTS network interface, which implements the UMTS Non-Access Stratum (NAS) functions, are described.

The paper is organised in six parts. Section II introduces the multimedia applications selected for the ARROWS testbed. Section III presents the rationale for the ARROWS QoS architecture. Section IV describes the UMTS terminal architecture, which is biased towards the compatibility between the IP and UMTS worlds from flow and QoS points of view. Section V gives an example of implementation procedures. Finally, Section VI presents the main conclusions.

II. MULTIMEDIA APPLICATIONS

In UMTS four traffic classes have been identified: conversational, streaming, interactive and background [2]. One main distinguishing factor between these classes is how delay sensitive the traffic is. The conversational class is meant for very delay-sensitive traffic, while the background class is delay tolerant. Moreover, when comparing the conversational and streaming classes, the former requires a tight bound on delay and stringent control of the delay jitter. This is mostly due to the fact conversational traffic is symmetric, while streaming is highly asymmetric and therefore it is possible to use buffers for smoothing out jitter. In conversational services, this would increase the delay acceptable for a natural human conversation, turning the communication awkward.

In ARROWS, one application representative of each UMTS traffic class was selected: Videoconference (conversational), Video streaming (streaming), Web browsing (interactive) and Email (background). All applications were required to satisfy three characteristics: 1) be widely used; 2) be open source, so that extensions to IPv6 or the incorporation of new QoS features could be easy; 3) have port for LINUX.

VIC (video conference tool) and RAT (robust audio tool) were selected as departing applications for Videoconference. Although designed for multicast environments, they are configured in ARROWS as point-to-point (unicast). Both applications rely on RTP (Real Time Transport Protocol), the IETF protocol for the transport of real-time data, including audio and video. It can be used for media-on-demand as well as for interactive services such as Internet telephony. RTP consists of a data and a control part. The latter is called RTCP (Real Time Control Protocol). Both protocols use the services provided by the UDP protocol that, in turn, uses IP.

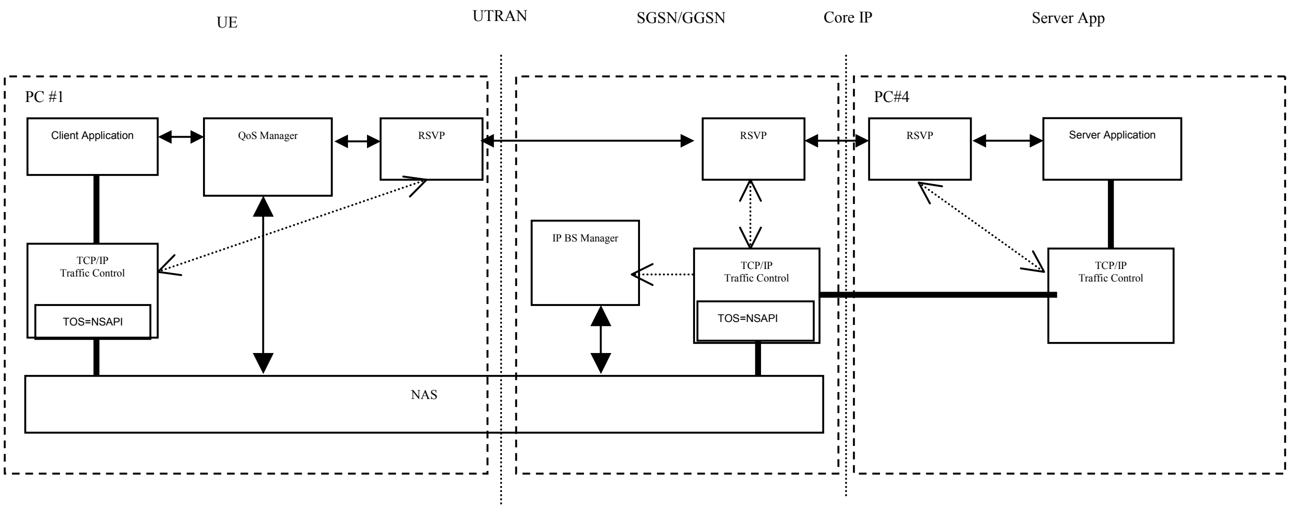

Figure 1 – ARROWS QoS Reference Architecture

VIC supports the H.261 and H.263 video codecs. H.261 is a standard designed for data rates multiple of 64 kbit/s, while H.263 was designed for low bitrate communication (less than 64 kbit/s). RAT supports various codecs, such as G.711 PCM (64 kbit/s), G.726 ADPCM (16-40 kbit/s), LPC (5.6 kbit/s) and GSM (13.2 kbit/s). When used for an audio-video telephony call, these applications generate two real-time and bi-directional IP flows (audio and video) that have to be adequately transported through the UMTS transport services, that is, Radio Access Bearers (RAB) and Packet Data Protocol (PDP) Contexts, as discussed on Section III.
For Video streaming it was decided to use a layered coding scheme that generates two streams with base and enhancement information, respectively. The application that is used is the one developed by MPEG4IP project, which deploys video over the RTP/UDP/IP protocol stack. When playing a stream, two unidirectional and real-time IP flows must be transported over the UMTS network.
Web browsing at the terminal requires a browser, that is, an HTTP client. Email, at the mobile terminal, requires both POP3 and IMAP clients. The three application protocols (HTTP, POP3 and IMAP) use the TCP/IP stack. No real-time requirements are envisaged for the IP flows they generate.
Mozilla is being used as departing point for these applications.

III. QOS ARCHITECTURE REQUIREMENTS

The terminal architecture presented in this paper was, in the first place, driven by the ARROWS testbed requirements. Among other requirements, the terminal should: 1) support only IP based applications and, for that reason, was required to attach only to the UMTS Packet Switched domain; 2) to concentrate mainly on the session management functions of the UMTS Non-Access Stratum, thus activating and deactivating PDP Contexts with QoS; 3) be compliant with the 3GPP specifications.
The 3GPP recommendations suggest that QoS must be first understood from an end-to-end point of view [3]. Considering that all data will be transported as IP datagrams, the obvious departing point was to consider this problem as a standard IP QoS problem where a set of networks are interconnected and, one of them, the access network, is UMTS.
In this approach, the network elements dealing with IP (terminal equipment, GGSN and server) need to cooperate and manage their packet queues taking into account the link level technology in use, so that the packets generated by the applications can be transported with some predictable and differentiated guarantees. Two well-known approaches for dealing with IP QoS are the differentiated services (DiffServ) model, which reuses the old Type of Service (ToS) field of the IP header, and the integrated services (IntServ) model. The IntServ model supports, among others, a Guaranteed Service class aimed at providing packets with a maximum transfer delay between source and destination hosts. This was considered well adapted to the testbed requirements, but does not mean any commitment to the adoption of the IntServ model in backbone IP networks.
The global QoS solution adopted in the project is then that depicted in Figure 1.

A. UMTS Bearers

Compared to GSM and other early generations of mobile networks, UMTS provides a new and important feature: the negotiation of the properties of a radio bearer. Attributes that define the characteristics of the transfer may include throughput, transfer delay and error ratio. UMTS allows a user/application to negotiate the bearer characteristics that are most appropriate for carrying information.
Bearer negotiation is initiated by an application, while renegotiation may be initiated by the application or by the network (e.g., in handover situations). It must be noted that UMTS like other wireless networks cannot provide hard QoS guarantees.
To exchange data packets with external PDNs (Packet Data Networks), the UE must apply for one or more addresses used in the PDN, e.g., an IP address when the PDN is an IP network. In the case of Figure 1, it will be the IP address associated with the NAS driver interface; this is called a PDP address. For each session, a so-called PDP

context is created, which describes the characteristics of the session [4].

Each PDP context is associated to a RAB (between the UE and the SGSN). When more than one PDP context exists, the other PDP contexts must have associated a Traffic Flow Template (TFT). A TFT consists of up to eight packet filters. Each filter contains a valid combination of the following attributes: Source Address and Subnet Mask, Protocol Number (IPv4) / NextHeader (IPv6), Destination Port Range, Source Port Range, IPSec Security Parameter Index (SPI), Type of Service (IPv4) / Traffic Class (IPv6) and Mask, Flow Label (IPv6).

The format and semantics defined for the PDP Context take into account existing reservation protocols such as RSVP [5]. Some UMTS Bearer parameters follow the IETF defined token bucket algorithm, which simplifies the mapping of the QoS parameters within an all-IP architecture.

The interface between the UMTS Bearer and the upper layer (IP) is supported by the Non Access Stratum (NAS). One of the NAS service access points is the Session Management (SMREG-SAP).

Procedures for PDP context activation, deactivation and modification are available at the SMREG-SAP [6]. This SAP exists at the UE and GGSN, but the primitives on each side are quite different. The UE primitives can pass the QoS parameters, unlike the GGSN primitives. Therefore, unless other indirect mechanisms are used, it is possible to activate a PDP Context with a specific QoS only from the UE side. Moreover, if an activation request from the GGSN is accepted (PDP Context activated) it will not be acknowledged.

These two aspects will be dealt with by the QoS Manager, as discussed below.

B. Reservation Procedures

In the ARROWS testbed, RSVP is used to request specific QoS guarantees from the network for particular application data streams or flows.

Four possible RSVP end-to-end signalling sequences are considered in 3GPP recommendations. The solution adopted in ARROWS was to trigger the Create PDP Context Request message after the RESV message.

The UMTS resources are scarce and normally more expensive than the IP backbone ones. Therefore, the approach is to reserve enough resources on the IP backbone so that on the UMTS network a safe margin is available for allocating the remaining resources necessary to meet the applications end-to-end QoS bounds.

In Figure 2 a positive reservation is shown by means of a Message Sequence Chart. The QoS Manager handles this negotiation process from both IP and UMTS layers.

A scenario where data flows from the server to the terminal is exemplified. In a first step, client and server get synchronised by means of an application protocol and both request RSVP for a new session. Then, the data flow generator (the server) requests RSVP to generate a PATH message that is routed as a normal IP packet and indicates to the flow receiver (the client) the flow characteristics and the delay data will suffer. Based on that information, the client requests RSVP to send a RESV message that is used by IP routers to reserve bandwidth. Since PDP Contexts must be activated from the terminal side, a RESV_CONF message confirming the availability of resources in all routers is expected before the new secondary PDP Context is activated.

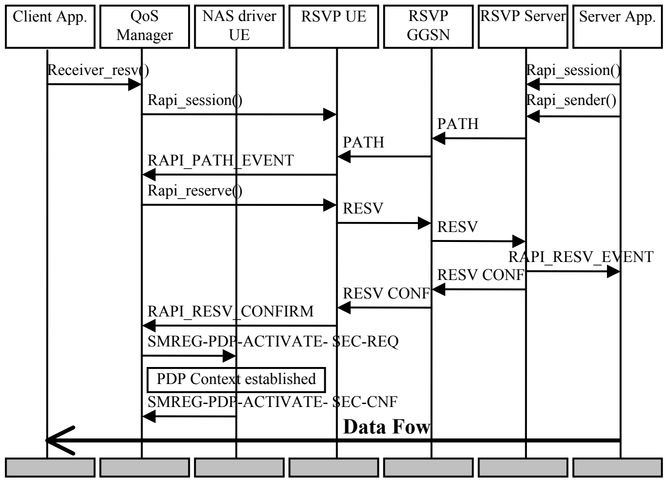

Figure 2 – A positive MSC reservation example

C. QoS Manager

The QoS Manager module is required mainly due to the fact that RSVP lacks support for renegotiation requests from the network. This module only exists in the UE; in the rest of the IP network RSVP mechanisms are used in the normal way.

The QoS Manager has four main functions: 1) handling the renegotiation requests from the network, 2) offering a light API to UE applications, 3) mapping between the IntServ QoS parameters (RSVP) and the UMTS Bearer (PDP Context) QoS attributes, and 4) multiplexing of data flows (RTP/UDP/IP) into the same PDP Context.

The light API aims at providing a simple way of adapting applications so that they can be QoS aware. This API is not as complete as, for example, the RSVP API (RAPI) but is lighter and simple to implement.

A question arises: if there were not such a limitation in RSVP, would the QoS Manager be needed? Probably yes. At first sight, the UMTS bearer could be treated as a normal link, and some changes would be needed in the RSVP at the UE and the GGSN to handle the QoS mapping. In this case the application at the UE would need to implement an RSVP API (e.g., RAPI). But a problem occurs when a reservation is requested by the UE towards the Server, i.e., the RSVP at the GGSN would request a PDP Context activation (link layer) through the NAS at the GGSN. As mentioned when discussing the UMTS Bearers, the primitive responsible for this does not pass the QoS parameter nor an acknowledgment is received in the case of a successful PDP Context activation, which is essential for the completion of the RSVP reservation.

IV. TERMINAL ARCHITECTURE

The architecture of the UMTS terminal that supports IP based services was designed according to the requirements identified in the previous section and is represented in Figure 3.

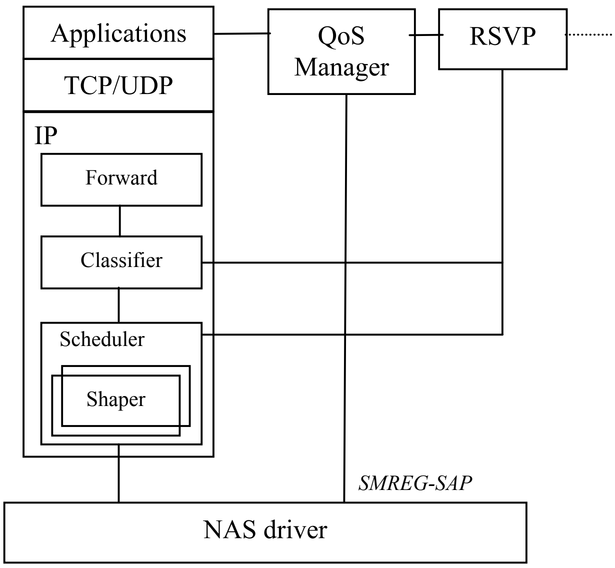

Figure 3 - UMTS Terminal Architecture

A. Functional blocks

Forward includes the IP look-up routing tables and the encapsulation of transport level segments in IP datagrams.
Classifier filters the packets and places them in different queues according to their characteristics. TCP/UDP and IP header fields, such as source port or source IP address, can be used as criteria to classify the packets.
Shaper, simply said, consists of a queue for an IP flow. A queue has properties associated with it, such as bandwidth. A flow must be shaped so that it does not violate the QoS previously negotiated for the associated PDP Context (in the NAS driver).
QoS Manager is the module responsible for negotiating the activation, modification and deactivation of PDP contexts, passing the NAS the desired QoS parameters. It performs the mapping between RSVP QoS parameters and PDP context QoS parameters, as well.
RSVP implements the RSVP protocol that, in ARROWS, is used to guarantee end-to-end QoS to IP flows that traverse both the UMTS and the IP backbone networks.
NAS driver implements Non Access Stratum functions such as session management and mobility management. Within ARROWS the NAS driver is implemented as a software module due to the integration methodology adopted in the testbed. However, in a real scenario the approach could be: on the user plane, the module would be offered as a standard UNIX network interface (umt0) and would be able to exchange datagrams with the IP layer. On the control plane, the NAS module would be offered as a character device driver (/dev/nas0), through which messages for establishing and terminating PDP Contexts would be exchanged.

B. IP Traffic Control

After a PDP context has been negotiated, the terminal may start sending and receiving IP datagrams. However, the terminal may have more than one PDP context activated, each with its own QoS parameters. It is, therefore, necessary to direct the packets to the proper PDP Context, schedule the packets according to their priorities, and shape the traffic so that the flow is compliant with the QoS previously negotiated for that PDP Context. A typical traffic control configuration is shown in Figure 4.

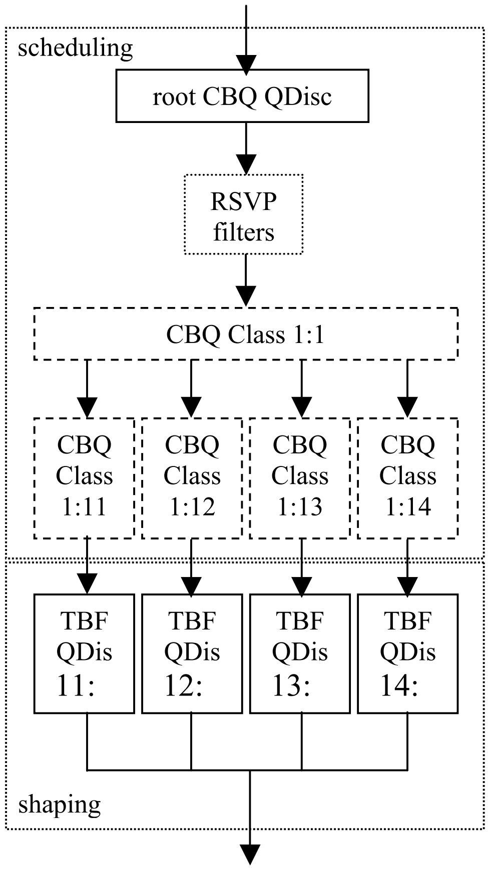

Figure 4 - IP Traffic Control

Scheduling and shaping of the flows are implemented with CBQ (Class Based Queueing) [7] and TBF (Token Bucket Flow) [8] disciplines, respectively. The location of the filters is also shown. For each PDP Context, one leaf class on the CBQ queuing discipline (e.g., 1:11) is created. Each class has one TBF queuing discipline associated instead of the default generic one.
Once the packet is sent to NAS how can it know to which PDP Context the packet belongs? The solution adopted was to use the ToS field of the IP datagram. The TFT associated to a Secondary PDP Context can be given a list of ToS. In ARROWS, there is a one to one relationship between one ToS value and one Secondary PDP Context. This solution will ease the migration of the proposed QoS architecture towards, for instance, DiffServ.

V. IMPLEMENTATION EXAMPLE

In this section the activation of a video streaming service is provided as an example. The video stream is coded using a multilayer codec, such as MPEG-4 (scalable profile) or H.263+. The server tries to make two reservations one for the base layer flow and the other for the enhancement layer flow. The base layer is mandatory, i.e., it must be received by the player (at the UE); on the other hand the receiver may negotiate different bit rates for the enhancement layer, but this flow is delivered with a lower priority. For the base layer a 32 kbits/s bearer is reserved at the IP and UMTS layers. For the enhancement layer a bearer ranging from 16 to 96 kbit/s in steps of 16 kbit/s can be negotiated or if not required or no resources are available, no bearer is reserved at all. In this way it is possible to adapt the video quality depending on the user requirements or the resources available.

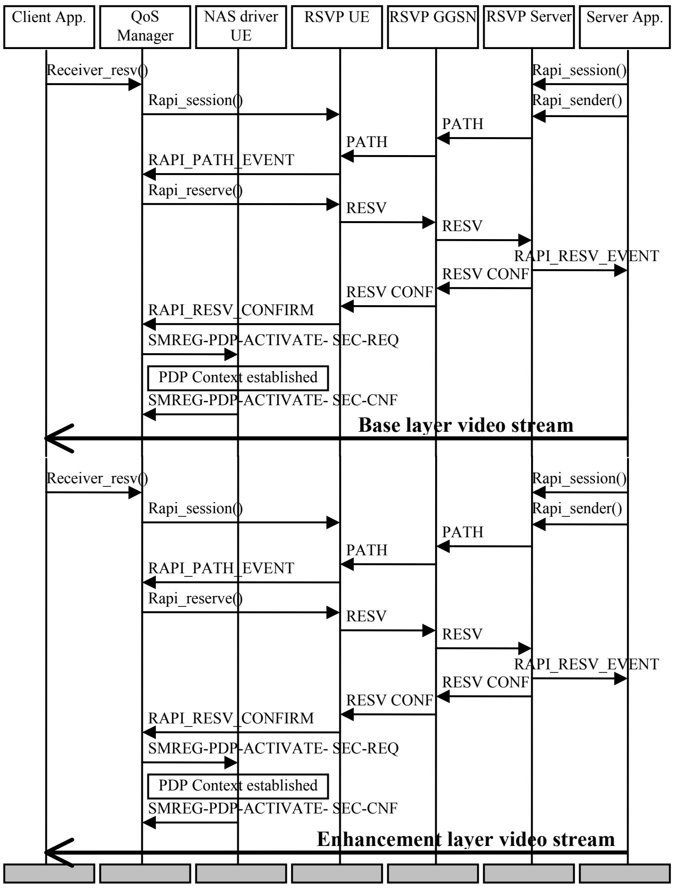

Figure 5 - Activation of a video streaming service

The QoS Manager handles operations such as network renegotiations during a video streaming session. The activation procedures are illustrated in Figure 5, which is similar to Figure 2.

VI. CONCLUSIONS

This paper reports some results of the work carried out within the European R&D project ARROWS and describes the architecture of an UMTS terminal that supports IP based services. The terminal is required to support 4 services (Videoconference, Video streaming, Web Browsing and Email) that are representative of relevant UMTS traffic classes. Videoconference and Video streaming use the RTP/UDP/IP protocol stack and each service generate two real-time flows with QoS requirements. The other services are less QoS demanding and use the traditional TCP/IP stack. The UMTS protocol stack, implementing the well-known Non-Access Stratum, is expected to be available as a module offered with two interfaces: the umt0 network interface for IP packets (user plane) and a character device driver used to establish and terminate RABs (control plane).

The terminal consists of a set of blocks and two of them, the QoS Manager and the IP (QoS), are particularly relevant. The first is responsible for mapping IP into UMTS QoS parameters, to (de)activate PDP Contexts, to (de)multiplex flows as well as to manage the overall resources. The second, besides shaping IP flows, is also responsible for marking the ToS field of the IP packets so that they can be easily classified and inserted in the appropriate PDP Context.

Furthermore, a representative signalling example is given for the case of the video streaming service, where PDP Context negotiation follows the IP level (RSVP) negotiation.

ACKNOWLEDGMENTS

The authors wish to thank the support given by the IST European research programme and their partners within the ARROWS consortium: Universitat Politecnica de Catalunya, University of Limerick, Telefónica I+D and Telecom Italia Lab.

REFERENCES

[1] IST ARROWS project http://www.arrows-ist.upc.es

[2] 3GPP TS 23.107 V5.0.0, "QoS Concept and Architecture", April 2001.

[3] 3GPP TS 23.207 V5.0.0, "End-to-end QoS Concept and Architecture", June 2001.

[4] 3GPP TS 23.060 V5.0.0, "General Packet Radio Service (GPRS); Service description", Jan. 2002.

[5] Paul White, "RSVP and Integrated Services at the Internet: a Tutorial", IEEE Comm. Magazine, Vol. 35, No. 5, May 1997, pp. 100-106.

[6] 3GPP TS 24.007 V4.1.0, "Mobile radio interface signalling layer 3; General aspects", Dec. 2001.

[7] Sally Floyd, "Notes on CBQ and Guaranteed Service", 1995.

[8] K. Dovrolis, M. Vedarn, P. Ramanathan, "The Selection of the Token Bucket Parameters in the IETF Guaranteed Service Class", Technical Report, Department of ECE, University of Wisconsin-Madison, Nov. 1997.