Toward the QoS support in 4G wireless systems*

A. L. Ruscelli[1] and G. Cecchetti[2]

[1] Scuola Superiore S. Anna `a.ruscellii@sssup.it`
[2] CNIT - Scuola Superiore S. Anna `g.cecchetti@sssup.it`

Summary. This paper presents a novel approach to support *Quality of Service* for wireless multimedia applications in the context of 4G wireless systems. Adopting a *Service Oriented Architecture*, it is inspired to *Open Wireless Architectures* (OWA), building a suitable framework over the top of the heterogeneous wireless MACs. It lets to enhance the existing QoS support provided by standard MAC protocols and it uses the *contract model* to guarantee QoS, taking into account the applications requests. It negotiates dynamically *Application Level Contracts* which will be translated seamlessly in *Resource Level Contracts* for the underlying network services. It receives the feedback by underlying network services to adjust the scheduling algorithms and policies to provide soft guarantees. The framework comprises *QoS Manager*, *Admission Control*, *Enhanced Scheduler*, *Predictor* and *Feedback System*. In particular, the QoS manager component is a middleware between applications and lower network layers and it is able to dynamically manage available resources under different load conditions in a transparent manner to application level.

.

Key words: 4G networks, multimedia communication networks, QoS Management, Open Wireless Architecture, Service Oriented Architecture.

1 Introduction

The increasing diffusion of multimedia communications involves the need to manage advanced and wideband multimedia services for which 4G systems provide a mix of concepts and technologies. Some of that are *evolutionary* because they are derived from 3G, while other are *revolutionary* because they are typical of this novel solution. In particular 4G networks improve the 3G

---

* This work has been carried out within the framework of the VICOM project founded by the Italian Ministry of University and Research (MiUR) and in part within FRESCOR EU project (Contract n. 034026

networks approach suggesting a new wireless architecture suitable to the future wireless service provisioning. The latter will be characterized by global mobile access that implies the convergence of the wireless mobile and wireless access in an open, common, flexible and expandable platform. In this context 4G mobile technologies [1] find in *Open Wireless Architecture* (OWA) [2] or *Converged Broadband Wireless Platform*, the suitable model for realizing global mobile access, high quality of service, simple, seamless, automatic access to media services for voice, data, message, video, world-wide web, etc, utilizing an horizontal communication model. This architecture includes baseband signal processing, RF, Networking, OS and application parts so that the same end equipment can flexibly work in the wireless access domain as well as in the mobile cellular networks, with optimal spectrum efficiency and resource ma-nagement. Moreover, the converged broadband wireless system necessarily has to take into account Quality of Service support. It is essential for several multimedia applications like VoIP, video conference call, audio and video strea-ming, contents distribution, Internet services and real time control services. The QoS support for OWA needs to be functional for integrated wired and wireless access modes using a common methodology and offering a differentiated service according to strict latency/throughput applications requirements, while the used medium offers time and space varying communication conditions. It turns out that the variability of available radio resources does not allow the network to provide hard QoS guarantees. Instead, the network must provide soft QoS guarantees constrained by a minimum channel quality. Some of these guarantees regards: *delay*, *delay jitter*, *packet loss ratio*, *throughput*, *bandwidth.* In particular the QoS provision must take into account the support done by each single access mode, however leaving space to build blocks for a full Quality architecture. In this context one trend is to use an adaptive QoS system with a relative QoS differentiation [3] based on different priority classes of Differentiated Service architecture to deliver multimedia data [4]. Another remarkable point of view is to introduce a cross-layer design with adaptive QoS assurance for multimedia transmissions [5], [6].

In this article we present a novel framework to provide a comprehensive Soft QoS support for multimedia traffic streams, inspired both to cross-layer architecture idea and QoS differentiation and it allows to interface seamlessly multimedia applications with lower layers of a wireless network. The wireless application framework is composed by QoS Manager and Scheduling Subsystem (Admission Controller, Scheduler, Predictor and Feedback mechanism). We specifically focus on QoS Manager which is a middleware between applications and lower network la-yers and it is able to dynamically manage available resources under different load conditions in a transparent manner to application level. It accepts the different QoS requests from the various multimedia applications and it translates these in the specifics of each involved medium access standard, handling time-varying network conditions, heterogeneous traffic streams and link layer resources.

2 The Cross Layer Framework

The framework has a cross-layer architecture (Fig. 1) composed by a middleware for QoS, the *QoS Manager*, and the scheduling subsystem. The former transparently manages the communication levels for applications while the latter consists of some building blocks to regulate the heterogeneous networks MAC layer. This subsystem may vary depending on the particular MAC(s) used (e.g. IEEE 802.11, IEEE802.16, MIMO, mobile public networks, wired networks, etc.). The QoS manager is independent both to application and MAC. Using this approach it lets one possible solution to convergence question, providing an open communication gateway architecture, which represents high level abstraction that lets practitioners to concentrate on the specification of the application requirements. Furthermore, it adopts a Service-Oriented Architecture which allows it relies mostly on the services provided and on the required applications. From the user perspective the framework allows an easy and simple access to multimedia services, hiding the complexity of the lower MAC levels of the different networks. The contract based scheduling implemented lets to the applications to dynamically specify its own set of complex and flexible execution requirements, written as a set of a service contracts for different resources, which are negotiated with the underlying implementation. To accept a set of contracts the QoS Manager has to check, as part of the negotiation, if it has enough resources to guarantee all the specified minimum requirements while keeping guarantees on all the previously accepted contracts negotiated by other application components. Eventually it adapts the requirements to available resources. If a result of this negotiation is accepted, the scheduling subsystem reserves enough capacity to guarantee the minimum requested resources and it reclaims any spare capacity available to share it among the different contracts that have specified their desire or ability for using additional capacity. The contract also contains Quality of Service tuning parameters that may be used by QoS manager.

1.

2.1 QoS Manager

QoS manager [7] is a middleware layer that mediates between application and underlying components of this framework. Different applications specify different sets of high level parameters (e.g., Multimedia Streaming, VoIP, signaling protocol and file transfer have different parameters and performance indicators). The set of high level QoS requirements of the application will be specified through an *Application Level Contract* (ALC). The QoS manager acts as a proxy: it translates the high level QoS requirements of the application into the resource allocations, it computes transmission parameters values and it negotiates them with admission control. The set of low level resource requirements produced by QoS manager will be called *Resource Level Contract* (RLC). Actually the underlying network may be heterogeneous, it may vary

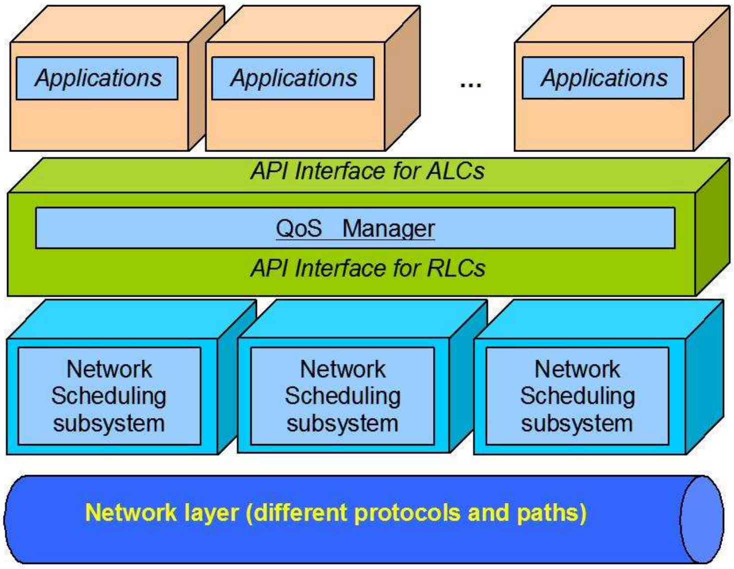

Fig. 1. Cross−Layer Architecture

in topology and standards, offering a completely variable scenario. For this reason, QoS manager has to interact with different scheduling subsystems, one for each different standard. Each subsystem has an admission control, a scheduler, a predictor and a feedback control. We can assume without lack of generality that each protocol does not interfere with other ones. When QoS manager interacts with a subsystem it provides the appropriate parameters and it takes into account the specific protocol used. We have implemented the QoS manager as a *"two-side" Application Program Inter-face* (API). The upper side interfaces applications while the bottom side interacts with scheduling subsystem. The applications can call the following functions of QoS manager:

```
int request_ALC (struct *ALCspec);
int modify_ALC (struct *ALCspec);
int cancel_ALC (struct *ALCspec);
```

An application uses `request_ALC` to negotiate an ALC with QoS manager. The latter checks the requirements specified with `struct *ALCspec through` *Admission Control subsystem* and returns the result 0 if the ALC is accepted, 1 if the ALC is modified or 2 if the ALC is rejected. If the ALC is modified, the application can cancel the current contract issuing `cancel_ALC`. At any time an application can request to modify the ALC by `modify_ALC` . This function behaves like `request_ALC`. To abort a contract an application use `cancel_ALC`, which should return 0 unless an error is occurred. The API bottom side is used by QoS manager to interact with the underlying levels. It consists of three functions for every protocol managed by QoS manager:

```
int request_RLC_proto (struct *RLCspec);
int cancel_RLC_proto (struct *RLCspec);
int get_RU_proto (struct *RU_proto);
```

where proto may be any supported network protocol (e.g. for 802.11e the function names are: `request_RLC_80211e` and `get_RU_80211e`). The first function is called by QoS manager to negotiate a RLC with the scheduling subsystem and it returns 0 if the RLC is accepted, 1 if the RLC is modified or 2 if the RLC is rejected. If the RLC is modified and ALC can still be satisfied, the QoS manager adjusts ALCspec and return it to the application requesting the corresponding contract, while if the RLC is modified but the ALC cannot be satisfied the QoS manager cancels the RLC through a `cancel_RLC_proto` and then notifies to the application that it cannot accept the ALC. RLCs are also canceled when ALC expires. The function `get_RU_proto` is used by QoS manager to query the scheduling subsystem about protocol resource utilization. The returning `RU_proto` lets the QoS manager to enhance its negotiation capability. Moreover QoS manager performs other actions useful to optimize QoS support:

- it adapts automatically the resource allocation to dynamic changes in the requirements of the application, tuning service parameters: when an application wants to change the contract profile, the QoS manager contacts again the corresponding admission control service and negotiates a new RLC. It perform a so-called adaptive resource allocation;
- it adapts dynamically the resource allocation in order to optimize the resource utilization without sacrificing on QoS requirements;
- it maintains as much as possible the resource allocation for each application as close the minimum that is needed to fulfill the ALC;
- if an overload occurs (e.g. due to varying network conditions or if a more important QoS request is received), it can decide to change one or more ALCs to degrade the QoS level of one or more applications by a call-back notification so that the application itself can adapt its QoS requirements.

2.2 The Scheduling Subsystem

The scheduling subsystem is composed by the admission control, the scheduler and the feedback mechanism.

Admission control verifies if there are sufficient resources for medium access to satisfy QoS manager requests. It computes the theoretical new bandwidth utilization and it checks if it is admissible without degradation of preexistent transmissions. The response is sent back to the QoS manager. If the instance request is successful a RLC is established and the QoS manager can communicate transmission parameters to the corresponding scheduler. The general admission test used is:

$$\sum_{i=0}^{N} \frac{Q_i}{P_i} \leq 1. \tag{1}$$

where $Q_i \equiv C_i/r_i$ is the average time budget of the medium which is reserved to the i^{th} network station transmitting within each period P_i; r_i is the physical bit rate assumed for admission control computations of the i^{th} traffic stream, C_i are the bytes transmitted during the P_i and *Ulub* is least upper bound utilization factor computed for the worst-case available bandwidth. If there is not enough bandwidth to serve the new request, three different admission control policies exist which act as follows:

- *saturation policy*: the highest possible budget is assigned to the task so that the total resource utilization does not exceed *Ulub*,
- *compression policy*: in respect of the established ALCs, all the RLCs are recomputed ("compressed") so that we can make new space for the new request,
- *reject policy*: the transmission is rejected.

The **scheduler** manages each transmission for each admitted flow and it assigns dynamically both the period P_i and transmission duration to follow the channel variability and streams characteristics. We propose a scheduler which can handle Traffic Stream (TS) with Soft Real Time guarantees [8] with special regard to VBR flows. VBR flows are supported by assigning transmission duration in agreement to the effective temporal demands. The assignment of P_i is dynamic, so it lets to increase the transmission frequency of the applications having in queue traffic with tightening requirements of QoS. The scheduler is also able to reclaim the unused time of nodes which have exhausted their transmission before the end of their transmission duration and then it assigns that time to the nodes which have still useful data to transmit.

Delay or advance of the transmission with respect to the pre-agreed rate (in terms of bytes which have been anticipatively used or have not been transmitted by node) are formalized as the scheduling error $\varepsilon_i^{(k)}$, defined, at the k^{th} time instant, as the difference between the cumulated bytes to transmit $z_i^{(k)} \equiv kC_i^{(k)}$ and the bytes actually transmitted $z_i^{(k)}$:

$$\varepsilon_i^{(k)} \equiv z_i^{(k)} - \overline{z}_i^{(k)} \tag{2}$$

The dynamic equation for the evolution of the scheduling error for the i^{th} real-time data flow is:

$$\varepsilon_i^{(k+1)} \equiv \varepsilon_i^{(k)} + C_i^{(k)} - \gamma_i^{(k)} Q_i^{(k)} \tag{3}$$

where $\gamma_i^{(k)}$ is the actual channel speed.

The **feedback** mechanism senses the effective information acknowledged by nodes. It eventually uses the information provided by a predictor to vary transmission parameters of the scheduler in order to respect hard and soft deadlines. It is responsible to minimize the scheduling error.

The rapidity of this action can be improved turning on special weights w_i for each traffic stream (TS_i). The feedback system can compensate little variations of network conditions without the intervention of admission control to establish new RLCs. During *normal condition*, if

$$\sum_{i=0}^{N} \frac{Q_i}{P_i} \leq 1. \tag{4}$$

is kept, feedback system controls the scheduling error assigning:

$$\forall i,\ Q_i^{(k)} \triangleq \widetilde{Q}_i^{(k)} = \frac{C_i^{(k)} + \alpha_i \varepsilon_i^{(k)}}{\rho_i^{(k)}}$$

where $\widetilde{Q}_i^{(k)}$ is the required assigned budget to compensate the scheduling error, $\alpha_i \in]0, 1]$ and $\alpha_i \varepsilon_i^{(k)}$ is a fraction of the current scheduling error for each TS_i and $\rho_i^{(k)}$ is the predicted channel speed at the physical layer.

During *overload condition* the allocated budget to each TS_i is decreased. For example, if the feedback scheme uses a weighted distribution, for each TS_i, $\widetilde{Q}_i^{(k)}$ is decreased of an amount proportional to the weight w_i assigning:

$$\forall i,\ Q_i^{(k)} \triangleq \widetilde{Q}_i^{(k)} - \frac{w_i \widetilde{Q}_i^{(k)}}{\sum_{j=1}^{N} w_j \widetilde{Q}_j^{(k)}} \left(\sum_{j=1}^{N} \widetilde{Q}_j^{(k)} - U_{lub} T_i \right)$$

where

$$\frac{w_i \widetilde{Q}_i^{(k)}}{\sum_{j=1}^{N} w_j \widetilde{Q}_j^{(k)}}$$

is the percentage of decreasing.

3 Conclusions

In this paper we have presented a description of a cross layer framework to integrate QoS support for 4G systems under time varying network conditions and different traffic specifications. It provides an interface to QoS support mechanisms for any applications with tightening guarantees and temporal boundaries. This framework lets applications establish contracts with QoS manager that administers the available resources from under-lying subsystems. The resulting QoS service is an improvement for applications running over wireless networks. The QoS manager acts as a proxy towards different network sub-systems which manage different wireless network protocols. Using this approach it lets one possible solution to convergence question and it takes a response to simultaneous need, in 4G systems, of ubiquity and diversity that imply flexibility and individuality, providing an open communication gateway architecture, which represents high level abstraction that lets practitioners to concentrate on the specification of the application requirements.

References

1. J. Shelper, “1g, 2g, 3g, 4g,” *TechColumn*, April 2005.
2. W. Lu and J. Hu, “Open wireless architecture - the core to 4g mobile communications,” *CIC Journal on China Communications*, Apr 2006.
3. C. Dovrolis, D. Stiliadis, and P. Ramanathan, “Proportianal differentiated services: Delay differentiation and packet scheduling,” *IEEE/CM Trans. Networking*, vol. 10, pp. 12–26, Feb. 2002.
4. J. Shin, “Quality-of-service mapping mechanism for packet video in diffentiated services network,” *IEEE Trans. Multimedia*, vol. 3, pp. 219–231, 2001.
5. A. Alwin, “Adaptive mobile multimedia networks,” *IEEE Pers. Commun*, vol. 3, no. 2, pp. 34–51, 1996.
6. M. Mirhakkak, “Dynamic bandwidth management and adaptive application for a variable bandwidth wireless environment,” *IEEE J. Select. Areas Communn*, vol. 19, pp. 1984–1997, 2001.
7. T. Cucinotta, L. Abeni, G. Lipari, L. Marzario, and L. Palopoli, “QoS management through adaptive reservations,” *Real-Time Systems Journal*, vol. 29, no. 2-3, Mar 2005.
8. J. Stankovic, K. Ramamritham, M. Spuri, and G. Buttazzo, “Deadline scheduling for real-time systems,” K. A. Publishers, Ed., Boston, 1998.