The DAIDALOS Architecture for QoS over Heterogeneous Wireless Networks

Gustavo Carneiro[1], Carlos García[2], Pedro Neves[3], Zhikui Chen[4], Michelle Wetterwald[5], Manuel Ricardo[1], Pablo Serrano[2], Susana Sargento[3], Albert Banchs[2]

[1]INESC Porto, {gjc, mricardo}@inescporto.pt
[2]Departamento de Telemática, Universidad Carlos III, {cgg, pablo, banchs}@it.uc3m.es
[3]Instituto de Telecomunicações, Universidade de Aveiro, pneves@av.it.pt, ssargento@det.ua.pt
[4]Universität Stuttgart, zhikui.chen@rus.uni-stuttgart.de
[5]Mobile Communications Department, Institut Eurécom, michelle.wetterwald@eurecom.fr

Abstract—**Next generation networks will include a diversity of wireless technologies for the network access. In the DAIDALOS project, an architecture that provides seamless QoS support, mobility, security and multicast, among other features, is being designed for such a heterogeneous wireless environment. This paper focuses on the wireless QoS part of this architecture: it is a modular architecture consisting of an the Abstraction Layer, which provides a generic interface to the upper layers and handles the technology independent functionality, and the Abstraction Layer Drivers, which implement the technology specific QoS functions. Abstraction Layer Drivers have been developed for the technologies 802.11, 802.15.1, 802.16 and TD-CDMA. The architecture further includes a source channel adaptation module that, by adapting the amount of resources reserved to the characteristics of the wireless channel, allows maximizing the overall performance of the wireless network.**

Index Terms—**modular architecture for QoS, wireless technologies, source channel adaptation**

I. Introduction

The DAIDALOS project [1] aims at providing an integrated architecture for multiple network technologies, both wired and wireless, with quality of service capabilities under a common authentication, authorization, accounting, auditing and charging (A4C) framework and in a secure communication environment. The integration of all these technologies and functionalities represents a major multi-disciplinary effort being undertaken by DAIDALOS.

One of the main challenges in the design of the DAIDALOS architecture is the provisioning of end-to-end QoS guarantees in such a heterogeneous wireless environment. In this paper we describe how this issue is dealt with in DAIDALOS, with a special focus on the wireless QoS architecture. The main design guidelines and merits of the architecture proposed are outlined as follows:

- A modular architecture that separates the technology specific and independent parts has been designed, which facilitates the future inclusion of new wireless technologies into the architecture.
- The architectural designed is flexible to allow the concatenation of several wireless technologies in the access.
- Automatic learning techniques are used which minimizes the configuration requirements and facilitates operation.
- The architecture is integrated with the core wired QoS architecture to provide users with end-to-end QoS guarantees.
- A solution integrated with mobility protocols has been designed to ensure that QoS is not disrupted during handoffs.
- Technology specific enhancements and configuration algorithms have been developed and designed for the various wireless technologies considered.
- Source channel adaptation applications are used in order to optimize the wireless resources usage. These applications interface with the QoS architecture to maximize the overall wireless performance.

This paper focuses on the wireless QoS part the Daidalos architecture. Other aspects of this architecture, including wired QoS, security, A4C, mobility and multicast, and their interactions, are described elsewhere [1].

The rest of the paper is structured as follows. In section 2 a description of the wireless QoS architecture and its interfaces with wired QoS and mobility is described. The Abstraction Layer, which implements the technology independent functionality, is described in Section 3. The Abstraction Layer Driver implements technology-specific functionality; a short overview of this module for each of the technologies implemented, namely 802.11, 802.15.1, 802.16 and TD-CDMA, is given in Section 4. Section 5 describes the source channel adaptation module and its interaction with the rest of the architecture. Finally, conclusions are given in Section 6.

II. Architecture

Daidalos QoS main goal is to provide an end-to-end QoS solution with strict guarantees independently of the wireless technology that is being used to access the network. Such a high level of support and transparency implies that strong and reliable integration methods are developed.

In the wireless QoS architecture, two different scenarios are distinguished: the single-hop and the two-hop scenarios. The single-hop scenario is composed of a set of wireless technologies, namely, 802.11e, 802.15.1 and TD-CDMA. In this case, the Access Points (AP) of each technology are directly connected to the Access Routers (AR) of the wired part of the Access Network (AN). The two-hop scenario is a concatenation of two wireless technologies, namely, 802.16

[5] in the first hop and 802.11 or 802.15.1 in the last hop. The 802.16 solution is used as a backhaul link for the 802.11 and 802.15.1 networks. It provides long range connections (30 km) with high throughputs (70 Mbps). The Base Station (BS) of the 802.16 network is directly connected to the AR.

To accomplish the demands of a transparent end-to-end QoS architecture, a set of modules and interfaces, presented in Figure 1, have been defined and are being implemented. The figure represents a two-hop scenario. Particularly, four modules have been defined. One is the **Abstraction Layer (AL)** module, which is responsible for the signalling part of the QoS architecture. It implements the functionality of resource management in the QoS architecture. The solution adopted for resource management has been designed with the following objectives: capability of performing QoS reservation and resource querying from the AR to remotely located APs, support for concatenation of multiple wireless technologies, and ability to work without requiring a priori centralized knowledge of the L2 topology. It is composed of three variations of the same basic module working in the Mobile Nodes (MN), APs and ARs. The **AL Driver** module is responsible for the QoS support for a particular technology in a single network link; it is the module that directly communicates with the specific technology, translating general QoS parameters, like TSpec and RSpec [9], to technology specific QoS. The **Source Channel Adaptation** (SCA) module, located in the MN and the AR, is used to enhance the application performance over wireless. Finally, the **TD-CDMA** module, located in the TD-CDMA AR and MN, implements the AL and the AL driver functionality in TD-CDMA.

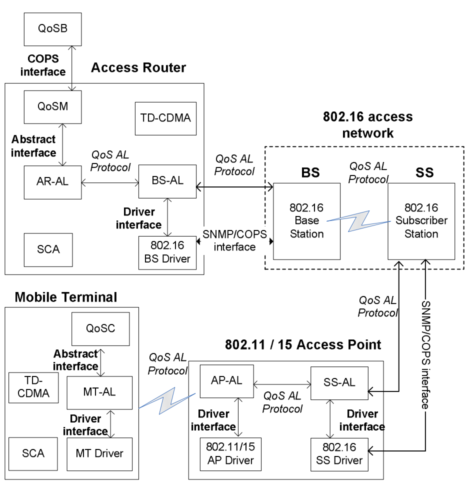

Figure 1 Access network modules, interfaces and protocols

Besides the previously referred modules, a set of interfaces has also been defined. The **Abstract Interface (AI)** in the AR is the interface between the AL in the AR and other modules in the same node, namely QoS Manager (layer 3 entity located in the AR which is responsible to reserve layer 3 resources and to communicate the reservation requirements to the AL) and the mobility related modules. It offers a set of primitives to reserve QoS resources in the LAN, query available bandwidth in APs, and receive QoS degradation notifications. The AI in the MN is the interface between the AL and the QoS client (layer 3 entity located in the MN which queries layer 3 QoS reservations); it offers primitives to, receive QoS, bit error rate information, and to provide notifications of newly created connections, with respective QoS and Context Information data. The Driver Interface is used to translate abstract QoS to technology specific QoS. There are also SCA related interfaces used to configure SCA parameters.

Briefly, the support of end-to-end QoS works as follows. When the AL in the AR receives a notification from the QoS Manager to reserve resources for a new flow in the current access network, through the AI, it triggers a request message from the QoS AL protocol that transfers the layer 2 QoS reservation parameters throughout the AR towards the MN. The MN takes a decision about the reservation request and sends back a response message through the QoS AL protocol to the AR. Each element receiving this message performs the reservation in the wireless link, through the AL Driver Interfaces. In the specific case of 802.16 Driver, the QoS configuration is performed through SNMP or COPS. When receiving the result of the reservation process, the AL in the AR informs the QoS Manager.

A. *Layer 3 QoS architecture*

Figure 1 includes an element in the access network denoted as QoS Broker. This element performs admission control and manages network resources in the access network; it controls the network routers according to the active sessions and their requirements. Moreover, this is the element that triggers, through COPS, the QoS Manager of the ARs, in the sender and receiver domains, for the reservation of the layer 2 wireless resources [6].

The interaction between the layer 3 and layer 2 QoS architecture is as follows. When a new flow is requesting admission, the QoS Broker receives the information (refer to [6] for details) containing the QoS characteristics of the reservation to be performed. The QoS Broker checks the availability of resources, and if the flow can be accepted through the information on the network layer resources, it triggers the QoS Manager to reserve the required resources. The QoS Manager reserves the required bandwidth and buffer space in its queues, and triggers the layer 2 QoS reservation process, through the AI, to perform the reservations in the wireless link. When the QoS Manager receives an answer of this reservation process result, it communicates the result to the QoS Broker. The QoS Broker can then accept the new flow in its network.

B. *QoS with mobility*

One of the main challenges in the QoS support is its integration with mobility. For the mobility purposes, extensions to the IPv6 fast mobility [7] are used. To provide seamless mobility between different technologies, the resources in the specific wireless link need to be reserved before the MN actually moves. For this purpose, in the handover preparation phase, both the layer 3 and layer 2 resources are pre-reserved in the new access network. For this reason, when the mobility module in the new AR receives a mobility request of a MN, it contacts the AL and sends information on the flow, QoS parameters and AP to attach.

The AL in the AR triggers the pre-reservation process. If the resources are correctly pre-reserved, the AL in the AR receives a positive notification and contacts the mobility module in the AR. The MN is then authorized to perform handover, and it may attach to the new network. At this stage, the reservation is confirmed as in the session setup process.

III. Abstraction Layer

The services offered by the QoS Abstraction Layer can be classified into four groups: 1) QoS reservation; 2) resource querying; 3) L2 QoS notifications; and 4) handover. These services are following described.

A. QoS reservation

The main service offered by the QoS AL consists in the creation of *QoS connections*. A QoS connection is a virtual channel between a MN and an AR, provisioned with QoS guarantees such as bit rate and delay. The QoS AL exists only at the control plane, as a signalling protocol. For that reason, no additional header is inserted into a data packet which, in Daidalos, consists of an IPv6 packet transported in some layer 2 frame. The IPv6 Flow Label field is reused in a context local to an access network, and each value identifies uniquely a connection in a given access network.

The service of QoS reservation is offered at the AR by the primitive `AL-CNX-ACTIVATE-REQ`, which takes as parameters, among others, the L2 address of the destination end point, and the desired QoS attributes. The QoS parameters include a Tspec and an Rspec, with the same meaning as described in [9]. The `AL-CNX-ACTIVATE-REQ` primitive triggers the signalling described in Sec. III.E, and returns a *connection identifier*, if the reservation succeeds. This identifier is used to mark the flow label field of the data packets that will be associated to the connection, and to modify and deactivate the QoS connections; it is also included in the QoS degradation notifications.

B. Resource Querying

There is a primitive to request a report of the available (free) resources in the access network, the `AL-RESOURCE-QUERY`. Its only parameter, `DestAddr`, identifies the "path" for which resources are to be queried. `DestAddr` can be an L2 address of a MN (request a single report for all APs in the path towards the given MN), an AP (request a single report for all APs in the path towards the given AP, including the destination AP itself), or a broadcast address (request reports for all APs in the AN, one report for each unique path towards an edge AP). Currently, the only reported parameter is the available bandwidth.

C. L2 QoS Notifications

The primitive `AL-CNX-INDICATION` is issued from the AL at both AR and MN to indicate that the AL was forced to modify the QoS for a specific connection due to changing conditions in the wireless medium. The primitive includes the connection identifier and the Rspec as parameters. This primitive is important, for it allows link adaptation by applications, and may serve as trigger for higher-level handover decisions.

D. Handover

There are a couple of primitives defined to support smooth handover between APs. AL-CNX-PREPARE-REQ works just like AL-CNX-ACTIVATE-REQ, except that it takes as extra parameter the address of an AP. It instructs the given AP (and all other APs in between) to start preparing QoS resources for a MN. Later on, when the MN executes the handover to the new AP, it issues an AL-HANDOVER-EXECUTE primitive to activate the resources previously prepared.

E. The Protocol

The mapping from primitives to Protocol Data Units (PDUs) is straightforward. The QoS AL PDUs are transmitted as Ethernet frames (although other L2 networks can be supported with little additional effort) with a new protocol type. It is inspired by the RSVP protocol, in the sense that it is also based on in-band signalling and soft-state. Moreover, just like in RSVP Path messages (host's IP addresses as destinations and intermediate routers intercepting the messages), the QoS AL primitives have the MN as destination address (or an AP, in some cases) and intermediate APs intercept these messages to implement the QoS AL protocol, before forwarding them again to the original destination.

By using the MN address as destination address of the signalling frames, several design goals are achieved, namely: 1) path discovery, 2) support of concatenated APs, and 3) support of transparent dynamic reconfiguration of the network topology. This all comes "for free" with IEEE 802.3 Learning Bridges [10], implemented by switches and wireless APs. The forwarding "indecisions" of IEEE 802 bridges, as we name it, were also solved. The protocol was designed to avoid committing anything on reception of request messages, but to always wait for a response, to avoid acting on reception of messages received only because the MAC forwarding table of a bridge has expired (and the bridge decided to broadcast it).

IV. Abstraction Layer Drivers

The AL Driver is the technology-dependent part of the AL module, which translates abstract QoS to technology specific QoS (based on the technology beneath). This division, done via the Drive Interface, allows the integration of almost any communication technology into the access network.

A. 802.11e

The WLAN AL driver is based on the EDCA access mechanism of the upcoming 802.11e standard [2], which provides service differentiation on the basis of a number of configurable parameters (namely, AIFSi, $CW_{min}i$, and $CW_{max}i$). The remaining challenge with this mechanism is the design of admission control and configuration guidelines in order to provide QoS guarantees. These are currently subject of ongoing research [3], [4].

The WLAN AL driver consists of the following modules (Figure 2):

- Driver controller: this is the core module, located at the AP. It executes the admission control and reconfiguration algorithms, using the measurements provided by the monitoring manager and the QoS requests from the AL.

Decisions on acceptance/rejection of QoS requests, and the computation of the parameters configuration in each case, are performed by this module.

- Monitoring modules: the monitor client at the MT gathers network performance information and conveys it to the monitor manager which collects the information of all MT's and provides it to the driver controller.
- Configuration module: the configuration manager sends the configurations computed by the driver controller to the configuration clients at the MT's which configure the 802.11e driver with the given parameters.

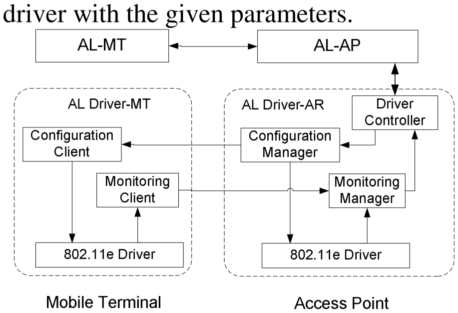

Figure 2 802.11e AL Driver modules

B. 802.15.1

The Bluetooth QoS AL driver is responsible for translating abstract QoS requests from the QoS AL to Bluetooth configuration primitives. It is integrated with the Bluetooth PAN profile [11], which already takes care of session management and creates a best-effort connection between a MN (PANU) and an AP (NAP). The driver is to be implemented on top of the Linux kernel and BlueZ stack.

The Bluetooth QoS AL driver is initially launched by the PAN daemon running in both MN and AP, and at that point registers itself to implement QoS for the newly created '`bnep`*n*' network interface. From this point on, new QoS AL connection requests are directly mapped into L2CAP links by the driver.

This sounds simple enough. However, some building blocks in the Bluetooth stack are missing, namely QoS mapping between L2CAP and ACL layers, support for multiple concurrent L2CAP links between PANU and NAP, and, at BNEP layer, a QoS mapping engine, which uses IPv6 Flow Label to select an output L2CAP link.

Regarding L2 QoS notifications, the trigger used is the HCI event "QoS Violation Event". This event is sent from the Bluetooth device whenever the requested QoS can no longer be achieved. After discovering the new level of QoS, an `AL-CNX-INDICATION` primitive is issued by the driver and transported by the QoS AL to MN and AR.

To implement `AL-RESOURCE-QUERY`, the driver in the AP keeps track of all L2CAP channels created so far and, considering the packet types used for each connection, computes the occupied and free slots, thus an estimation of available bandwidth in the piconet can be derived.

C. 802.16

The 802.16 driver modules are responsible to enforce the set of requested QoS parameters in the 802.16 wireless links. Particularly, they perform the communication between the AL module and the 802.16 system. Basically, there are two instances of the 802.16 driver module: one is responsible for the communication with the Base Station (BS) equipment (802.16-BS driver) and the other is responsible to establish the communication with the Subscriber Station (802.16-SS driver) premises (SS). This set of modules and interfaces are represented in Figure 3.

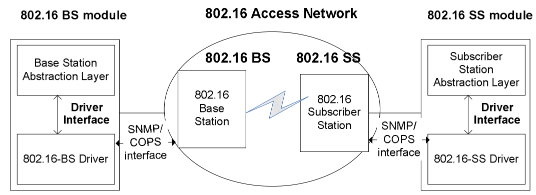

Figure 3 802.16 AL Driver modules

The 802.16-SS Driver will be responsible for the activation of the requested service flows (uplink and downlink) in the 802.16 wireless links with the set of QoS parameters which were included in the primitive sent by the AL module. Additionally, the 802.16-SS Driver is also responsible for the change and/or deactivation of the service flows. On the other side, the 802.16-BS Driver is not able to manage the service flows. It is responsible to perform the resource queries in the 802.16 wireless links and provide these to the upper entities. The 802.16-BS Driver and the 802.16-SS Driver have two interfaces: one interface with the AL module (Driver Interface) and one interface with the 802.16 equipment (SNMP/COPS interface). The Driver interface is used to establish the communication between the AL and the drivers (802.16-BS and 802.16-SS). The communication primitives are sent between these two modules using this interface. When a message from the QoS AL protocol (issuing the resource reservation or querying for resources) arrives to the AL (in the SS or BS, depending on the purpose), it sends a primitive to the 802.16 drivers. The content of the primitive is decoded and a new message is built.

D. TD-CDMA

The TD-CDMA AL driver re-uses the Moby Dick project architecture and implementation. It is enhanced to comply with the AI. It provides a single-hop architecture through the TD-CDMA Radio Interface. It can be divided into three main components.

The ALs located in the AR and in the MT take care of the signalling received through the AI and, after the necessary adaptation, forward it to the corresponding TD-CDMA Driver.

The AR TD-CDMA Driver, located in the AR, performs the mapping of the primitives received from the AL, based on IP technology, to the UMTS-TDD operation. In the architecture inherited from Moby Dick, the UMTS Access Stratum, implementing the Radio Interface, is directly inter-connected to the IP protocol stack through the TD-CDMA driver. This driver plays the role of the Non Access Stratum for the radio interface. In the control plane, the driver maps the IP parameters of the flow, triggers the allocation/de-allocation of the radio channels (or radio bearers) and sets up some classifiers to drive the data traffic towards these radio resources. In the user plane, the driver sorts and relays the IP packets towards the corresponding radio channels.

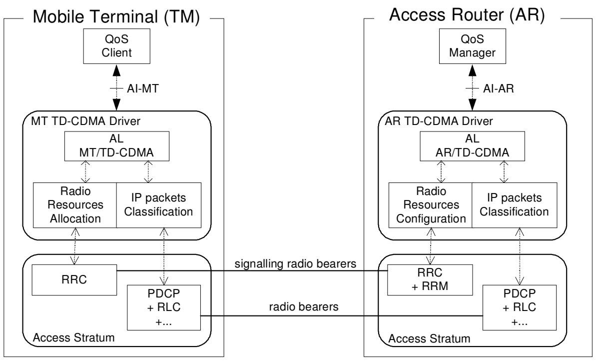

Figure 4 TD-CDMA AL Driver modules

The MT TD-CDMA driver, located in the MT, performs the allocation of resources at the MT level and classifies the packets in a symmetric manner.

Both drivers communicate through the UMTS signalling radio bearers and trigger the radio bearer procedures of the RRC protocol, as described in [8] to configure the radio channels.

V. Source Channel Adaptation

As with any wireless communication, a reliable delivery mechanism must be provided. Generally, two strategies are used to guard against the unreliable signal channel: forward error correction (FEC) and automatic repeat request (ARQ). In real-time communications, the retransmitted packets arriving at a media decoder may also be discarded due to late arrival. Therefore, for real-time communications, ARQ is not preferred. This section proposes a QoS strategy using the following four techniques:

- The latest video codec standards, the ISO/IEC video codec standard H264/AVC [12], which provides an enhanced bit error resilience capability when compared with previous codec standards. The encoded data can be partitioned into three sections; enabling the sensitive header information and resilient picture coefficients to be separated.
- UDP-Lite [13], which can deliver erroneous packets and has the functionality for dealing with erroneous packet payloads. It can also cope better with lost packets in the above codec by checksumming the partitioned header information.
- Protocol header compression, RoHC [14], which reduces the IP overhead allowing network providers a faster return on investment for their 3G/B3G/4G networks whilst, at the same time, enabling real–time services by improving the IP packet latency over bandwidth-limited links.
- Forward error correction, which is very important for eliminating the need for retransmission because of the unreliable communication environment and long transmission delays involved. Retransmission of data severely degrades the overall data throughput performance of the channel, especially in real-time communications.

SCA module studies and implements the optimized customizations of error-resilient techniques of multimedia over an unreliable wireless communication environment based on co-operation with optimised RoHC/FEC, UDP-Lite for recent code.

VI. Conclusions

This paper describes a modular architecture which provides seamless QoS support over different wireless technologies for the access network, and describes the appropriate interface with core network to perform end-to-end QoS. The architecture is composed by two main modules: Abstraction Layer, which provides communication with the core network to perform management, and Abstraction Layer driver, which is technological dependent and performs QoS operations regarding technology.

The proposed architecture provides the following features: Scalability: any new coming access technology can be merged in the architecture just adding the correspondent AL-driver module; Support for complex scenarios: the two-hop capability allows concatenation of two different technologies to extend the access network; and Integration with other functionalities: appropriate interfaces have been defined in order to provide compatibility with mobility, security and multicast.

Acknowledgment

The work presented in this paper was partially funded by the EU project IST-2002-506997 “Daidalos” [1].

References

[1] Daidalos IST Project: Daidalos: “Designing Advanced Interfaces for the Delivery and Administration of Location independent Optimised personal Services”. (FP6-2002-IST-1-506997). http://www.ist-daidalos.org

[2] 802.11e standard IEEE 802.11e/D6.0, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS), Draft Supplement to IEEE 802.11 Standard, October 2003.

[3] A. Banchs, "Analysis of the Delay Distribution in 802.11 DCF: Step Towards End-to-end Delay Guarantees in WLAN", 5th International Workshop on Quality of future Internet Services (QofIS'04), Barcelona, Spain, September 2004.

[4] A. Banchs and L. Vollero, "A Delay Model for 802.11e EDCA", accepted to IEEE Communications Letters.

[5] IEEE 802.16e/D4-2004, “IEEE Draft for Local and Metropolitan Area Networks – Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems – Amendment for Physical and MAC L. for Comb. Fixed and Mob. Op. in Licensed Bands”, August 2004.

[6] Rui Prior et al, “Heterogeneous Signaling Framework for End-to-end QoS support in Next Generation networks”, HICCS-38, Hawaii International Conference of System Sciences, Hawaii, January 2005.

[7] R. Koodli (ed.), *Fast* Handovers for Mobile IPv6, Internet Draft, October 2003.

[8] 3GPP TS 25.331: "RRC Protocol Specification"

[9] S. Shenker and J. Wroclawski, "General characterization parameters for integrated service network elements." RFC 2215, Sept. 1997.

[10] I. T. Telecommunications, "Iso/iec final dis 15802-3 (ieee p802.1d/d17) may 25, 1998.

[11] Bluetooth SIG, “Personal Area Networking Profile”, February 14, 2003.

[12] T.Wiegand and G.Sullivan, “DRAFT ISO/IEC 14496-10 AVC”, *Draft ITU-T Recommendation H.264*. March 2003

[13] L-A. Larzon, M. Degermark, S. Pink, et al, “The Lightweight User Datagram Protocol (UDP-Lite)”, *IETF RFC 3828*, July 2004.

[14] C. Bormann, C. Burmeister, M. Degermark, et al, “Robust Header Compression (RoHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed”. *IETF RFC 3095,* July 2001.