Next Generation Network Provider Architecture Demonstrator

Victor Marques[1], Carlos Parada[1], Pedro Gonçalves[2], Rui L. Aguiar[2,3], Francisco Fontes[1]
[1] Portugal Telecom Inovação, P-3810-106 Aveiro, Portugal
[2] Instituto de Telecomunicações, P-3810-193 Aveiro, Portugal
[3] Universidade de Aveiro, P-3810-193 Aveiro, Portugal

Abstract

This paper presents a next generation network demonstrator currently under deployment in Aveiro. The demonstrator supports mobile IP with fast handovers with integrated QoS and AAAC issues considerations. A monitoring system provides network information adequate for QoS management, and a specially built QoS Broker is able to manage multiple types of Access Routers. The demonstrator has been tested with some applications, and is currently under refinement.

I. INTRODUCTION

IP networks are assuming an increasing importance with most of the services traditionally supported on the circuit-based networks being ported to these "next generation networks". Telecom operators, with telephony service as its traditional main income, are now putting a large effort on the provisioning of this and new multimedia services on IP networks, preparing themselves to satisfy the needs and will of the increasing more demanding users and companies. On another hand, the equipment vendors are also very enthusiastic with the new frontiers open by the mass use of IP and what this may bring in terms of business in a near future. Together, telecom operators, vendors, research institutes and universities are quite active in the research and development forums that deal with next generation networking and services.

However, the provision of services with real time constraints, such as voice telephony, brings a new set of problems that did not exist in previous (circuit-based) networks. Another issue being raised in these new networks is terminal and user mobility. Real time communications must not suffer any degradation during handover. Quality of Service (QoS) provision and control is thus a problem to face.

The QoS problem appears when the resources are scarce and the applications requirements are high, which is the case, for instance, when using multimedia applications in radio based access networks. In this sense, the providers must be able to differentiate the QoS among the several types of services and even between different users. This control is only effective if the key control entities have a real time view of the network status in terms of resource usage, availability and performance. This paper presents a demonstrator of next generation networks. This demonstrator implements an advanced architecture, developed under the framework of the IST project Moby Dick (PTIn is a partner in the project) [1]. This architecture has been developed from a provider point of view, and is able to grant user access to services based on the specific contract with the provider, and control resources in real time by means of the integrated use of a QoS Broker and a Monitoring system.

On the next section we present the network demonstrator, describing its main elements and the overall operation. On section III we present a detailed description of the QoS System, specially the QoS Broker. Following this section, the Network Monitoring system is also described in detail. Finally, on section V we present our main conclusions.

II. NETWORK DEMONSTRATOR DESCRIPTION

The overall 4G architecture underlying this paper is IPv6-based, supporting seamless mobility between different access technologies. The target technologies envisaged are Ethernet (802.3), for wired access and Wi-Fi (802.11b), for wireless LAN access. A W-CDMA (the physical layer of UMTS), for cellular access can also be introduced at a later stage, since the W-CDMA access router development is in a terminal phase.

Fig. 1 presents a conceptual view of the network demonstrator installed at the Institute of telecommunications and Portugal Telecom Inovação premises. The main entities on this network are the AAAC server, which deals with user authentication, authorisation, accounting and charging, the QoS Broker that deals with the service admission control per user and per service and performs overall network configuration, the network monitoring system, with an EO Manager (Operation Element Manager) and a EMs (Measurement Elements), the Access Routers, the Home Agent (collocated with the AAAC and the DNS and also with core routers functions) and the terminals. Also, two wireless access points are present at the moment.

Mobility is a key problem in this environment, because inter-technology handovers are supported. Thus mobility cannot be simply handled at the physical layer, having to be implemented at the network layer. An "IPv6-based" mobility mechanism is used for interworking, and no internal mechanisms for handover, both on the wireless LAN and on the W-CDMA, is used. So, in fact, the 802.11 nodes are handled as in ad-hoc mode, and IPv6-protocols are used to handle movement between different cells(similarly, no mobility mechanisms will be supported on W-CDMA cells, and the same IPv6 protocols will handle the movement between cells).

In this scenario, multiple network functions/sub-systems were identified and are demonstrated:

- Physically supporting the mobility of the terminals, over multiple technologies;
- Guaranteeing planned QoS levels to specific traffic flows;

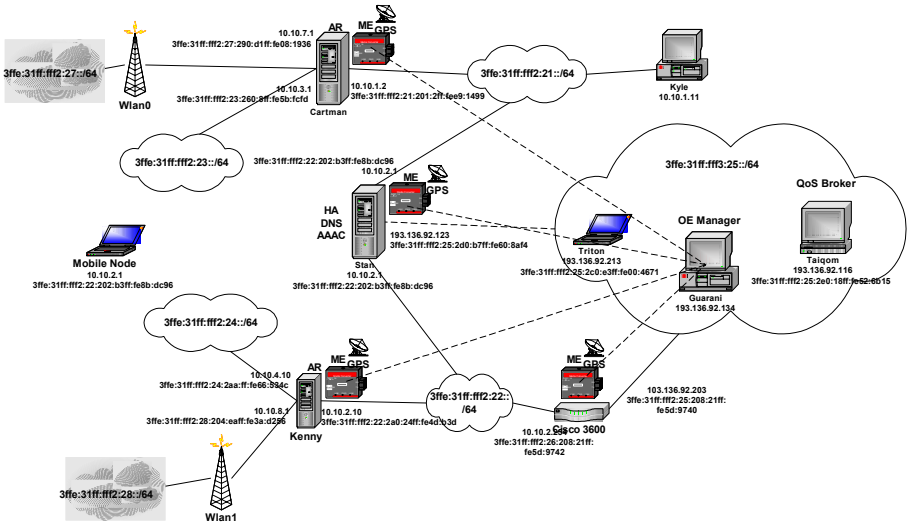

Fig. 1 – Network Demonstrator

- Supporting inter-operator information interchange, for multiple-operator service provision;
- Realizing appropriate monitoring functions, for providing information to the service operator about network and service usage and network level of operation.

The whole network, including management functions and applications has been implemented with the IPv6 (IP version 6) protocol stack over Linux environments. The implementation relies on MIPL (Mobile IP for Linux), together with Fast Mobile IP extensions. Also developed were all other network and stack entities required to allow the mobile terminals to seamlessly operate in this heterogeneous environment: QoS, AAAC and Monitoring sub-systems are responsible of serving each user according to the SLA previously negotiated, operating respectively at the network level and at the service level, using a differentiated services strategy; security has also been integrated in the network on the user access link [1]. The software for these entities was developed resorting to a mix of modifying existing implementations and internally developed modules.

III. QOS SYSTEM

A hard constraint of our architecture is the simultaneous support of mobility and QoS. We developed an innovative usage of QoS Brokers (QoSB), incorporated in a traditional DiffServ approach to be able to control and manage available resources in an efficient way, even for mobile users. The QoS Broker interacts with the AAAC system during the (required) user registration phase receiving from it all relevant user-specific information for QoS provisioning, the Network View of the User Profile (NVUP). This NVUP contains information describing the services subscribed by the user in the MN. The QoS Broker may then perform SAC (Service Admission Control) decisions on every service request done by the user's terminal based on the NVUP and on the network state. For that, the QoS Broker also interacts with the ARs in its QoS domain. These interactions are required both for AR's QoS configuration and for service authorisation. "Services" are requested by simple DSCP marking on outgoing packets, with each DSCP corresponding to a different contracted service. To start using a contracted service, the MT just needs to start sending packets marked with the DSCP for that service. When a new traffic flow from the MT is trying to gain access to the Core Network, the QoS Manager (on the AR) sends a resource request to the Qos Broker that after analysing the network resource availability and network monitoring results accepts or denies the access by performing devices (ARs) configuration. Policing is performed, following the COPS outsourcing model (selecting or modifying the appropriate PHB). When a user initiates a FHO to move to a new AR, it is also the QoS Manager that starts the QoS-aware handover procedure, involving the old and new ARs, and the old and new QoS Brokers.

Fig. 2 presents the main QoS system entities and their main building blocks.

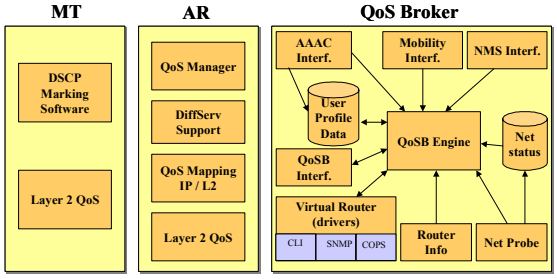

Fig. 2 – QoS architecture basic components

The MT has a DSCP Marking Software, a function that marks the outgoing traffic with the relevant DSCP for the network QoS-aware service being used. A L2 QoS function covers specific physical-layer QoS aspects, dependent on the access technology (relevant mostly for radio access).

On the Access Router a L2 QoS function also exists, with similar responsibilities. This is associated to a QoS Mapping function. This block includes the mapping between DiffServ QoS parameters and the appropriate parameters of the access network. Traditional DiffServ Support is also present, for usual diffserv shaping and policing on the traffic to be sent

from and to the MT. The critical QoS function of the AR is the QoS Manager. This block performs control functions related with shaping, policing, per-hop-behaviour (PHB) and mobility. It also sends periodically reports of the use of resources to the QoS Broker.

A. QoSBroker

QoSBroker is the entity responsible for controlling the state of network resources. It monitors the network edges for incoming and outgoing resource reservation and utilization.

It searches in an internal database named NetworkDB for the AR's where configuration actions are required, and queries the format (CLI, SNMP, COPS) that should to be used to communicate. Then the QosBroker reads the commands required to make the configuration action as well as the additional information such as the MIB or the management remote login password.

The Qos Broker internal architecture presented in Fig. 2 can be briefly described:

- Interfaces:
  a. *AAACInterface is the interface used to receive the NVUP information from the AAAC system;*
  b. *RouterInterface is the interface used by the AR to send the service requests to the QoSBroker. It has several drivers that allow the QoS Broker to communicate with AR using several protocols: CLIDriver, is a remote login API developed to enable remote configuration actions in the AR; COPSDriver, is the driver used for communication with the AR (Linux and Cisco IntServ enabled routers); SNMPDriver, a driver that enables communication with network devices MIB's;*
  c. *QBrokerInterf interfaces the other Qos Brokers. It is used in the handover processes to send relevant user information to the new QosBroker;*
  d. *NMSInterf is the interface where the QoS Broker receives network status information from the monitoring system;*
  e. *RGInterface is an API that communicates with a Radio Gateway in order to open radio channels;*
- NetProbe is an entity that monitors the network devices load state. The information collected is then inserted in the NetworkDB;
- RouterInfo is an entity that monitors the running state of the network devices. The information collected by this entity is placed into the NetworkDB. It also discovers new elements in the QosBroker.
- QBrokerEngine is the core of the QosBroker. It takes the SAC decisions and coordinates the lunch and the killing of the threads running in the QosBroker. QEngine implements several internal entities:
  a. *UserProfile manages all the information related to the QoS profiles defined by the AAAC system, and the registered users information. It also implements an internal database named UserProfileDB that keeps all the information needed by UserProfile;*
  b. *NetStatus (i) manages all the information related with the service authorization and network device load; (ii) takes the SAC decisions for the QEngine and receives all the service requests; (iii) it implements an internal database named NetStatusDB that is a subset of the information of NetworkDB. NetStatus.*
  c. *RouterAttendant receives all the requests from the AR and parses the messages before sending them to NetStatus. It can read requests sent using the COPS protocol.*
  d. *RouterAnalyser is an entity used by NetStatus to check the load status of some network device.*
  e. *RouterConfigurer is the entity that executes the configuration actions of the network elements. It uses the Network devices information, present in the NetworkDB, to know how to communicate with them (COPS, SNMP, CLI) and which commands should be used to perform a particular configuration action in a network element of some device, according to its model and vendor.*
- Databases
  a. *NetworkDB (i) keeps the information that describes the QosBroker network topology; (ii) lists the network elements, its characteristics, its interfaces, and the information needed to configure them.*
  b. *NetStatusDB has a small subset of the NetworkDB information and is stored in memory by NetStatus. Is used to perform faster SAC decisions;*
  c. *UserProfileDB stores all the information describing the services QoS parameters. Keeps also the information of the user's NVUP and information of all services that users had subscribed to its network provider.*

IV. NETWORK MONITORING SYSTEM

Maintaining quality in an IP network, guaranteeing differentiated levels of service according to agreements established with customers is a hard task to perform, when compared with traditional circuit switching networks, and yet in its infancy. Even if several mechanisms have been proposed and are mature enough to be deployed in commercial networks.

The main motivation for the utilization of a measurement system by a Service Provider (SP), is the capability to evaluate the QoS provided to the clients in the long term, as well as interact with management/control systems like QoS Brokers in the short term.

A number of parameters that can be used in the specification of an SLA (an SLA will include more service and networks aspects, like availability, time to solve a problem, etc.) were defined by IETF on its WG IPPM [3]. From those, four are of particular importance, and commonly accepted as being the basic ones for network operation analysis: bandwidth, delay, jitter and error rate.

The IPPM WG is working on the definition of these metrics as well as defining clear rules to accomplish them. With this work, the WG intends to standardise the metrics, to allow the different parties, ISP and customers, to have a clear and uniform vision about what means a given result.

In the long term, statistical parameters can be calculated over these metrics, in order to assess the behaviour of the network, being very important to evaluate if the SLAs were

accomplished or not. This could also be a good tool for network dimensioning by the SP.

But the most interesting capability of a measurement platform is on the short term (real time). Here, the information collected by the platform contribute to the setup of a base of knowledge that will be immediately used by a QoS Broker, in order to trigger the control and management policies, according to the situation of the network on that time and, therefore, take decisions on acceptance, (re)negotiation and termination of communications.

The *ping* tool, e.g., measures the round-trip delay of a communication but most of the times, and particularly when being applicable to asymmetric IP communications, the one-way measurements is more important.

About Metrics

The metrics can be divided in three groups: the singletons, the samples and the statistics.

The singletons represent the result of a single measurement performed. The sample metrics represent an ordered list of singletons obtained using a certain sampling method (e.g. Constant, Uniform, Poisson, etc) to send probing packets. The statistic metrics are the result of performing different statistic calculations over a concrete sample (the most used are maximum, minimum, average, percentile or reverse percentile). The singleton metrics can be divided in two kinds: One-Way (OW) and Round-Trip (RT). The most important metrics as well their names (as specified on the IETF) are the Delay **[5][7]**, **IPDV [4]** (IP Packet Delay Variation time - also known as jitter) and the **Loss [6].**

Derived metrics can be achieved from the previous three sorts of metrics. Examples of derived metrics are loss periodicity, loss distances or connectivity period.

Measurement Models

There are basically two measurement models: active measurement and passive measurement. The **active measurement** assumes that for measurement purposes, additional packets are introduced into the network in order to simulate some sorts of traffic usually present on the network. On the other hand, the **passive measurements** use the existent packets on the network. Each of them has advantages and disadvantages and should be used according to the different situations.

Measurement Points (MPs)

Among the several locations where a measurement point may be placed we outline the Customer networks, the Services networks, the Access networks and the Core networks, because of their special interest.

The measurements can be performed between two points in any of those generic places. For example, in the most suitable case when we want to measure the end-to-end QoS, the MPs should be located on the customer networks. However, this situation is not always possible, especially when we are talking about a small customer. For this reason, the best approach should be use MPs on the access network.

Another important point of measurement is located on the services network. This allows measuring the network parameters when a service is accessed. In the case where the services network has separated networks, different MPs can be identified.

The Service Providers can already use the MPs on the core network for internal purposes to evaluate the behaviour of some portions of the backbone. This way, they can take advantage of this platform for monitoring or dimensioning purposes.

A. Measurement Platform Architecture

The architecture of a measurement system (IPProbe) is illustrated on Fig. 3. There can be identified two main components.

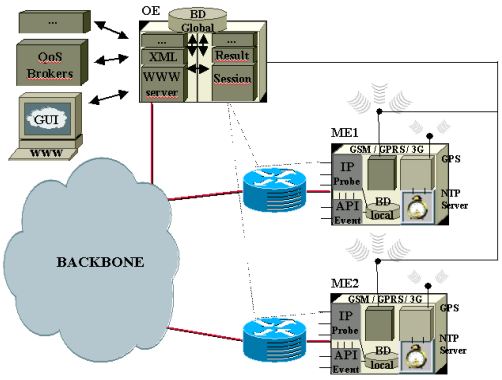

Fig. 3 – Architecture of a measurement platform (IPProbe)

The **Measure Element** (MEs), which objective is accomplish the measurement itself, sending and receiving packets (active measurement) with a given characteristics, and saving the results obtained, in order to be lately collected.

The **Operation Element** (OE) has the main objective of interacting with the measure component and with external parties, either users or other management/control platforms like for example QoS Brokers. It provides the graphical user interface, the interface to external entities and programs the MEs, defining how measurements and results reporting should be done

Each of those components is mapped on different physical equipments. This way, in a complete platform, many small Measurement Equipments will exist, which implement Measurement Component functionalities - MEs in short - and, Operation Equipments, which implement Operation Component functionalities - in short OEs. Typically, there are several MEs being managed by one OE.

As the MEs are the equipments that perform measurement tasks, they will be deployed on the MPs as described below. The OE should be located typically on the farm network of the Service Provider, or in any other place in the case it belongs to a big customer.

Measurement Equipment

The Measurement Equipment (ME) should encompass some hardware and software blocks in order to accomplish several functionalities. Those functionalities can be divided in two blocks: the Control Block, which allows the interaction between the OE, and, the Measurement Block, which perform the measurement tasks itself.

The **Control Block** should communicate with the OE, receiving the information about the tests to be performed, what we call Sessions and, sending the results obtained, either the partial or the complete ones. This communication is made through a special Ethernet interface (the Management interface), or through a backup access (although it could be used usually), using public wireless networks like GSM/GPRS or UMTS. The information received by the OE (Sessions definition) is saved on a local database, as well as the information to be sent to the OE (Results obtained).

The **Measurement Block** should perform the measurements with other MEs, acting as a sender, receiver or both. In this block, an essential component is the GPS receiver so that every MEs could have the same synchronised time. This is fundamental in One-way metrics, specially for the delay, when the tag on the sending time and on the receiver is set by different MEs. The NTP protocol could also be used, but the degree of synchronisation achieved in not the same, because the NTP packets are under the same effects as the traffic we want to measure.

Operation Equipment

The Operational Equipment can be divided on three main components: the OE2ME, which makes the interaction with the ME, the EO2Interf, which provides interaction to the external world, and, the Global DB, which saves all the information related to every MEs it manages.

The **OE2ME** should deal basically with two things, the session configurations on the MEs and the achievement of the results. For the communications between MEs and the OE, a proprietary protocol is used, either for sessions and results.

The **OE2Interf** makes the interface to the user as well as to the external entities. For this reason it should provide beyond the web interface, an easy and very known interface such as XML in order to ease the integration.

Typically, the sessions should be received via a graphical interface (web based), and the session results could be driver either for a web interface or any other external entity.

The **Global DB** is used to save all the information related to all session belonging to all MEs managed by itself. Besides, the database is also used by the OE2Interf and the OE2ME to communicate and exchange values (among other signalling).

Features

These are the main feature that IPProbe platform offers.

- Measurement of all metrics defined on the IETF.
- Provision of a web interface for session's configuration, as well as for results visualisation.
- Complete customisation of the packets to be measured (protocols, addresses, sizes, etc.).
- Configuration of the sampling methods, time intervals between probing packets, starting and ending times, duration, etc.
- User authentication.
- Wireless access for accessing the results, behind the common control interface.
- Access to the results by other entities, taking advantage of the XML API provided.
- Path discovery functionality (like traceroute).
- NTP server.
- Interface with QoS controlling unit (QoS Broker)

V. CONCLUSIONS

The described infrastructure is able to demonstrate the operation of an IP-based multi-service provider network. This demonstrator is able to test the behaviour of different QoS approaches and policies to support multiple service mixes on the network. The demonstrator is also able to support integrated and differentiated services approaches simultaneously and with different operator policies.

The demonstrator is quite effective in providing services from an operator perspective. Services can be defined, allocated to specific users, and their impact on network behaviour and management can be easily assessed. The monitoring facilities are able to provide specific performance results, which are integrated in the network management. These same facilities can be used to provide assessment results on specific management policies concerning configuration of network devices for specific QoS metrics. The monitoring system is able to provide a real time network status view, which is essential for an optimal network management and control.

VI. REFERENCES

[1] Moby Dick: http://www.ist-mobydick.org

[2] Victor Marques, et al, "A Simple QoS service provision framework for beyond 3rd generation scenarios", 10th International Conference on Telecommunications ICT'2003, Papeete, French Polynesia, February, 2003

[3] IP Performing Metric (IPPM) Worrking Group, http://www.ietf.org/html.charters/ippm-charter.html

[4] C. Demichelis, P. Chimento, Draft: draft-ietf-ippm-ipdv-07 - IP Packet Delay Variation Metric for IPPM (work in progress), http://www.ietf.org/internet-drafts/draft-ietf-ippm-ipdv-07.txt, February 2001.

[5] G. Almes S. Kalidindi, M. Zekauskas, Request for Comments: 2679 - A One-way Delay Metric for IPPM, http://www.ietf.org/rfc/rfc2679.txt, September 1999

[6] G. Almes S. Kalidindi, M. Zekauskas, Request for Comments: 2680 - A One-way Packet Loss Metric for IPPM, http://www.ietf.org/rfc/rfc2680.txt, September 1999.

[7] G. Almes S. Kalidindi, M. Zekauskas, Request for Comments: 2679 - A Round-trip Delay Metric for IPPM, http://www.ietf.org/rfc/rfc2681.txt, September 1999