

# QoS Management in Fixed Broadband Residential Gateways

C. Guerrero, J. Garcia, F. Valera, A. Azcorra

University Carlos III of Madrid  
Telematic Engineering Department  
Avda. de la Universidad, 30  
E-28911 Leganés  
Spain

E-mail: {guerrero,jgr,fvalera,azcorra}@it.uc3m.es

**Abstract:** QoS management is nowadays a mandatory feature in current broadband residential gateways developments. The interconnection between different QoS domains has to be treated into different steps in order to provide a reliable end-to-end QoS solution. The scenario analyzed in this paper is the mapping between QoS requirements in residential users connected to a broadband access network across a multiservice broadband access gateway. Different approaches to provide QoS in the access network are discussed as well as their impact in the design of a residential gateway. An architecture of a gateway based on IMS (IP Multimedia Subsystem) as SIP-based signaling domain for multimedia services is presented with the corresponding adaptation to a broadband fixed access scenario according to Next Generation Networks (NGN) standardization. Finally, the implementation of a prototype of the QoS-enabled gateway, based on the Click! modular router [1], is described to demonstrate end-to-end QoS provisioning for multimedia services. This prototype allows us the demonstration of (1) an innovative way of extending gateway device functionalities using Click! and (2) the feasibility of residential gateway architecture proposed. The work presented in this paper has been developed within the framework of the 6<sup>th</sup> Framework Programme IST MUSE [2] project.

**Keywords:** NGN, SIP, IMS, fixed broadband, access network, residential gateway.

## 1 Introduction

Next Generation Networks (NGN) are considered multiservice networks based on packet switching technology, and basically using the IP protocol as the only available technology to provide end to end connectivity. During these last years the first elements for NGN networks have emerged: the 3GPP Release 6 has defined with the IP Multimedia Subsystem IMS [3] a first instantiation of the NGN architecture in the mobile field, which will be detailed in next releases. Another important example of a

NGN architecture may be found in the scenario of a multiservice broadband fixed access network where the IST MUSE European project [2] is focused on.

The 3GPP IMS is rapidly becoming the de facto standard for real-time multimedia communications services. Although the IMS was originally specified for 3G generation mobile networks, it also provides excellent service deployment architecture for any fixed or wireless network, and all IP-based networks such as WiFi, corporate enterprise networks, residential LANs or the public Internet. IMS standard define open interfaces for session management, access control, mobility management, service control and billing. The use of SIP [4] as the main signaling protocol in IMS allows independent software developers to leverage a broad range of third party applications servers, media servers and SIP-enabled end user devices to create next generation services. There are many experiences of implementing IMS in mobile environments, but this work considers a contribution in using IMS principles and architecture in a fixed broadband access scenario, where some considerations and redesigns have to be performed in order to adapt the IMS model to this new scenario. Recently, ETSI-TISPAN [5] standardization work is focused on migrating IMS to a fixed access network scenario in the context of an overall NGN architecture. MUSE project is studying the possibility of using IMS adaptation from TISPAN standardization as an alternative for QoS provisioning model in the access network.

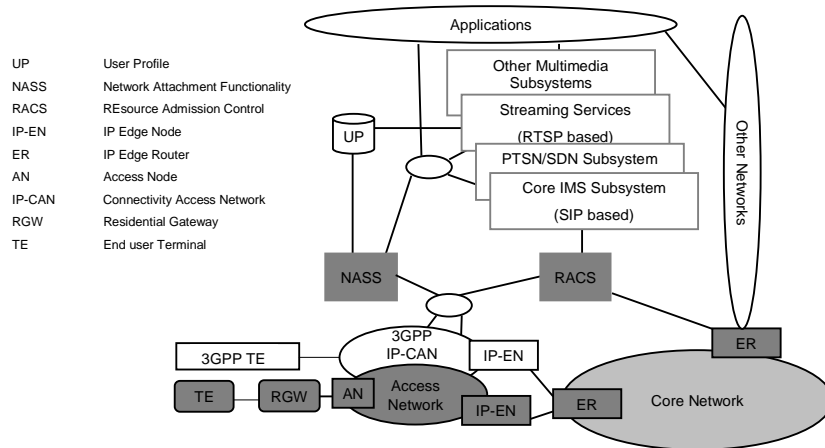
It is important not to forget the QoS viewpoint of the end-users. Their assumed business role is to pay for a certain service or application subscription (i.e. VoIP, video on demand or Internet browsing) instead of paying for a class of service (always on with guaranteed bandwidth rate). The way of charging them is expecting to be based on services usage instead of network usage and in order to allow this, resource availability has to be checked for every service application request, and not only during the subscription. Several models, considering key aspects like the entity that requests QoS needs or how the resources are provided and charged, are considered and discussed in section 2 that describes the standardization work in fixed broadband access networks in the framework of NGN.

These assumptions let us present in section 3 an innovative scenario with the deployment of an IMS infrastructure in a fixed broadband access network by mainly focusing on the residential gateway (RGW) that interconnects the residential end-users to the corresponding service providers. Finally, a prototype of this RGW lets us demonstrate the feasibility of supporting some QoS scenarios using an innovative Click! modular router platform approach. The conclusions of this experience of prototyping a RGW architecture based on a still open standardization process are presented in conjunction with future work in these topics.

## **2 Standardization in fixed broadband access networks**

The main standardization body that is contributing to the definition and dissemination of today and future telecommunications networks for broadband fixed and mobile access is the ETSI-TISPAN in the framework of NGN, trying to facilitate the convergence of network and services supporting both users and services nomadism and mobility. NGN enable different business models across access, core network and

service domain. SIP will be the call and control session protocol and 3GPP Release 6 IMS will be the base for NGN IMS. It enables any IP access to Core IMS and other subsystems from different domains (mobile, home and corporate). Internetworking towards circuit switched voice traditional services is considered too. Service providers use NGN architecture to offer real-time and non real-time communications services between peer-to-peer or client-server configurations.



**Figure 1. TISPAN-NGN architecture in fixed-mobile broadband access**

Figure 1 shows an overview of TISPAN-NGN architecture in Release 1 [5] where new key network elements for the fixed scenarios are included: the Access Node (AN) that interconnects the customer premises network to the access network; the IP Edge Node (IP-EN) that terminates L2 connections; the Resource and Admission Control Subsystem (RACS) provides to applications a mechanism to request and reserve resources from the access network and the Network Attachment Subsystem (NASS) that provides authentication and autoconfiguration services.

To ensure QoS aware NGN service delivery, two architectures for dynamic QoS control are considered in the standardization process. The first one is the *guaranteed QoS* model, where the services are delivered with previously reserved resources. The RACS performs admission control in the access network through throughput control and traffic policing. The other model is *relative QoS*, which implies traffic class differentiation (DiffServ) by means of separate queues dedicated to particular IP traffic classes and by performing priority scheduling between these queues in the IP-EN and the access network.

Support of other models like best effort networks or statically provisioned networks are not considered by RACS. The architecture supports both QoS control architectures – *guaranteed* and *relative* – allowing the access provider to select the most suitable QoS architecture for its needs. When QoS differentiation is used (*relative QoS*), DiffServ marking/remarking shall be performed at the IP-EN. DiffServ marking may be performed also by the RGW for uplink traffic, considering that the network operator controls the RGW in the customer premises network. For

*guaranteed QoS control*, enforcement of QoS admission control decisions (throughput control and traffic policing) shall be performed in the IP-EN and the RGW. At this point it is important to remark that TISPAN-NGN standards do not consider the requirement to provision QoS control in the RGW and it is considered for further study [6], [7] and [8]. Another interesting issue considered in the standards is the resource reservation mechanisms. Two models are defined: the *proxied QoS reservation request with policy push* and the *CPE-request QoS reservation with policy pull*. The main difference between these two models is whether the end user terminal equipment (TE) (or the RGW on behalf of it) is capable or incapable of sending explicit QoS requests. In the *proxied QoS reservation request*, the TE does not support native QoS signalling mechanisms. When the end user invokes a specific service using a SIP based signalling, the RACS is the responsible for QoS authorization (policy control) and resource reservation. The TE, in the *CPE-request QoS reservation* model, is capable of sending QoS requests over dedicated signalling in the user plane. The RACS sends an authorization token to the TE through the signaling channel.

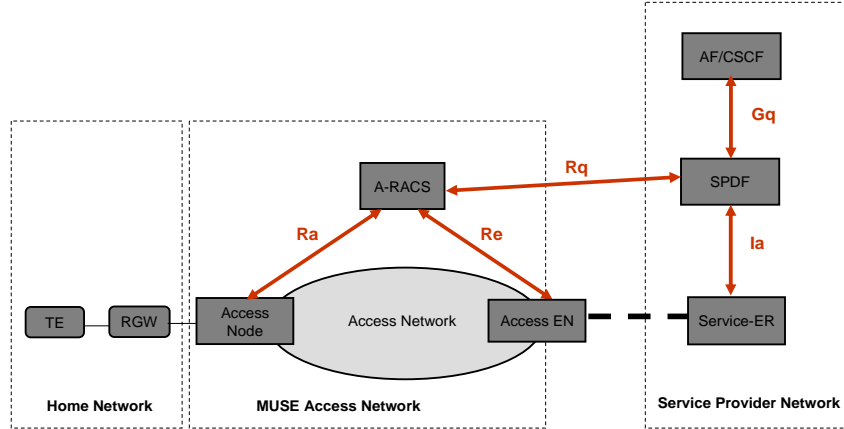
As a conclusion, it is important to emphasise that the current standard is still not mature enough in the QoS control from the end user side in the scenario of a fixed broadband access scenario and the design and implementation work of a RGW with QoS support is an innovative approach that demonstrates the possibility of implementing the architectures that are being currently standardized in a real scenario.

### 3 Residential gateway architecture

The QoS support in the RGW described here is being developed in the framework of MUSE project [2]. Next subsection describes the broadband access scenario considered in MUSE emphasizing some concepts like network entities, business roles and QoS models. A detailed description of the RGW QoS support in a set of scenarios considering some of the described QoS models is presented afterwards.

#### 3.2 MUSE broadband access scenario

MUSE project is designing the architecture of a multiservice broadband access network and it is studying the feasibility of TISPAN-NGN standardization for QoS solutions. Figure 2 represents the broadband access scenario in MUSE where we can identify the main roles involved in the architecture. There are three identified network segments: the *Home Network* where QoS provisioning is out of scope of both MUSE and TISPAN standardization. This segment is critical in end to end scenario and the RGW is the network element in which we are going to focus our work. The Access-RACS (A-RACS) in the *Access Network* segment plays an important role in QoS provisioning. It is remarkable that the interfaces (“Ra” and “Re”) between A-RACS and AN and EN are still in standardization process (by TISPAN).



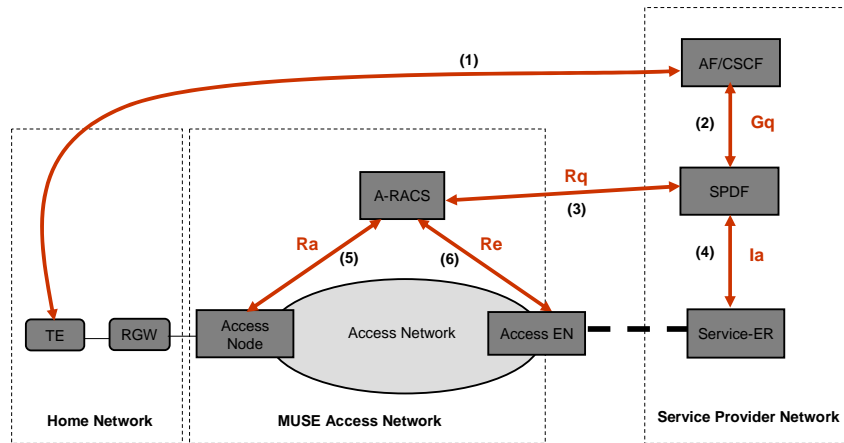
**Figure 2. MUSE broadband access scenario**

Other interfaces relevant from QoS provisioning viewpoint in the access network are “Rq”, “Gq” and “Ia” that interconnects the *Service Provider Network* segment with the *Access Network* segment. The AF/CSCF (Application Function/Call and Session Control Function) is the element that offers the required resources to the applications. SPDF (Service Policing Decision Function) perform policy decisions based on policy set-up information obtained from the AF/CSCF via the “Gq” interface and authorizes QoS resources to the AF and edge routers (Service-ER and Access-EN) via “Rq” and “Ia” interfaces. Considering that the access network QoS model is TISPAN-NGN compliant according to Figure 2, and that the standardization work is not focused on the RGW QoS support, we are going to define several scenarios in which QoS facilities are deployed in the RGW. Extensions to other possible QoS models that are not covered in TISPAN-NGN R1 standardization are being studying in MUSE project. Specifically, a *service oriented model* is considered as relevant in a fixed broadband scenario like MUSE one. In this model, the service provider is the responsible of requesting the resources to the access network provider, and in consequence, the responsible of configuring the RGW, or TE, without any specific QoS signaling flow at application level. Another QoS model in MUSE, that it is compliant with TISPAN-NGN models, is the *application signaling based model* where the end user is capable of sending standard SIP-based signaling to the service provider in order to request services using two different procedures depending on whether SIP requests support QoS information via SDP extensions or not [9].

### 3.2 QoS management in the residential gateway

This section describes the RGW QoS support in the *application signaling based* model. The facilities provided by SIP-based signaling in order to extend services requests with QoS information and the massive use of this signaling protocol in TEs lets us consider this QoS model for the RGW architecture to be prototyped and tested. Two scenarios have been identified:

**Scenario 1. Signaling Relay Scenario (SRS)** where the RGW detects and transparently relays the end user SIP-based signaling to the corresponding IMS node (AF/CSCF) in the service provider network. In that case, the RGW transparently treats SIP flows as any other end user data flows, mapping them to the corresponding preconfigured QoS class in the RGW autoconfiguration phase. Figure 3 represents this scenario where (1) the TE sends a SIP-based signaling to the AF/CSCF located in the service provider and starts the end-to-end session characteristics negotiation. The RGW detects that traffic and transparently sends it to the access network. (2) AF/CSCF by the “Gq” interface indicates the resources associated to the service requested in previous step to the SPDF/RACS.(3) and (4) the SPDF/RACS checks if the QoS characteristics negotiated by the end users can be delivered by the respective access network. This checking is based on several sources of information: view of all network resources (preconfigured and already in use) and the user profile and subscription information. If the access network cannot deliver the requested QoS, the involved SPDF can modify the QoS characteristics. After successful end-to-end negotiation, the SPDF/RACS authorizes the use of the resources for the session, configuring the QoS resources to the corresponding nodes in the access network completing the steps (4), (5) and (6) in the Figure 3. Finally, the IP media flow is established considering the provisioned resources in all the network elements at each segment involved in the end to end communication.



**Figure 3. Signaling Relay Scenario (SRS)**

**Scenario 2. Signaling Proxy Scenario (SPS)** where the RGW supports SIP-based signaling on behalf of the TEs at home, by generating the signaling associated to upstream and downstream traffic and acting as a signaling proxy, requesting the corresponding QoS. The RGW is capable of identifying end user flows that do not correspond to any SIP-based terminal. In order to authorize this flow and to provide the resources in the access network, the RGW acts as a signaling proxy sending a SIP based signaling to the service provider. To harmonize the end-to-end QoS architecture, it could be useful to support a non-IMS terminal at home or even a SIP

terminal but not full 3GPP/IMS compliant or without the QoS extensions in SDP needed for the QoS model negotiation. In all these cases, a signaling proxy has to be performed by the RGW but with differences depending on the TE SIP capabilities. Figure 4 shows this scenario where a legacy terminal sends a session request (1a). The RGW detects that this particular request (without signaling or with SIP-based signaling without QoS extension) has to be proxied and acts on behalf of the TE sending the corresponding SIP-based signaling to the AF/CSCF (1b). Next steps are similar than in the previous scenario displayed in Figure 3. Both scenarios have a strong performance overhead in the treatment of signaling messages in the RGW and it is important to measure the performance of a RGW prototype in the case of SIP-based signaling in comparison with user data messages. This prototype and the results are described in next section.

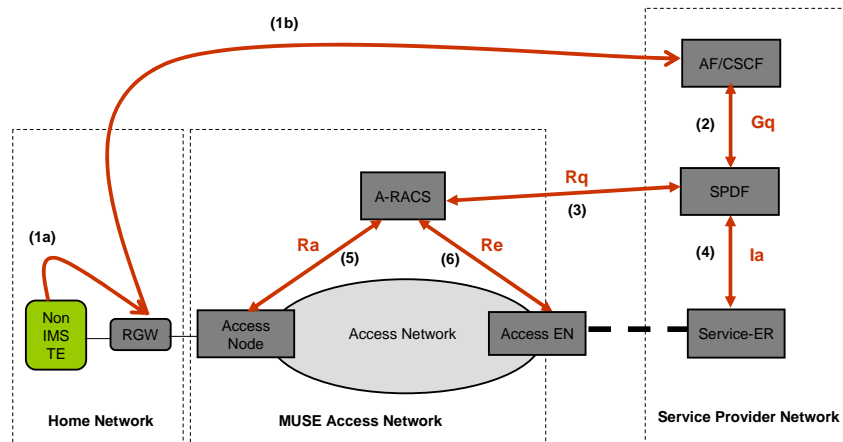


Figure 4. Signaling Proxy Scenario (SPS)

## 4 The prototype

The RGW scenarios previously described are going to be developed using the Click! modular router platform that let us demonstrate the capability for extending functionalities to the RGW node. The prototype described in this section is a starting point for analyzing a flexible platform so as to study and propose RGW architectures, mainly focusing on the QoS support within the framework of a fixed broadband access in the MUSE project.

Click! [1] is a modular software router developed by MIT, the ICSI and UCLA. A Click! router is an interconnected collection of modules called *elements* in the Click! terminology and an element is a C++ implementation of a specific functionality. There are elements to communicate devices, to modificate packets, to program dropping policies and packet scheduling. It is quite simple to construct a router with Click! since the only thing to do is to specify the elements connections using a specific Click! script language. Click! can be executed in two different modes: *User-level* and *Linux module*. In the *User-level*, the Click! is another application with their

restrictions (see Figure 5). A patch must be applied to the kernel sources in order to use the kernel mode. When the Click! is installed as a *Linux module*, the kernel chain is changed, so all incoming packets will enter to the Click! router first. The user can then fully control packets transmission from/to the kernel. Changing the routing table, and creating a virtual device, all packets can be force to go through Click!. Outgoing packets (packets from applications to the network) do not necessarily have to pass through the Click! router in case it is not needed (see Figure 6).

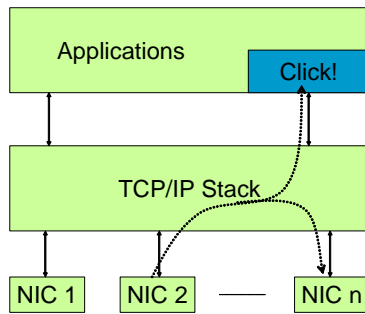


Figure 5. Click at the user level

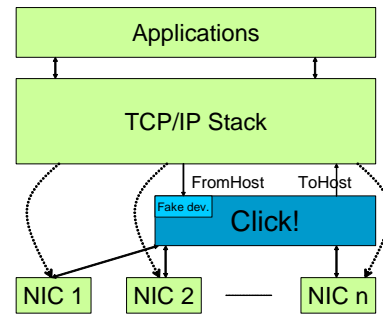


Figure 6. Click as a Linux module

Important Click! functionalities to be considered in order to select this platform are IPv4/IPv6 datagram processing, extensibility, maintenance, performance and network connections availability. Click! uses Linux drivers, so there are no practical problems with hardware compatibility since Linux has a big hardware compatibility list with network cards. As a negative point to be remarked, Click! has no elements to work with MPLS, VLANs, VPLS, etc., so, it does not accomplish these functionalities nowadays and there seems to be working groups around it. Based on a previous study (comparing the use of iptables [10], netlink, libpcap [11] and Click!), we will use Click! to implement the RGW prototype. Click! is not a complete software so far, due to its lacks in IPv6 functionalities, the impossibility to work with the 2.6 new Linux kernel and its deficiencies to directly process layer 2 frames other than Ethernet frames but our idea is to create a first stage prototype using the Click! elements available now, and enhance it with new functionalities in a second stage.

#### 4.1 The hybrid model

For the RGW prototype, software is required capable of capturing all packets at layer 2 level, modifying them, re-injecting them into the network, sending them to upper layers, etc., so it was decided to use the Click! modular router software (more precisely the 'Click' module). Although we chose Click!, it may not be mandatory or desirable to develop new applications at Linux kernel level because programming new applications at the kernel level is sometimes very difficult and the creation of new hardware and software network applications is also desirable, and they should be as independent as possible from low level packet facilities. To overcome these



problems, it was decided to create a new *hybrid model* where neither pure Click! nor pure application level programs will be developed but a combination of these two ones. Figure 7 depicts this *hybrid model* with three main boxes: **Click!** is the Click! software router working at kernel level. It will receive every packet, wrap them inside a new UDP packet and forward them up to the Manager or Process application (this will be configured by the Manager). **The Manager** will receive fresh packets from the Click! module and process them. Depending on the packet characteristics, the Manager could configure the Click! module to forward the same kind of packets to a certain process. **Processes P1..Pn** are the user level applications developed to perform certain functions.

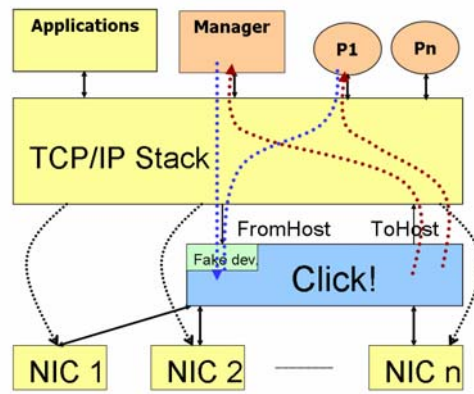


Figure 7. Hybrid model

This model must be tested to assure that it can possibly be used and that it is not suffering any serious performance problem. When an application is programmed at the Click! module level the time a frame spends crossing the Linux kernel is saved. This is why the delay imposed by the Manager must be estimated to validate the hybrid model. The main intention of the hybrid model is to help the programmer to develop RGW applications in an easy and fast way, but there are some issues that must be validated. In these tests, we try to measure the additional delay introduced by the hybrid model due to the transmission of the packet from the Click! layer to the application layer and back.

#### 4.2 Delay introduced by the Manager Application

This scenario tests whether the use of a user-level application called Manager does slow down frame management or not (should it really reduce the performance it could always be possible to manage the frames inside the Click! module, without passing them to the user level although the flexibility of the development at the application layer would be lost). For this test Click! has been installed in a computer with two different configurations. *Direct connection* where frames are encapsulated in an UDP packet by Click!, and then they are sent again directly to the same interface where they came from.

The other one is *Manager connection* where frames are also encapsulated in an UDP packet, but now they are sent to the Manager. This process can be carried out by a fake interface called *fake0* (for example). When the Manager receives a frame, it returns it to the source machine through Click!. In both cases, packets had the same size and were sent by the same source machine. In order to perform the test, a large number of streams of 1000 packets have been sent, with different sizes in each experiment. Information was collected by the source machine with a sniffer application (*Ethereal*). Table 1 shows the results obtained in these tests.

| Packet size | Direct Connection | Manager Connection | Gap          |
|-------------|-------------------|--------------------|--------------|
| 100 bytes   | 120 $\mu$ s       | 250 $\mu$ s        | +130 $\mu$ s |
| 540 bytes   | 200 $\mu$ s       | 330 $\mu$ s        | +130 $\mu$ s |
| 1060 bytes  | 290 $\mu$ s       | 430 $\mu$ s        | +140 $\mu$ s |
| 1440 bytes  | 365 $\mu$ s       | 500 $\mu$ s        | +135 $\mu$ s |

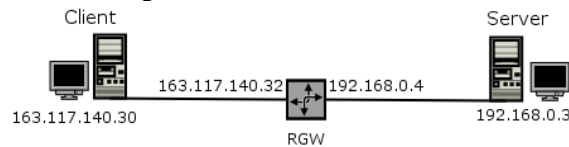
**Table 1. Delay introduced due to the Hybrid Model**

Taking into account this result it can be concluded that the usage of the Manager increases the time around 130-140  $\mu$ s, (this is a packet size independent result). Nevertheless, the Manager will not always directly resend packets, because sometimes it has to send packets to another user-level applications or Click! modules (through a fake interface for example). Then, time used for managing frames could be similar in both cases (Click! handling or Manager handling).

### 4.3 Supported load in the Hybrid Model

The aim of this test is to probe the RGW load capacity when the hybrid model is used. Two PCs were connected to the *compact* device working as a RGW as showed in Figure 8. The RGW implements the NAT (Network Address and Protocol Translation) in three different ways:

- Linux using the iptables functionality. The Linux kernel must be configured to support the iptables module. To set the NAT table the command `iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE` must be invoked. It is also necessary to set the `ip_forwarding` behaviour.
- NAT functionality implemented at the Click! level. It is easy to create the NAT functionality using Click! elements.
- NAT functionality implemented at the application level (Manager). The Click! level wraps the received frame in an UDP packet and sends it to the TCP/IP stack using the ToHost Click element. The frame is received by the Manager at the application level which performs the NAT functionality sending the modified frame to the Click! level again.



**Figure 8. Load testing scenario**

Results are shown in Figure 9 where the iptables values are omitted due to their similarity with Click ones. The most important comments extracted from these results are that there exists a maximum packet generation rate depending on the size of the packet. The client can not generate the nominal interface rate and this value decreases when packet size also decreases. If the Iperf program is launched using 1470 bytes as the packet size, the maximum rate is 95,1 Mbps. For packet lengths of 850 bytes, just 92,7 Mbps can be generated and this value is reduced to 75,1 Mbps for 200 bytes packets. The iptables and Click scenarios show similar results. In the Manager scenario, the results are similar to the other ones while 40 Mbps rate is not reached and packets size is above 850 bytes. For higher inputs or lower packet sizes, the performance decreases drastically. It is important to note that in the hybrid model (Manager scenario) just some packets will go up to the application level. Just signaling and fresh (not configured flows) packets must be processed by the Manager and we expect a lower rate than 40 Mbps for this kind of packets. Another important point is the packet size. Normally, data packets are bigger than 200 bytes, so we just must take care about the signaling packets. For example, for SIP messages the worst case is for INVITE (mean of 465 bytes) and OK messages (mean of 388 bytes). More complex SIP messages where the header is extended with SDP QoS information, the size of these messages are between 838 and 1024 bytes.

Another interesting test is to change the RGW device by a more powerful equipment than the compact one. The new device has a Pentium 4 2,4 GHz processor and 512 Mbytes of RAM (the same than the compact one). Figure 10 depicts the results obtained in this new test where, as the previous one, the iptables results are omitted due to its similarity with the Click experiments. The results are notably better than the obtained with the compact device and it proves the importance of a powerful processor. In another model, where all frames were treated at the application level, a high performance device is firmly recommended. Nevertheless, in the Hybrid Model that we proposed, where just signaling and fresh frames are treated at the application level, the compact device is capable enough so as to process the estimated traffic.

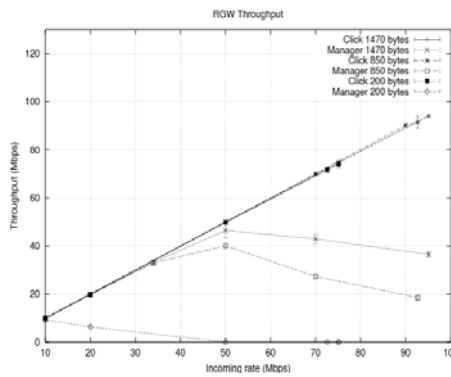


Figure 9. RGW throughput

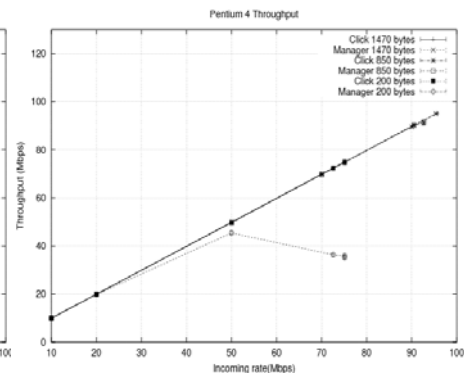


Figure 10. Pentium 4 throughput

## 5 Conclusions

In this paper it is showed an innovative software architecture of a RGW node with support of QoS functionalities in a flexible and extensible way by using the Click! platform. Click! lets us to process traffic at different levels as it is described in several models and we are interested in treating the QoS signaling traffic (SIP-based) separately from data plane traffic. The differentiation of SIP traffic managed in Click! allows us to support dynamically QoS facilities in the RGW. Several QoS signaling scenarios have been proposed following the drafted TISPAN-NGN standardization in Release 1. This work contributes to offer an open testbed for probing their feasibility in an european broadband access scenario where the RGW is considered a critic node when end-to-end QoS is provided.

Future research will follow not only the standardization process but also the technological trends in segments like home and access networks where the RGW plays an important role in the mapping of QoS architectures.

## Acknowledgements

This article has been partially granted by the European Commission through the IST-507295 Multi-Service Access Everywhere (MUSE) project.

## REFERENCES

- [1] The Click Modular Router Project. [www.pdos.csail.mit.edu/click](http://www.pdos.csail.mit.edu/click).
- [2] MUSE (Multi-Service Access Everywhere). IST European Project. [www.ist-muse.org](http://www.ist-muse.org).
- [3] 3GPP TS 23.228: IP Multimedia Subsystem, Stage 2, Release 6, January 2003.
- [4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Shooler. IETF RFC 3261. SIP: Session Initiation Protocol. June 2002
- [5] DES/TISPAN-02007-NGN-R1 NGN Functional architecture for NGN Release 1
- [6] DES/TISPAN-02021-NGN-R1. NGN Functional Architecture; Network Attachment. Draft 2005.
- [7] DES/TISPAN-02020-NGN-R1. Functional Architecture; Resource and Admission Control Subsystem (RACS). Stable Draft 2005.
- [8] DTS/TISPAN-05002-NGN Interface Protocol Req. Definition; QoS Control in Access Networks. Draft 2005.
- [9] M. Handley and V. Jacobson. IETF RFC 2327. SDP: Session Description Protocol. April 1998.
- [10] The netfilter/iptables project. <http://www.netfilter.org/>
- [11] Lawrence Berkeley National Labs, libpcap, Network Research Group, URL: <http://www.tcpdump.org/>