Authors' contact information:

Marco Platania[1]
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274004
E-mail: platania@dis.uniroma1.it

Roberto Beraldi
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274018
E-mail: beraldi@dis.uniroma1.it

Giorgia Lodi
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274057
E-mail: lodi@dis.uniroma1.it

Leonardo Querzoni
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274057
E-mail: querzoni@dis.uniroma1.it

Roberto Baldoni
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274014
E-mail: baldoni@dis.uniroma1.it

[1] Corresponding author

Noname manuscript No.
(will be inserted by the editor)

Supporting NGNs Core Software Services: a Hybrid Architecture and its Performance Analysis

Marco Platania · Roberto Beraldi · Giorgia Lodi · Leonardo Querzoni · Roberto Baldoni

Received: date / Accepted: date

Abstract In the last few years telco providers are striving to migrate their services from the traditional Public Switch Telephone Network (PSTN) to so called Next Generation Networks (NGNs) based on standard IP connectivity. This switch is expected to produce a cost degression of 50% for CAPital EXpenditure (CAPEX), while OPerating EXpences (OPEX) remains fairly stable due to network management and energy costs. At the same time, the instantiation of new telco services (Voice over IP (VoIP), video conferencing, etc.) and the support of third party applications (such as support to smartphone applications, etc.) are expected to produce a big increase of the load of a telco provider at the core level. The goal of this work is to show how management and energy costs can be effectively reduced by leveraging autonomic approaches to move some NGN services toward the telco network edge while still providing Quality of Service (QoS) levels comparable with those provided by a traditional fully-managed infrastructure. This is done by taking into consideration the increase of the load of such services that is expected to raise by one order of magnitude in the near future. Specifically, we propose a hybrid architecture letting telco administrators reduce the number of servers in the provider managed network by exploiting home devices in the computation and by organizing them in a self-configuring Peer to Peer (P2P) system; in this way it is possible to reduce the overall system and operational costs. Our claims are supported by an experimental study based on both simulations and theoretical models that analyze the trade-off between the number of servers and home devices in order to guarantee a service within QoS constraints. Experiments are carried out on a realistic model that abstracts the lookup procedures within

Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy
Tel.: +39-06-77274004
Fax: +39-06-77274002
E-mail: {*platania* | *beraldi* | *lodi* | *querzoni* | *baldoni*}@dis.uniroma1.it

the NGN of a big telco provider (i.e., finding the IP address of a given unique user profile).

Keywords Autonomic systems · Peer to Peer · Next Generation Network · VoIP

1 Introduction

In the last few years, worldwide telco operators have been striving to develop solutions over Next Generation Networks (NGNs), that are packet-based networks that use multiple broadband, Quality of Service (QoS)-enabled technologies in which service-related functions are independent of the underlying transport-related technologies [1].

With respect to the traditional Public Switc Telephone Network (PSTN), NGNs exhibit a telephone traffic cost reduction and provide a higher number of services such as web call centers, e-commerce, home banking, video conferences, as well as supporting third party applications such as smartphone applications, social network applications, etc. In the near future, these applications are expected to increase by one order of magnitude the resources requested to the telco core network. Despite this raising load, communication protocols currently used in NGNs, such as H.323 and SIP (Session Initiation Protocol) [2], rely on a collection of central servers that manage all client requests in case, for example, a user's contact information has to be retrieved and a connection with that contact has to be established. These centralized architectures drive up the telco operational costs by adding servers to cope with the increased load as well as intruducing scalability issues in the long run.

Looking closely to the network infrastructure of a large telco provider used for deploying such communication protocols, (e.g. Telecom Italia, AT&T, France Telecom, etc.), this is usually organized as a static tree with servers installed in different geographic domains (leaves of the tree). These servers gather regional traffic and forward it to clusters of other servers for users contacts retrieval purposes. Due to the employment of dedicated system administrators in each domain [3], the majority of the system costs are represented by maintenance and configuration. In addition, when the system load increases, new servers need to be added to avoid the creation of bottlenecks. Several load balancing systems have been proposed to smartly split the load including servers arranged along a Distributed Hash Table (DHT) [3].

In this paper we describe the outcome of a project, funded by *Telecom Italia* [4] that focuses on the design and the simulation of a load balancing architecture providing telco core services supporting up to 10 million customers. On one hand, this architecture aims to reduce operational costs, while on the other hand, it has to be able to cope with a load of the core services of one order of magnitude higher than the one currently handled by telco providers. The architecture specifically analyzes the contact management service, that is, considering a unique client IDentification number (ID), the service returns

the *current* IP address of the users. Such a service can actually be considered as a building block for applications deployed over a telco provider network (Voice over IP (VoIP), video conferencing, Instant Messaging) and for a myriad of third party applications developed for smartphones or social networks. We consider a telco network as composed by a core network managed by the provider and an unmanaged edge. The former is populated by powerful servers, while the latter is actually a DHT formed by devices installed at the user's premises, namely Home Gateways. Thus, the load of contacts requests is balanced between managed and unmanaged networks in order to respect the QoS of the core service and to reduce management costs. We study how the number of servers can be reduced by including opportunistical home gateways in the computation[1]. The study starts with a test of performance of specific servers and home gateways, then such real data are used as an input to a simulation of the contacts lookup service over a 10 million customer network. At the same time we build an analytic model of the architecture to validate the simulation results.

Our study assesses that the hybrid architecture can reduce the number of servers of one order of magnitude passing from a managed network of thirty servers (actual size of the inter-regional core infrastructure for a mid-large size telco operator) to a few units when combined with an edge network of four million home gateways. From the point of view of the telco provider, this translates into a substantial reduction of (i) the OPerating EXpenses (OPEX) for running the system (maintenance, management, and energy costs), and (ii) the CAPital EXpenditure (CAPEX) for the smaller number of servers in the core network. In this paper we do not discuss energy savings in the core network routers because it is well known by providers that the actual energy saving is on the server farm. The increase of power consumption between an idle core router and a 100% traffic loaded router is 1% [5]. Considering that the cost of energy is on average 20% of the cost of the installed devices [5], the additional cost due to the 100% traffic load on a router is 0.2%, while the cost of energy for a server is roughly $50 for $100 of deployed equipment (so 50% compared to 0.2%).

The rest of this paper is structured as follows. Section 2 describes a background in the area of VoIP and NGNs and a number of works similar to our proposed solution. Section 3 presents the system model, while Section 4 illustrates the solution we propose. Section 5 describes the theoretical model that allows us to model the telco system infrastructure. Section 6 presents an experimental evaluation we have conducted in real settings and using the designed theoretical model. Finally, section 7 concludes this paper highlighting some future works.

[1] A user can switch the device on/off while the telco operator can manage it remotely if the device is switched on.

2 Background

The evolution of voice services offered by telco operators has been driven by the disruptive technological changes that happened in their core networks. The first of these changes happened with the migration of the PSTN from a strongly hierarchical structure based on the widespread usage of analog exchanges, to a more modern digital network based on the usage of virtual channels. The next evolutionary step is happening today: telco networks are slowly moving their voice services from the existing circuit switching PSTNs to IP-based packet switching networks that promise services with adequate quality at lower costs. These new networks are usually referred to as Next Generation Networks.

An NGN decouples QoS-enabled transport technologies from service-related functions that are independent of the underlying transport-related technologies. A key characteristic of an NGN is its supports to generalized mobility that will allow consistent and ubiquitous provision of services to users. To this end, a variety of identification schemes that can be resolved to IP addresses for the purposes of routing in IP networks has to be provided [1]. A fundamental identification scheme is mapping a user profile to its current IP address.

For example, in SIP [2], when a user comes online, it registers to a server called Registrar, and informs that it is able to accept incoming calls. Proxy servers are used for user lookup procedures: if the requested user is outside of a proxy's domain, the request is forwarded to another proxy closer to the target. Redirect Servers are used to manage possible user handovers, providing an alternative route if the target has moved, while a Location Server is used as a database containing a list of bindings between SIP-URIs and user locations.

Throughout this paper, we refer to a generic *location service* that maps the user profile to an IP address as detailed later in the next section. In general, implementing the service on a single server machine exhibits a number of limitations:

- **Scalability**: when the number of connected users increases, servers may become overloaded; this, in turn, has a serious impact on the time needed for a lookup procedure.
- **Maintenance**: each maintenance operation needs to be performed both on the client and the server side.
- **Availability**: if the network infrastructure is damaged, for example due to battlefields or natural disasters, the Internet Service Provider (ISP) could be unable to provide access to users.

Scalability and self-* properties of peer to peer (P2P) technology were attractive for several solutions that have been proposed to replace the standard SIP lookup procedure by an existing P2P protocol [6]; typically, the lookup procedure is performed over a DHT. For example, Johnston in [7] presents a SIP-using-P2P algorithm in which users' location information is directly stored in the DHT, without using traditional SIP Registrar and Location servers. In [3] the authors distinguish between super-nodes, i.e., nodes with high capacity (bandwidth, CPU, memory) and availability (uptime, public IP address), and

ordinary nodes. Super nodes are organized in a DHT, i.e.: Chord [8], while ordinary nodes maintain a connection with a single super node. All users' information is stored on the DHT; thus, all lookup procedures are performed on it. SIP is used as an underlying protocol for locating users, joining the DHT, registering users, call setup and instant messaging.

SOSIMPLE [9] combines the traditional SIP and SIMPLE, a set of SIP extensions for Instant Messaging. All nodes in SOSIMPLE have the same responsibility (i.e., no super nodes) and are organized in a DHT based on Chord.

In [10], the authors present an interworking of two different systems: IMS, i.e. the new adopted and highly centralized architecture in NGN, and P2PSIP. Differently from our work, in which we propose an integrated solution of a centralized and a distributed network, [10] proposes an interconnection through proxy servers between two separate networks. In addition, it describes just the architecture, while our work comes along with an experimental and theoretical evaluation.

Other works in literature [11] [12] use a two layer architecture with super peers and resource constrained devices as in our case. While we use a DHT connecting small devices, [11] uses the DHT for connecting super peers that maintain point-to-point connections with home devices. This architecture is basically used for content exchange, media streaming, reliable storage and device management. The authors claim the possibility of using small devices to reduce the load in the super peer's network; but differently from our work, they do not evaluate the benefit of having an edge network in terms of core management and traffic reduction.

In [12] the authors instead devise a protocol for ensuring locality properties in both layers of the architecture. Super peers and resource constrained devices maintain a fixed number of intra- and extra-layer connections, while a scheduling algorithm is used to disseminate a video stream at a rate very close to the average upload bandwidth of participating nodes.

Differently from [11] [12], in [13] the authors use a multi-tree overlay network for media content delivery. Source and inner nodes have higher bandwidth availability, while leaf nodes have scarce available bandwidth. The tree structure is quite rigid, with inner nodes organized in levels: each node is connected to all other nodes in the same level plus additional links to the lower level. Experimental results show how a real prototype of this architecture validates a theoretical model that the authors use for describing node behavior in terms of the number of out-links and bandwidth usage. However the rigidity of the structure requires a central dedicated server that reorganizes the overlay in case of node arrivals/departures. This approach clearly poses scalability issues in large scale systems.

From the assessment of the state of the art, we can conclude that all these works provide a detailed description of the architectures and the implemented functionalities related to a single service (i.e., VoIP, multimedia streaming); however, none of them presents an analysis of the performances of those architectures. In contrast, our work aims at (i) describing how the system infras-

tructure is structured and the telco services (VoIP, Instant Messaging, video conference, ...) can be provided, and (ii) discussing an extensive experimental study we have carried out that points out the performance of our system in terms of users information lookup latency. To the best of our knowledge, this type of evaluation is the first attempt in this context.

Few other works focus on performance evaluation: systems such as Skype [14], [15] and other VoIP applications, such as MSN [16] and Google Talk [17].

In [18] the authors propose a hybrid architecture composed by a centralized core network of powerful servers and a P2P system populated by home devices, with the aim of minimizing management and costs while still providing a good QoS. This paper, based on results in [18], provides an in-depth examination of that solution by adding (i) a theoretical model and its validation for predicting average message latency based on the load imposed on system devices, and (ii) the study conducted for the assessment of the realistic model used in the experimental analysis. In addition, we illustrate how to configure the system architecture through the application of the theoretical model in order to minimize the core network size. Finally, we conclude the paper highlighting some interesting future work that can improve the performance of the architecture.

3 System model

The internal architecture of an NGN can be described in terms of two Network Access Points (NAPs): Access Routers (ARs) and Home Gateways (HGs). An AR is a powerful server playing the role of NAP for a wide geographical area. We consider ARs deployed over the whole national landscape. Instead an HG is a small device installed at users' homes by telco providers. It is characterized by limited computational resources and scarce available memory. We consider the IP architecture of an NGN depicted in figure 1, where core services are identified and applications supporting third party services are hosted on a telco network.

Core service model We model a *location service* that is accessed through a *lookup* and an *update* primitive. Specifically, the $lookup(UserID)$ primitive retrieves the user profile associated to a given UserID. The user profile contains at least the current IP address of the user. For example, in a VoIP application the UserID corresponds to the phone number and the lookup procedure provides the *phone-number IP address* mapping that is exploited on a per-call basis. In another example, push-based applications use lookup to implement the push information delivery model, e.g., the Apple MobileMe adopts a push technique to send information (mail, contacts and calendar) to an iPhone, iPad, or iPod Touch.

The lookup primitive implements an exact match semantics: it returns either a single user's profile whose associated UserID fully matches the one provided by the issuer, or an empty result if the profile does not exist. The

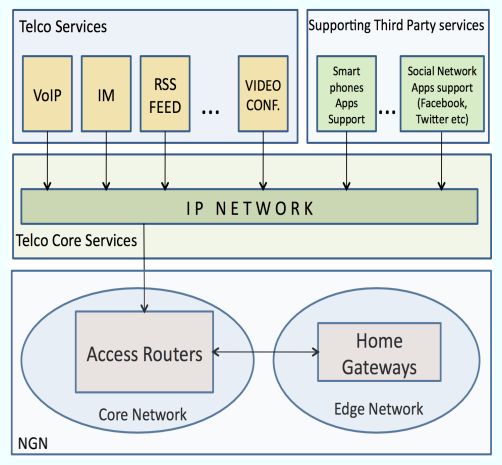

Fig. 1 Expected Services offered by a telco provider

lookup operation performance is critical for many applications. For example, in VoIP the lookup time is directly translated into the end user's Quality of Experience (QoE) and should be kept typically below 200 ms [3]. In general, the lookup primitive should guarantee a short lookup time for QoS requirements enforcement (i.e., low latency), limited control message overhead, low memory and processors consumption on the NAPs, dependability (i.e., no false responses on lookup requests).

The $update(UserID, UserProfile)$ procedure is used to periodically refresh the user profile associated with the UserID. It is issued by a user every time his/her IP address changes. However, a user renews his/her data once per hour even if the associated IP address doesn't change. A similar mechanism has been presented in [2].

User model We consider a system composed of a fixed number of users accessing the IP network via a NAP. We distinguish between fixed and mobile users. Fixed users access the network via their own HG or through an AR (for example because they are using dial-up connections). Mobile users, on the other hand, always get the access through an AR. Once a user has joined the IP network, he/she sends his/her new profile to the location service and refreshes it periodically.

In addition, mobile users can roam in a geographic area and consequently their NAP (AR) can change. In this case, mobile users must update their profile with the IP address of their new NAP. Thus, the only impact of the mobile users handovers on our architecture is a profile update each time they change NAP.

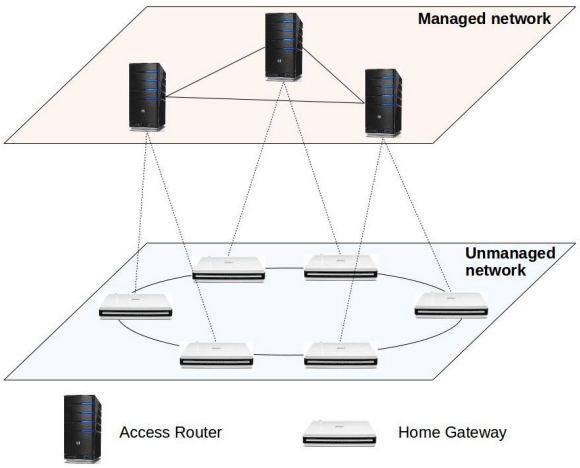

Fig. 2 Hybrid architecture for load balancing

4 Hybrid architecture for balancing users requests

A telco provider usually implements the location service exploiting ARs: all the users' lookup and update messages are sent to ARs and the server storing the required user profile triggers a disk access that represents the main service load.

In the proposed solution we move part of the location service load on a service implemented that exploits the HGs subsystem. In the NGN the location service is very likely to increase in volume and this can cause performance penalties that can in turn translate into a scalability problem. By moving part of this load on the HGs we aim at mitigating this problem. More precisely, the system architecture we propose is organized in two levels: a *managed core* level, populated by ARs connected through a clique, and an *unmanaged edge* level, populated by HGs, connected through a DHT. Due to its simple structure and popularity, we have chosen Chord [8] as the DHT. In this paper, we consider a simplified version of the core network, composed by servers only. In a real setting it may be composed by several data centers connected through network devices. Both the clique and DHT are implemented over an IP network. In addition, each HG maintains an IP connection with a single AR randomly assigned by the telco provider. Figure 2 illustrates a sketch of the overall architecture.

Both ARs and HGs store user profiles; the profiles are replicated in k copies, with the constraint that at least one of them is stored in both the managed and unmanaged networks. We assume that each AR stores a fraction of the user profiles and uses an in-memory hash table to rapidly redirect a lookup request towards the AR that contains the full user profile.

Thus, each AR knows where a user profile is stored in the clique, and the lookup procedure costs $O(1)$ messages. In contrast, each HG stores just a

limited number of user profiles (due to its available memory), and the lookup procedure costs $O(log(n))$ [8] messages in the worst case.

The service works as depicted in figure 3: a telco provider customer issues a user profile request through the $lookup(UserID)$ primitive, either to the managed network or to the unmanaged edge depending on his/her NAP (AR or HG).

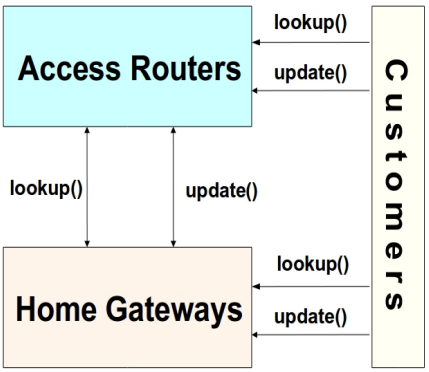

Fig. 3 The system architecture

All lookup requests are first managed by the core network: if a user accesses the system via an HG, this device invokes the $lookup(UserID)$ primitive on the AR that is directly connected. When an AR receives a lookup request (directly from a customer or an HG), it inserts the message in the *Incoming Messages* queue (see figure 4). Then, the AR extracts the message and accesses the hash table to find the destination. If the destination is itself, then it enqueues the message into the *Access to memory* queue; otherwise, with probability p, that message is redirected to the DHT subsystem through a $lookup(UserID)$ call, and with probability $1 - p$ to the destination AR.

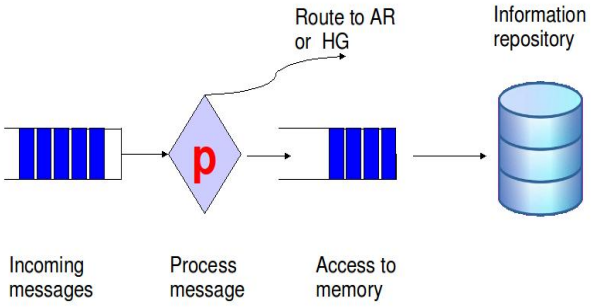

Fig. 4 Internal representation of a Network Access Point

The NAP (AR or HG) that stores the requested user profile directly replies to the customer that issued the request (i.e., the $lookup(UserID)$ procedure returns).

As for lookups, even updates can be issued on the managed or unmanaged network depending on the customer NAP. $update(UserID, UserProfile)$ procedures invoked on HGs are both executed on the DHT and forwarded to the core network. In the same way, $update(UserID, UserProfile)$ procedures issued on ARs are both executed on the managed network and forwarded to the DHT, so as to update all copies of the requested profile and guarantee data consistency.

The motivations behind the choice of this architecture are twofold: (i) we exploit ARs memory availability and bandwidth to connect ARs through a clique. This ensures a fast lookup latency; (ii) we use a distributed system among HGs in order to download part of the traffic from the clique, thus reducing the time spent by a message in the AR's *Access to Memory* queue; this helps us in preserving QoS in terms of latency and avoiding message losses. The use of a DHT as a distributed system contributes to minimize the time necessary to localize a resource.

5 Theoretical model

In this section we describe a theoretical model that can be used in order to predict the mean messages delay as a function of the load imposed on ARs and HGs. We express the load on a device as the ratio between incoming and processed messages in a time unit. Without loss of generality, we consider NAPs equipped only with the *Access to Memory* queue; the motivation is that the disk access time is two orders of magnitude greater than the extracting message time. Thus, the time spent by a message in the *Incoming Messages* queue can be considered negligible.

The *Access to Memory* queue is modelled as an M/M/1 message queue [19], that is a single-server First In First Out (FIFO) queue model in which arrivals are a Poisson process (as, for example, telephony calls arrival [20]), the service time is exponentially distributed, and the buffer length is infinite. The model is characterized by two main parameters: (i) μ, which is the number of processed messages in a time unit; (ii) λ, which is the number of incoming messages in a time unit. The ratio $\rho = \frac{\lambda}{\mu}$ is called *traffic intensity*. In order to guarantee queue stability, it must be $\rho < 1$.
As our aim is to define a theoretical model that computes the average message latency for both ARs and HGs, we use the well-known Little's theorem in the queue model:

$$W = \frac{1}{\mu(1-\rho)} \tag{1}$$

where W is the average time spent by a message in a queue.

For the sake of simplicity, in our analysis we consider the managed network consisting of a single AR: a lookup message is queued just in the AR containing the requested profile that is reached at most in two hops (we neglect transmission delay in the core network). Thus, the time spent by a message in the queue is determined by:

$$W_{AR} = \frac{1}{\mu_{AR}(1 - \rho_{AR})} \tag{2}$$

In contrast, for the unmanaged network we consider intermediate steps before reaching the target HG: the average number of hops in Chord is $\frac{1}{2}\log_2 N$ [8], with N that represents the network size. Thus, formula (1) in case of HGs becomes

$$W_{HG} = \frac{1}{2}\log_2 N_{HG}\Delta t_{hop} + \frac{1}{\mu_{HG}(1 - \rho_{HG})} \tag{3}$$

where N_{HG} is the number of HGs in the network and Δt_{hop} the mean transmission delay per DHT hop.

ρ_{AR} and ρ_{HG} represent the messages' load over an AR and an HG, respectively. In particular, ρ_{AR} depends upon the update and the fraction $1 - p$ of lookup in a time unit. Since the users are HGs randomly assigned to an AR, we can consider the overall load split uniformly over the ARs. Thus, ρ_{AR} is

$$\rho_{AR} = \frac{\alpha(1 - p) + \beta}{N_{AR}\ \mu_{AR}} \tag{4}$$

where α is the total number of lookups, β the total number of updates and N_{AR} the number of ARs in the managed network.

Similarly, ρ_{HG} depends on the fraction p of lookup messages redirected by AR connected to it, and on the fraction $\frac{1}{N_{HG}}$ of updates coming from that AR (an AR issues an update procedure over the DHT by randomly choosing the target HG). Thus, ρ_{HG} is

$$\rho_{HG} = \left(\frac{\alpha}{N_{AR}N_{HG}}p + \frac{\beta}{N_{HG}}\right)\frac{1}{\mu_{HG}} \tag{5}$$

All the parameters used in these formulas are explained in Table 1

Finally, the average time spent by a message in system queues is determined as follows:

$$W = (1 - p)\ W_{AR} + p\ W_{HG} \tag{6}$$

6 Evaluation

In this section we describe an evaluation of our solution, first validating the earlier introduced theoretical model through an experimental analysis on real devices, and then simulating a telco services in a large scale system.

μ_{AR}	number of messages processed by AR/sec
λ_{AR}	number of incoming messages to AR/sec
ρ_{AR}	λ_{AR}/μ_{AR}
μ_{HG}	number of messages processed by HG/sec
λ_{HG}	number of incoming messages to HG/sec
ρ_{HG}	λ_{HG}/μ_{HG}
N_{AR}	number of ARs in the network
N_{HG}	number of HGs in the network
α	number of lookups
β	number of updates
Δt_{hop}	mean transmission delay per DHT hop
p	traffic redirection probability

Table 1 Parameters of the model

6.1 Theoretical model validation

In order to validate our theoretical model we run several simulations on two real devices; namely, a 3 GHz QuadCore PC acting as an AR equipped with 4 GB RAM and 64-bits Linux Ubuntu 9.04 Operating System (OS), and a router Linksys WRT54GL acting as an HG equipped with 200 MHz CPU speed, 16 MB RAM and Linux OpenWRT Kamikaze 8.09.2 OS.
We have separately evaluated ARs and HGs, fixing $p = 0$ and $p = 1$ in (6), respectively. In both cases, we set the time unit to 1 second.

AR evaluation In order to evaluate the behaviour of ARs, we have considered the network populated by ARs only. The AR service has been implemented by installing a C server program and a MySQL database on the PC. The C program has two threads: the first thread is used to insert incoming messages in the queue, and the second is used to extract the top-queue message and access to the database. The database is populated by 2 million entries, each of which containing a pair $< userID, IPaddress >$. Even if we conducted experiments on 10 million users, we set the number of entries in the database to 2 million by considering that users' profiles are distributed on several ARs (each server stores just a portion of them). Each message contains the $userID$ of the requested user profile. We first have found the number of messages processed by an AR in a second. To this end, we have used a second (identical) PC acting as a client. This PC sent messages to the server for several minutes, then it stopped. The server queued incoming messages and when the client stopped functioning, it started the second thread in order to extract messages and access the database for finding all the entries with the $userID$s specified in those messages. We repeated this operation 10 times and the resulting average number of messages processed by the AR in a second was 238.

We ran a second experiment in order to evaluate the average latency per single message; we set $\mu_{AR} = 238$ and we varied ρ_{AR}. The server ran the

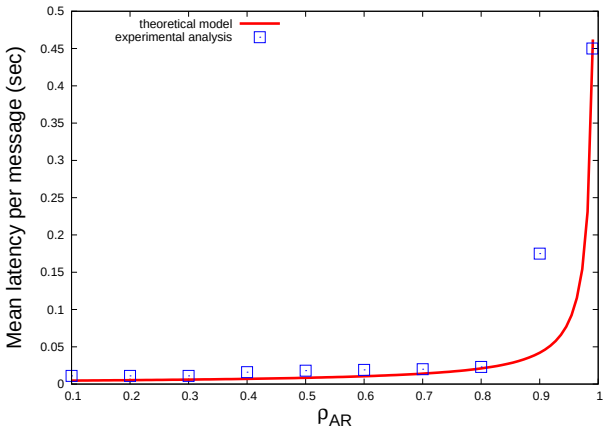

Fig. 5 Average response time for AR under different traffic intensity

threads concurrently, whereas the client was sending messages at a rate $\lambda_{AR} = \rho_{AR}\ \mu_{AR}$ per second, with $\rho_{AR} \in (0; 1)$.

Figure 5 compares equation (6) with $\mu_{AR} = 238$ and $p = 0$, with the results obtained by averaging 10 times a 5-minute test on the real PC. As illustrated in Figure 5, the two curves are very close to one another, thus resulting in the validation of the proposed theoretical model. The experimental curve diverges from the theoretical one for $\rho_{AR} = 0.9$, only. This is due to the fact that this point represents two extreme cases: a case in which the queue of the AR is always full ($\rho_{AR} = 1$), and the case in which that queue is always empty ($\rho_{AR} = 0.8$). Hence, $\rho_{AR} = 0.9$ oscillates between these two cases, producing a divergence from the theoretical value. Note that in figure 5 we omitted the standard deviation for each point of the experimental curve due to its small values.

HG evaluation In order to evaluate the behaviour of the HGs, we have considered the network populated by HGs only. In contrast to using a database to store information, we created a text file containing pairs $< userID, IPaddress >$. In order to find the number μ_{HG} of messages processed in a second we used the same method previously described for ARs, using the router as server. The result we obtained was $\mu_{HG} = 130$.
We repeated the same AR experiment in order to evaluate the average latency per single message, by setting $\mu_{HG} = 130$ and varying ρ_{HG}. The comparison between the theoretical and experimental analyses is shown in figure 6; in this case, we set Δt_{hop} to 0.03 seconds [21] and N_{HG} to 200000.

Even in the case of the HGs, our experiment validates the theoretical model. As in figure 5, we omitted the standard deviation for each point of the experimental curve due to its small values.

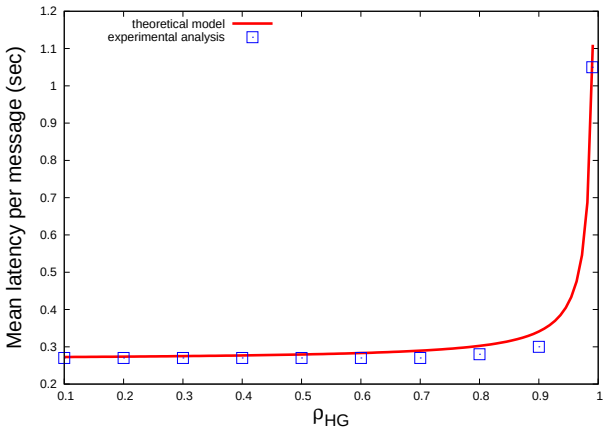

Fig. 6 Average response time for HG under different traffic intensity

6.2 Simulation

We also ran a simulation study aiming at assessing the trade-off between the number of ARs and HGs that could lead to a significant managed network size reduction, while providing services within QoS constraints in terms of messages latency. To this end, we used realistic data collected in the experimental analysis to simulate our model on a large scale system. These data concern: (i) the *service time* (i.e., the time needed for extracting a message from the queue and accessing user information); (ii) the transmission delay in the provider managed network.

Finally, following the specifications we obtained from the telco provider, we did not consider the impact of the churn rate as home gateways are switched on most of the time. This implies that the churn rate is negligible.

Test details We ran our tests again over a 3 GHz QuadCore PC with 4 GB RAM and 64-bits Linux Ubuntu 9.04 OS. Both the architecture and VoIP service are implemented using OMNeT++ v. 4.0 64-bits, a C++ component-based simulator [22]. We ran simulations by varying the number of ARs in the set $\{5-30\}$; the number of users is 10 million: 60% of them are mobile, while the remaining 40% are fixed. Remember from section 3 that the only impact of mobile users on our architecture is the rate at which they update their profile, depending on the frequency they roam and the possibility to change NAP. In addition, the number of HGs is fixed to 4 million. All these values were required by the telco provider as they reflect the real users' setting in our country. Thus, we simulate a scenario in which all fixed users are equipped with an HG. Currently, not all customers are provided with a home gateway. Anyway, due to the continuous effort telco providers are making toward the reduction of management costs, we expect that in a few years, the simulated scenario will be adopted in practice. For each configuration *number of ARs* -

number of HGs, we evaluated the mean message latency. The *service time* was obtained averaging the results of 10000 runs obtained by running the client-server program described in section 6 on the real PC and the Lynksis router; for each single run we computed the time spent to those devices for accessing their information repository and retrieving the requested profile. This time follows a Gaussian distribution, with mean value equal to 4.2323 milliseconds and standard deviation equal to $3.91626\ 10^{-3}$ milliseconds for ARs, and mean and standard deviation for HGs equal to 2.05195 and $1.0759\ 10^{-1}$ milliseconds, respectively. Communication channels have been divided into *fast* and into *slow*. A fast channel is used to connect ARs within the managed network and HGs to ARs; a slow channel, however, is used to connect HGs in the DHT. For sake of simplicity, we model both fast and slow channels as a Gaussian distribution. The Mean value and standard deviation for fast channels are computed averaging 10000 Round Trip Time (RTT) values obtained by running a simple client-server program on two real PCs over the GARR-G network [23] at Sapienza University. These values resulted in 2.44949 and $2.19\ 10^{-1}$ milliseconds, respectively. The mean value and standard deviation for slow channels were set to 30 and 2.569 milliseconds, as assessed in a previous study on a WAN environment [21].

All the following results are the average of 5 different tests on the same scenario; that is, the number of unique profiles in the network is 10 million; each profile has $k = 2$ copies, one stored in the managed network and one in the unmanaged network.

We simulated one hour of the day with a maximum request rate RR, fixed 3000 lookup/sec. This value was obtained by considering the number of telephony calls in a peak hour for a population of 1 million of fixed [24] and mobile [25] users. For a population of 10 million users, we assumed the number of calls open up to approximately 300. In order to accomodate the growing number of applications using such services as depicted in figure 1, we estimated an aggregate RR of an order of magnitude greater than the one expected for phone calls.

Results Figure 7 shows the mean lookup latency varying the number of ARs in the managed network and the probability p of traffic redirection in the set $\{0, 0.1, 0.5, 0.95\}$. As stated above, the number of HGs is 4 million. The parameter that helps in reducing the managed network size is p. Redirecting a lookup message avoids it having to wait for a long in the AR's *Access to Memory* queue, due to the presence of update and other lookup messages. Augmenting p from 0 to 0.95, the number of required ARs in the managed network is more than halved. In addition, the figure shows an interesting trade-off between the number of ARs and p: a higher p value helps to reduce the managed network size; however, it imposes an overhead due to lookup procedures over the DHT. Hence, a telco provider should properly set p accordingly with the managed network size that is willing to manage and the QoS level that it wants to provide.

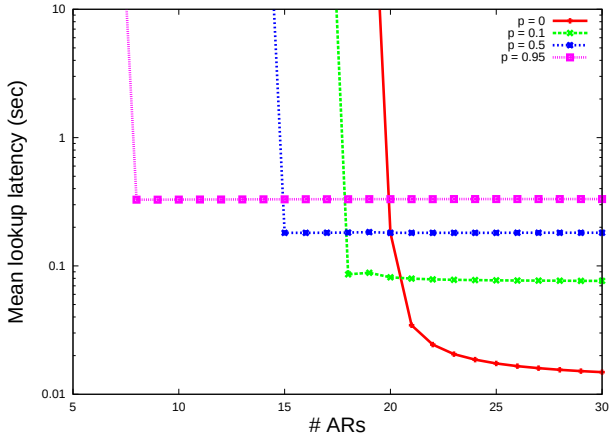

Fig. 7 Mean lookup latency with 4M HGs varying the number of ARs

Figure 8 shows how experimental results are confirmed by the theoretical model. Formula 6 represents a powerful tool for a telco provider to determine the composition of the system's internal infrastructure (number of ARs versus number of HGs) in order to guarantee services within the desired QoS constraint. All the simulation parameters included in 6 are listed in 2.

μ_{AR}	238 messages/sec
μ_{HG}	130 messages/sec
N_{AR}	$\{5-30\}$
N_{HG}	4 millions
Δt_{hop}	30 milliseconds (mean value)
p	(0;1)

Table 2 Values used in the theoretical model

Finally, we use function 6 to compute the minimum number of ARs and HGs in order to guarantee a service with a mean lookup latency of 200 milliseconds [3]. Results are reported in table 3. They prove that when the probability p is very high (i.e., $p = 0.95$), the number of ARs in the network are reduced drastically. In this case, the size of the DHT has a big impact on the mean lookup latency, due to the high number of lookup procedures performed on it. Hence, a small DHT has to be used (i.e., $10k$ HGs), in order to reduce the mean number of hops to retrieve a user profile and, in turn, the lookup latency. On the contrary, when p is lower (i.e., $p = 0.1$, $p = 0.5$) the lookup latency is mainly determined by ARs; thus, a higher number of servers in the managed network is required. In this case, the DHT size has much less impact due to the limited number of lookup procedures performed on it.

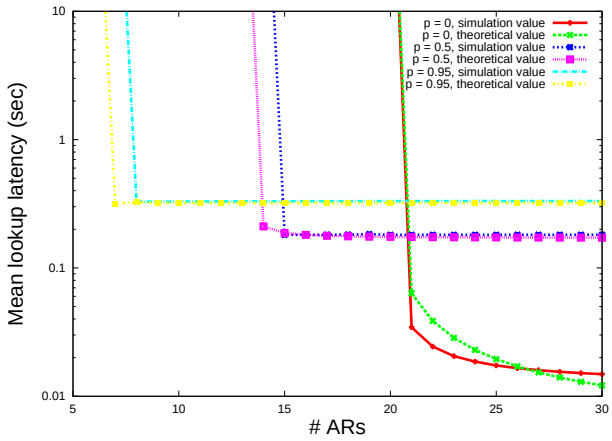

Fig. 8 Comparison between simulation and theoretical results with 4M HGs varying the number of ARs

p	HGs	ARs
0.1	$100k$	19
0.5	$100k$	14
0.95	$10k$	8

Table 3 Configurations ARs-HGs to guarantee a lookup latency of 200 milliseconds

7 Conclusion

In this work, we describe a hybrid architecture for supporting telco and third party services in NGNs. In particular, to embrace the expected developments of these services in the near future, we considered an environment in which the number of requests per second is one order of magnitude greater than the one generated by current VoIP services. This load is balanced between a core network done by a set of servers and an edge network done by home gateways arranged as a DHT. We have shown that the DHT can be used to reduce the number of servers significantly while meeting specific QoS requirements; this, in turn, implies a reduction of the operational (energy and management) costs from the operator (despite the increased load). Simulation results have been provided considering 10 million users and 4 million home gateways. In addition, we have provided a theoretical model of the system that validated the simulation and, thus, can be used as a powerful (and simple) tool by a telco designer to estimate the number of servers and of home gateways necessary to maintain a given quality of service in the telco infrastructure.

An improvement of the presented architecture that we are pursuing at the time of this writing exploits the locality property: instead of having a flat DHT we can use a hirarchical approach as described in [26]. In this case, the whole DHT is partitioned in domains composed of nodes that are close in the physical network. Each domain is a smaller DHT in which nodes maintain ad-

ditional extra-domain links. One of the benefits of this architecture is latency improvement: local traffic can be routed in a single domain so as to reduce the number of hops over the DHT.

In this paper we considered a simplified core network architecture, composed by servers only. In a real setting it can be composed by several data centers deployed over the national landscape and interconnected among them through network devices (bridges, routers, switches). Data centers are composed by servers, each of which runs several virtual machines depending on the current load (number of connected users, traffic) imposed on the network. In a future work we would like to do an analytic and experimental performance evaluation in such a scenario. This has to take into account the possibility that the number of virtual machines may vary with time depending on the system load. This translates into an IP addresses management issue concerning the consistency of $< virtual\ machine\ IP\ address, physical\ machine >$ mappings managed by network devices [27]. The same problem arises when a network device fails; in this case some data centers may be disconnected and their virtual machines must be allocated on available physical servers in other data centers. In our architecture, this problem is exacerbated by the need to also manage the IP addresses of Home Gateways.

Acknowledgements This work has been supported by Telecom Italia. The authors are indebted to Simone Ruffino and Marco Marchisio (Telecom Italia) for discussions and comments on a prelimiary draft of this work and for helping us in defining the system model.

References

1. ITU-T Rec. Y.2001: General Overview of NGN, 2004.
2. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), 2002.
3. Singh, K., Schulzrinne, H.: Peer-to-Peer Internet Telephony Using SIP. NOSSDAV '05: Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video, 2005.
4. Telecom Italia, `http://www.telecomitalia.it`
5. Avaya Secure Router 4134 vs. Cisco ISR 3845: WAN Router Performance, Power Consumption and TCO, `www.avaya.com/usa/resource/assets/whitepapers/Tolly210111AvayaSR4134.pdf`
6. Fan Pu, P2P architecture for IP telephony using SIP. Helsinki University of Technology, TKK T-110.5190. Seminar on Internetworking, May 2006. `http://www.tml.tkk.fi/Publications/C/21/Pu_ready.pdf`
7. Johnston, A.: SIP, P2P, and Internet Communication. IETF Internet Draft, March, 2006. `http://tools.ietf.org/html/draft-johnston-sipping-p2p-ipcom-02`
8. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, page 160, 2001.
9. Bryan, D. A., Lowekamp, B. B., Jennings, C.: SOSIMPLE: A Serverless, Standards-based, P2P SIP Communication System. Appears in AAA-IDA 2005 IEEE `http://www.cs.wm.edu/bryan/pubs/bryan-AAA-IDEA2005.pdf`

10. Marocco, E., Manzalini, A., Sampò, M., Canal, G., Interworking between P2PSIP Overlays and IMS Networks–Scenarios and Technical Solutions, 2007.
11. Kellerer, W., Despotovic, Z., Michel, M., Tarlano, A., Berndt, H., P-Gate: Pushing the Operator Service Platform towards the Network Edge, ICIN Conference 2009, Bordeaux, France, 2009.
12. Efthymiopoulos, N., Christakidis, A., Denazis, S., Koufopavlou, O., LiquidStream Network dependent dynamic P2P live streaming, Peer-to-Peer networking and applications, Springer, 2010.
13. Muller, J., Magedanz, T., Fiedler, J., NNodeTree: A Scalable Peer-to-Peer Live Streaming Overlay Architecture for Next-Generation-Networks, Network Protocols and Algorithms, ISSN 1943-3581, Vol. 1, No. 2, 2009.
14. Skype, http://www.skype.com
15. Guha, S., Daswani, N., Jain R.: An Experimental Study of the Skype Peer-to-Peer VoIP System. In Proceedings of IPTPS, 2006.
16. Chiang, W.H. and Xiao, W.C. and Chou, C.F.: A Performance Study of VoIP Applications: MSN vs. Skype. In Proceedings of MULTICOMM, 2006.
17. Barbosa, R. and Kamienski, C. and Mariz, D. and Callado, A. and Fernandes, S. and Sadok, D.: Performance evaluation of P2P VoIP application. ACM NOSSDAV, volume 7, 2007.
18. Baldoni, R., Beraldi, R., Lodi, G., Platania, M., Querzoni, L., Moving Core Services to the Edge in NGNs for Reducing Managed Infrastructure Size, in Proceedings of the 6th International Conference on Networks and Services Management (CNSM), Niagara Falls, Canada, 2010.
19. Kleinrock, L.: Queueing Systems: Volume 1: Theory, 1975, John Wiley & Sons New York.
20. Erlang, A. K.: The Theory of Probabilities and Telephone Conversation, Nyt Tidsskrift for Matematik B, 20 (1909) 33-39; English translation in: The Life and Work of A. K. Erlang (The Copenhagen Telephone Company, Copenhagen, 1948).
21. Baldoni, R., Marchetti, C., Virgillito, A.: Impact of WAN Channel Behavior on End-to-end Latency of Replication Protocols, In Proceedings of European Dependable Computing Conference, 2006.
22. OMNeT++, http://www.omnetpp.org/
23. Garr: the Italian Research and Academic Network http://www.garr.it/stampaGARR/materiali/leaflet_RETE_GARR_ENG.pdf
24. Iversen, V.B., Glenstrup A.J., Rasmussen, J.: Internet Dial-Up Traffic Modelling, NTS-15, Fifteenth Nordic Teletraffic Seminar, Lund, Sweden, August 22-24, 2000, http://www.diku.dk/~panic/articles/NTS15-InternetDialUp.pdf
25. Bregni, S. and Cioffi, R. and Decina, M.: An Empirical Study on Time-Correlation of GSM Telephone Traffic. IEEE Transactions on Wireless Communications, volume 7, number 9, pages 3428–3435, 2008.
26. Ganesan, P., Gummadi, K., Garcia-Molina, H., Canon in G major: designing DHTs with hierarchical structure, in Proceedings of 24th IEEE International Conference on Distributed Computing Systems, 2004.
27. Greenberg, A. G., Hamilton, J. R., Jain, N. Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P. Sengupta, S., VL2: a scalable and flexible data center network, in Proceedings of ACM SIGCOMM 2009, pp. 51-62.

Marco Platania is a PhD student at Sapienza - Università di Roma. He obtained the Laurea degree in computer engineering from the same institution in 2008 with a work on internal clock synchronization in presence of byzantine faults in WAN environment. His research activity covers peer to peer systems, middleware and event-based systems. As member of the MIDLAB research group, he is involved in Italian and European projects about diagnosis and

monitoring of critical infrastructures, peer to peer solutions for next generation networks and Data Distribution Service in ultra large scale systems.

Roberto Beraldi received the Laurea degree in computer science from the University of Calabria, Cosenza, Italy, in 1991, and the PhD degree in computer science in 1996. He is an assistant Professor at Department of Computer and System Science, Sapienza - Università di Roma, since 2002. He has published more than 60 peer-reviewed papers in various fields, including computer networks, wireless networks, and distributed systems. He also participates in many research projects and regularly serves as a reviewer for international conferences and journals in the above areas.

Giorgia Lodi is a Post-doc researcher at Sapienza - Università di Roma. She received the Ph.D. degree in Computer Science from the University of Bologna (Italy). From 2001 to 2002 she was a research associate of Computer Science at the University of Newcastle upon Tyne (U.K.). In the recent past she published peer-reviewed papers in various computer-science fields including security, complex event processing systems, application server middleware technologies, SLA management systems, and wireless and mobile networks.

Leonardo Querzoni is an Assistant Professor in Computer Science at Sapienza - Università di Roma. He obtained a PhD in computer engineering from the same institution with a work on efficient data dissemination though the publish/subscribe communication paradigm. In the recent past he published several peer reviewed papers in several fields including large scale and dynamic distributed systems, publish/subscribe data diffusion, wireless and sensor networks. He has been invited to chair the student forum for the 7th European Dependable Computing Conference and regularly serves in the technical program committees of many conferences in the field of dependability, event-based communications and autonomic systems.

Roberto Baldoni is a Full Professor in Computer Science at Sapienza - Università di Roma. He conducts research (from theory to practice) in the fields of distributed, pervasive and p2p computing, middleware platforms and information systems infrastructure with a specific emphasis on dependability and security aspects. Roberto Baldoni is a member of the IFIP WG 10.4, of the steering committees of ACM DEBS, DSN and of the editorial boad of IEEE TPDS.