Moving Core Services to the Edge in NGNs for Reducing Managed Infrastructure Size

TECHNICAL REPORT 8/2010 - MIDLAB 2010

Roberto Baldoni, Roberto Beraldi, Giorgia Lodi, Marco Platania, Leonardo Querzoni
Dipartimento di Informatica e Sistemistica "A. Ruberti"
Sapienza - Università di Roma
via Ariosto 25, 00185, Rome, Italy

***Abstract*—Telco providers are in the phase of migrating their services from 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 CAPEX, while OPEX remains fairly stable due to network management and energy costs. At the same time we are expecting a big increase of the load of a telco provider at the core level due to the istantiation of new telco services (VoIP, video conferencing etc) and to the support of third parties services (such as support to smart phone applications, etc.). 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 QoS levels comparable with those provided by a traditional fully-managed infrastructure. This is done taking into consideration the increase of the load of such services that is expected to raise of one order of magnitude in the close 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 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 the NGN of a big telco provider (i.e., finding the IP address of a given unique user profile).**

I. Introduction

Next Generation Network (NGN) is a packet-based network able to provide Telecommunication Services to users and able to make use of multiple broadband, QoS-enabled transport technologies and in which service-related functions are independent of the underlying transport-related technologies [2].

Worldwide telco operators are striving to develop new solutions over NGNs, which exhibit a telephone traffic cost reduction and provide a higher number of services such as web call center, e-commerce, home banking, video conference, as well as supporting third party applications such as smartphone applications, social network applications etc. These applications are expected to increase the resources requested to the telco core network of one order of magnitude in the close future. Despite this raising load, almost all NGN communication protocols such as H.323 and SIP (Session Initiation Protocol) [1] rely on a collection of central servers that manage all clients requests in case, for example, a user contact information has to be retrieved and a connection with that contact has to be established. These centralized architectures grow 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 (e.g. Telecom Italia, AT&T, France Telecom etc.), it is usually organized as a static tree with servers installed in different geographic domains (leaves of the tree). These servers are used to respond to regional user contacts requests and forward those requests to a cluster of other servers (root of the tree) made available for user contact retrieval purposes. In these networks, the majority of the system costs are represented by maintenance and configuration, due to the employment of dedicated system administrators in each domain [3]. If the load increases, servers are added to the system to prevent the creation of bottleneck. Several load balancing systems have been proposed to smartly split the load including some where servers are arranged along a DHT systems [3].

In this paper we describe the outcome of a project, funded by *Telecom Italia* [7], which focuses on the design and the simulation of a load balancing architecture providing telco core services supporting up to 10 millions of customers. This architecture, on one hand, aims to reduce operational costs and, 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 handled by the current telco systems. The architecture specifically analyzes the contact management service, that is, considering a unique client ID, the service returns the *current* IP address of the users. This service is actually a basic building block of many applications deployed on the top of the telco provider network such as VoIP, video conference, Instant Messaging, etc... Moreover, the contact service also supports a myriad of applications that are developed by third parties from smartphone applications to social network ones. We consider a telco network formed by a servers'(core) network, managed

This work has been supported by Telecom Italia.

by the telco provider, and an edge network. The latter is actually a DHT formed by devices installed at user premises, namely Home Gateways. The load of contacts' requests is then balanced between the managed network and the edge one in order both to respect the QoS of the core service and to reduce the management costs. We study how the number of servers can be reduced by including opportunistically 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 millions customer network. At the same time we build a analytic model of the architecture to validate the simulation results.

The performance analysis shows that our 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 this small managed network is combined with an edge network of four millions home gateways. From the point of view of the telco provider, this translates into a substantial reduction of (i) the OPerating EXpences (OPEX) for running the system, namely maintenance, management, and energy costs, and (ii) the CAPital EXpenditure (CAPEX) for the smaller number of servers forming the core network. In the 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% [20]. Considering that the cost of energy is on average 20% of the cost of the installed devices [20], 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 II describes a background in the area of VoIP and NGNs and a number of works similar to our proposed solution. Section III presents the system model, while Section IV illustrates the solution we propose. Section V describes the theoretical model that allows us to model the telco system infrastructure. Section VI presents an experimental evaluation we have conducted in real settings and using the designed theoretical model. Finally, section VII concludes this paper highlighting some future works.

II. 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 Public Switched Telephone Network (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 (NGNs).

A NGN decouples QoS-enabled transport technologies from service-related functions which are independent of the underlying transport-related technologies. A key characteristic of a NGN is its supports to generalized mobility which will allow consistent and ubiquitous provision of services to users. To this end, a variety of identification schemes which can be resolved to IP addresses for the purposes of routing in IP networks has to be provided [2]. A fundamental identification scheme is mapping a user profile to its current IP address.

For example, in the Session Initiation Protocol (*SIP*) [1] 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 users lookup procedure: 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 users handover, 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 users location.

Throughout this paper we refer to a generic *location service* that maps user profile to 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 client and 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.

P2P is an attractive alternative for addressing these drawbacks. In fact, several solution has been proposed to replace the standard SIP lookup procedure by an existing P2P protocol [9]; typically, the lookup procedure is performed over a Distributed Hash Table (DHT). For example, Johnston in [5] presents a SIP-using-P2P algorithm in which users location information are directly stored in the DHT, without using traditional SIP Registrar and Location servers. In [3] 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 [4], while ordinary nodes maintain a connection with a single super node. All users' information are stored on the DHT; thus, all lookup procedures are performed on it. SIP is used as underlying protocol for locating users, joining the DHT, registering users, call setup and instant messaging.

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

SOSIMPLE [8] combines the traditional SIP and SIMPLE, a set of SIP extensions for Instant Messaging (IM). All nodes in SOSIMPLE have the same responsibility (i.e., no super nodes) and are organized in a DHT based on Chord.

In [21] 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, [21] 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.

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); however, none of them presents an analysis of the performances of those architectures. In contrast, our work aims at (i) describing how the system infrastructure is structured and the telco services (VoIP, Instant Messaging, video conference, ...) that 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.
Just other few works focus on performance evaluation; however only of systems such as Skype [6], [11] and other VoIP applications, such as Google Talk [13] and MSN [12].

III. System model

The internal architecture of a 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. An HG instead 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 are identified core services and applications supporting third party services hosted on a telco network.

a) ***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 UserID corresponds to the phone number and the lookup procedure provides the *phone-number IP address* mapping which is exploited on a per-call basis. As another example, push-based applications use lookup to implement the push information delivery model, e.g., the Apple MobileMe adopts push technique to send information (mail, contacts and calendar) to iPhone, iPad, or iPod Touch.

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

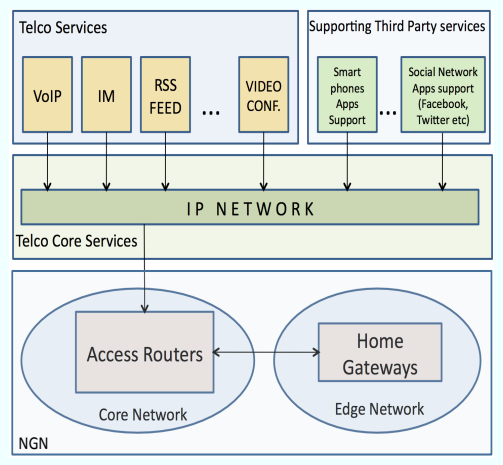

Fig. 1. Expected Services offered by a telco provider

is critical for many applications. For example, in VoIP the lookup time is directly translated into the end users Quality of Experience (QoE) and should be kept typically below 200 ms [3]. In general, the lookup service should guarantee 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 to 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 mechanism similar to the one presented in [1].

b) ***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.

IV. 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; 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 exploiting 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 service 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*

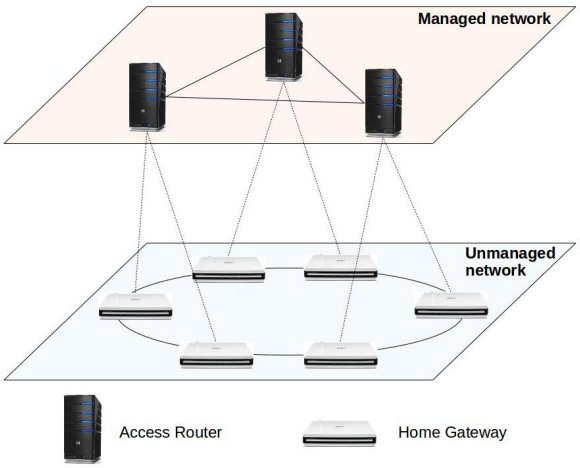

Fig. 2. Hybrid architecture for load balancing

level, populated by HGs, connected through a DHT. Due to its simple structure and popularity, we have chosen Chord [4] as DHT. Both the clique and DHT are implemented over an IP network. In addition, each HG maintains an IP connection with a single AR, assigned at random 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 $O(log(n))$ [4] messages in the worst case.

The service works as follows: all lookup requests generated by users connected through ARs and HGs are sent to the core network. When an AR extracts a lookup message from the *Incoming Messages* queue (see figure 3), it 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, and with probability $1-p$ to the destination AR.

Even update messages are sent to the core network. When an AR extracts an update message from the *Incoming Messages* queue, it accesses the hash table to find the destination AR. Hence, the *Access to memory* queue contains all the update requests for profiles stored in the local disk and only a fraction of lookup requests that it can directly serve. Updates are not redirected over the DHT. However, in order to guarantee profiles consistency, when an AR updates a user profile, it executes an update procedure over the DHT. 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)

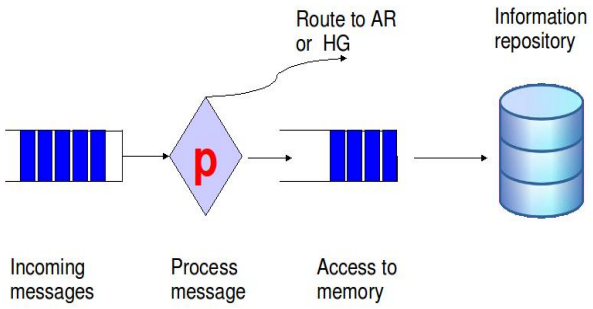

Fig. 3. Internal representation of a Network Access Point

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 preserve QoS in terms of latency, and avoiding message losses. The use of a DHT as distributed system contributes to minimize the time necessary to localize a resource.

V. Theoretical model

In this section we describe a theoretical model that can be used in order to predict the mean messages delay as 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 order 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 a M/M/1 message queue [14], that is a single-server FIFO queue model in which arrivals are a Poisson process (as, for example, telephony calls arrival [15]), 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, which 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:

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 I
PARAMETERS OF THE MODEL

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

In contrast, for the unmanaged network we consider intermediate steps before reaching the target HG: the average number of hops in Chord is $\frac{1}{2}\log_2 N$ [4], 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 from the update and the fraction $1-p$ of lookup in a time unit. Since users are HGs are randomly assigned to an AR, we can consider the overall load splitted uniformly over 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 choosing at random 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 I

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

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

VI. 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.

A. 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 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 evaluated separately ARs and HGs, fixing $p = 0$ and $p = 1$ in (6), respectively. In both cases, we set the time unit to 1 second.

*c) **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 on the PC a C server program and a MySQL database. 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 millions of entries, each of which containing a pair $< userID, IPaddress >$. Each message contains the $userID$ of the requested user profile. We first have found the number of messages processed by a 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 run 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 run the 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 4 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 4, the two curves are very close 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 by 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 4 we omitted the standard deviation for each point of the experimental curve due to its small values.

*d) **HG evaluation**:* In order to evaluate the behaviour of HGs, we have considered the network populated by HGs only. In contrast to use 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 is $\mu_{HG} = 130$.
We repeated the same AR experiment in order to evaluate the

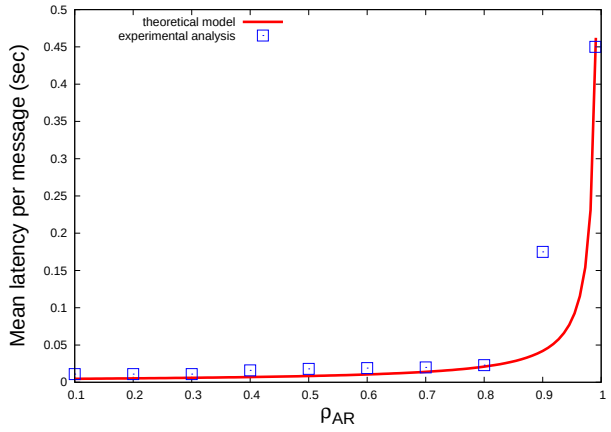

Fig. 4. Average response time for AR under different traffic intensity

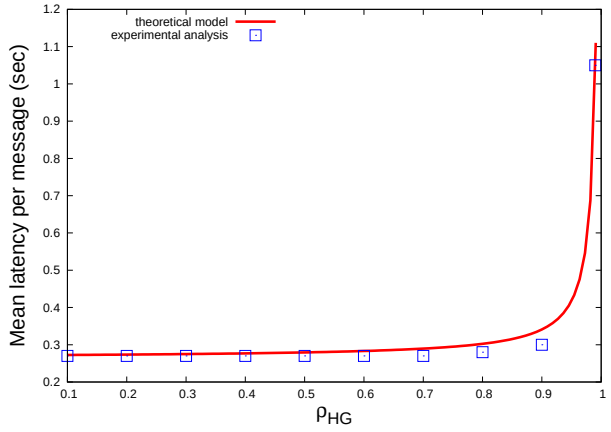

Fig. 5. Average response time for HG under different traffic intensity

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 5; in this case, we set Δt_{hop} to 0.03 seconds [16] and N_{HG} to 200000.

Even in the case of HGs, our experiment validates the theoretical model. As in figure 4, we omitted the standard deviation for each point of the experimental curve due to its small values.

B. Simulation

We also run 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 didnt consider the impact of churn rate as home gateways are switched on most of the time. This implies that the churn rate is negligible.

e) ***Test details****:* Our tests run again over a 3 GHz Quad-Core 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 [17]. We run simulations by varying the number of ARs in the set $\{5 - 30\}$; the number of users is 10 millions: 60% of them are mobiles, while the remaining 40% are fixed. In addition, the number of HGs is fixed to 4 millions. 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 an home gateway. Anyway, due to the continuous effort telco providers are making toward the reduction of management costs, we expect that in few years the simulated scenario will be adopted in practise. 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 running on the real PC and the Lynksis router the client-server program described in section VI; we computed for each single run 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 in *fast* and *slow*. A fast channel is used to connect ARs within the managed network and HGs to ARs; a slow channel, instead, is used to connected 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 RTT values obtained running a simple client-server program on two real PCs over the GARR-G network [19] 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 [16].

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 millions; 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 maximum request rate RR, fixed 3000 lookup/sec. This value is obtained by considering the number of telephony calls in a peak hour for 1 million population of fixed [18] and mobile [10] users. For a population of 10 millions users, we assume the number of calls growing approximately up to 300. In order to accomodate the growing of applications using such services as depicted in figure 1, we estimate an aggregate RR of an order of magnitude greater than the one expected for phone calls.

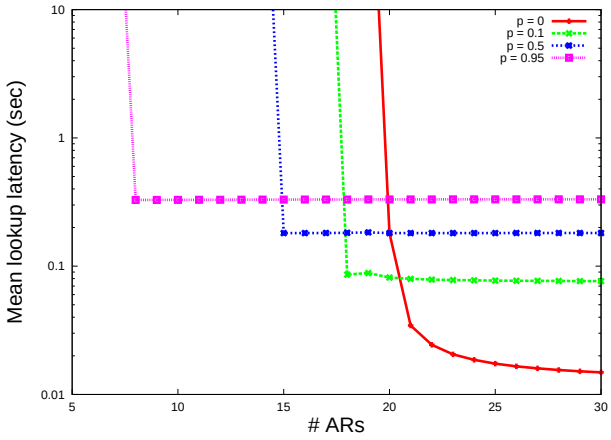

Fig. 6. Mean lookup latency with 4M HGs varying the number of ARs

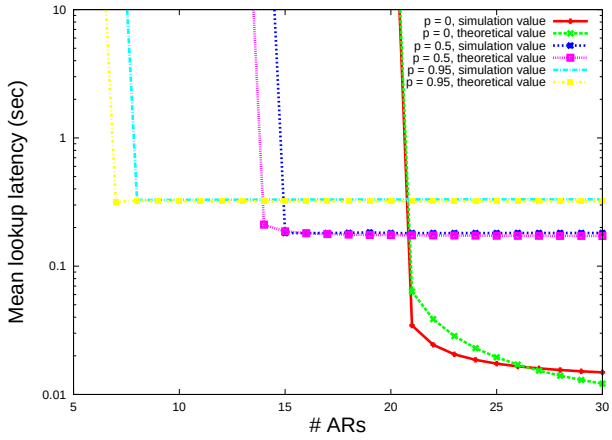

Fig. 7. Comparison between simulation and theoretical results with 4M HGs varying the number of ARs

f) ***Results****:* Figure 6 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 millions. The parameter that helps in reducing the managed network size is p. Redirecting a lookup message avoids it 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 number of ARs and p: an 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 maintain and the expected QoS standard.

Figure 7 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 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 II.

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 II
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 mean lookup latency of 200 milliseconds [3]. Results are reported in table III. They evidences that when the probability p is very high (i.e., $p = 0.95$), the number of ARs in the network reduces 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 procedure performed over 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 an 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 over it.

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

TABLE III
CONFIGURATIONS ARS-HGS TO GUARANTEE A LOOKUP LATENCY OF 200 MILLISECONDS

VII. 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 close future, we considered an environment in which the number of requests of lookup 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) cost from the operator (despite the increased load). Simulation results have been provided considering 10 million users and 4 millions of home gateways. Finally, we have provided a theoretical model of the system that validated the simulation and, thus, it 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.

REFERENCES

[1] 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)

[2] ITU-T Rec. Y.2001: General Overview of NGN (2004)

[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] 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)

[5] Johnston, A.: SIP, P2P, and Internet Communication. IETF Internet Draft, March, 2006. http://tools.ietf.org/html/draft-johnston-sipping-p2p-ipcom-02

[6] Skype, http://www.skype.com

[7] Telecom Italia, http://www.telecomitalia.it

[8] 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

[9] 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

[10] 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)

[11] Guha, S., Daswani, N., Jain R.: An Experimental Study of the Skype Peer-to-Peer VoIP System. In Proceedings of IPTPS, 2006.

[12] 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

[13] 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.

[14] Kleinrock, L.: Queueing Systems: Volume 1: Theory, 1975, John Wiley & Sons New York.

[15] 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).

[16] 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

[17] OMNeT++, http://www.omnetpp.org/

[18] 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

[19] Garr: the Italian Research and Academic Network http://www.garr.it/stampaGARR/materiali/leaflet_RETE_GARR_ENG.pdf

[20] www.avaya.com/usa/resource/assets/whitepapers/Tolly210111AvayaSR4134.pdf

[21] Marocco, E., Manzalini, A., Sampò, M., Canal, G., Interworking between P2PSIP Overlays and IMS Networks–Scenarios and Technical Solutions, 2007