Peer-to-Peer Netw. Appl.6 (2013) 2, pp. 175–193

DOI: 10.1007/s12083-012-0151-9

Validation of H-P2PSIP, a scalable solution for interoperability among different overlay networks

Isaias Martinez-Yelmo · Roberto Gonzalez-Sanchez · Carmen Guerrero

I. Martinez-Yelmo (🖂)
Universidad de Alcalá, Campus Universitario N-II, km. 31,5. E-28871 Alcalá de Henares, Madrid, Spain e-mail:
isaias.martinezy@uah.es ; URL: http://it.aut.uah.es/∼isaias/

R. Gonzalez-Sanchez · C. Guerrero Universidad Carlos III de Madrid, Av. Universidad 30. E-28911 Leganés, Madrid, Spain

R. Gonzalez-Sanchez e-mail: rgonza1@it.uc3m.es

C. Guerrero e-mail: guerrero@it.uc3m.es

Abstract This paper reports the results of experiments from an implementation of H-P2PSIP, a hierarchical overlay architecture based on the ongoing work in the IETF P2PSIP Working Group. This architecture allows the exchange of information among different independent overlay networks through the use of a two-layer architecture based on super-peers and hier-archical identifiers. The validation of this proposal is based on a Linux based real implementation where we have used four different scenarios with 1,000 peers in order to perform different experiments. We have obtained results for different parameters such as rout-ing performance (number of hops), delay, routing state (number of overlay routing entries) and bandwidth consumption.

Keywords Hierarchical overlay, P2PSIP , Implementation, Validation, Interconnection, Interoperability

1 Introduction

New paradigms are arising on the Internet to offer cost-effective and scalable mechanisms for Content Delivery Networks (CDNs), Cloud Computing or distributed services. Peer-to-Peer networks emerged last decade to address this problem in a distributed and scalable way. Indeed, Peer-to-Peer networks make possible to place and recover resources over the Internet in a distributed way creating an overlay network. Some examples are Chord [44] or Kademlia [36]. The last one is especially popular in file-sharing applications, such as eMule,[1] with its KAD network [43] or trackerless Bittorrent[2] [15]. Finally, this kind of networks is becoming also popular in the datacenters of very popular websites such as Facebook [26].

Nevertheless, although Peer-to-Peer applications are popular nowadays, some open issues have not been addressed yet. Probably, one of the most challenging issues is to define mechanisms that allow the interoperability among different Peer-to-Peer networks. The Internet community is making some efforts to bridge this existing gap. The P2PSIP[3] WG is developing a protocol (RELOAD [24]), that allows the implementation of any overlay network [5] and specially

[1] http://www.emule-project.net/
[2] http://www.bittorrent.com/
[3] http://datatracker.ietf.org/wg/p2psip/

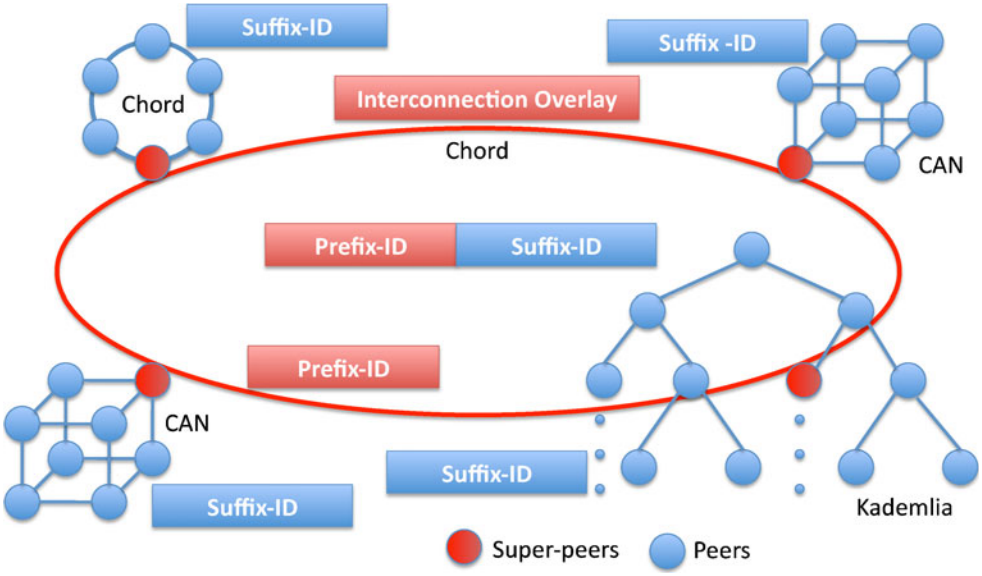


Fig. 1 H-P2PSIP

focuses on structured Peer-to-Peer networks.[4] This fact will lead to the unification and simplification of the development and management of Peer-to-Peer based applications and services. Nevertheless, no solution has been seriously considered to allow the interoperability among different overlay networks. This feature is of real importance since it can help to make possible the interoperability among different based overlay/peer-to-peer distributed services such as a distributed SIP replacement [23], IPTV [25] or content sharing in CDNs or Cloud computing [47]. These interoperability properties are really promising since open new possibilities such as Service Provider migration (i.e. from one distributed Cloud service to other one) or something even better, to make possible the collaboration among different platforms. This technology would help to make possible the download of resources from different providers or to allow load balancing and backup properties among different distributed cloud computing services in a transparent way.

The H-P2PSIP architecture [28, 31] aims to address the open issue of interoperability among different overlay networks in RELOAD. Further details about this proposal are provided in Section 3. As a summary, we can say that it consists of a two-level hierarchical overlay architecture based on super-peers that allows the exchange of information among different heterogeneous overlay networks; Fig. 1 illustrates an example with DHT networks. This proposal has been mathematically analysed [30, 32] and evaluated using simulation tools [28, 29, 31]. However, due to the complexity of the overlay technologies because of their decentralised architecture, a real implementation based on a real TCP/IP stack is essential to perform a complete and solid validation. We present in this article the first implementation of a hierarchical overlay architecture with these interoperability characteristics at the best knowledge of the authors. This implementation has allowed us to study different relevant aspects such as routing efficiency of queries, size of overlay routing tables in peers or scalability issues in terms of bandwidth consumption. Moreover, a brief comparison with our previous simulation results has been performed in order to study the differences between a simulation and a full implementation. Preliminary results of this work were presented in [33].

The obtained results demonstrate that H-P2PSIP is able to save more than a 20% of the hops needed to reach a destination with respect to a flat counterpart. Furthermore, super-peers, which are the worst case, need to manage around 80% of the traffic needed in a flat overlay network while the memory used to maintain their overlay routing tables is around 40% smaller. Therefore, the proposed solution is not only efficient in terms of interoperability properties; but it also has good scalability properties.

The structure of the paper is as follows. Section 2 summarizes the related work with respect to P2PSIP

[4] Although RELOAD considers overlay networks in general, it is especially focused on structured Peer-to-Peer networks, mainly Distributed Hash Tables (DHTs). Thereby, we use from this moment both terms overlay network and peer-to-peer network with the same meaning. Each term is selected depending on the context where it is used.

and hierarchical overlay networks. A detailed description of H-P2PSIP is provided in Section 3. Section 4 explains the main development steps to implement this proposal as well as the different scenarios and setups used for its validation. The outcome of our experiments is detailed in Section 5 and the lessons learnt after this work are included in Section 6. Finally, a summary of the conclusions obtained from this work is included in Section 7 followed by a detail of the future research lines in Section 8.

2 Related work

2.1 P2PSIP and the RELOAD protocol

Our proposal is very closely related to the IETF P2PSIP Working Group and the RELOAD protocol [24]; thus, we deem necessary first to include a proper description of this ongoing work. RELOAD is a lightweight binary protocol that is suitable for peers that need to manage a lot of connections and resources (CPU, bandwidth, etc.). Its main objective is to provide the necessary mechanisms to develop and implement any overlay network in order to facilitate the creation of distributed systems. It has a modular design (see Fig. 2) that allows reusing most of its components for different overlay networks. The *Topology Plugin* is responsible for the implementation of the overlay network itself. Hence, a different plugin is necessary for each different overlay network. This plugin takes care of the operation and maintenance tasks of each overlay network. Consequently, it inherits some of the typical overlay network primitives such as Join, Leave or Update. The RELOAD architecture has also a *Storage* module that takes care of managing the placement and replication policies of the information stored in the overlay network. Hence, it takes care about Store, Fetch and Remove operations in the overlay network. An API communicates this *Storage* module with the *Topology Plugin* since the *Storage* module decides where to store the information, but only the *Topology Plugin* knows how the peers are organized in order to reach them. Additionally, the *Message Transport* component is responsible for the end-to-end management of the exchanged information and deals with issues such as retransmission or duplicates. Finally, the *Forwarding and Link Management* component delivers all messages from all modules. This layer provides NAT traversal support and churn management to identify missbehaving/disconnected peers. NAT traversal support is based on the Interactive Connectivity Establishment (ICE) protocol [41]. In order to take full advantage of the integration of ICE with RELOAD, there are two important fields in any RELOAD message:

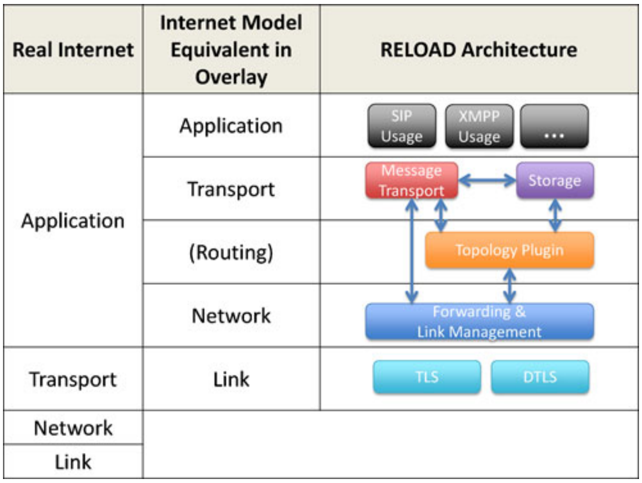


Fig. 2 RELOAD architecture

Destination-list It is a list of intermediate peers that must be used to forward a query. This destination list helps to route overlay messages through STUN and TURN servers to perform NAT traversal operations.

Via-list It is a list used to get a response path, which is symmetric to the path of the request. It stores all intermediate hops of an operation from the source to the destination. The RELOAD protocol is based on a request-response exchange. Thereby, it is strongly recommended to use bidirectional symmetric paths since response messages can reuse the NAT traversal path created previously to deliver the response to the origin.

The main advantage of this modular design is the fact that the *Topology Plugin* can change but the rest of functionalities remain unaffected and as a result it is not necessary to reimplement them. Figure 2 presents the role of the different modules with respect to the Internet architecture. We can appreciate how all layers have their equivalence at the overlay/application level. However, one important feature is missing: there is no mechanism that allows the exchange of information among different overlay networks, which is addressed by H-P2PSIP.

2.2 Hierarchical peer-to-peer networks

There are many publications related to hierarchical peer-to-peer networks. One of the first and more relevant works is presented in [19], which included an explanation and evaluation of the benefits of a hierarchical approach. Later, other publications related to hierarchical peer-to-peer networks have been published with different proposals and points of view.

Hieras [48] minimizes the delay experienced by queries in a multilevel hierarchy. Each peer can participate in different groups inside the Hieras hierarchical architecture. Peers are grouped in Hieras according to the delay among peers. The higher the delay among peers, the higher the level of the hierarchy where the formed group is placed. Queries are launched in the lowest level of the hierarchy and if a resource is not found, the next level will be used. The idea under this scheduling of queries is to find the fastest path among different peers. The main drawback of Hieras is the increase of routing state and maintenance traffic in all peers, since each peer needs to maintain each hierarchy level where it participates. Furthermore, we can observe in [38] how this approach can be extended to other structured peer-to-peer networks.

Canon [18] also optimizes the delay of different structured peer-to-peer networks, but it intentionally limits the overlay routing entries maintained by peers to avoid their overload. Unfortunately, the design of this hierarchical architecture is heavily coupled with the DHT used to build the hierarchy. All groups must support the same DHT network since specific modifications must be performed to adapt them to the Canon proposal. These modifications focus on merging the same kind of peer-to-peer networks in a larger overlay.

In order to overcome some of the drawbacks of the previous proposals, Cyclone [1] was designed. This proposal builds a hierarchical DHT overlay network where different DHT networks can exchange information among them. The different DHT networks are interconnected through a top-level overlay network very similar to Kademlia [36]. If an item is stored in other overlay network, the top-level hierarchy is used to reach other peer-to-peer networks until the desired item is found. Nevertheless, the routing is suboptimal since different overlay networks must be crossed until the overlay network containing the desired item is reached. It would be desirable to efficiently use the top-overlay network to directly reach the overlay containing the desired item.

Other proposals take a completely different approach and propose gateways to exchange information among different overlay networks [9, 10, 17] instead of merging them in a hierarchical structure such as previously mentioned proposals. Gateways have the advantage of enabling the communication between pairs of peer-to-peer networks but a different set of rules is necessary to enable the communication between each different pair of peer-to-peer networks.

2.3 Evaluation platforms for peer-to-peer systems

There are two main approaches for evaluating real developments of peer-to-peer applications and distributed systems: distributed testbeds and emulation platforms. The well-known distributed research testbed Planetlab[5] [12] belongs to the first category. Planetlab allows the validation of real peer-to-peer applications through geographical distributed virtual machines, which run the desired applications. Nevertheless, Planetlab has disadvantages [40], which become bigger in large-scale experiments where the slice management mechanism (resource availability per experiment) of Planetlab is undisclosed [11]. Furthermore, there is no guarantee of obtaining results as a representation of current Internet since most of Planetlab nodes are connected to high-speed networks belonging to universities and research facilities [2], which does not correspond with real Internet. In addition to Planetlab, we have the emulation platforms Emulab [21] and Modelnet [45] in the second category. These emulation platforms allow the emulation of real networks by describing connectivity graphs and defining the characteristics of the links in terms of bandwidth, delay, error probability or any other predefined characteristic.

We analyse four different metrics in this paper to validate the behaviour of our design. The first three ones are the number of hops, the number of entries in the peer-to-peer tables and the associated traffic, which are not heavily correlated with the underlying network topology (further details are in Section 5). Obviously, the network topology would have some effect, for instance in the refresh mechanisms of the overlay routing tables but its effectiveness has been proved previously in multiple and different conditions. Additionally, there is a fourth metric, the delay, which depends on the underlying network topology. Therefore, the dilemma consists on choosing the best option between Planetlab or the emulation platforms. The emulation platforms can be configured to emulate previously worldwide network measurements such as those ones provided by the IEPM PingER project [35], giving a reasonable, but limited, representation of the current underlying network in Internet. Unfortunately, Planetlab cannot give either a real representation of current Internet. Furthermore, it must be considered that the repetition of the experiments in Planetlab is not straightforward,

[5] http://www.planet-lab.org/

the background traffic and the availability of nodes cannot be controlled. These facts make difficult the repeatability of the experiments to get a correct statistical validation of the results (with enough independent experiment repetitions under similar conditions), which is a disadvantage to consider. Indeed, the generalization of the results from the Planetlab platform is not straightforward.

For all previous reasons, we have not considered Planetlab in our testbed and we use the Modelnet emulation platform in conjunction with the measurements of the IEPM PingER project to validate our implementation of H-P2PSIP. This setup allows us to validate our implementation under a close conditions corresponding to a worldwide Internet peer-to-peer application with a reasonable scalability and operativeness. Modelnet is demonstrated to be a good option since the research community is using it to validate many flavours of peer-to-peer applications. Some examples are peer-to-peer online games [49], peer-to-peer video on demand [46], peer-to-peer networks [16, 27] peer-to-peer file-systems [8] or the Tor emulation platform, which is also based on Modelnet [4].

3 H-P2PSIP

H-P2PSIP [28, 31] defines a hierarchical overlay network composed by two levels of hierarchy; an example is given in Fig. 1. The purpose of this two-level hierarchy is the exchange of information among different overlay networks based on the RELOAD protocol. In H-P2PSPSIP, the different inner overlay networks can exchange information among them. This exchange is obtained through the Interconnection Overlay, which is placed on its own in the upper level of the hierarchy. The Interconnection Overlay, which can be based on any overlay network, is used to route among different overlay networks when a peer from one overlay network wants to retrieve information from a different one. In order to properly build the hierarchy, H-P2PSIP uses a hierarchical space of identifiers composed by Hierarchical-IDs (see Fig. 3). A Hierarchical-ID contains two concatenated IDs: a Prefix-ID and a Suffix-ID. The Prefix-ID is used to route queries in the Interconnection Overlay among different overlay networks. This fact implies that all peers and resources belonging to the same peer-to-peer network share the same Prefix-ID. On the other hand, the Suffix-ID is used to find a resource inside an overlay network belonging to the lower level of the hierarchy. Thus, if a resource from other overlay network wants to be obtained, the query is routed to the desired overlay network through the Prefix-ID in the Interconnection Overlay. Finally, the desired resource in the destination overlay network is found through the Suffix-ID.

Prefix-ID (n-bits)	Suffix-ID (m-bits)

Generic URI: owner@example.com/ID
Prefix-ID = hash(example.com)
Suffix-ID = hash(owner@example.com/ID)

Fig. 3 Hierarchical-ID and Resource-ID/URI mapping

3.1 Hierarchical-ID generation

An important point is how to generate the Hierarchical-IDs in a predefined well-known way. We consider that usually most of the resources have an associated URI (e.g., `owner@example.com/ID`) that can be used to identify them. In previous URI, *owner* identifies the responsible of a resource if it exists, *example.com* is the domain associated to the resource and *ID* identifies the resource itself (e.g. a path in an URL).

Node-ID generation A Node-ID identifies each node participating in an overlay. The Prefix-ID must be the same for peers belonging to an overlay network and the Suffix-ID has to be unique for each of them. Thus, we obtain the Prefix-ID by hashing the domain name where the peers belong (see Fig. 3). However, the Suffix-ID generation is more complex and it depends on the desired security level. An option is to apply a hash function to some unique parameter (e.g., an IP address), but this option is not completely secure [14]. A secure solution implies a central authority that assigns unique random Node-IDs with signed certificates.

Resource-ID generation The Prefix-ID is generated in the same way as in the Node-ID, we hash the domain name of the URI. However, the Suffix-ID must be generated in a deterministic way and associated to the resource itself. Our proposal is to generate the Suffix-ID applying a hash to the whole URI that identifies the resource (see Fig. 3). Ergo, once the mapping between the URIs and Hierarchical-IDs is established, any resource can be stored with its Resource-ID and the original URI in order to perform disambiguation in resources sharing the same Resource-ID. If allowed in the overlay network policy, resources without an owner could be stored and the owner term would not be used in the URI.

3.2 Super-peer role

Hierarchical-IDs have been defined, but we need to establish the mechanism to manage them properly. We use super-peers that are attached to both inner overlays and the Interconnection Overlay (see Fig. 1). If a peer needs information from another overlay, it forwards the query to a super-peer from its overlay network. Hence, only super-peers carry out the necessary operations to route queries among the different overlays. Thus, super-peers must perform additional maintenance operations with respect to legacy peers; the management of the Interconnection Overlay and the forwarding of queries have a cost but legacy peers do not have to support it.

Two different scenarios related to the provision of super-peers can be considered: an operator scenario and a community network scenario. In an operator scenario, the operators provide high availability well-known super-peers with high bandwidth, CPU, memory and storage capabilities. Super-peers track all signalling among the different overlays and SP can use this information with charging purposes. On the other hand, the community network scenarios match with decentralised infrastructures without a central authority. Thus, some mechanism is necessary to select the most suitable super-peers among all available peers [37, 39, 42]. These proposals can be combined with Scribe [6] or similar proposals [7] to recollect and disseminate the necessary related super-peer information among all peers inside an overlay network.

3.3 URI based routing and signalling flow

Our proposal must be compatible with the assumptions made in the development of RELOAD. One of these assumptions relates to the fact that any hash function can be used in any overlay network. Therefore, our proposal based on Hierarchical-IDs will not work if different hash functions are used in each inner overlay network since resources in a remote overlay could not be found. In order to overcome this problem, we propose an URI based routing for queries belonging to other overlay networks. The Hierarchical-ID is replaced by the whole URI in Fetch operations. Thus, the appropriate Suffix-ID can be calculated by the super-peers in the destination overlay according to the hash function used in that overlay. This enhancement allows using different hash functions in each overlay maintaining the utility of Hierarchical-IDs. Furthermore, it is only necessary in queries between different overlay networks. Thereby, the original RELOAD requirement is preserved. An example of the associated signalling with the URI routing mechanism is presented in Fig. 4. First super-peers publish their information with a store operation in the Interconnection Overlay (message 1). In addition, peers also publish their information inside their own peer-to-peer networks (message 2). If a peer performs an inter-domain query, it sends the whole URI to its super-peer (message 3). Its super-peer calculates the Prefix-ID and requests the information from destination super-peer (message 4); the response contains the desired information (message 5). Request from peer in domain.b is forwarded to destination super-peer (message 3 is reused with an updated Via-List). Later, super-peer from destination peer-to-peer network generates the correct Suffix-ID according to the hash function used inside its peer-to-peer network and it launches the query inside its peer-to-peer network (message 6). If the item is found, it is forwarded

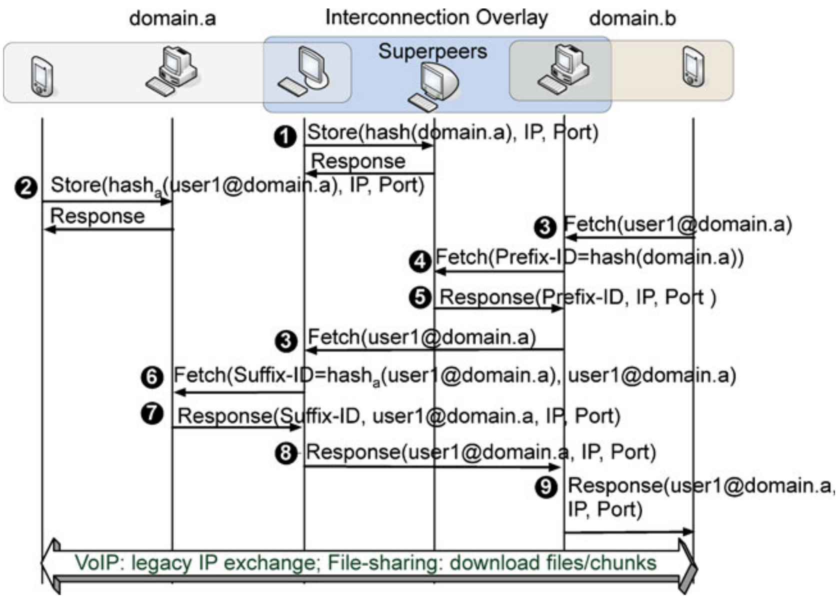

domain.a
Interconnection Overlay
domain.b
Superpeers
1 Store(hash(domain.a), IP, Port)
Response
2 Store(hash$_a$(user1@domain.a), IP, Port)
Response
3 Fetch(user1@domain.a)
4 Fetch(Prefix-ID=hash(domain.a))
5 Response(Prefix-ID, IP, Port)
3 Fetch(user1@domain.a)
6 Fetch(Suffix-ID=hash$_a$(user1@domain.a), user1@domain.a)
7 Response(Suffix-ID, user1@domain.a, IP, Port)
8 Response(user1@domain.a, IP, Port)
9 Response(user1@domain.a, IP, Port)
VoIP: legacy IP exchange; File-sharing: download files/chunks

Fig. 4 H-P2PSIP signalling

to the destination through super-peers making use of the Via-List (messages 7, 8 and 9).

3.4 Characteristics of H-P2PSIP

Considering previous works presented in Section 2, our proposal might be considered similar to some of them to some extent, however there exist important differences. With respect to Cyclone [1], our proposal only needs super-peers to maintain the Interconnection Overlay, while every peer has to do it in Cyclone. Moreover, query routing in our proposal is more efficient with respect to Cyclone since the destination overlay network is reached directly through the Interconnection Overlay without visiting intermediate overlays. Rings [38] is more similar to our proposal but it can only support one kind of overlay at the same time. However, our proposal can support any desired overlay network, which gives a great flexibility for the design of any desired service. Furthermore, it has been proved analytically that our proposal has also good scalability properties [28, 31]. We suppose a logarithmic complexity of the necessary resources with respect to the number of participating nodes according to the most used peer-to-peer networks (Chord and Kademlia). The size of overlay routing tables in peers participating in the hierarchical overlay is $O(\log_B M)$, where M is the number of peers in that overlay. On the other hand, super-peers have to maintain two overlay routing tables: an overlay routing table of size $O(\log_B M)$ for their own overlay and an overlay routing table of size $O(\log_B K)$ (K is then number of overlays participating in the hierarchical overlay architecture) for the Interconnection Overlay. Thus, super-peers maintain $O(\log_B(M \cdot K))$ overlay routing entries in average. This size is the same one that peers should support in a flat overlay counterpart. Thereby, the only real drawback of super-peers is the bandwidth and CPU consumption to forward and process queries that belong to other overlays, since the bigger overlay routing table would be also necessary in a flat overlay network.

4 Implementation and testbed setup

4.1 Implementation

When this work started, RELOAD protocol was in an early stage but its main functionalities and requirements had already been defined. Thus, there was not any available implementation. In order to overcome this problem, instead of using RELOAD, we use the Peer-to-Peer Protocol (P2PP[6]) [3], which is one of the precursors of RELOAD and many of its features are included in RELOAD. Hence, P2PP offers a good trade-off between not starting the implementation from scratch and a proof of concept of our proposal with a protocol with many similarities with respect to RE-LOAD. The available P2PP protocol implementation has available Chord, Kademlia and Bamboo peer-to-peer networks. We have performed the necessary modifications to use Kademlia in our H-P2PSIP im-plementation. This selection is based on the fact that our previous simulation studies were also based on Kademlia and we make a comparison with them later. Furthermore, the use of Kademlia in all inner overlay networks as well as in the Interconnection Overlay allows the comparison of our hierarchical proposal with respect to its flat counterpart, which is also interesting. Future work will also consider including Chord and Bamboo.

Several modifications were needed to adapt the available implementation to our convenience. The original design only supported legacy flat overlay networks. Thus, we added additional support to manage Hierarchical-IDs properly. This means including the fields of Prefix-ID and Suffix-ID and the necessary logic to manage them accurately in both Node-IDs and Resource-IDs. Furthermore, the routing of queries related with Prefix-ID from other overlays implies their forwarding to super-peers. Therefore, a super-peer role and a bootstrap based mechanism have been implemented. In order to provide the super-role, a multi-thread solution running two different instances of a Kademlia overlay network is used where the major problems for this design were the synchronization and data exchange between them. Indeed, this is a great difference with respect to simulation tools, which most of them are based on a single thread due to their simplifications and assumptions. The source code is available at http://hdl.handle.net/10016/8382.

4.2 Scenario setup

It is key to validate any proposal with a relevant and meaningful testbed. Thus, the selection of an appropriate scenario setup has the same relevance as the implementation itself. A summary of the setup is presented in Table 1, but all the parameters are detailed in the following paragraphs.

[6] http://www1.cs.columbia.edu/∼salman/peer/

Table 1 Testbed setup

Peer level	
Behaviour:	VoIP service
Queries:	Poisson distribution [20] Average: 10 queries per hour per peer
Overlay level	
Number of peers:	1,000
Overlay network:	Kademlia [36]

Churn rate:	Negative binomial distribution [43]
	r = 17
	p = 0.005

Peer geographical distribution:	See Table 2
Network level	
Type: Emulator:	Worldwide emulated network
Emulation level:	Modelnet [45] Links among different countries and access links
Emulation parameters:	PingER project [35]
Virtual machine level	
Virtual machine framework:	Proxmox 1.4
Modelnet virtual machines:	4 FreeBSD VMs
OpenVZ containers:	20 Linux VMs
Hardware level	
Number of machines:	6
CPU:	Intel core 2 quad Q9550
RAM:	12 GB
Network interface:	Gigabit Ethernet (GbE)
Connectivity:	Switch GbE
Experiment level	
Number of repetitions:	Equal or greater than 10
Statistical validation:	95% Confidence Intervals
Confidence interval error:	Most of cases less than 10%

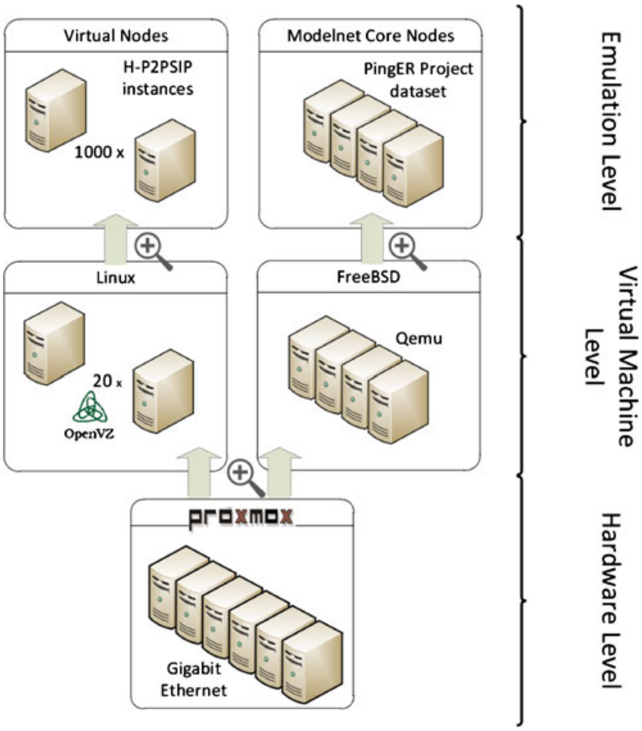


Fig. 5 Experiments infrastructure

4.2.1 Equipment setup

6 machines with Intel Core 2 Quad Q9550 CPU and 12 GB of RAM compose the hardware equipment in conjunction with a GbE switch. High speed GbE LAN is used to avoid bottlenecks and unexpected results in the exchange of traffic between the different machines. Over this infrastructure, Proxmox VE 1.4[7] in cluster configuration is installed in order to provide a Virtual Machine (VM) infrastructure to create the desire scenario and take advantage of all the available resources. We need a VM infrastructure since the emulation software, Modelnet [45], only runs on 32 bits machines, which cannot take advantage of all available RAM memory. In these experiments, 24 VMs are used in order to have enough scalability considering our hardware constraints. The first 20 VMs are OpenVZ[8] VMs responsible for hosting the Modelnet virtual nodes where our application is executed. The other 4 VMs are qemu[9] VMs where the Modelnet software is installed to emulate a global worldwide network topology. A scheme of the used infrastructure is given in Fig. 5.

4.2.2 Modelnet setup

Modelnet is an emulation software that helps to recreate real networking conditions based on the setup of networking attributes among different links. These attributes are bandwidth, delay, error probability or packet drop probability. The information provided by the IEPM PingER project [35] is used as input for the Modelnet software to emulate the main core links between different countries as well as their characteristics and properties. Indeed, PingER provides most of the attributes managed by Modelnet, but a data format translation is necessary. Thus, these values are introduced in Modelnet to have a reasonable representation of the current Internet. This setup allows the validation of the developed implementation in a scenario closer to

[7] http://www.proxmox.com/

[8] http://wiki.openvz.org

[9] http://wiki.qemu.org/

the real Internet with a high number of nodes, which is something quite difficult to achieve.

Modelnet has two types of nodes: core nodes and edge nodes. Core nodes perform the emulation of the network that interconnects the edge nodes, which run our implementation. We use 4 core nodes and 1,000 edge nodes in our scenario. A peer for our experiments runs in each edge node and 50 edge nodes run in each OpenVZ VM as completely independent instances. This configuration was selected after different performance and scalability benchmarks in order to avoid any possible bottleneck; the CPU was the limiting parameter.

4.2.3 Hierarchical overlay network setup

Modelnet provides the underlying infrastructure of our scenario. The next step is to create the infrastructure of our hierarchical overlay network in a representative way. Our underlying network provided by Modelnet and the PingER project gives us a global vision of the Internet across a set of different countries from all continents around the world. Thereby, this must be taken into account in our study. We have considered different setups to study the locality effects of the nodes placement. The experiments are based on 1, 5, 10, 20, 30 and 40 interconnected overlay networks. Furthermore, four different scenarios have been considered taking into account the distribution of peers in the different overlay networks as well as their geographical point of attachment. These scenarios are summarised in Table 2. We have the scenario setup that fits with our *analytical model assumptions* [28, 31]. This fact means that all inner overlays have the same size and nodes are placed randomly over the available countries from PingER project. The second scenario setup considers *local overlays*. This scenario has also inner overlays with the same size but all nodes in an overlay are in the same country. We want to quantify the locality effects associated to this setup, but a fair comparison is necessary. The used methodology is as follows. The same countries used for the *local overlays* setup are used for the *analytical model assumptions* setup. This fact means that one country is considered for one inner overlay, five countries for five inner overlays, ten countries for ten overlays and in this way consecutively. This mechanism ensures the same under-layer network conditions for both setups in all cases. The selected countries for these experiments are those ones with a greater number of Internet users according to the reports [13, 22]. Additionally, we have other two remaining setups. The *different size overlays setup* aims to study a more realistic scenario where inner overlays have different sizes. Thus, a more general scenario is considered removing one of our previous assumptions. Moreover, we have a *local different size overlays setup* where all nodes in an overlay are attached to the same country. The same methodology used in the first two setups is used to choose the countries for these setups. Furthermore, the size of the inner overlays in the local different size overlays setup is proportional to the number of Internet users according to the reports previously considered. This is imposed by the fact that all nodes in an overlay must be attached to the network of the same country. Consequently, these same sizes have been used later for the *different size overlays setup* but with a random placement among the countries. We consider this procedure the most appropriate to obtain fair meaningful comparisons.

4.2.4 Peers setup

After defining how the peers are distributed in the hierarchical overlay network, it is necessary to define the behaviour and characteristics of these peers. All peers run a Kademlia peer-to-peer network [36]. The session time distributions and churn rate of peers are configured according to the measurements in [43]. However, we do not take exactly these values; we adapt them proportionally to our population of 1,000 peers. Thus, churn rate of peers is a negative binomial distribution with an average scaled to the number of peers in our experiment. We suppose that peers are participating in a VoIP service so their store their location in formation in their own inner Kademlia network. Therefore, a query generation rate following a Poisson distribution (the typical one used in telephony) with average of 3.3 queries per second is used, which leads to having approximately 10,000 queries in each experiment.

Moreover, the ρ_{ii} parameter is used to model the probability of performing a query inside the own overlay instead of any other external overlay network. Super-peers run Kademlia both in the inner overlay and

Table 2 Different scenarios setups

		Nodes placement	
		Random	Same country
Size of overlays	Equal size	Analytical model assumptions	Local overlays
	Different size	Different size overlays	Local different size overlays

in the Interconnection Overlay. However, super-peers are not exposed to churn. We make this simplification for several reasons. In a SP based VoIP scenario, it is reasonable that the operators provide high redundant available super-peers since they want to control all outgoing signalling with charging purposes. There are already several mechanisms for the selection and management of super-peers where resilience and performance are considered, see Section 2. Finally, the main objective of this work is to validate that H-P2PSIP is feasible under reasonable and realistic conditions. We consider that the assumption of an operator based VoIP scenario is plausible since this is one of the main reasons for the creation of RELOAD and the P2PSIP group in the IETF. Thus, we postpone super-peer management in community networks scenarios for future work.

4.3 Experiments setup

Each experiment for any configuration of the previous setups is divided in two phases. Firstly, a transitory phase limited to 30 minutes is performed. During this period of time, the average number of peers needed to realise the experiment joins the hierarchical overlay network. Later, we have the second phase, which corresponds to the stationary state and it has duration of 60 minutes. During this steady state, the negative binomial distribution for join and departure rates of peers is used as well as the Poisson distribution for the query generation rate of peers. This last phase is used to collect the results that are shown in the next section. Our experiments are repeated several times (at least 10 times) in order to obtain results with representative 95% confidence intervals and a smaller than 10% error, in most cases, with respect to the estimated average of the results. The recollection of all the results took close to a couple of months.

5 Results

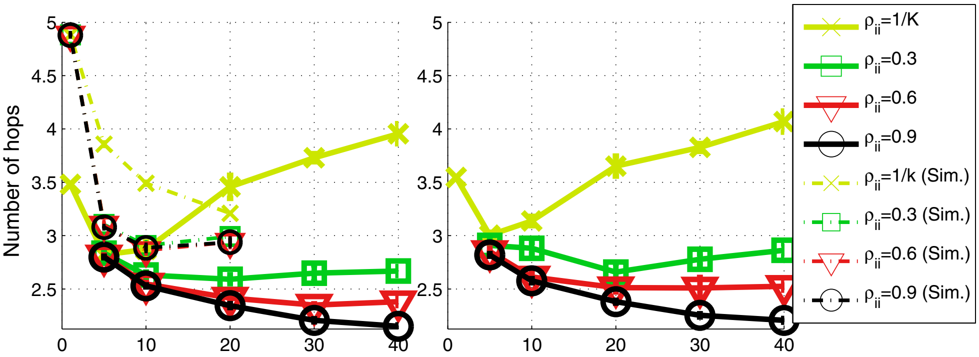
This section presents the results obtained for the different proposed configuration setups. Preliminary work was published in [33] with a subset of the results related to the *Analytical model assumptions* setup (see Section 4.2.3). This section completes those results and also considers new other three more realistic scenarios. All figures present the parameter under study in the y-axis and the different number of overlay networks (K) in the x-axis. All figures present the results for different values of ρ_{ii}, which represents the probability of making a query inside the own peer-to-peer network of a peer. Furthermore, subfigure (a) usually corresponds to those setups where peers are randomly placed in some of the available countries and subfigure (b) with the setups where peers of an overlay network are in the same country.

5.1 Equal size overlay networks setups

We present in this section the results obtained when all overlay networks have the same number of peers. In addition to the obtained results, we also present our previous results based on a simulation tool [28, 31] to compare both kinds of results. Implementation results are shown in continuous lines and simulation results are depicted with dashed lines.

5.1.1 Overlay routing performance

Figure 6a shows the number of hops made to reach the destination when peers are randomly distributed between the different available countries for each number of available inner overlay networks. An important result is the fact that the expected number of hops to reach the destination in a query is close and slightly



(a) Analytical model assumptions

(b) Local overlays

Fig. 6 Average number of hops

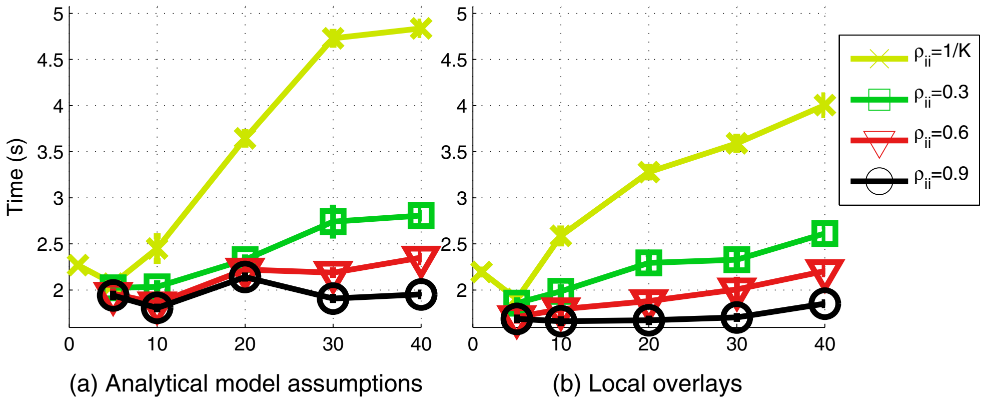
better with respect to our previous simulations [28, 31]. We explain this behaviour later. In addition, if queries are randomly distributed among all peers in the hierarchical overlay architecture ($\rho_{ii} = 1/K$—continuous line with cross markers), we observe how the num-ber of hops starts to be 20% smaller in comparison with the flat counterpart (first point of the plot that corresponds to $K = 1$). Furthermore, if the number of overlay networks increases (K), the number of hops increases slightly but very close with respect to the flat counterpart routing performance. This increment is produced because if K increases, $\rho_{ii} = 1/K$ decreases and consequently the probability of looking in other overlays increases. Thus, the average number of hops increases since queries to other overlays need extra hops to reach the super-peer and cross the Interconnection Overlay). On the other hand, if ρ_{ii} increases ($\rho_{ii} > 1/K$), the number of hops decreases since extra hops to reach other overlay networks are unnecessary. Therefore, the greater ρ_{ii}, the smaller the impact of increasing K in the routing performance. Additionally, we can see in Fig. 6b the number of hops that are needed to reach the destination if all peers of an overlay are in the same country. We observe that the plot is very similar to Fig. 6a. This result agrees with the fact that the overlay topology is built with independence of the underlying network topology.

5.1.2 Delay

Figure 7 shows the average delay suffered by queries in both previously mentioned setups. A certain correlation exists between these results and the results in Fig. 6. This fact makes sense since a greater number of hops usually leads to a higher delay, but not always. A small delay is observed if the number of overlay networks is also small. This result is produced because of the used methodology to create the different setups (Section 4.2.3). The same number of countries than existing overlay networks is used in each experiment. Thus, if the number of overlay networks increases, more countries are available and some of them have bad connectivity properties with a high population such as India. Therefore, the delay increases more than expected since it is not only proportional to the number of hops, it depends also on the underlying network as in a real environment. Furthermore, if $\rho_{ii} > 1/K$, the average delay is reduced considerably since the number of queries to other overlay networks is reduced and fewer hops are needed, which implies a smaller delay. If we compare the delay in the *local overlays* setup (Fig. 7b) with respect to the *analytical model assump-tions* setup (Fig. 7a), we observe certain differences motivated by the different underlying network topology in each scenario. Delay in overlays with peers in the same country is smaller as expected and especially for $\rho_{ii} = 1/K$ (case with a higher number of queries among different domains).

5.1.3 Overlay routing state

Not only overlay routing performance and delay are important parameters, it is also interesting to study the load sustained by peers to maintain the overlay network in order to route the queries correctly. The size of overlay routing tables in peers is a good approximation to the effort undergone by them. A bigger number of overlay routing entries implies higher memory, CPU and bandwidth consumption to maintain these entries fresh. The number of overlay routing entries of each peer has been periodically collected during the experiment and the average of these values is showed in Fig. 8 both for peers and super-peers. Results for *local overlays* setup and *analytical model assumption* setup are very similar. Thus, only the results from *analytical model assumption* setup are displayed to compare them



(a) Analytical model assumptions

(b) Local overlays

Fig. 7 Average delay

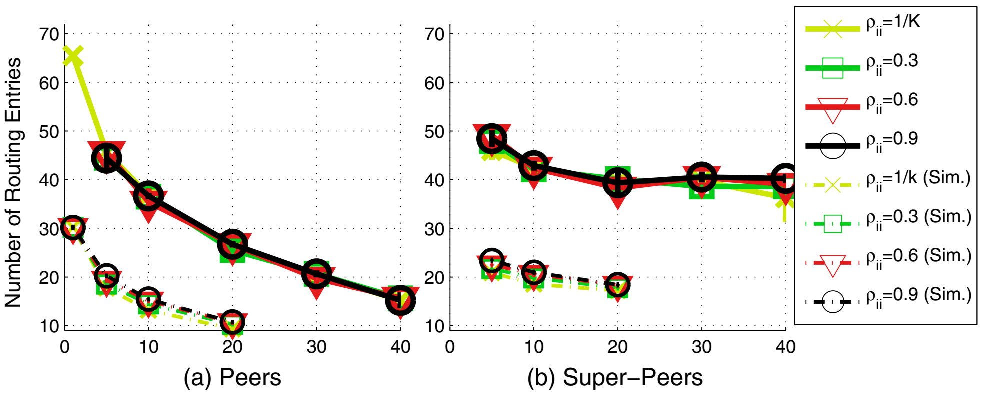


Fig. 8 Average number of entries in the routing tables

with the results of our simulation tool. Figure 8a shows the average number of routing entries in peers. We observe how the number of overlay routing entries decreases if the number of overlays increases. For $K =$ 5, this architecture saves more than 20% of the routing entries while for $K = 40$ this profit grows close to 75%. This effect is expected due to the limitations of our testbed, we have to maintain a constant number of peers in our experiments independently of the number of overlay networks considered. A bigger number of overlays implies fewer peers per overlay and smaller routing tables in peers. Additionally, we can see in Fig. 8a how the number of overlay routing entries is bigger than the values obtained in our previous simu-lations. Therefore, Kademlia implementation in P2PP stores in average a higher number of overlay routing entries than the Kademlia protocol in the simulator. However, both of them are valid since their values are among the theoretically expected results given in [36]. Thus, slight differences exist between the simulated and implemented overlay (bucket management, handling of events, etc.). Indeed, this fact explains why the num-ber of hops is smaller with respect to the simulation results (Fig. 6). More populated overlay routing tables in peers allow finding destination peers in fewer hops. Finally, we can also observe how ρ_{ii} has a negligible effect over the number of overlay entries because the routing tables must be maintained independently of this parameter and they depend on the number of peers. Figure 8b illustrates the overlay routing state information of super-peers. We see the average number of overlay routing entries taking into account both routing tables: own overlay network and Interconnection Overlay. The results are very similar to the ones obtained for peers but slightly bigger because of the additional entries from the Interconnection Overlay. In this case, the overlay routing state in super-peers is around 15% (instead of 20% of peers) smaller with respect to the overlay routing state of peers in a flat overlay.

These results validate that the hierarchical overlay architecture is more scalable than its flat counterpart. Furthermore, we can conclude that the obtained results from simulations are a reasonable approximation with respect to the results obtained in these experiments.

5.1.4 Overlay traffic

It is also important to measure the traffic exchanged by peers and super-peers. We measured periodically the sent and received traffic for each peer participating in the experiment to later calculate the average of the obtained values. Figure 9a shows the traffic supported

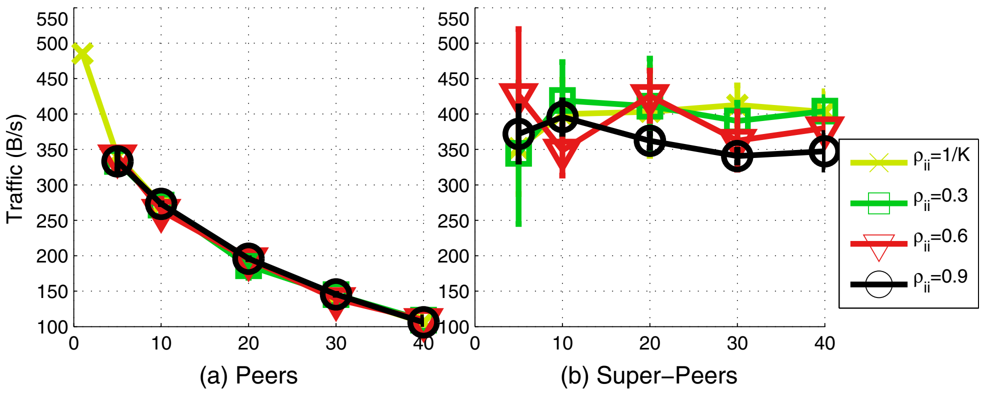


Fig. 9 Average supported traffic

by peers inside their overlay network; we see how the traffic decreases when the number of overlays grows. In this case, for $K = 5$ the traffic is around 25% less than in the flat counterpart, while this enhancement grows to 80% when we have 40 different overlay networks. This result is expected; we found a similar behaviour in previous results. If the number of overlays increases, the number of peers per overlay decreases. Thus, less traffic for maintenance operations and queries is necessary. In Fig. 9b, we see the traffic supported by super-peers. This traffic is larger than traffic supported by peers (Fig. 9a) because they also have to support the traffic associated to the Interconnection Overlay. Additionally, they need to forward all queries among the different domains. However, we find how the traffic supported by super-peers is 20% smaller rather than peers in its flat equivalent overlay counterpart ($K = 1$ in Fig. 9a), which reveals the better scalability properties of H-P2PSIP. The confidence intervals for small number of overlays are larger in comparison with respect to all previous results. The number of super-peers is small and important asymmetries in the traffic supported by super-peers are found. Thus, a higher variance exists in the results, which makes the intervals bigger.

5.2 Different size overlay networks setups

This section presents more realistic scenarios where different number of peers exists in each overlay network, further details can be found in Section 4.2.3.

5.2.1 Overlay routing performance

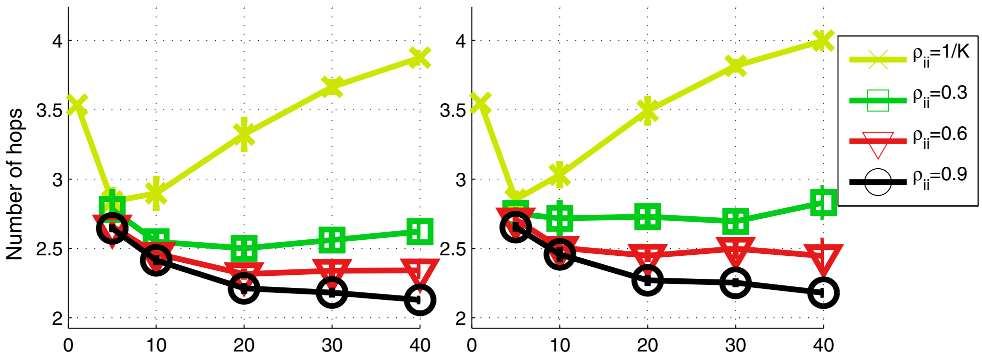
If we compare Fig. 10 with Fig. 6, we find that results are quite similar. However, results for overlays with different size are better (slightly smaller) than the scenario with overlays of the same size. More differences were to be expected, but we find how the larger number of hops in bigger overlays is compensated by those overlays with a small number of peers, which need fewer hops. Nevertheless, these results do not imply that overlays with different sizes will always produce a better performance in our architecture. It only means that the sizes used in this setup give better results with respect to the equal size setup. We can also see how there is no difference in the results with respect to the placement of peers as it was observed in Fig. 6.

5.2.2 Delay

We can appreciate in Fig. 11 , like in previous setups (Fig. 7), how the *Local different size overlays* setup has smaller delays than the *Different size overlays* setup. This behaviour is produced by the underlying network topology, which is favourable for the *Local different size overlay* setup. However, there exist differences between them. We can observe how the experienced delay for different size overlay network (Fig. 11) is slightly bigger than the obtained delay for equal size overlay networks (Fig. 7), especially for experiments with 20 and 30 overlay networks. Part of this effect is produced by the fact that some of the countries introduced in the positions 11 to 20 have big delays inside their own networks or with their neighbours (e.g. South Africa and India). Therefore, their penalty effect is distributed in the case where all peers are randomly distributed between equally sized overlays and countries; this is Fig. 7a. However, this fact does not happen in Fig. 11a where a country like India has worse quality links but its population is very large. Thus, a negative impact on the experienced delay is obtained.

5.2.3 Overlay routing state

We must also check the effort made by a peer to be part of the hierarchical overlay architecture for the distribution of peer-to-peer networks under study. The



(a) Different size overlays

(b) Local different size overlays

Fig. 10 Average number of hops

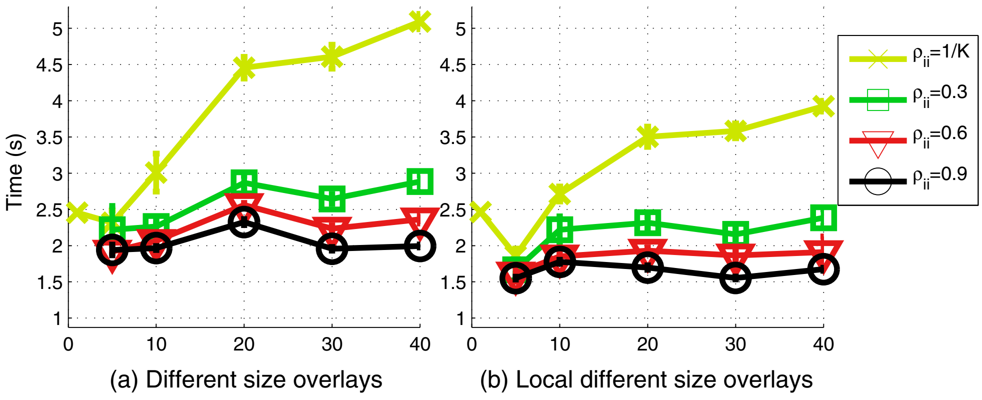


Fig. 11 Average delay

first measure taken is in relation to the average number of entries in the overlay routing tables of peers and super-peers; Fig. 12 shows these results. We can see how the average number of entries decreases for peers when the number of overlays grows up with independence of ρ_{ii} as expected. Nevertheless, if we compare results in Fig. 12 with Fig. 8, we find some differences. On the one hand, the number of overlay routing entries is higher in peers in the scenarios considering overlays with different sizes (Fig. 12a with respect to Fig. 8a). This fact is due to the existence of bigger overlays, and as a consequence, greater overlay routing tables are necessary. Peers in smaller overlays need smaller overlay routing entries but in average they cannot compensate the effort by peers in the bigger ones. On the other hand, the overlay routing state in super-peers is different (Fig. 12b). We find also how the number of routing entries is smaller with respect to peers in a flat overlay network ($K = 1$ in Fig. 12a) but not always bigger than the number of overlay routing entries for peers in the same experiment (see points for $K = 5, 10, 20$). In order to figure out the reason for this behaviour, we plot the size of the two different overlay routing tables in super-peers. If we compare Fig. 12a with Fig. 13a, we find how the number of overlay routing entries associated to the inner overlay is bigger in peers than in super-peers for experiments with a smaller number of overlays. This result is counterintuitive, since they should be similar. In order to explain this, we must go deep inside the behaviour of the Kademlia protocol. In our implementation, super-peers do not perform queries but Kademlia uses queries to populate and refresh the overlay routing tables inside its iterative lookup process. Therefore, our super-peers only populate the inner overlay routing table with a long-term stabilization mechanism and their inner overlay routing table is underpopulated for this reason (further details in [36]). This fact especially happens in scenarios with small number of overlays where overlay networks are bigger and especially if we have some overlays bigger than other ones. This last fact populates the overlay routing tables in the legacy peers due to the larger number of queries and neighbours inside their own overlay, but this effect does not occur in super-peers. Finally, we find in Fig. 13 how the size of the overlay routing table of the inner overlays decreases in a slower proportion with respect to the increase of the size in the Interconnection Overlay routing table. Thus, the overall number of routing entries in super-peers increases for a high number of supported overlays, but

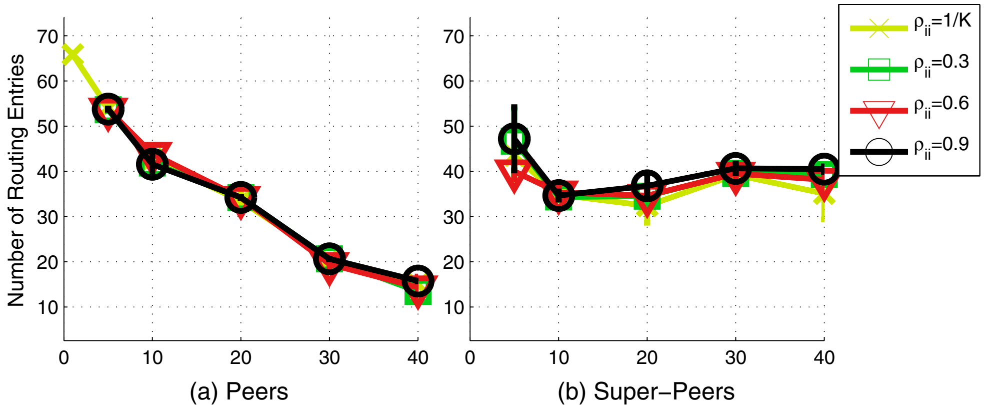


Fig. 12 Average number of entries in the routing tables

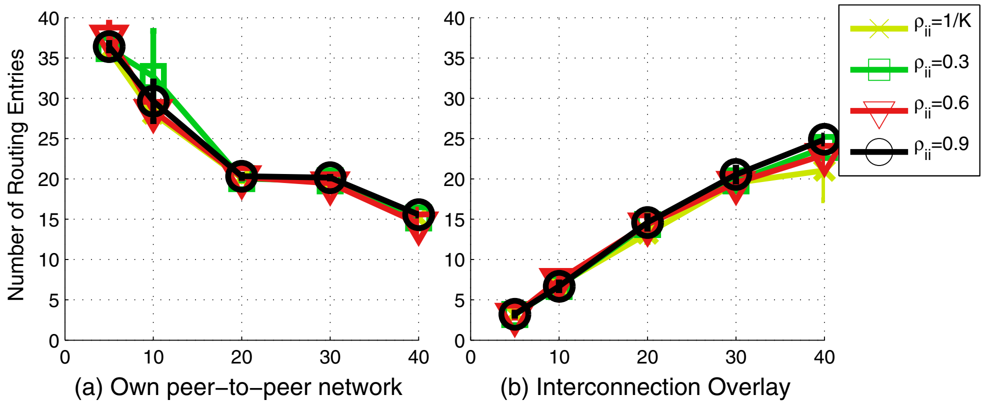


Fig. 13 Average number of entries in super-peer routing tables

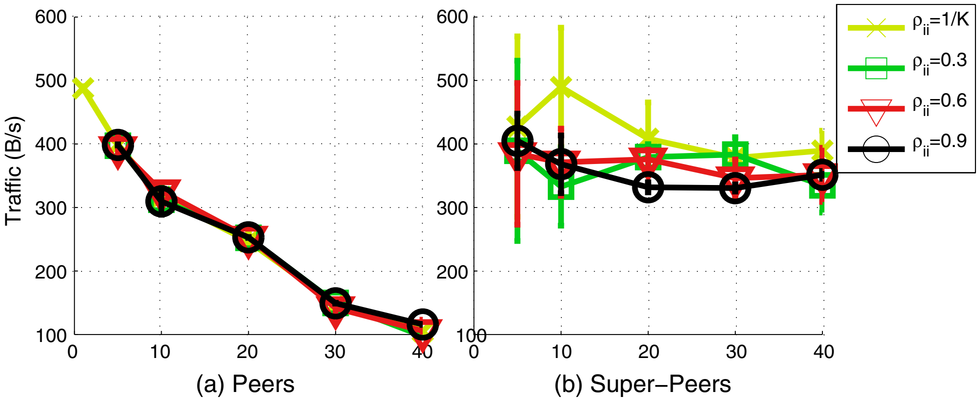


Fig. 14 Average supported traffic

they are smaller with respect to the equivalent flat counterpart.

5.2.4 Overlay traffic

Figure 14 shows peers and super-peers traffic consumption under the study scenario. Traffic in peers decreases when the number of overlays increases. On the other hand, traffic in super-peers remains constant but having big confidence intervals (the explanation is the same one given in Section 5.1.4). If we compare these results with the other ones in the Fig. 9, we can see how the traffic in peers is higher when the peers are not equally distributed among the different overlays. Larger overlays imply higher maintenance traffic and small overlays cannot compensate the traffic in big ones.

6 Lessons learnt after the implementation and verification process

After the implementation and validation of the H-P2PSIP proposal, we want to share some of the knowledge obtained through this long process.

The definition of a good scenario setup to obtain meaningful results is a very complete task. It is impossible to make a validation in a complete real scenario, which can completely represent current Internet as well as the system behaviour, its parameters and conditions. Furthermore, in order to allow the reproducibility of the results and help the research community to properly evaluate the validation, a complete full parameterization of the experiments is necessary. This is a very hard task to achieve. It would be easier if some flexible and standard mechanism would exist to perform this task. Some effort exists[10] [34], but its adoption by the research community is hard. We have defined thoroughly all our parameters and assumptions, and a summary is found in Tables 1 and 2.

Unfortunately, the definition of a good setup does not assure complete good results. If a bottleneck exists in some part of the scenario, the results will not be valid. Special attention must be paid to the aggregated traffic of the different virtual machines inside a physical computer or in the aggregated traffic of different physical computers over some specific destination or link. Indeed, this step is the reason of having a GbE network

[10] http://heim.ifi.uio.no/plageman/Site_3/Benchmarking.html

to interconnect the testbed instead of a Fast Ethernet network. Unpredicted queuing delays can lead to uncontrolled behaviours since unplanned timer timeouts in any layer can seriously modify the final results. Thus, it is very important to check if any unpredicted bottleneck is affecting the experiments.

In small environments, ad-hoc configurations are possible. Nevertheless, if large-scale experiments are being designed, it is important to automate all possible procedures. We have an ad-hoc XML parsing tool that extracts all the necessary actions to configure the desired setup from a XML file.

Finally, a single realization of an experiment is meaningless and it is absolutely necessary to make several replications of an experiment to assure the validity of the results. In our case, we calculate the corresponding confidence intervals of the different results to proof their validity.

7 Conclusions

This paper presents the validation through a real implementation of H-P2PSIP that allows the interoperability among different peer-to-peer networks. This design is based on the ongoing work on the RELOAD protocol [24] in the IETF P2PSIP WG while the implementation is based on one of its prequels, the P2PP protocol [3]. The experiments to perform this validation are based on different scenario setups with 1,000 peers. These scenario setups are based on the Modelnet emulation framework, which is used to create an emulated worldwide topology based on the data provided by the PingER project. The peers are distributed in the underlying topology according to several policies and different overlay networks sizes according to Internet population in different countries. This helps to study the locality effect of overlays where their peers are attached to the same country; improvements, especially in the delay, are observed due to the favourable underlying network. The obtained results demonstrate the ability to interconnect different overlay networks making use of an Interconnection Overlay maintained by super-peers. With independence of its interoperability properties, the scalability properties of the proposal can be confirmed. Savings of a 20% in the routing performance and 40% in the overlay routing entries are obtained with respect to its equivalent flat counterpart. Furthermore, super-peers, which represent the worst case, only consume 80% of the bandwidth with respect to peer in the flat overlay network. Hence, we can say that our solution is scalable. Consequently, the interconnection capabilities and scalability properties as well as the performance of H-P2PSIP are demonstrated through the presented thoughtful experiments. Moreover, we can say that the simulation tools give reasonable results and can be useful as a guidance to estimate some of the aspects of real peer-to-peer implementations.

8 Future work

After the detailed emulation experiments presented in this paper, further work can be realised in relation to this proposal. Our work is based on a hierarchical Kademlia overlay network in order to perform a fair comparison with a Kademlia flat overlay network. However, H-P2PSIP allows the use of any peer-to-peer network if necessary. Thus, a validation with different peer-to-peer networks will increase the understanding of the behaviour of our proposal and the comparison of different peer-to-peer networks under similar conditions. This work is also based on the assumption of a VoIP operator based scenario where a high availability of super-peers is guaranteed. However, there are scenarios where this fact is not feasible, i.e. community networks. Therefore, super-peer management strategies for structured peer-to-peer networks around H-P2PSIP are also an interesting future research line.

Acknowledgements We would like to acknowledge the anonymous reviewers for their insightful comments and Felix Gomez-Fernandez for his collaboration in this work. This research was supported in part by the European Commission Seventh Framework Programme under grant agreement n 25774 (TREND Network of Excellence), Comunidad de Madrid grant S-2009/TIC-1468 (MEDIANET project) and Spanish MICINN grant TEC2011-29688-C02-02 (eeCONTENT project).

References

1. Artigas M, Lopez P, Ahullo J, Skarmeta A (2005) Cyclone: a novel design schema for hierarchical dhts. In: Fifth IEEE international conference on peer-to-peer computing, 2005, P2P 2005, pp 49–56
2. Banerjee S, Griffin T, Pias M (2004) The interdomain connectivity of Planetlab nodes. In: Passive and active network measurement, pp 73–82
3. Baset S, Schulzrinne H, Matuszewski M (2007) Peer-to-peer protocol (P2PP). Internet draft draft-baset-p2psip-p2pp-01.txt
4. Bauer K, Sherr M, McCoy D, Grunwald D (2011) Experimentor: a testbed for safe and realistic tor experimentation. In: Proceedings of the 4th conference on cyber security experimentation and test, CSET'11. USENIX Association, Berkeley, CA, USA, pp 7–7. http://dl.acm.org/citation.cfm?id=2027999.2028006

5. Bryan D, Matthews P, Shim E, Willis D, Dawkins S (2010) Concepts and terminology for peer to peer sip. Internet draft draft-ietf-p2psip-concepts-03.txt
6. Castro M, Druschel P, Kermarrec AM, Rowstron A (2002) Scribe: a large-scale and decentralized application-level multicast infrastructure. IEEE JSAC 20(8):1489–1499
7. Castro M, Druschel P, Kermarrec AM, Rowstron A (2003) Scalable application-level anycast for highly dynamic groups. In: Group communications and charges. Lecture notes in computer science, vol 2816. Springer, Berlin, pp 47–57
8. Chen X, Chu X, Liu J (2010) A lightweight emulator for bittorrent-like file sharing systems. In: 2010 IEEE International Conference on Communications (ICC), pp 1–5. doi:10.1109/ICC.2010.5502796
9. Cheng L (2007) Bridging distributed hash tables in wireless ad-hoc networks. In: Global telecommunications conference, 2007, GLOBECOM '07. IEEE, pp 5159–5163
10. Cheng L, Ocampo R, Jean K, Galis A, Simon C, Szabo R, Kersch P, Giaffreda R (2006) Towards distributed hash tables (de)composition in ambient networks. In: Large scale management of distributed systems. Lecture notes in computer science, vol 4269. Springer, Berlin, pp 258–268
11. Chun B, Chun B, Spalink T (2003) Slice creation and management. Tech. rep., Intel Research Berkeley and Princeton University
12. Chun B, Culler D, Roscoe T, Bavier A, Peterson L, Wawrzoniak M, Bowman M (2003) Planetlab: an overlay testbed for broad-coverage services. SIGCOMM Comput Commun Rev 33(3):3–12
13. C.I.A. (2012) Country comparison: internet users. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html
14. Douceur J (2002) The sybil attack. In: Druschel P, Kaashoek F, Rowstron A (eds) Peer-to-peer systems. Lecture notes in computer science, vol 2429. Springer, Berlin, pp 251–260
15. Erman D, Ilie D, Popescu A (2009) Bittorrent session characteristics and models. River Publishers Series in Information Science and Technology. Special Issue in Traffic Engineering, Performance Evaluation Studies and Tools for Heterogeneous Networks 1:61–84
16. Fonseca P, Rodrigues R, Gupta A, Liskov B (2009) Full-information lookups for peer-to-peer overlays. IEEE TPDS 20(9):1339–1351. doi:10.1109/TPDS.2008.222
17. Furtado P (2007) Multiple dynamic overlay communities and inter-space routing. In: Databases, information systems, and peer-to-peer computing. Lecture notes in computer science, vol 4125. Springer, Berlin, pp 38–49
18. Ganesan P, Gummadi K, Garcia-Molina H (2004) Canon in G major: designing DHTs with hierarchical structure. In: Proceedings of the 24th international conference on distributed computing systems (ICDCS'04) (ICDCS '04). IEEE computer society. Washington, DC, USA, pp 263–272
19. Garces-Erice L, Biersack EW, Ross KW, Felber PA, Urvoy-Keller G (2004) Hierarchical P2P systems. In: Euro-Par 2003 parallel processing, vol 2790, Springer, pp 1230–1239
20. Hersent O, Petit J, Gurle D (2005) Beyond VoIP protocols: understanding voice technology and networking techniques for IP telephony. Wiley. http://books.google.com/books?id=ceXyTuXhjyMC
21. Hibler M, Ricci R, Stoller L, Duerig J, Guruprasad S, Stack T, Webb K, Lepreau J (2008) Large-scale virtualization in the emulab network testbed. In: USENIX annual technical conference. USENIX Association, pp 113–128
22. Internet usage statistics (2012) http://www.internetworldstats.com/
23. Jennings C, Lowekamp B, Rescorla E, Baset S, Schulzrinne H (2011) A sip usage for reload. Internet draft draft-ietf-p2psip-sip-06.txt
24. Jennings C, Lowekamp BB, Rescorla E, Baset S, Schulzrinne H (2011) Resource location and discovery (reload) base protocol. Internet draft draft-ietf-p2psip-base-16.txt
25. Ko SK, Kim YH, Oh SH, Lee BT, Avila VP (2009) Iptv usage for reload. Internet draft draft-softgear-p2psip-iptv-02.txt
26. Lakshman A, Malik P (2010) Cassandra: a decentralized structured storage system. ACM SIGOPS Oper Syst Rev 44(2):35–40
27. Leonini L, Rivière E, Felber P (2009) Splay: distributed systems evaluation made simple (or how to turn ideas into live systems in a breeze). In: Proceedings of the 6th USENIX symposium on Networked Systems Design and Implementation, NSDI'09. USENIX Association, Berkeley, CA, USA, pp 185–198. http://dl.acm.org/citation.cfm?id=1558977.1558990
28. Martinez-Yelmo I, Bikfalvi A, Guerrero C, Cuevas R, Mauthe A (2008) Enabling global multimedia distributed services based on hierarchical dht overlay networks. IJIPT International Journal of Internet Protocol Technology 3(4):234–244
29. Martinez-Yelmo I, Bikfalvi A, Guerrero C, Cuevas R, Mauthe A (2008) Enabling global multimedia distributed services based on hierarchical dht overlay networks. In: The Second International Conference on Next Generation Mobile Applications, Services and Technologies, 2008, NGMAST '08, pp 543–549. doi:10.1109/NGMAST.2008.42
30. Martinez-Yelmo I, Cuevas R, Guerrero C, Mauthe A (2008) Routing performance in a hierarchical dht-based overlay network. In: 16th Euromicro conference on parallel, distributed and network-based processing, 2008, PDP 2008, pp 508–515. doi:10.1109/PDP.2008.79
31. Martinez-Yelmo I, Bikfalvi A, Cuevas R, Guerrero C, Garcia J (2009) H-P2PSIP: Interconnection of P2PSIP domains for global multimedia services based on a hierarchical dht overlay network. Comput Networks 53(4):556–568. Special Issue on Content Distribution Infrastructures for Community Networks
32. Martinez-Yelmo I, Guerrero C, Cuevas R, Mauthe A (2009) A hierarchical p2psip architecture to support skype-like services. In: 17th Euromicro international conference on parallel, distributed and network-based processing, 2009, pp 316–322. doi:10.1109/PDP.2009.27
33. Martinez-Yelmo I, Gonzalez R, Guerrero C (2010) Benefits of an implementation of h-p2psip. In: The second international conference on advances in P2P systems, (AP2PS 2010). IARA
34. Martinez-Yelmo I, Seoane I, Guerrero C (2010) Fair quality of experience (qoe) measurements related with networking technologies. In: Wired/wireless internet communications. Lecture notes in computer science, vol 6074. Springer, Berlin, pp 228–239
35. Matthews W, Cottrell L (2000) The PingER project: active internet performance monitoring for the henp community. Commun Mag 38(5):130–136
36. Maymounkov P, Mazieres D (2002) Peer-to-peer systems: first international workshop, IPTPS 2002, Cambridge, MA, USA, 7–8 Mar 2002. Revised papers, Lecture notes in computer science, vol 2429/2002, chap. Kademlia: a peer-to-peer information system based on the XOR metric. Springer, pp 53–65
37. Min SH, Holliday J, Cho DS (2006) Optimal super-peer selection for large-scale P2P system. In: International conference

on Hybrid Information Technology, 2006, ICHIT'06, vol 2, pp 588–593

38. Mislove A, Druschel P (2005) Providing administrative control and autonomy in structured peer-to-peer overlays. In: Peer-to-peer systems III, pp 162–172
39. Mizrak AT, Cheng Y, Kumar V, Savage S (2003) Structured superpeers: leveraging heterogeneity to provide constant-time lookup. In: Proceedings of internet applications, WIAPP 2003, pp 104–111
40. Rhea S, Chun B, Kubiatowicz J, Shenker S (2005) Fixing the embarrassing slowness of opendht on Planetlab. In: Proc. of the second USENIX workshop on real, large distributed systems, pp 25–30
41. Rosenberg J (2010) Interactive Connectivity Establishment (ICE): a protocol for Network Address Translator (NAT) traversal for offer/answer protocols. RFC 5245 (proposed standard)
42. Sanchez-Artigas M, Garcia-Lopez P, Skarmeta AFG (2008) On the feasibility of dynamic superpeer ratio maintenance. In: P2P '08: proceedings of the 2008 eighth international conference on peer-to-peer computing. IEEE Computer Society, Washington, DC, USA
43. Steiner M, En-Najjary T, Biersack EW (2007) A global view of kad. In: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on internet measurement. ACM, New York, NY, USA, pp 117–122
44. Stoica I, Morris R, Liben-Nowell D, Karger D, Kaashoek M, Dabek F, Balakrishnan H (2003) Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans Netw 11(1):17–32
45. Vahdat A, Yocum K, Walsh K, Mahadevan P, Kosti´c D , Chase J, Becker D (2002) Scalability and accuracy in a large-scale network emulator. SIGOPS Oper Syst Rev 36(SI):271–284
46. Vratonjic´N, Gupta P, Kneževi´c N , Kostic´D, Rowstron A (2007) Enabling dvd-like features in p2p video-on-demand systems. In: Proceedings of the 2007 workshop on peer-to-peer streaming and IP-TV, P2P-TV '07. ACM, New York, NY, USA, pp 329–334. doi:10.1145/1326320.1326326
47. Wang J, Shen J, Meng Y (2009) Content sharing usage for reload. Internet draft draft-shen-p2psip-content-sharing-00.txt
48. Xu Z, Min R, Hu Y (2003) Hieras: a dht based hierarchical P2P routing algorithm. In: International conference on parallel processing, 2003. Proceedings, pp 187–194
49. Zhou Z, Wang H, Zhou J, Tang L, Li K, Zheng W, Fang M (2006) Pigeon: a framework for testing peer-to-peer massively multiplayer online games over heterogeneous network. In: Consumer communications and networking conference, 2006, CCNC 2006. 3rd IEEE, vol 2, pp 1028–1032. doi:10.1109/CCNC.2006.1593194

Isaias Martinez-Yelmo holds a PhD in telematics from the University Carlos III of Madrid since March of 2010. He also holds a MSc in telematics as well as a MSc in telecommunications (both also expedited by University Carlos III of Madrid in 2007 and 2003, respectively). He is currently assistant professor in the Department of Automatics at the University of Alcala in Spain. His current research interests are Peer-to-Peer networks, Content Delivery Networks, Next Generation Networks, Wireless technologies and Internet Protocols. He has participated in several Spanish and European funded projects related with previously mentioned topics. He is member of the IEEE and ACM societies. He has also participated as referee in relevant conferences and journals (INFOCOM, Computer Networks, Computer Communications, IEEE Transactions on Computers, ACM Transactions on Multimedia Computing Communications and Applications, etc). Furthermore, he has been member of the Technical Program Committee of the 17th IEEE International Conference on Networks (ICON2011). More details are available at http://it.aut.uah.es/~isaias/

Roberto Gonzalez-Sanchez was born in Madrid, in 1984. He obtained his MSc in telematics engineering and MSc in telecommunications engineering at University Carlos III of Madrid (Spain) in 2011 and 2009, respectively. He is member of the Telematic Engineering Department and NETCOM research group since 2010, where currently he is Teaching Assistant and PhD student. He has been teaching during the last 2 years in different degrees (Telecommunication Systems, Audiovisual Systems, etc.).

Carmen Guerrero received her telecommunication engineering degree in 1994 from the Technical University of Madrid (UPM), Spain, and her Ph.D. in computer science in 1998 from the Universidade da Coruna (UDC), Spain. She is currently associate professor since 2003 in the Telematics Engineering Department at the Universidad Carlos III de Madrid. She has been involved in several national and international research projects related to green networking, future Internet, content distribution and Internet measurements. More details are available at http://www.it.uc3m.es/carmen.