Computer Networks 53 (2009) 556–568

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

H-P2PSIP: Interconnection of P2PSIP domains for global multimedia services based on a hierarchical DHT overlay network

Isaias Martinez-Yelmo [a,*], Alex Bikfalvi [b], Ruben Cuevas [a], Carmen Guerrero [a], Jaime Garcia [a]

[a] Universidad Carlos III de Madrid, Av Universidad 30, 28911 Leganés, Spain
[b] IMDEA Networks, Av del Mar Mediterráneo 22, 28918 Leganés, Spain

ARTICLE INFO

Article history:
Available online 21 November 2008

Keywords:
P2P
P2PSIP
Hierarchical overlay
Multimedia
VoIP

ABSTRACT

The IETF P2PSIP WG is currently standardising a protocol for distributed multimedia services combining the media session functionality of SIP and the decentralised distribution and localisation of resources in peer-to-peer networks. The current P2PSIP scenarios only consider the infrastructure for the connectivity inside a single domain. This paper proposes an extension of the current work to a hierarchical multi-domain scenario: a two level hierarchical peer-to-peer overlay architecture for the interconnection of different P2PSIP domains. The purpose is the creation of a global decentralised multimedia services in enterprises, ISPs or community networks. We present a study of the routing performance and routing state in the particular case of a two-level distributed hash table hierarchy that uses Kademlia. The study is supported by an analytical model and its validation by a peer-to-peer simulator.

1. Introduction

Nowadays, the provisioning of multimedia services is one of the most important objectives of ISPs in order to provide new and attractive value added services. However, due to their requirements, these services lack a wide deployment at the moment.

Although some applications like Skype[1] [1–3] are really successful, they are not easy to design and implement. In spite of the fact that the Session Initiation Protocol (SIP) [4] has been developed and standardised for this purpose, it has an important constraint: it depends on centralized infrastructures. This is a problem in some scenarios where it is not feasible to use a server-based infrastructure.

On the other hand, it is expected that in the near future, handheld devices will support more multimedia services. Considering that the number of multimedia terminals is expected to increase in a high proportion with respect to the total number of mobile devices, scalability problems could exist because a central entity would not be able to manage such a large number of terminals. However, it could be argued that the computational power of handheld devices is limited and they could not support P2PSIP. Nevertheless, if it is considered that year-by-year handheld devices increase their capabilities (according to Moore's Law), we can suppose that these devices would have the necessary capabilities to support the P2PSIP protocol. In any case, since the cost of computational power cannot be ignored and the resources needed for a terminal must be limited as much as possible, a decentralised architecture as lightweight as possible is necessary. Furthermore, there is another key point for the development of a decentralised architecture: the proliferation of community networks is imposing a solution that is easy to manage. This is difficult to achieve in a centralised topology for administrative reasons. By contrast, peer-to-peer overlay networks are flexible enough to support the dynamic environment of community networks without a central entity.

* Corresponding author. Tel.: +34 916248795; fax: +34 916248749.
E-mail addresses: imyelmo@it.uc3m.es (I. Martinez-Yelmo), alex.bikfalvi@imdea.org (A. Bikfalvi), rcuevas@it.uc3m.es (R. Cuevas), guerrero@it.uc3m.es (C. Guerrero), jgr@it.uc3m.es (J. Garcia).

[1] http://www.skype.com.

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.09.028

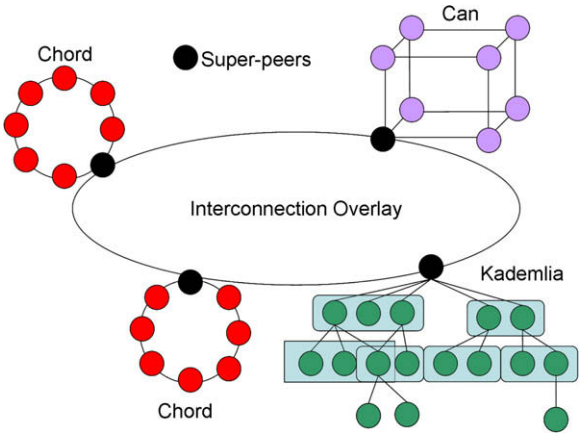


Fig. 1. Hierarchical overlay architecture.

Although there are many solutions to support decentralised multimedia services, the new approach of the IETF P2PSIP[2] working group is gaining supporters. P2PSIP [5] defines a peer-to-peer overlay-based solution that facilitates a decentralised architecture. It is expected to standardise a flexible protocol [6] able to support most of the existing peer-to-peer networks. The key concept of this solution is to provide a protocol that allows the implementation of any distributed hash table (DHT) overlay network like Kademlia [7], Chord [8] or Content Addressable Network (CAN) [9]. However, the design of this protocol does not consider yet the possibility of interconnecting different P2PSIP domains in order to provide services between them. Our proposal, called Hierarchical-P2PSIP (H-P2PSIP) and illustrated in Fig. 1, creates a hierarchical topology where different P2PSIP domains deploy their own overlay network and are interconnected through an interconnection overlay. The nodes forming this interconnection overlay are peers with extra capabilities from each domain, called super-peers. In order to have inter-domain connectivity, every domain must have at least one super-peer. When a peer searches for a resource (an item, service or reference) if the resource is not in the same domain, the peer performing the search will ask its super-peer to route the query to the appropriate P2PSIP domain. To support both inter and intra-domain P2PSIP routing, we use a Hierarchical ID formed by a Prefix ID for the routing in the interconnection overlay and a Suffix ID for the routing in each P2PSIP domain.

Some of the advantages of this architecture are the network isolation and the improved scalability that are intrinsic to the hierarchical architectures [10]. A potential drawback is the super-peer overload [11] in comparison with a flat topology.

In Section 2 this paper presents the details of the ongoing work on P2PSIP. Section 3 describes the hierarchical architecture for H-P2PSIP: the structure and the management of the Hierarchical ID, the data location and storage on the different peers, and the functionality of the P2PSIP protocol in the hierarchical scenario. In Section 4 we analyse the performance of H-P2PSIP with a mathematical model and in Section 6 we validate the model with experimental simulations. Related work is presented in Section 7 and the conclusions of this work are summarised in Section 8.

2. P2PSIP

2.1. General overview

The target of P2PSIP WG is to develop a peer-to-peer version of the SIP protocol called P2PSIP, which can use any DHT-based peer-to-peer network to locate resources, services and users in a decentralised way. The motivation of this work comes from the necessity of having a standard for developing Skype-like decentralised multimedia applications.

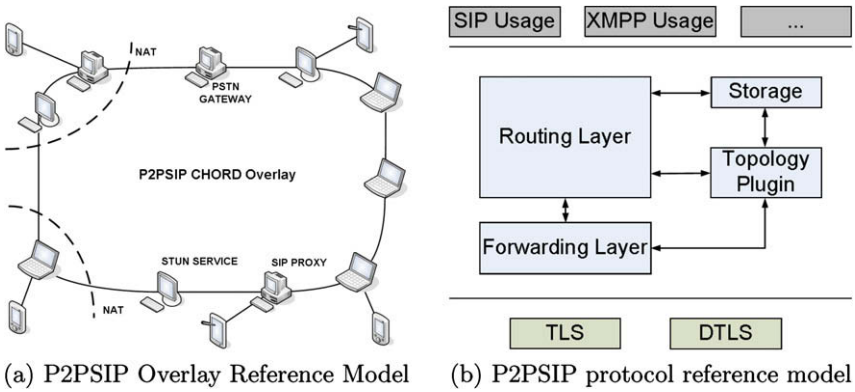
P2PSIP WG is chartered to develop protocols and mechanisms for the use of SIP in environments where the service of establishing and managing sessions is mainly handled by a collection of intelligent end-points, rather than centralised SIP servers. However, the scope of P2PSIP is not limited to a distributed replacement of SIP by overriding the proxy and registrar SIP servers, but it can also be used for other purposes (for example file sharing) or in combination with other signalling protocols.

Fig. 2a presents the P2PSIP overlay reference model using the basic concepts from [5]. P2PSIP protocol is designed to support any type of DHT-based network. Each deployed overlay network is identified by an overlay name and the participants in this architecture can support two profiles: peers and clients. Peers are active node participants in the overlay network and they are uniquely identified by a Node ID (e.g. the computers and laptops in Fig. 2a). On the other hand, clients are entities that use the resources offered by the peer-to-peer overlay network but they do not participate in the network maintenance. This role is reserved to and should be used only by devices with very limited capabilities, such as the handheld devices in Fig. 2a.

The information stored in the peer-to-peer network is made of resources records associated with the resources existing in the network. These resources are uniquely identified by a Resource ID and they can store services provided by peers identified by a Node ID. Because these peers and their services are usually identified by names in Uniform Resource Identifier (URI) format, we need to define a mechanism that maps the user and service URI to their ID. However, the details of this mapping depend on each implementation and are independent of the functionality offered by the P2PSIP protocol. In addition, the protocol must support the basic primitives of a peer-to-peer overlay network such as joining, bootstrapping, resource allocation and maintenance, while maintaining the connectivity between peers and clients (even in NAT scenarios). Finally, all these requirements increase the complexity of the solution.

To summarise, P2PSIP re-implements the proxy and registrar functionality of SIP in a decentralised fashion. The user and service information is distributed among all peers in the peer-to-peer overlay network, instead of stor-

[2] http://www.p2psip.org.



(a) P2PSIP Overlay Reference Model (b) P2PSIP protocol reference model

Fig. 2. P2PSIP reference models.

ing it in the registrar and proxy servers. The requests for this information are also handled by the overlay infrastructure. The advantages of P2PSIP include the elimination of the single points of failure (because of its decentralised nature) and reducing the costs as it does not require any dedicated equipment. For situations when interoperability between P2PSIP and conventional SIP entities [12] is needed, a proxy SIP service is used that is announced in all P2PSIP domains that support this service.

2.2. Ongoing design

Based on the requirements for the P2PSIP protocol presented previously, the RELOAD protocol [6] has been recently proposed as a working group draft. One of the most relevant decisions is the adoption of a binary protocol instead of a character based protocol, resulting in a lightweight protocol suitable for peers that have to manage a lot of connections and resources (CPU, bandwidth, etc). The protocol is based on a modular design that supports different overlays and applications (see Fig. 2b).

In this case the P2PSIP infrastructure can be used for any application purpose, such as locating SIP user information or instant messaging. When secure connections are needed, the protocol can use TLS [13] or DTLS [14]. RELOAD is divided into different blocks, making easier the explanation of its functions. The topology plug-in is responsible for implementing the DHT overlay algorithm. This is connected with the routing layer with the purpose of routing the different messages through the overlay (joins, leaves, etc). A Storage module handles the storing of resources in the overlay and it is connected with the topology plug-in (to determine the replication policy) and to the routing layer (that determines the next hop). Finally, a forwarding layer delivers the messages, crossing NATs if necessary using the Interactive Connectivity Establishment (ICE) [15] protocol based on STUN and TURN servers. An additional connection between the forwarding layer and the topology plug-in is used by the forwarding layer to notify when a peer is not reachable, triggering maintenance operations such as updating the routing table.

The performance of an overlay network is closely related to the routing layer, which can support iterative or recursive routing. However, recursive routing is preferred because in most cases it would incur a lower delay that is also closely related with the requirement of supporting NAT in a transparent way. When recursive routing is used, a peer forwards a message to the next hop according to its routing table. Because only peers that have been directly contacted are added to the routing table, a peer has all the IP addresses to reach a next hop before checking the connectivity with an ICE exchange. Therefore, when possible, the messages are forwarded without contacting an ICE exchange since a cache of previous ICE exchanges can be used. On the other hand, if iterative routing is used, most probably the next hop is not known by the peer performing the operation, caching is not feasible and therefore an ICE exchange is performed every forwarding resulting in an undesirable impact on the delay.

The following components are used when routing messages. The first is the Node ID, currently defined as a 128 bit element. A variable length field could be useful and we advocate this option for reasons we explain in Section 3. Resource IDs are expected to have a variable length of maximum 255 bytes. If the Resource ID is longer in length than the Node ID, then it should be truncated to the Node ID length for storage and fetch operations. The overlay messages contain two additional data structures: the destination list and the via list. The destination list allows specifying a list of intermediate peers and can be used to avoid unnecessary ICE exchanges. The via list is used to get a response path symmetric to the request path (Fig. 3a). Another option would be that the contact info of the peer sending the message is included to allow a direct response (Fig. 3b). Although this seems to be more efficient from the point of view of delay, this is not necessarily true because the total delay depends on both the delay on the direct path and on whether an ICE exchange is necessary to find a pair of valid locators between requester and responder. Thus, when using the via list there is a higher probability that the ICE exchange is not necessary since the result should be cached in advance.

Regardless on the mechanism used on the return path, once the information is retrieved, the next step depends exclusively on the application level. If multimedia applications are being developed the end-points can proceed with

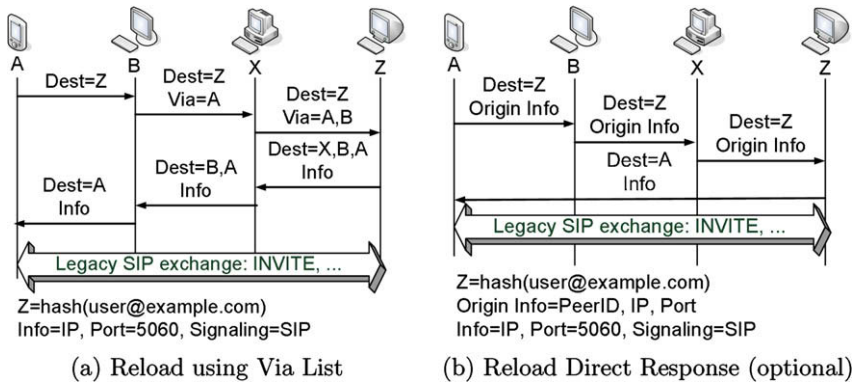


Fig. 3. Different request-response models.

the establishment of the multimedia session. For this scenario Fig. 3a and b show a SIP exchange where the negotiation of these session parameters is performed. For other types of applications, the underlying SIP exchanged is replaced by another suitable protocol.

Finally, once the communication is possible, any new information can be stored in the overlay network as resources. There are no restrictions upon the type of this information which depends on the application that uses the P2PSIP protocol (e.g. location information of users, supported and signalling protocols, etc).

3. H-P2PSIP

3.1. Hierarchical space domain of identifiers

In order to support the hierarchical P2PSIP architecture (H-P2PSIP), we define a hierarchical space of identifiers containing Hierarchical IDs (see Fig. 4). Each Hierarchical ID is composed by two part IDs: a Prefix ID with *n* bits and a Suffix ID with *m* bits. The Prefix ID is used for the routing in the interconnection overlay between the different P2PSIP domains, whereas the Suffix ID is used for routing queries only in the own P2PSIP domain of a peer. This design justifies that a variable length for Node IDs in P2PSIP since any mapping function with independence of its length can be used to generate the Hierarchical ID. This Hierarchical ID can be used either as Node ID or Resource ID.

3.2. H-P2PSIP service mapping

One of the main problems in a decentralised architecture is the mapping between the available information and/or services and the peers in the system. If we consider a multimedia environment based on P2PSIP, it is clear that resources are identified with URIs, for example `resource@example.com`. In order to map this URI to the Hierarchical ID, the Prefix ID is obtained by applying a hash to the domain of the URI: `Prefix ID = hash(example.com)`. The Suffix ID is obtained from the hash of the whole URI: `Suffix ID = hash_a(resource@example.com)`. The hash functions `hash` and `hash_a` can be identical or different. Once the mapping between the URIs and the Hierarchical ID has been established, the resource is stored in a tuple composed by the Resource ID, the original URI and the resource information by the peer having the closest Node ID. Depending on the DHT protocol, this tuple can be replicated to other peers in some way.

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

Fig. 4. Hierarchical ID.

The content of the resource information can vary depending on the application scenario. In the case of a VoIP application, it can be the user location, supported protocols and codecs. In the case of services, it can be configuration parameters. For instance, a streaming server must define at least the supported protocols and the IP address-port tuple.

3.3. H-P2PSIP basic operation

After resources have been mapped to identifiers and a criterion for their storage has been defined, H-P2PSIP defines a method to locate these resources. The behaviour of this method is divided in two cases. In the first case the search of a resource is bounded to the P2PSIP domain of the requester. This case is really simple since the search for resources is done inside the P2PSIP domain and it is identical to the flat peer-to-peer overlay using only the Suffix ID. In this situation, the Prefix ID of the resource must be equal to the hash of the associated URI domain. This hash is known by all the peers belonging to that P2PSIP domain.

However, if a resource is stored in a different domain the operation is more complex. For instance, this case can correspond to a VoIP call from a user in a P2PSIP domain to another user in a different P2PSIP domain. In order to obtain the resource (e.g. location) of the desired user, it is necessary to obtain the contact information published in the other P2PSIP domain. The first step in the search is to find a peer that can request information from other P2PSIP domains. These peers are the super-peers and there are several mechanisms [16,17] that can be used to select them. These mechanisms can be integrated in the maintenance protocol of the DHT used in the domain. In each P2PSIP domain there exists at least one super-peer,

although it is desirable to have several super-peers for redundancy.

Since all the peers in a domain know at least one super-peer, they can send a query to the super-peer in one hop. When the super-peer receives the query, it will search in the interconnection overlay for any of the super-peers that are responsible for the target Prefix ID, and once this information is retrieved, the query is forwarded to one of these super-peers. When the super-peer of the destination P2PSIP domain receives the query, it forwards the query inside its domain. If the query reaches a peer that has the desired resource, then the peer replies in way that is compliant with the P2PSIP protocol [6].

An example of the signalling on the proposed hierarchical scenario is shown in Fig. 5. Several aspects are taken into account in order to understand the signalling flow. First of all, when the peer in `domain.b` requests the information of `user1@domain.a`, the query in the Fetch message is plain text. Plain text is used since a peer in a domain does not have to know what hash function is used in the interconnection overlay and what hash function is used in other P2PSIP domains. Thus, the super-peer in `domain.b` performs `hash(domain.a)` in order to obtain the information of the super-peers in `domain.a` through the interconnection overlay. Inside this information, the hash used in the other domain (*hash_a*) is included and a request for the desired item can be built as `hash_a(user1@domain.a)`. Some of the peers taking care of the desired Resource ID answer to the super-peer from `domain.a`, which then forwards this information to the super-peer from `domain.b`. Finally, the super-peer from `domain.b` sends the desired Resource ID to the peer from `domain.b`. Once this flow finishes, a SIP negotiation can be initiated for IM, VoIP or video conference. Fig. 5 illustrates a subset of the real flow. The figure omits the intermediate hops in each overlay and ICE exchanges, if any are needed.

3.4. Advantages and disadvantages of the H-P2PSIP architecture

The H-P2PSIP proposal has several advantages. First, the operations or primitives of the DHT used in H-P2PSIP are not modified. Only some changes are needed in the maintenance operations to include the selection and update of super-peers [16,17]. Furthermore, the routing state does not increase compared to a flat overlay network, because the number of maintained peers is only increased up to the number of peers from each P2PSIP domain. Hence, the number of the routing entries is limited by the number of peers in a domain, although connectivity with other P2PSIP domains is available. If we consider that the routing state in a peer-to-peer network usually depends on the logarithm [18] of the number of peers, we have that the routing state in our approach is $O(\log_B M)$ where M is the number of peers in a domain. If we compare this routing state with a unique flat P2PSIP domain that contains all P2PSIP domains, we obtain that the number of peers in the flat overlay is $M \cdot K$ where K is the number of domains. Thus, the routing state is increased up to $O(\log_B(M \cdot K))$. If the number of registered P2PSIP domains increases, the routing state is higher. This effect is not desirable, especially in the case of P2PSIP enabled handheld devices where the resources are limited.

Other approaches like [19] or [20] propose more complicated hierarchical architectures in order to obtain short delay overlays but their solutions also imply an increment of the routing state, which is not suitable for our scenario. In the hierarchical case, we show that the routing state is reduced for the same number of peers, while having a comparable routing performance.

The only drawback of this approach is the possible overload of super-peers [11]. Nevertheless, this overload is smaller than in other proposals [21,22], where the super-peers must store all peers that depend on them. This fact

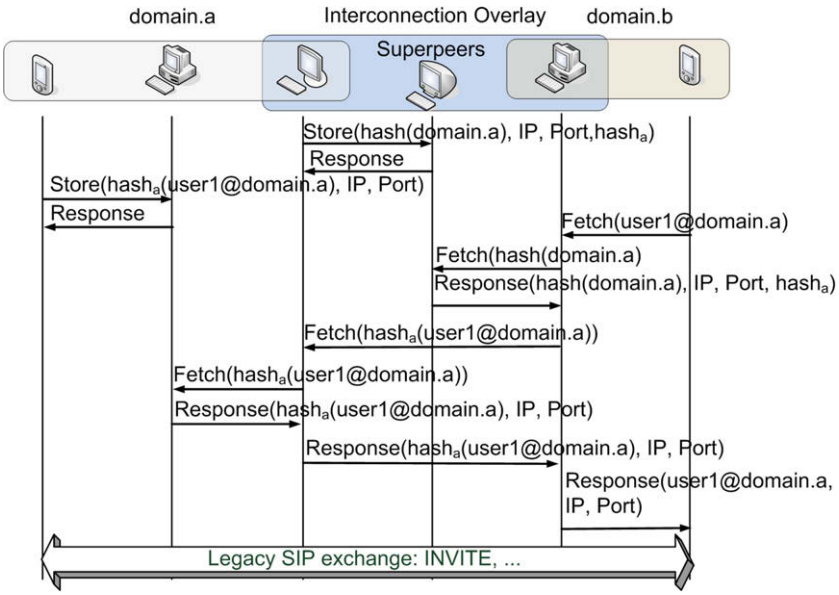


Fig. 5. H-P2PSIP signalling.

implies the maintenance of a larger amount of information, which is less scalable. Actually, they have to maintain two routing tables: a routing table of size $O(\log_B M)$ for their P2PSIP domain and a routing table of size $O(\log_B K)$ for the interconnection overlay. In this case, the state maintenance in super-peers is $O(\log_B M) + O(\log_B K) = O(\log_B(M \cdot K))$. This value is the same as in [20] and smaller than in [19] with the difference that the state in these proposals is maintained by all peers in the hierarchical overlay and in our case only in the super-peers.

To summarise, despite of the higher load experienced by super-peers, the proposed architecture allows global connectivity between different P2PSIP domains without increasing the routing state for peers, which could be a critical point especially for the computational power of handheld devices.

4. Routing performance in H-P2PSIP

This section studies the routing performance in a system based on H-P2PSIP. We have taken the work from [23] and we have improved the analysis with a more formal approach.

In the next list, there is a definition of the parameters for the analytical model:

- K: The number of P2PSIP domains.
- M_k: The number of peers in a P2PSIP domain k.
- N: All the peers from all the P2PSIP domains. In our case, it is considered that a peer cannot be attached to multiple P2PSIP domains, hence $N = \sum_{i=1}^{K} M_i$.
- S_k: The number of super-peers in a P2PSIP domain k.
- ρ_{ij}: The probability of launching a query from the P2PSIP domain i to the P2PSIP domain j.
- $C(x)$: The number of hops needed to find a super-peer in the interconnection overlay depending on the number of super-peers x. This value depends on the type of overlay used in the interconnection overlay.
- $D_k(x)$: The number of hops needed to find a peer in a flat overlay of type k as function of the number of peers x belonging to the P2PSIP domain.

We assume that all the peers in a P2PSIP domain know their super-peers from the interconnection overlay. This assumption implies that only *one hop* is needed to reach the super-peer. The routing performance inside a P2PSIP domain does not change and is the same as in a flat overlay network. However, if a query must be routed to other domain, it would be in any case one hop to any of the super-peers. The worst case happens when all the super-peers of a domain are attached to the interconnection overlay. Since the number of attached super-peers increases, the number of hops to search a resource in the interconnection overlay increases. Nevertheless, this increment is marginal: between one and three hops, depending on the number of super-peers per domain and the overlay used for the interconnection overlay.

Taking into account the above definitions, we obtain the routing performance (RP) of this DHT-based hierarchical overlay networks. First of all, we define the cost of finding a peer in each overlay:

- $D_k(M_k)$: The cost of finding a peer in its own domain.
- $C\left(\sum_{k=1}^{K} S_k\right)$: The cost of finding a super-peer in the interconnection overlay.

If the probability of obtaining an item in a domain from its super-peer is considered negligible and because the average number of peers in a P2PSIP domain is N/K with $N \gg K$, the average routing performance experienced by a peer in P2PSIP domain i can be written as follows:

$$RP_i = \rho_{ii} \cdot D_i(M_i) + \sum_{j=1, j\neq i}^{K} \rho_{ij} \cdot \left[1 + D_j(M_j) + C\left(\sum_{k=1}^{K} S_k\right)\right]. \tag{1}$$

The first term of the sum is the cost of searching something in the P2PSIP domain of a peer, whereas the second term is the cost for the searches in the other P2PSIP domains.

The average number of hops is given by the next expression:

$$RP = \frac{1}{N} \cdot \sum_{i=1}^{K} M_i \cdot RP_i. \tag{2}$$

If the number of peers is the same in all P2PSIP domains, we have:

$$RP = \frac{1}{K} \cdot \sum_{i=1}^{K} \cdot RP_i. \tag{3}$$

Because we have assumed that the number of peers is equal in all P2PSIP domains and each lookup in the overlay is considered randomly independent, we obtain that the probability of looking for a peer attached to other P2PSIP domain is equally distributed among all the foreign P2PSIP domains. In addition, the probability of looking for a peer in the own domain is different from the one of looking for a peer in other P2PSIP domains. Thus, the inter-domain query probability is $\rho_{ij} = \frac{1-\rho_{ii}}{K-1}$ and we can express Eq. 1 as follows:

$$RP_i = \rho_{ii} \cdot D_i(M) + \sum_{j=1, j\neq i}^{K} \cdot \frac{1-\rho_{ii}}{K-1} \cdot \left[1 + D_j(M) + C\left(\sum_{k=1}^{K} S_k\right)\right]. \tag{4}$$

This relation is useful for some type of scenarios like VoIP in community networks where $\rho_{ii} > \rho_{ij}$, which implies that calls between peers of the same community are more frequent. For other services where the lookup probability in the own P2PSIP domain is the same as for foreign P2PSIP domains $\left(\rho_{ii} = \rho_{ij} = \frac{1}{K}\right)$, we get Eq. 5 (which is a simplified version of Eq. 4):

$$RP_i = \frac{1}{K} \cdot D_i(M) + \sum_{j=1, j\neq i}^{K} \cdot \frac{1}{K} \cdot \left[1 + D_j(M) + C\left(\sum_{k=1}^{K} S_k\right)\right]. \tag{5}$$

Finally, if the same overlay is used in all P2PSIP domains the sum can be eliminated from Eq. 5:

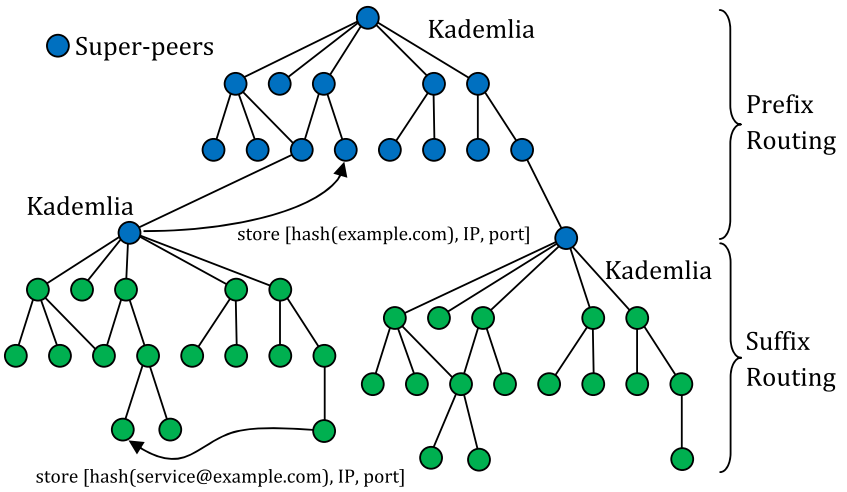


Fig. 6. Hierarchical Kademlia overlay network.

$$RP_i = \frac{1}{K}\cdot D(M) + \frac{K-1}{K}\cdot\left[1 + D(M) + C\left(\sum_{k=1}^{K} S_k\right)\right] = D(M) + \frac{K-1}{K}\cdot\left[1 + C\left(\sum_{k=1}^{K} S_k\right)\right]. \quad (6)$$

5. H-P2PSIP in Kademlia

In this section, we study the H-P2PSIP routing performance and routing state in the case when a Kademlia overlay [7] is used in all the P2PSIP domains and also on the interconnection overlay (see Fig. 6). Kademlia has been selected, because it is one of the most used DHT overlays in peer-to-peer applications like eMule[3], BitTorrent[4], etc.

Summarising, Kademlia is an overlay network which has a routing performance and a routing state with a logarithmic dependency on the number of peers from the overlay. These results are due to its XOR distance-based routing algorithm.

In order to verify the efficiency of our solution, when the Kademlia protocol is used, we use the next equality: $C(x) = D(x) \sim \log_B x + c$. We substitute this expression in Eq. 6 because the validation is performed via simulation with a setup similar to the conditions which are valid for this expression. Therefore:

$$RP = RP_i \sim \log_B(M) + c + \frac{K-1}{K}\cdot\left[1 + \log_B\left(\sum_{k=1}^{K} S_k\right) + c\right]. \quad (7)$$

If $K \gg 1$ and taking into account the properties of the logarithm, we can write:

$$RP = RP_i \sim \log_B(M) + c + 1 + \log_B\left(\sum_{k=1}^{K} S_k\right) + c = 1 + \log_B\left(M\cdot\sum_{k=1}^{K} S_k\right) + 2c. \quad (8)$$

In Fig. 7 we can see the routing state taking into account up to 10^4 peers. The x-axis is the number of peers and the y-axis represents the number of hops. To determine the routing performance of a Kademlia-based P2PSIP domain, we have to see how many peers belong to the overlay in order to see the required number of hops. The same method can be applied to the interconnection overlay if we consider $S_k = 1$. Furthermore, the total number of hops for the overlay can be estimated considering all the peers in all the P2PSIP domains.

Since the routing state must also be taken into account, the number of entries depends on the number of peers and on the setup parameter B. The number of routing in a super-peer is $O\left(\log_B\left(M\cdot\sum_{k=1}^{K} S_k\right)\right)$. If a flat overlay is used to connect all peers in different P2PSIP domains, peers would need $O(\log_B(M\cdot K))$ routing entries, but using the hierarchical architecture, peers only need $O(\log_B M)$. Therefore, the routing state savings are significant if many P2PSIP domains are interconnected.

6. Validation of the H-P2PSIP in Kademlia

This section is dedicated to the validation of H-P2PSIP using a hierarchical Kademlia overlay model. The objective is to validate the analytical model of the routing performance for this architecture and to evaluate the state size needed by peers and super-peers to maintain the proposed architecture.

The simulator for this study has been the PeerfactSim.KOM[5] simulation engine [24], which is a packet-level discrete event-based simulator written in Java. In order to facilitate the simulation of large scale peer-to-peer networks, the simulator uses a simple packet latency model between nodes that is the equivalent of the cumulative propagation, forwarding and queuing delay. However, it does not consider some details such as the processing time and the bandwidth of links (links are over-provisioned).

[3] http://www.emule-project.net/.
[4] http://www.bittorrent.com/.
[5] http://peerfact.kom.e-technik.tu-darmstadt.de/.

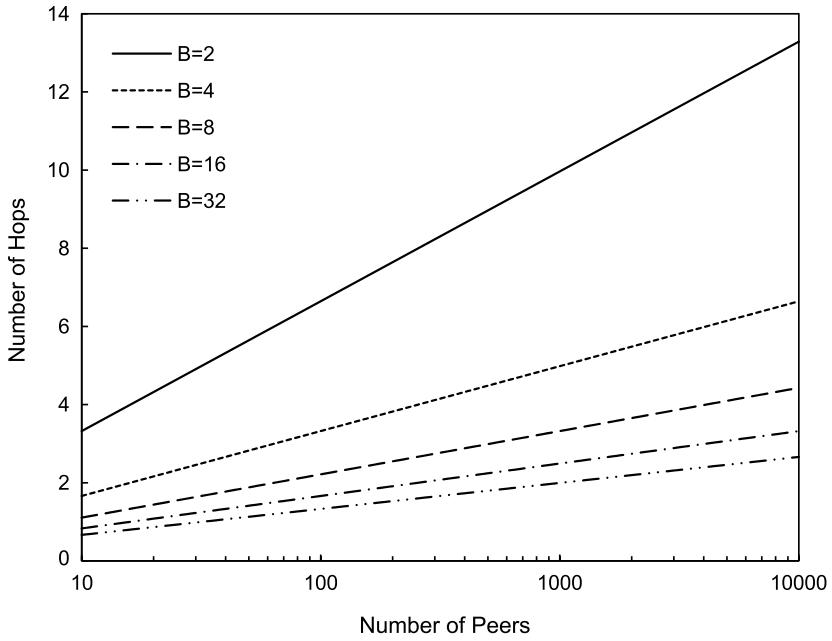


Fig. 7. Routing performance.

6.1. Simulation setup

To run the experiments, we implemented a prototype of the hierarchical Kademlia protocol and a network scenario generator on top of the simulator engine. The objective was to generate peer-to-peer network models similar to the behaviour of real life Kademlia peers. For this we assumed network scenarios with an average number of peers between 10 and 10,000 and the following number of domains: 1 (i.e. a pure Kademlia network), 5, 10 and 20. The peers were uniformly distributed among the domains. In addition, each domain has a super-peer that facilitates the connection of the domains through the interconnection overlay. Only one super-peer ($S_k = 1$) is placed since the routing performance penalty is marginal as has been explained in Section 4 and the complexity of the simulation increases a lot. Additionally, the stability of super-peers can be assured as in Skype [3] with some mechanism like [11,16,17]. The management of the super-peers is not included in the study and it constitutes future work. Thus, we do not consider churn in super-peers and only churn in peers in the manner we explain in the next paragraph.

Each peer executes four types of operations: joining when it attaches itself to the peer-to-peer overlay; storing a key-value pair; lookup when searching for a previously stored key in the attempt to find the value and leaving. In order to have scenarios closer to reality, we used an existing study of the Kad implementation of Kademlia [25] that measures the peer behaviour in terms of churn rate and up-time distributions. Their findings conclude that in a file-sharing Kad network peers arrive and leave with a negative binomial distribution, while the peer session time is similar to a Weibull distribution. Additional details can be found in [26–28]. This setup can be considered as a medium-high churn rate scenario since the Kad network is used in eMule and BitTorrent applications where the churn is not at all negligible. Thus, our scenario is a worse case study compared to the real situation that occurs in multimedia applications like Skype [1–3].

Due to the simulation constraints (such as simulation duration, required computing resources, etc.) each simulation scenario has two phases. The first is a transitory phase, during which the total number of peers reaches the average targeted in each scenario. This phase does not consider the Kad peers behaviour, since in a real Kad network the arrival and the leaving rate are the same. In the second phase, the peers join and leave the peer-to-peer network at the rate given in [25] with a negative binomial distribution (approximately one peer every two seconds). In this phase, the average number of peers in the network is the number of peers at the end of the first phase. Because the results from the Kad study were given for a flat Kademlia network, in the hierarchical case, arriving peers are randomly assigned to any of the existing domains with a uniform distribution.

During a session each peer performs a store that is the equivalent to storing its own URI in the peer-to-peer network, and a number of lookup operations that are the equivalent to searching for the URI of other peers. Assuming that the lookups follow the behaviour of the user contacting other peers, we used a Poisson distribution to model them, at an average rate of one call every 10 min. The transitory first phase was limited to 30 min, while the stationary second state spanned up to two hours. As in Kademlia, a maintenance operation was run by each peer every hour after their arrival, in order to refresh their routing tables and republish stored values to neighbour peers. Measurements were taken only during the second phase.

In relation with the setup of the Kademlia overlay, the protocol has been configured with $B = 2^b = 2$, $k = 20$ and $\alpha = 1$. The reason for using $\alpha = 1$ is to facilitate the comparison with other overlays that cannot easily parallelize their operations. Determining the performance for higher values of α is planned for future work. The value of k is used for the size of the buckets and also for the number of replicas of each item inside the overlay.

6.2. Simulation results

This section presents the results obtained with the PeerfactSim.KOM simulator and with the simulation setup that was explained in the previous section. In order to increase the accuracy, each scenario was simulated ten times. With these measurements, we obtained a very low standard deviation and narrow 99% confidence intervals, indicating that for the large number of peers used, their IDs are uniformly distributed in the hash space.

Fig. 8 illustrates the global routing performance, i.e. the average number of hops a peer experiences when locating a stored value (the URI of another peer for H-P2PSIP). Since by scenario design, the majority of lookups are inter-domain lookups, the function representing the total number of hops is the sum of three terms. The first is the number of hops performed in the domain of the requester. For inter-domain lookups this is always 1, because we assumed that all peers know their super peers. The second term is the number of hops performed in the interconnection overlay. This is a logarithmic increasing function with the number of domains. The last term is the number of hops from the destination domain between the super-peer and the peer that contains the desired resource. On average, this is a logarithmic increasing function with the number of peers inside the domain.

For each set of results, the experiments considered a fixed number of peers, N, and several values for the number of domains, K. Consequently, on average the number of peers inside each domain, M, is inverse proportional to K because peers are uniformly assigned to an existing domain. The routing performance in terms of number of hops is bounded by Eq. 8, which is a constant since it only depends on N. The obtained results are smaller than the theoretical limit due to the information replication.

In Figs. 9 and 10 we analyse the routing performance separately inside the domain and interconnection overlay. As expected, the number of hops needed in the interconnection overlay (see Fig. 9) is roughly the same for any number of peers, since it only depends on the number of domains, K. In addition, the logarithmic dependency with K can be observed through the large increase in the number of hops from one domain to five domains and the same increase between 5, 10 and 20 domains (the same difference when doubling the number of domains, hence a linear increase on a logarithmic scale).

Likewise, in Fig. 10 we can see the reduction in the number of hops needed in the cluster when increasing the number of domains, since this results in a proportional decrease in number of peers inside the cluster, $M = \frac{N}{K}$. For the same K and we obtain a logarithmic dependency with N (linear on a logarithmic scale).

Finally, in Figs. 11 and 12 we illustrate the average number of routing entries that have to be stored by the peers in the routing tables used inside the domain and in the interconnection overlay, respectively. This is important since it has a direct correlation with the necessary memory that we want to reduce in the case of mobile devices and to justify our hierarchical solution. The results for a single domain (i.e. a flat overlay) serve as a reference, in which case the memory required for the interconnection overlay

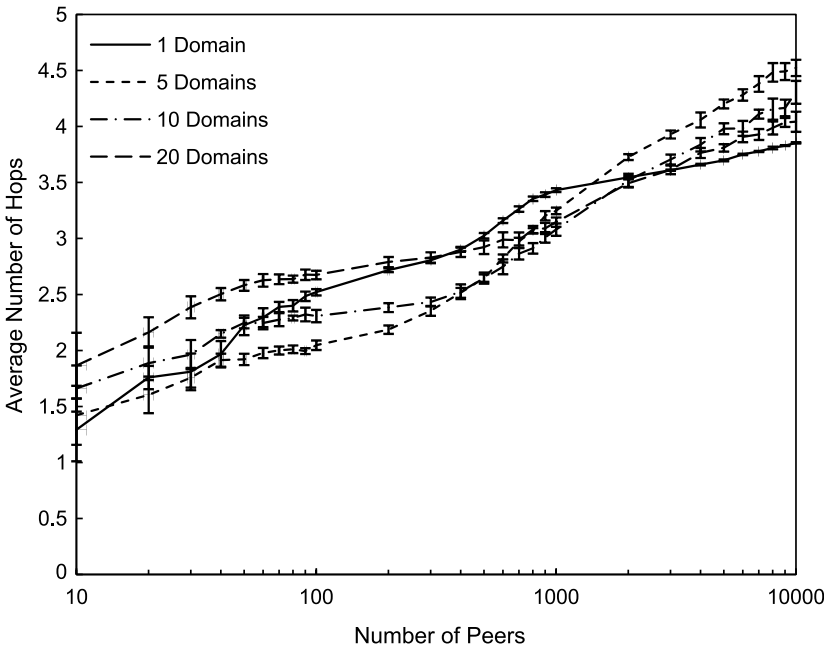


Fig. 8. Global routing performance for value lookups.

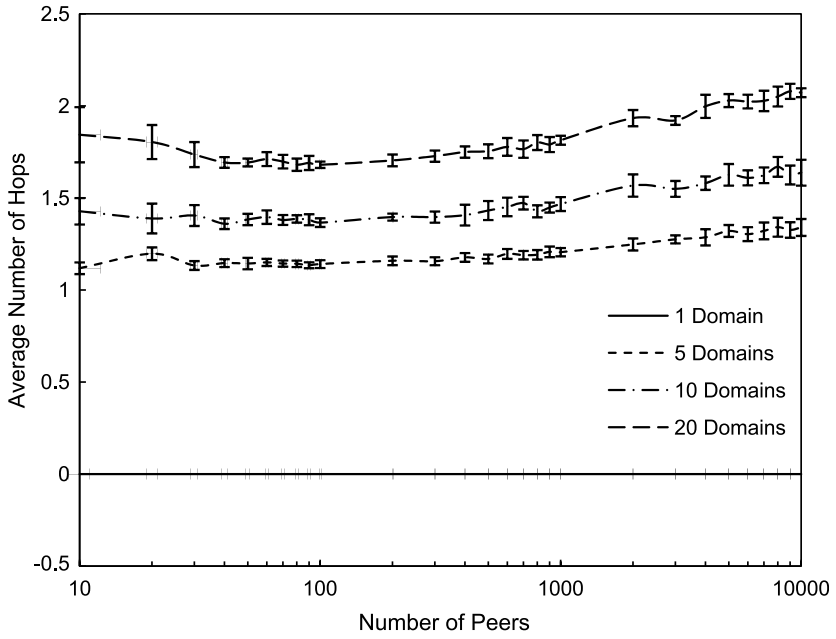


Fig. 9. Interconnection overlay routing performance for value lookups.

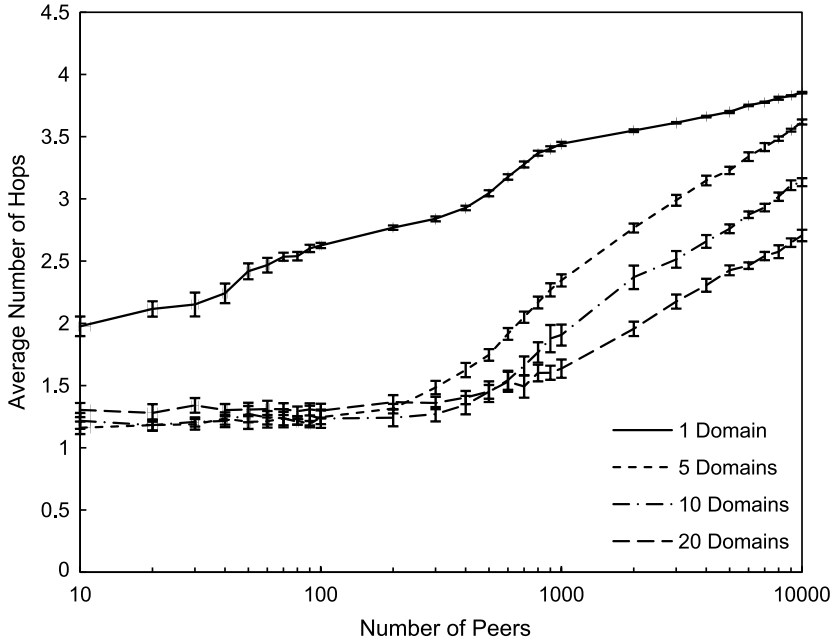


Fig. 10. Domain routing performance for value lookups.

routing table of the only super-peer is zero. However, with a modest increase in the number of domains and their associated routing state (up to 20 – see Fig. 11), we obtain a significant reduction (approximately 50%) in the average used routing entries by the peers (see Fig. 12).

We can observe that the number of routing entities lies between the expected value for a Kademlia node, $entries \in [(B-1)\cdot \log_B T, k\cdot(B-1)\cdot \log_B T]$ according to [7] (where T is the number of peers in the considered overlay, M or K in our case), and have an increasing monotonic dependency with the number of peers inside a domain and in the interconnection overlay, respectively.

7. Related work on hierarchical overlay networks

Peer-to-peer overlay networks usually require $O(\log_B N)$ peer hops to reach the desired destination and $O(\log_B N)$ routing entries to maintain the desired structure. This com-

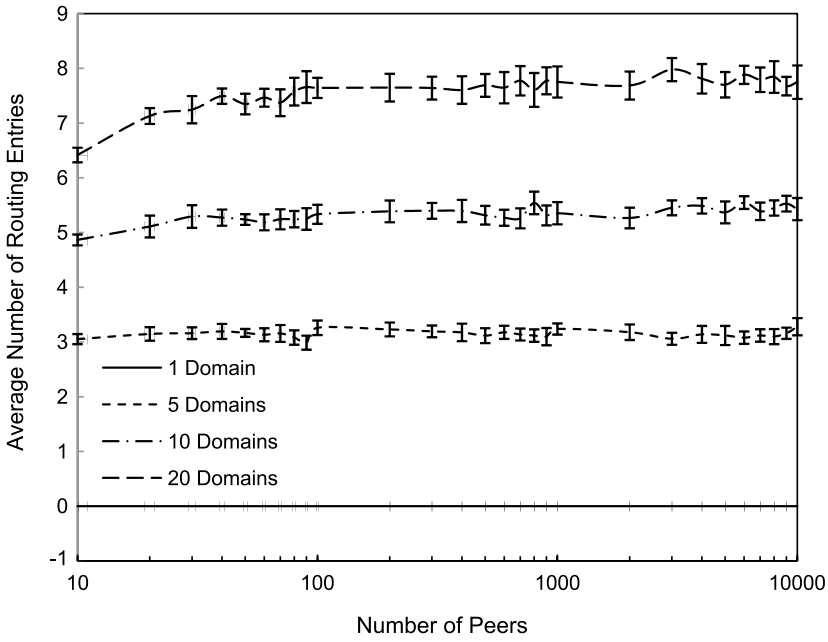


Fig. 11. Average number of entries in interconnection overlay routing tables.

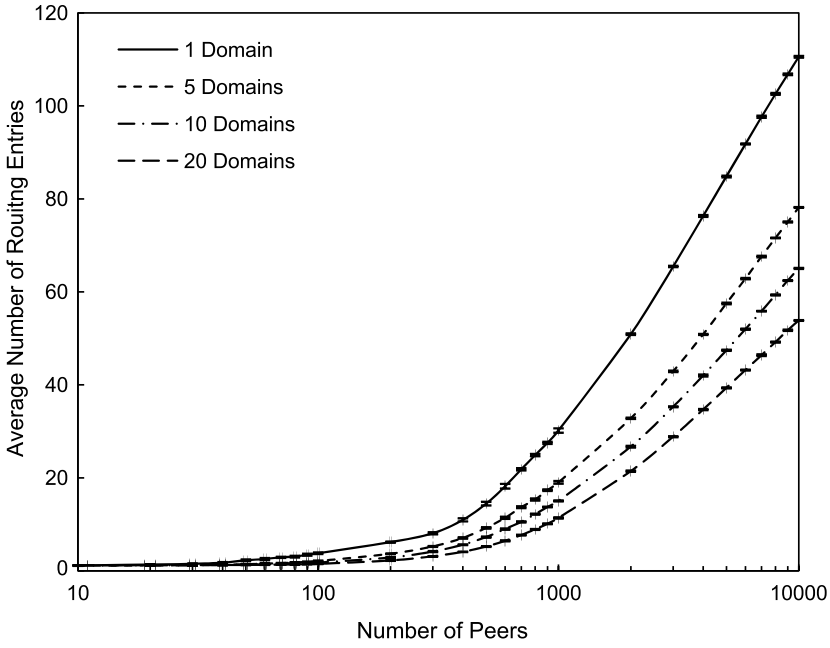


Fig. 12. Average number of entries in peers routing tables.

plexity ensures good scalability but it is desirable to further improve it, especially if the decentralised applications want to be deployed in handheld devices that have limited resources. The most representative example is VoIP.

Studies related to hierarchical overlay networks try to improve the canonical overlay networks. When a hierarchical architecture is considered, it is necessary to take into account the different trade-offs that arise with these types of architectures [29]. It is demonstrated that these architectures have benefits [30,31] in comparison with the canonical counterparts.

One approach is to delegate all the work to super-peers [21,22]. They maintain the overlay network and perform all the necessary actions and peers only have to register their information with their super-peers.

Other studies are focused on decrementing the delay in peer-to-peer overlay transactions. In [19] a low delay hierarchical overlay network based on Chord is proposed. The

drawback is the high routing state capacity needed (memory, CPU and bandwidth) because *all* the peers in the overlay are attached to *all* the levels in a *n-level* hierarchy. A less aggressive design with the same objective is presented in [20] where a hierarchical structure is built with the constraint of limiting the maintenance cost to the canonical (flat) counterpart. In addition to Chord, there is also related work in hierarchical CAN architectures [32,33].

Our approach allows building, in a simple way, a hierarchical overlay network due to the definition of the Hierarchical ID. Furthermore, since the routing in the interconnection overlay (based on the Prefix ID) is independent with respect to the routing in the P2PSIP domains (based on the Suffix ID), a great flexibility is given to the architecture. This flexibility comes from the fact that any overlay network can be deployed in the interconnection overlay or in the P2PSIP domains with independence of the deployment in any of the domains.

Although cross-connectivity between the P2PSIP domains is obtained, peers do not see any penalty and only super-peers may become overloaded. Thus, an improvement to the previous work is obtained due to its simplicity, low cost and efficiency.

8. Conclusions

The objective of the H-P2PSIP architecture proposed in this paper is to enable the interconnection between different P2PSIP domains in order to support global multimedia services. This solution provides a tool for the easy development of decentralised multimedia architectures since they can provide a more scalable solution than centralised architectures. Furthermore, this decentralised architecture is suitable when central servers cannot be located in a well-known location, such as in the case in community networks.

In H-P2PSIP the peers (members of a community network) connect to their local P2PSIP domain. In each P2PSIP domain an overlay is maintained and at least one super-peer is selected to represent the domain. Between the super-peers an interconnection overlay is maintained that assures the connectivity between the different domains. The routing in each P2PSIP domain is based on a Suffix ID, while the routing in the interconnection overlay uses a Prefix ID. If `Prefix ID = hash(example.com)` and `Suffix ID = hash_a(user@example.com)`.

We perform an analytical study of the routing performance in terms of number of hops needed to locate a particular resource in the peer-to-peer overlay, proving that we obtain approximately the same values as in the flat counterpart. However, since in a hierarchical topology peers from each domain do not store any kind of routing information about peers outside their domain, we obtain a lower routing state that is determined only by the number of peers in the domain.

Finally, we simulate the H-P2PSIP scenario using the PeerFactSim.Kom simulator [24] in order to validate the analytical model of the routing performance and routing state. In order to simulate a realistic scenario, a churn has been setup according to the results in [25]. The adoption of a hierarchical architecture gives about the same routing performance when compared with a global flat overlay network and is much lower than the theoretical limit due to replicas stored at both domain and interconnection level. The routing state in peers is decreased when increasing the number of domains while connectivity between all domains is still assured through the super-peers.

As future work, we would like to study the inclusion of a specific mechanism for the selection of super-peers [11,16,17] and to analyse its impact in the performance of the H-P2PSIP architecture.

Acknowledgements

This work has been supported by the European Commission under the IST Content NoE[6] (FP6-2006-IST-038423), by the Regional Government of Madrid under the BioGridNet[7] project No. (CAM, S-0505/TIC-0101) and by the Ministry of Science and Innovation under the CONPARTE project No. (MEC, TEC2007-67966-C03-03/TCM).

References

[1] S.A. Baset, H.G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol, in: INFOCOM 2006, Proceedings of the 25th IEEE International Conference on Computer Communications, April 2006, pp. 1–11.
[2] S. Guha, N. Daswani, R. Jain, An experimental study of the skype peer-to-peer voip system, in: IPTPS 2006, 2006.
[3] Dario Rossi, Marco Melia, Michela Meo, A detailed measurement of skype network traffic, in: In IPTPS 2008, 2008.
[4] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, RFC 3261 (Proposed Standard), June 2002, Updated by RFCs 3265, 3853, 4320, 4916.
[5] D. Bryan, P. Matthews, E. Shim, D. Willis, Concepts and terminology for peer to peer sip, Internet Draft draft-ietf-p2psip-concepts-02.txt, July 2008.
[6] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, H. Schulzrinne, Resource location and discovery (reload), Internet Draft draft-ietf-p2psip-reload-00.txt, July 2008.
[7] P. Maymounkov, D. Mazieres, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002, Revised Papers, Lecture Notes in Computer Science, Chapter Kademlia: A Peer-to-peer Information System Based on the XOR Metric, Springer, vol. 2429, 2002, pp. 53–65.
[8] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for internet applications, IEEE/ACM Trans. Network. 11 (1) (2003).
[9] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Schenker, A scalable content-addressable network, in: SIGCOMM'01, ACM Press, New York, NY, USA, 2001. pp. 161–172.
[10] H.A. Simon, The architecture of complexity, in: MIT Press, editor, The Sciences of the Artificial, 1981, pp. 192–229.
[11] B. Beverly Yang, H. Garcia-Molina, Designing a super-peer network, in: Proceedings of the 19th International Conference on Data Engineering, 2003, pp. 49–60.
[12] E. Marocco, D. Bryan, Interworking between p2psip overlays and conventional sip networks, Internet Draft Draft-marocco-p2psip-interwork-01.txt, March 2007.
[13] T. Dierks, E. Rescorla, The transport layer security (TLS) protocol version 1.1., RFC 4346, Internet Engineering Task Force, April 2006.
[14] E. Rescorla, N. Modadugu, Datagram transport layer security, RFC 4347, Internet Engineering Task Force, April 2006.
[15] J. Rosenberg, Interactive connectivity establishment (ice): a protocol for network address translator (nat) traversal for offer/answer protocols, Internet Draft draft-ietf-mmusic-ice-19.txt, October 2007.
[16] Su-Hong Min, J. Holliday, Dong-Sub Cho, Optimal super-peer selection for large-scale p2p system, in: International Conference

[6] http://www.ist-content.eu.
[7] http://www.biogridnet.org.

on Hybrid Information Technology, 2006, ICHIT'06, vol. 2, 2006, pp. 588–593.
[17] A.T. Mizrak, Yuchung Cheng, Vineet Kumar, S. Savage, Structured super-peers: leveraging heterogeneity to provide constant-time lookup, in: Proceedings of the Internet Applications, WIAPP, 2003, pp. 104–111.
[18] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes, Commun. Survey Tutorial IEEE 7 (2) (2005) 72–93.
[19] Zhiyong Xu, Rui Min, Yiming Hu, Hieras: a dht based hierarchical p2p routing algorithm, in: Proceedings of the International Conference on Parallel Processing, 2003.
[20] P. Ganesan, K. Gummadi, H. Garcia-Molina, Canon in g major: designing dhts with hierarchical structure, in: Proceedings of the 24th International Conference on Distributed Computing Systems, 2004, pp. 263–272.
[21] Luis Garces-Erice, Ernst W. Biersack, Keith W. Ross, Pascal A. Felber, Guillaume Urvoy-Keller, Hierarchical p2p systems, in: Proceedings of the ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par), 2003.
[22] S. Zoels, Z. Despotovic, W. Kellerer, Cost-based analysis of hierarchical dht design, in: Sixth IEEE International Conference on Peer-to-Peer Computing, P2P, 2006, pp. 233–239.
[23] Isaias Martinez-Yelmo, Ruben Cuevas, Carmen Guerrero, Andreas Mauthe, Routing performance in hierarchical dht-based overlay networks, in: Proceedings of the 16th Euromicro International Conference on Parallel Distributed and Network-based Processing, February 2008.
[24] V. Darlagiannis, A. Mauthe, N. Liebau, R. Steinmetz, An adaptable, role-based simulator for P2P networks, in: Proceedings of the International Conference on Modeling, Simulation and Visualization Methods, 2004, pp. 52–59.
[25] Moritz Steiner, Taoufik En-Najjary, Ernst W. Biersack, A global view of kad. in: IMC'07 – Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, New York, NY, USA, 2007, pp. 117–122.
[26] Moritz Steiner, Taoufik En-Najjary, Ernst W. Biersack, Exploiting kad: possible uses and misuses, SIGCOMM Comput. Commun. Rev. 37 (5) (2007) 65–70.
[27] Moritz Steiner, Ernst W. Biersack, Taoufik En-Najjary, Actively monitoring peers in KAD, in: IPTPS'07, 6th International Workshop on Peer-to-Peer Systems, Bellevue, USA, February 26–27, 2007.
[28] Moritz Steiner, Taoufik En-Najjary, Ernst W. Biersack, Analyzing peer behavior in KAD, Technical Report EURECOM+2358, Institut Eurecom, France, October 2007.
[29] Vasilios Darlagiannis, Andreas Mauthe, Ralf Steinmetz, Overlay design mechanisms for heterogeneous, large scale, dynamic P2P systems, J. Network Syst. Manage. Special Issue Distribut. Manage. 12 (3) (2004).
[30] M. Kwon, S. Fahmy, Toward cooperative inter-overlay networking, in: Proceedings of the IEEE ICNP, 2003.
[31] M. Kwon, S. Fahmy. Synergy: an overlay internetworking architecture, in: Proceedings of the 14th International Conference on Computer Communications and Networks, ICCCN 2005, pp. 401–406.
[32] Zhichen Xu, Zheng Zhang, Building low-maintenance expressways for p2p systems, Technical Report, Internet Systems and Storage Laboratory, HP Laboratories Palo Alto, 2002.
[33] Xiao-Ming Zhang, Yi-Jie Wang, Zhou. Jun Li, LNCS: Parallel and Distributed Processing and Applications, Research of Routing Algorithm in Hierarchy-Adaptive P2P Systems, Springer, Berlin, Heidelberg, 2007. pp. 728–739.

Isaias Martinez-Yelmo received a M.Sc. in Telecommunication Engineering in 2003 from University Carlos III de Madrid, and a M.Sc. on Telematics in 2007 from University Carlos III de Madrid and University Politecnica de Cataluña, both in Spain. He is research and teaching assistant in Telematics Engineering Department and Ph.D. student on Telematics at University Carlos III de Madrid since 2004. His research activities are focused on NGN networks, Peer-to-Peer overlay networks and Content Distribution Networks. He has been involved in several national and international research projects related with content distribution, overlay networks and broadband networks. Some of the recent research projects in which he has participated are EU IST CONTENT and Madrid Government BIOGRIDNET projects.

Alex Bikfalvi received a B.S. degree in telecommunications from Technical University of Cluj-Napoca, Romania in 2006 and he is currently pursuing a M.S. and a Ph.D. at Carlos III University of Madrid. He is now a research assistant at IMDEA Networks research institute in Madrid, Spain. His research interests include networking, peer-to-peer, protocol design and system software.

Ruben Cuevas got his M.Sc. in Telecommunication Engineering and M.Sc. in Telematic Engineering at Universidad Carlos III de Madrid in 2005 and 2007, respectively. Furthermore, he obtained his M.Sc. in Network Planning and Management at Aalborg University in 2006. Currently he is Ph.D. Candidate at the Telematic Department at University Carlos III de Madrid. His research interest includes Overlay and P2P Networks and Content Distribution.

Carmen Guerrero received the Telecommunication Engineering degree in 1994 from the Technical University of Madrid (UPM), Spain, and the Ph.D. in Computer Science in 1998 from the Universidade da Coruña (UDC), Spain. She has been an assistant (1994–2000) and assistant professor (2000–2003) at UDC. She is currently associate professor since 2003 at Universidad Carlos III de Madrid (UC3M), teaching computer networks courses. She has been involved in several national and international research projects related with content distribution, overlay networks, information retrieval, broadband access networks, network management and advanced network and multimedia real time systems. Some of the recent research projects in which she has participated are: CONTENT: Content Home Network and Services for Home Users, MUSE: Multiservice Access Everywhere and E-NEXT: Network of Excellence in Emerging Networking Technologies.

Jaime Garcia received the Telecommunications Engineering degree in 2000 from the University of Vigo, Spain and the Ph.D. in Telecommunications in 2003 from the University Carlos III of Madrid, Spain. He is currently an associate professor at Univ. Carlos III of Madrid having joined in 2002 and he has published over 25 papers in the field of broadband computer networks in magazines and congresses. He has been involved in several international and national projects related with protocol design, user localization, broadband access and signaling protocols like the EU IST MUSE project. Currently, he is working in the EU IST CONTENT and Madrid Government BIOGRIDNET projects.