Benefits of an Implementation of H-P2PSIP

Isaias Martinez-Yelmo, Roberto Gonzalez, Carmen Guerrero
Universidad Carlos III de Madrid
Av. Univerisidad 30. 28911 Leganes. Madrid. Spain
Email: {imyelmo,rgonzale,guerrero}@it.uc3m.es

***Abstract*—In this paper, we report on the results of experiments with an implementation of H-P2PSIP, which allows the exchange of information among different DHTs (Distributed Hash Tables) making use of a hierarchical architecture. This paper validates our previous H-P2PSIP proposal in an environment with a real TCP/IP stack close to a real scenario. The results show how the benefits of this real H-P2PSIP implementation in terms of routing performance (number of hops), delay and routing state (number of routing entries) are better than a flat DHT overlay network and how the exchange of information among different DHT overlay networks is feasible.**

***Keywords*-Hierarchical DHT, P2PSIP, Implementation, Evaluation, Performance.**

I. Introduction

The traffic related with Peer-to-Peer overlay networks has increased considerably last years. Thus, the research related with Peer-to-Peer in the Internet and how to measure accurately this kind of traffic [1] is currently a hot topic. Many applications generate Peer-to-Peer traffic in Internet, where the most relevant ones are probably eMule [2] with its KAD network [3], Bittorrent [4], [5] and Skype [6], [7]. Some of them are based on the same Peer-to-Peer network, such as trackerless Bittorrent or KAD network from eMule that make use of Kademlia [8], but they cannot establish communication among them although their purpose is the same.

H-P2PSIP [9], [10] presents an architecture based on two levels of hierarchy, which takes into account the ongoing design on the IETF P2PSIP Working Group. The P2PSIP WG aims to develop a protocol that allows the implementation of any Peer-to-Peer network but it does not consider the exchange of information between different domains based on different DHTs. The main usage expected for P2PSIP is a distributed replacement of SIP [11], [12], although P2PSIP has the flexibility to support other ones [13], [14]. Our proposal consist on a hierarchical DHT architecture that allows the exchange of information between different heterogeneous DHTs deployed in different domains, (see Figure 1). The idea is to support the exchange of information between different domains. In relation with a SIP usage, our hierarchical architecture replaces some of the proxy/registrar SIP functionalities.

This proposal is analysed mathematically and validated through simulation in [9], [10]. However, if we consider the complexity of Peer-to-Peer technologies due to their decentralised architecture, an implementation with a real TCP/IP stack is necessary to have a full validation of this proposal. In this paper, we report on the results obtained from a real implementation of our H-P2PSIP proposal. This implementation verifies that the exchange among different DHT overlays is possible without losing performance in terms of hops, delay or routing state. Furthermore, the hierarchical DHT architecture can be used to develop additional features and services with the flexibility offered by H-P2PSIP, since any DHT can be used if desired.

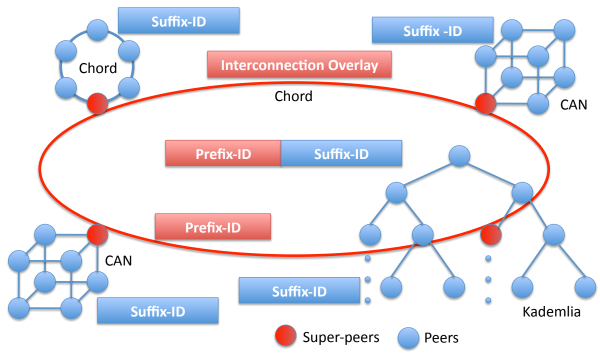


Fig. 1. H-P2PSIP architecture

The structure of this paper is as follows. Section II summarizes the related work with respect to the topic of hierarchical Peer-to-Peer networks. A description of H-P2PSIP is given in Section III, detailing the key points of the proposal and helping to understand the results provided in this paper. Section IV explains the main development steps to implement H-P2PSIP. Section V describes the scenario used to validate the implementation. The results obtained from our experiments are detailed in Section VI. Finally, Section VII summarises the conclusions and future work related with this paper.

II. Related Work

One of the first and more relevant publications in hierarchical Peer-to-Peer networks is the work in [15]. This paper describes the benefits of hierarchical Peer-to-Peer networks and presents a mathematical analysis. Subsequently, other publications related with hierarchical Peer-to-Peer networks have been published. For instance, Hieras [16] minimizes in a multilevel hierarchy the delay experienced by queries. Hieras groups each level of the hierarchy according to the delay among peers; the higher is the delay among peers, the higher is the level of the hierarchy where they are grouped. Queries are launched in the lowest level of the hierarchy and if a resource is not found, the next level is used. The main drawback of Hieras is the increase of routing state and maintenance traffic

 ISBN: 978-1-61208-102-1

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

Fig. 2. Hierarchical-ID

in all peers, since each peer needs to maintain each level of the hierarchy. Other issues related with hierarchical ring topologies are published in [17]. Additional improvements and algorithms that are more complex can be used if necessary. Cyclone [18] builds a hierarchical DHT overlay network in similar manner with respect to [16]. However, it assures that the number of links maintained by each peer is similar to the flat counterpart. Other approaches (i.e., Canon [19]) optimize the delay, but they limit the Routing State that is maintained by peers. Unfortunately, in most of these proposals, the design of the hierarchical architecture is heavily coupled to the DHT used to build the hierarchy. Our proposal, H-P2PSIP [9], [10], offers a great flexibility at this point since the architecture does not depend on the DHTs being used. In addition, different types of DHT networks can be used if necessary without compromising the performance of the proposed design. We demonstrate these advantages of H-P2PSIP with the implementation presented in this paper.

III. H-P2PSIP

H-P2PSIP [9], [10] defines a hierarchical DHT overlay network composed of two levels of hierarchy; an example is given in Figure 1. The purpose of this two-level hierarchy is the exchange of information among different domains where any DHT overlay network is deployed inside of them. The lower level is composed of different domains that want to exchange information among them. Each domain is independent of the others; therefore, the peers in each domain implement a DHT overlay network according to the domain preferences. Thus, the DHT overlay network can be different in each domain. On the other hand, the upper level is composed of only one DHT overlay network, named Interconnection Overlay. This Interconnection Overlay provides a service similar to a directory service among the different domains from the lower level of the hierarchy. The purpose of this Interconnection Overlay is to route the queries among different domains when a peer of one domain wants to retrieve information stored in other domain. This Interconnection Overlay can be based on any DHT overlay network. In order to support these functionalities, H-P2PSIP uses a hierarchical space of identifiers composed of Hierarchical-IDs (see Figure 2). 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 the different domains. This implies that all the peers or resources belonging to the same domain share the same Prefix-ID. On the other hand, the Suffix-ID is used only in the domain where a peer is attached and permits to localize any resource in the overlay network of that domain. Thus, this design allows the routing of queries among different domains. When searching for a resource in another domain, the query is routed to the desired domain using the Prefix-ID. Finally, the desired resource in the external domain is found through the Suffix-ID.

A. Hierarchical-ID generation

The next paragraphs explain how we generate the Node-IDs and Resource-IDs (based on Hierarchical-IDs) by applying hash functions to a regular URI. The starting point for this discussion is that each domain has a domain name (e.g., `example.com`) and its resources are identified by an URI (e.g., `resource@example.com`).

Node-ID generation. A Node-ID identifies each node participating in the overlay network of a domain. The generation of a Node-ID depends on the security level desired in the system. If we have a domain named `example.com`, the Prefix-ID is generated as follows: `Prefix-ID=hash(example.com)`. This construction of the Prefix-ID allows its generation if the domain that wants to be contacted is known in advance. On the other hand, the Suffix-ID must be different for each peer. Several mechanisms exist to generate this Suffix-ID. One option is to apply a hash function to some parameter (e.g., `Suffix-ID=hash(ip_address)`), but this option is not completely secure. A secured solution implies a central authority that assigns Node-IDs with signed certificates. The central authority uses a random number generator for Suffix-IDs and avoids unnecessary Node-ID collisions tracking the previously generated IDs.

Resource-ID generation. If we have a resource associated to a regular URI such as `resource@example.com`, a Hierarchical-ID can be created using this URI. The Prefix-ID in a Resource-ID must have the same value used in the Node-IDs. In fact, this is possible since the URI contains the domain where the resources are located. Therefore, the Prefix-ID must be: `Prefix-ID=hash(example.com)`. On the other hand, the Suffix-ID must be different for each resource if possible; thus, the Suffix-ID is generated hashing 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 Hierarchical-IDs is established, any resource can be stored with its Resource-ID and the original URI in order to perform disambiguation in resources with the same Resource-ID. Taking into account this Resource-ID generation, the information associated to a domain is stored in that domain since Resource-IDs are couple to each domain through the Prefix-ID. However, it is possible to store pointers to resources from other domains if necessary.

B. Super-Peer role in H-P2PSIP

Super-peers are necessary to interconnect the different domains and to route the queries among these domains; see Figure 1. We mentioned previously how the routing is based on the defined Hierarchical-ID, and now we explain how to use this Hierarchical-ID. In order not to overload most of the peers with additional tasks, we use super-peers to realize the necessary additional operations. Basically, super-peers forward the queries to the right external domain specified in the query. The super-peers must participate in the Interconnection

Overlay as well as in their own domain as regular peers. Taking into account the role of super-peers, their main tasks are enrolment and maintenance operations in the Interconnection Overlay and forwarding of inter-domain queries. Since super-peers have to manage the Interconnection Overlay, they have to manage an additional overlay routing table in addition to the regular that peers use to perform queries inside their own domain. The forwarding of queries implies a higher load in terms of CPU and bandwidth consumption in super-peers.

An open issue is the management of super-peers. H-P2PSIP suits very well to create a distributed VoIP signalling protocol where the location information is stored in the DHTs of the different domains (different usages can be also applied to H-P2PSIP if desired). Considering VoIP scenarios, two scenarios mainly require attention: operator-based scenarios or community networks based scenarios. In operator-based scenarios, operators want the control of super-peers to get track of whole signalling in order to charge users. Therefore, operators have to provide the necessary super-peers and take care of their full availability since they are the key for charging users. On the other hand, in community networks based scenarios, a mechanism is necessary to select the most suitable super-peers dynamically and perform load balancing among them. There are several published proposals on the selection of super-peers in a proper way [20]–[22]. These mechanisms can be integrated in the maintenance operations of any Peer-to-Peer protocol if necessary.

C. Signalling - URI based routing

The H-P2PSIP signalling is associated to all the operations in the overlay network. However, the most interesting operations are the storage and retrieval of resources. An example of the signalling on the proposed hierarchical scenario is shown in Figure 3 (the figure omits the intermediate hops in each DHT). Several aspects are taken into account in order to understand the signalling flow. The peer in `domain.a` performs the storage operation inside its domain. Later, peer in `domain.b` requests the information of `user1@domain.a`. Here, an important issue is that the query in the Fetch message contains the whole URI as plain text. This method allows rebuilding the Hierarchical-ID independently of the hash function used in the different domains. Thus, once the super-peer in `domain.b` receives the query, it performs a query of `hash(domain.a)` in order to obtain the information related with the super-peers in `domain.a` through the Interconnection Overlay. Inside this query, the hash used in the other domain ($hash_a$) is included and a request for the desired item is built as `hash_a(user1@domain.a)` and sent to the super-peer in `domain.a`. The super-peer from domain `domain.a` forwards the information to its overlay and waits for the answer. Once the answer is received, the super-peer from `domain.a` forwards the resource information to the super-peer from `domain.b`. Finally, the super-peer from `domain.b` sends the response to the peer that generated the original query. This response path is the same that the request path in order to be compliant with the ongoing design in the IETF P2PSIP WG.

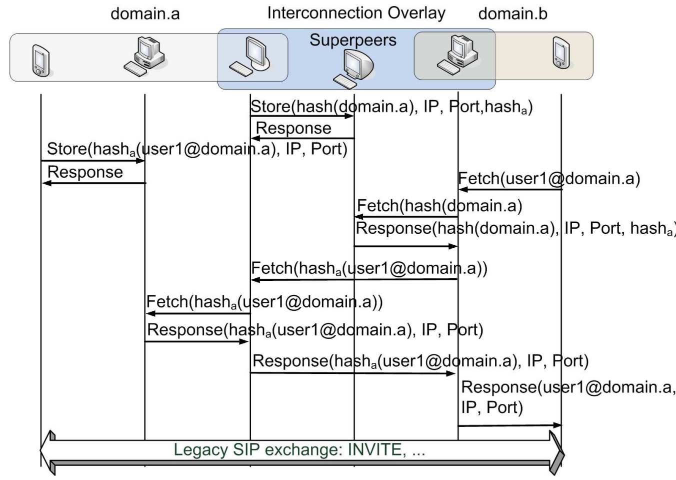


Fig. 3. H-P2PSIP Signalling

Additionally, the example shows a possible SIP usage of H-P2PSIP, since H-P2PSIP is used to locate the end-points of the communication, replacing the SIP proxy/registrar functionalities. Later, a direct legacy SIP point-to-point negotiation is established to setup the parameters for a VoIP call. Finally, we highlight that the signalling can be transmitted over TLS [23] /DTLS [24] to increase the security and to avoid URI tracking from adversaries.

D. Characteristics of H-P2PSIP

H-P2PSIP proposal has several advantages. In order to see these advantages, we compare it with a flat overlay network counterpart. First, the operations or primitives of the DHTs used in H-P2PSIP are not modified. The routing state in legacy peers does not increase with respect to a flat overlay network because the number of maintained peers per overlay network is the same. It is not necessary to store information of peers from other domains; peers only need to store their super-peer to reach them. Hence, the number of the peers in each domain limits the number of routing entries, although connectivity with other P2PSIP domains is possible, which was a prerequisite in the design rules. If we consider that the Routing State in a Peer-to-Peer network usually depends on the logarithm of the number of peers, we have the fact that the Routing State in our approach is $O(\log_B M)$ where M is the number of peers in a domain (B is the base of the ID space). If we compare this Routing State with a single flat P2PSIP domain that contains all P2PSIP domains, we obtain that the number of peers in the flat overlay network is $M \cdot K$, where K is the number of domains and the Routing State is increased up to $O(\log_B(M \cdot K))$. Thus, using the hierarchical architecture, legacy peers save $log_B(K)/log_B(M \cdot K) \cdot 100\%$ of overlay routing entries in comparison with the flat counterpart. In addition, the Routing State is independent with respect the number of P2PSIP domains, which is also interesting.

The drawback of this approach is the overload of super-peers [25]. This fact implies the maintenance of a larger amount of information with respect to the peers. Actually, our super-peers have to maintain two routing tables: a routing table of size $O(\log_B M)$ for their own domain and a routing table of size $O(\log_B K)$ for the Interconnection Overlay. In this case, the Routing State in super-peers is $O(\log_B M)+O(\log_B K) = O(\log_B(M \cdot K))$, this values corresponds with an equivalent flat overlay network with all peers of the different domains. Therefore, the only drawback with respect to a completely flat overlay network is the bandwidth and CPU consumption associated to the forwarding and processing of queries that belong to other domains.

IV. Implementation

Our implementation is based on the Peer-to-Peer Protocol (P2PP) [26], [27], from the University of Columbia. We choose this protocol because its implementation is available and it is being used as reference with other protocols in the IETF P2PSIP WG to design the ongoing P2PSIP protocol named RELOAD [28]. The full details of our implementation are too complex to be detailed here, but we highlight the key points that are absolutely necessary to provide to P2PP the functionalities of H-P2PSIP. The original design only supports legacy flat overlays. Therefore, it is necessary to add support to Hierarchical-IDs. In addition, it is also necessary to manage the new Hierarchical-IDs. Peers have to check if their queries target their own domain (same Prefix-ID). If the Prefix-ID is different, the query must be forwarded to a super-peer. We implement super-peer role by developing peers that are attached to the overlay network from their domain and the Interconnection Overlay. Super-peers forward the queries to the super-peer that belongs to the destination domain according to the defined signalling (Figure 3). One detail that is not implemented, in order to avoid more complexity in the development of our proposal, is the hashing method to map URIs to Hierarchical-IDs since this mapping is straightforward and it does not affect to the performance of our solution.

V. Scenario Setup

In order to test and develop our implementation, we have a configurable scenario based on network emulator software. The selected software is Modelnet [29] since offers an excellent infrastructure for the debugging and testing of distributed applications. Its emulation properties allow the repeatability of experiments, which helps in the debug process. In addition, it also allows emulating big core networks with different properties (delay, probability of error, etc); this feature allows evaluating the program in conditions closer to current Internet with a real TCP/IP stack.

A. Modelnet setup

Our implementation is tested in a Modelnet scenario running 1000 peers, which are grouped in a number of different domains ranging from 1 to 20 (all of them with the same number of peers). Scenarios with fewer peers are not considered since Peer-to-Peer networks are supposed to have at least thousands of peers. More peers are not considered due to the limitation of our computational resources. Each peer is deployed in an OpenVZ [30] virtual machine. The core network that interconnects the different peers is emulated according to the tools offered by Modelnet. Modelnet, depending on the computational capacity, can emulate any topology with different link properties. In order to have a topology as much realistic as possible, the measurements provided by the IEPM PingER project [31] are used to define the topology used in the experiments. Different developed representative countries are selected to build the core network of our experiment. With the information provided by the PingER project, the links among the different countries and their characterization (bandwidth, delay, losses, etc) are introduced in the Modelnet core emulator in order to have a reasonable representation of the current Internet. This setup allows the usage of a TCP/IP stack in an environment closer to the real world.

B. Peers setup

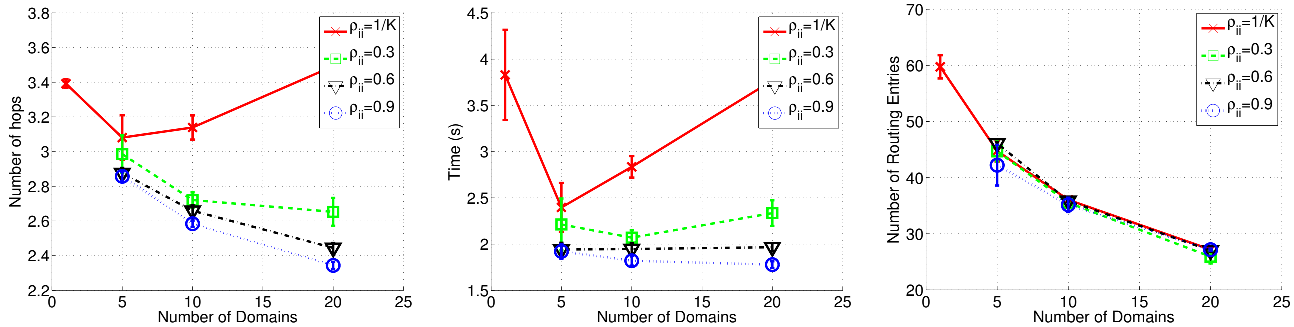
In our experiments, it is also necessary to reflect the peer behaviour. The session time distributions, and churn rate of peers are configured considering the measurements in [3]. However, we do not take the values in [3] directly but we adapt them to our population of 1000 nodes. Therefore, according to the previously mentioned paper, the churn rate is based on a negative binomial distribution. However, the average of the churn rate is scaled to the number of peers in our experiment. If we would not perform this operation, the churn rate would be very high for the number of peers in our experiments since the measurements in [3] are composed by a higher number of peers. For the query generation rate, a Poisson distribution is used since one of the most suitable scenarios for H-P2PSIP is VoIP and telephone calls, which are usually Poisson distributed. The protocol used in all the domains as well as in the Interconnection Overlay is Kademlia. This selection allows the comparison with the equivalent flat Kademlia counterpart of an overlay network with all peers in all domains. The Kademlia [8] implementation found in the P2PP protocol [27] has the following parameters, which are the same with respect to the original implementation: $B = 2$, implying that Kademlia performs routing based on a binary tree. The size of the buckets used to store the information of the routing entries for each level of the binary tree is $k = 20$. Kademlia can perform parallel queries to several peers when a query is being requested, the P2PP implementation uses $\alpha = 1$, thus the parallelization mechanism is not used. One super-peer per domain with high availability (churn does not affect super-peers) is used according to an operator-based scenario (operators take care of the high availability) of H-P2PSIP, which has been described previously.

VI. Results

This section presents the results obtained in scenario described previously; but, first of all, it is necessary to explain how experiments are conducted.

A. Experiments setup

The experiment is divided in two phases. First, we have a transitory phase limited to 30 minutes, during this time the



(a) Average number of hops (b) Average delay (c) Routing State in peers

Fig. 4. Results with respect to the number domains

average number of peers needed to realise the experiment join the hierarchical DHT network. The second phase, it is the stationary state and it has duration of 50 minutes. In this phase, 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. During this phase, the data are collected to obtain the results that are shown in the next section. The experiments are repeated several times (at least 10 times) in order to obtain results with representative 95% confidence intervals and an error smaller than 10% with respect to the estimated average of the results.

B. Routing Performance

Figure 4(a) shows the number of hops made to reach the destination with respect to the number of domains used to group the peers; this figure also shows this number of hops for different values of the intra-domain hit probability (ρ_{ii}). ρ_{ii} is the probability that a peer has of performing a query inside its own domain instead of any other external domain, this formulation comes from [9], [10]. The first important point is the fact that the expected number of hops is close and slightly better with respect to the simulations in [9], [10]. We explain this behaviour later. In addition, when the queries are randomly made among the different domains (ρ_{ii}=1/K), the red continuous line with cross markers), we observe how the number of hops starts to be smaller in comparison with the flat counterpart (first point of the plot that corresponds to K=1). Furthermore, as the number of domains increases, the number of hops increases slightly but very close with respect to the flat counterpart routing performance. This increment is produced since ρ_{ii}=1/K and the probability of looking in other domain increases, which implies an extra number of hops (one hop to reach the super-peer in addition to the number of necessary hops through the Interconnection Overlay). On the other hand, while the intra-domain hit probability increases (ρ_{ii}>1/K), the number of hops decreases since extra hops to reach other domains are unnecessary.

Figure 5(a) shows the number of hops made to reach the destination with respect to the intra-domain hit probability. This plot gives a view of the effects of the intra-domain hit probability. If the intra-hit domain probability increases, more queries are performed in the own domain of each peer and the average number of hops is smaller. Furthermore, the plot gives a clear comparison of the Routing Performance with respect to the flat counterpart, represented with a dotted blue line with round markers, that is clearly different. We see how the number of hops is smaller than the equivalent flat DHT.

We have also figures that show the average delay suffered by the queries in our scenario. Figure 4(b) represents the delay with respect to the number of domains and the figure includes the legend for different values of ρ_{ii}. We see that a certain correlation exists among the results in Figure 4(b) and Figure 4(a). When ρ_{ii}=1/K, the delay increases if the number of domains increases. The extra number of hops needed to reach the information in other domains causes this effect (Figure 4(a)). On the other hand, if ρ_{ii} increases, the average delay is reduced considerably since the number of queries to other domains is reduced and fewer hops are needed. Finally, Figure 5(b) shows how the delay depends more on ρ_{ii} rather than the number of domains.

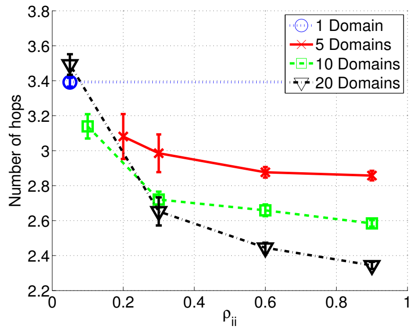
C. Routing State

In previous section, the Routing Performance and delay are analysed; nevertheless, it is also interesting to study the load sustained by peers to support the structure of the DHT overlay network in order to route the queries correctly. In order to do that, we collect the average number of routing entries of the peers in order to estimate the information that must be maintained by them. These routing entries not only consume memory on peers, they also imply a bandwidth consumption to maintain them updated. Therefore, the number of entries in peers also reflects partially the effort undergone by peers to assure the validity of their overlay routing information.

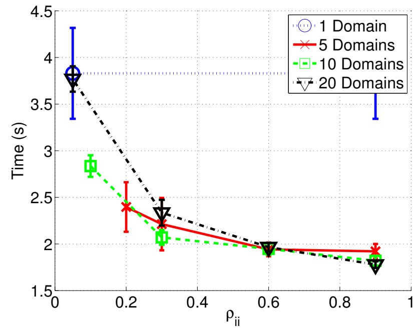
Figure 4(c) shows how the number of routing entries needed decreases if the number of domains increases. This effect is expected due to the limitations of our testbed, we must maintain a constant number of peers in our experiments. Thus, if the number of domains increases, the number of peers per domain decreases and consequently the number of routing entries per peer decreases.

On the other hand, Figure 5(c) demonstrates that the intra-domain hit probability (ρ_{ii}) has a negligible effect on the

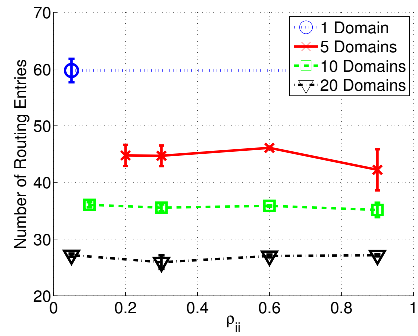
 ISBN: 978-1-61208-102-1



(a) Average number of hops

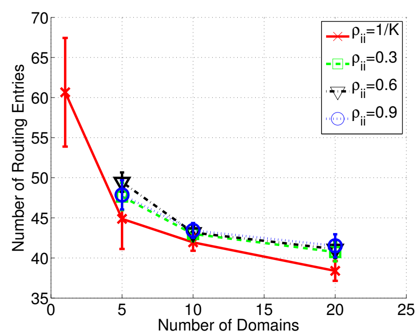


(b) Average delay

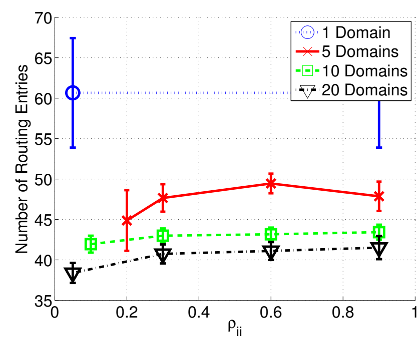


(c) Routing State in pees

Fig. 5. Results with respect ρ_{ii}



(a) With respect to the number of domains



(b) With respect to ρ_{ii}

Fig. 6. Routing State in super-peers

number of entries needed by every peer. The overlay routing tables must be maintained with independence of this parameter and they depend on the number of peers in the overlay network. If the plot is examined carefully, a slight decrease of the number of entries is observed for 5 domains and ρ_{ii}=0.9. However, it is appreciated in the figure how the confidence interval is bigger for this value (the error here is around 10%) and how the upper bound of the confidence interval is close to the other values and cannot be considered an atypical value. However, we plan to study this effect further in the future.

The number of routing entries in both figures is larger than the values obtained in the simulations. Therefore, the Kademlia implementation in P2PP stores in average a higher number of routing entries rather than the Kademlia protocol in the simulation. This fact explains the previously observed better routing performance of the emulated experiment with respect to the simulated experiment. More populated routing tables in peers allow finding the destination in fewer hops. This increment in the average number of routing entries is motivated by some differences in the procedure to update the overlay routing tables, which can be expected from different implementations. Nevertheless, the obtained results are considered valid since the number of routing entries is among the theoretically expected values [8].

In addition to these results, Figure 6(a) illustrates the Routing State information of super-peers. The super-peers must maintain the overlay routing table associated to the Interconnection Overlay, in addition to the own overlay. In fact, Figure 6(a) shows the average number of routing entries taking into account both routing tables. The size of the overlay routing table that belongs to the own domain is quite similar to the given in Figure 4(c). The extra number of entries corresponds to the overlay routing table of the Interconnection Overlay. The difference is not very large since the number of interconnected domains is small due to the limitations of our testbed. In Figure 6(a), we see how the number or routing entries decreases if the number of domains increases and how the intra-domain hit probability affects negligibly to these values in Figure 6(b). These results also give an overview about the requirements in peers to support our proposal.

VII. Conclusions and Future Work

This paper presents the results obtained from our implementation of H-P2PSIP based on the P2PP [27] protocol. The evaluation of this implementation is based on an experiment with 1000 nodes running on a Modelnet emulation framework using the information from PingER project.

In detail, if the queries are randomly distributed among the different domains, the required number of hops and delay increases if the number of domains increases, but they are always below the flat overlay counterpart. This fact is really interesting since the size of routing tables to achieve this performance is smaller (both peers and super-peers) in comparison with the equivalent flat overlay network. This fact is a good advantage

 ISBN: 978-1-61208-102-1

over the traditional flat overlay networks. However, the main advantage of our design is the fact that any DHT overlay network can be used in the H-P2PSIP architecture. Therefore, each domain chooses the overlay network that suits better its requisites, which offers a great flexibility for any deployment.

Our implementation demonstrates that the interoperability among different overlay networks is possible, although the price of this feature is the dependency on super-peers.

Our future work plan consists on performing measurements with our testbed to quantify the exchanged traffic depending on the number of domains to evaluate accurately the maintenance scalability of our solution with respect to the bandwidth perspective. In addition, we also plan to perform measurements with different number of peers per domain in order to be closer to a real operation environment. We also plan to study super-peer selection mechanisms for DHTs networks to make feasible our solution in community network scenarios.

VIII. Code Availability

The code of H-P2PSIP developed until this moment is available at http://hdl.handle.net/10016/8382

Acknowledgments

We thank A. Bikfalvi, M. Gramaglia, I. Soto and anonymous reviewers for their insightful comments to improve this paper.

References

[1] W. John, S. Tafvelin, and T. Olovsson, "Trends and differences in connection-behavior within classes of internet backbone traffic," *Passive and Active Network Measurement*, pp. 192–201, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-79232-1_20

[2] Y. Kulbak, D. Bickson, A. Prof, and S. Kirkpatrick, "The emule protocol specification," DANSS (Distributed Algorithms, Networking and Secure Systems) Lab School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Tech. Rep., 2005.

[3] M. Steiner, T. En-Najjary, and E. W. Biersack, "A global view of kad," in *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM, 2007, pp. 117–122.

[4] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, *Peer-to-Peer Systems IV, LNCS.* Springer, 2005, vol. 3640, ch. The Bittorrent P2P File-Sharing System: Measurements and Analysis, pp. 205–216, 10.1007/11558989_19. [Online]. Available: http://dx.doi.org/10.1007/11558989_19

[5] D. Erman, D. Ilie, and A. Popescu, "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*, vol. 1, pp. 61–84, 2009.

[6] S. A. Baset and H. G. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–11.

[7] D. Rossi, M. Melia, and M. Meo, "A detailed measurement of skype network traffic," in *In IPTPS 2008*, 2008.

[8] P. Maymounkov and D. Mazieres, *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, ser. Lecture Notes in Computer Science. Springer, 2002, vol. 2429/2002, ch. Kademlia: A peer-to-peer information system based on the XOR metric, pp. 53–65. [Online]. Available: http://www.springerlink.com/content/2ekx2a76ptwd24qt/?p=d9a88bd0609d4ac3902f147d1c183345&pi=0

[9] I. Martinez-Yelmo, A. Bikfalvi, R. Cuevas, C. Guerrero, and J. Garcia, "H-p2psip: Interconnection of p2psip domains for global multimedia services based on a hierarchical dht overlay network," *Computer Networks (Special Issue on Content Distribution Infrastructures for Community Networks)*, vol. 53, no. 4, Mar. 2009. [Online]. Available: http://hdl.handle.net/10016/3390

[10] I. Martinez-Yelmo, A. Bikfalvi, C. Guerrero, R. Cuevas, and A. Mauthe, "Enabling global multimedia distributed services based on hierarchical dht overlay networks," *International Journal of Internet Protocol Technology (Special Issue on Future Multimedia Networking)*, vol. 3, no. 4, Dec. 2008. [Online]. Available: http://hdl.handle.net/10016/4300

[11] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "A sip usage for reload," March 2010, internet Draft draft-ietf-p2psip-sip-04.txt.

[12] Y. Gao and Y. Meng, "A new sip usage for reload," July 2009, internet Draft draft-gaoyang-p2psip-new-sip-usage-00.txt.

[13] S.-K. Ko, Y.-H. Kim, S.-H. Oh, B.-T. Lee, and V. P. Avila, "Iptv usage for reload," July 2009, internet Draft draft-softgear-p2psip-iptv-02.txt.

[14] J. Wang, J. Shen, and Y. Meng, "Content sharing usage for reload," June 2009, internet Draft draft-shen-p2psip-content-sharing-00.txt.

[15] L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller, "Hierarchical p2p systems," in *Euro-Par 2003 Parallel Processing*, Springer, Ed., vol. 2790, 2004, pp. 1230–1239.

[16] Z. Xu, R. Min, and Y. Hu, "Hieras: a dht based hierarchical p2p routing algorithm," in *Parallel Processing, 2003. Proceedings. 2003 International Conference on*, 2003, pp. 187–194.

[17] A. Mislove and P. Druschel, "Providing administrative control and autonomy in structured peer-to-peer overlays," *Peer-to-Peer Systems III*, pp. 162–172, 2005. [Online]. Available: http://www.springerlink.com/content/XR2X7MXUFB2FRYEX

[18] M. Artigas, P. Lopez, J. Ahullo, and A. Skarmeta, "Cyclone: a novel design schema for hierarchical dhts," in *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on*, Aug.-2 Sept. 2005, pp. 49–56.

[19] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in g major: Designing dhts with hierarchical structure," *icdcs*, vol. 00, pp. 263–272, 2004.

[20] S.-H. Min, J. Holliday, and D.-S. Cho, "Optimal super-peer selection for large-scale p2p system," in *Hybrid Information Technology, 2006. ICHIT'06. Vol 2. International Conference on*, vol. 2, 2006, pp. 588–593.

[21] A. T. Mizrak, Y. Cheng, V. Kumar, and S. Savage, "Structured superpeers: leveraging heterogeneity to provide constant-time lookup," in *Internet Applications. WIAPP 2003. Proceedings.*, 2003, pp. 104–111.

[22] M. Sanchez-Artigas, P. Garcia-Lopez, and A. F. G. Skarmeta, "On the feasibility of dynamic superpeer ratio maintenance," in *P2P '08: Proceedings of the 2008 Eighth International Conference on Peer-to-Peer Computing*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 333–342.

[23] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246 (Proposed Standard), Internet Engineering Task Force, Aug. 2008, updated by RFC 5746. [Online]. Available: http://www.ietf.org/rfc/rfc5246.txt

[24] T. Phelan, "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)," RFC 5238 (Proposed Standard), Internet Engineering Task Force, May 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5238.txt

[25] B. Beverly Yang and H. Garcia-Molina, "Designing a super-peer network," in *Data Engineering, 2003. Proceedings. 19th International Conference on*, 2003, pp. 49–60.

[26] "P2PP webpage," last access date: 2010-07-13. [Online]. Available: http://www1.cs.columbia.edu/~salman/peer/

[27] S. Baset, H. Schulzrinne, and M. Matuszewski, "Peer-to-peer protocol (p2pp)," November 2007, internet Draft draft-baset-p2psip-p2pp-01.txt.

[28] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "Resource location and discovery (reload) base protocol," July 2010, internet Draft draft-ietf-p2psip-base-09.txt.

[29] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostic, J. Chase, and D. Becker, "Scalability and accuracy in a large-scale network emulator," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 271–284, 2002.

[30] "OpenVZ webpage," last access date: 2010-07-13. [Online]. Available: http://wiki.openvz.org

[31] W. Matthews and L. Cottrell, "The pinger project: active internet performance monitoring for the henp community," *Communications Magazine, IEEE*, vol. 38, no. 5, pp. 130 –136, may 2000.