

Routing Performance in a Hierarchical DHT-based Overlay Network

Isaias Martinez-Yelmo, Ruben Cuevas, Carmen Guerrero
Departamento de Ingeniería Telemática
Universidad Carlos III de Madrid
Av. Universidad 30. 28911 Leganés. Madrid (Spain)
{imyelmo,rcuevas,guerrero}@it.uc3m.es

Andreas Mauthe
Computing Department at InfoLab 21
Lancaster University
Lancaster LA1 4WA (UK)
andreas@comp.lancs.ac.uk

Abstract

The scalability properties of DHT based overlay networks is considered satisfactory. However, in large scale systems this might still cause a problem since they have a logarithmic complexity depending. Further, they only provide a one dimensional structure and do not make use on inherent clustering properties of some applications (e.g. P2PVoIP or locality aware overlays). Thus, structures based on a hierarchical approach can have performance as well as structural advantages. In this paper, a generic hierarchical architecture based on super-peers is presented where a peer ID is composed by a Prefix ID and a Suffix ID. Prefix ID is only routed at the super-peer level and the Suffix ID at the peer level. We specifically analyse the Routing Performance of this approach within the context of two specific overlays, viz. CAN and Kademia.

1 Introduction

DHT overlay networks have been extensively studied by the research community in recent years. Their basic properties are to maintain a structure between all the peers in the overlay in order to route queries to the destination. Some of these DHT overlays are Content Addressable Network (CAN) [15], Kademia [10], Chord [18] or Pastry [16]. Usually, these networks require $O(\log N)$ peer hops to reach the desired destination and $O(\log N)$ routing entries to maintain the desired structure [9]. A detailed comparison of these overlays with their characteristics, advantages and drawbacks can be found in [9].

Although DHT overlay networks are considered scalable because of their logarithmic increase in costs depending on the number of peers, it is also well known that hierarchical architectures can improve this for complex systems [17]. A hierarchical architecture can be used on an overlay network using the concept of super-peer. As shown in [1] this has advantages but also drawbacks.

In this paper a simple hierarchical architecture is presented where the IDs of the DHT network are composed by a Prefix ID and a Suffix ID. The Prefix ID is only used on the super-peer domain and the Suffix ID is used under the cluster of peers with the same Prefix ID. Over this design, we propose a very simple model to obtain the Routing Performance (R.P.) or number of hops that can be obtained with this architecture. Two case studies are presented, a hierarchical CAN overlay and a Kademia hierarchical overlay.

The rest of the paper is structured as follows. In Section 2 the related work associated with this paper is presented. Section 3 presents the proposed hierarchical architecture and the Routing Performance expression that can be obtained with this architecture. Two examples are given for this architecture, a hierarchical CAN overlay network in Section 4 and a hierarchical Kademia network in Section 5. Finally, in Section 6, the conclusions of this work are presented as well as the future work.

2 Related Work

A number of papers have been dealing with hierarchical overlay networks to improve the performance of canonical overlay networks. Thus, it is necessary to take into account the different trade-offs that arise with these types of architecture and what has to be addressed [4] when adopting a hierarchical approach. Nevertheless, the benefits of hierarchical architectures have been widely studied [7], [8].

In [6] and [22], it is explained how a hierarchical architecture based on a Chord overlay network can be used to improve the routing performance. In [6] super-peers save the information of all their leaves while peers just send keep-alives messages to their super-peer. This fact implies that super-peers are put more under stress (by maintaining pointers to the peers, keeping ongoing connections and query processing if it is necessary) if the number of peers increases. However, the main concern of this analysis is the consideration of the churn rate. On the other hand, in [22], an analysis of the costs of super-peers on a hierar-

chical structure is done. Again, peers in each cluster do not maintain any structure and depend only on their super-peer.

Nevertheless, additional work has been done on hierarchical overlay networks. In [19], a low delay hierarchical overlay network based on Chord is proposed. The drawback is the high maintenance state needed (memory, CPU and bandwidth) because all the peers in the overlay are attached to different levels of the hierarchy. A less aggressive design is presented on [5] where a hierarchical structured is built with the constraint of limiting the maintenance cost to the canonical (flat) counterpart. This approach improves the latency of the overlay. Furthermore, some simulation studies of hierarchical CAN overlay networks have been presented in the literature [20], [21]. These approaches present a better Routing Performance rather than the canonical overlay.

3 Hierarchical DHT-based Overlay Networks

The proposed hierarchical architecture is based on creating to different domains, the cluster domain and the inter-cluster domain. If a peer wants to retrieve information of a different cluster, it must route its query to super-peers which maintain the reachability between the different clusters.

The overlay network is divided on K different clusters with several overlay networks. Furthermore, on each overlay network, there is a super-peer that is also part of an interconnection overlay network between all the super-peers of the different clusters. Any cluster can be reached through this interconnection overlay using the super-peers. The IDs managed on this hierarchical overlay architecture have two fields a Prefix and a Suffix (cf. Figure 1).

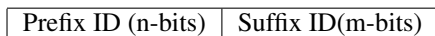


Figure 1. Hierarchical ID

Peers on a cluster use the Suffix ID for the cluster overlay network. On the other hand, the super-peers route the queries along the interconnection overlay using the Prefix ID that can be also considered as the cluster ID. Thus, if a peer needs to find a peer with a different Prefix ID to its own cluster, the query is sent to its super-peer. This super-peer can route the query to the super-peer that manages the cluster where is placed the destination peer.

A key factor for this architecture is that can be used for file sharing networks using the Prefix ID as a Group of Interest (GoI) ID based on social networking [3], [12], [13]. This approach can be also used in a P2P VoIP environment, especially in relation with the work on the P2PSIP IETF Working Group ¹ which is working on an open standard

¹<http://www.p2psip.org>

for p2p VoIP applications [2]. Our architecture can be applied to interconnect the different P2PSIP domains using the super-peer overlay where the Prefix ID plays the role of domain ID and the Suffix ID is the P2PSIP client ID.

This architecture reduces the information needed on the super-peers in comparison with [6], [22]. The information of all the peers on the cluster is not necessary to be stored by the super-peer, only those peers needed for the routing in the overlay of the cluster. Cluster peers obtain also advantages because they need to store and maintain less information that in the canonical counterpart, a cluster peer only has to take care about the peers of its own cluster. Thus, cluster peers have to maintain approximately an overlay of N/K peers (where K is the number of clusters), whereas in the canonical overlay the number of peers is N .

The reduction of routing entries with this architecture in the cluster peers is approximately $\frac{\log K}{\log N} \cdot 100\%$. This reduction not only implies less memory and CPU consumption, the bandwidth needed for the maintenance of the routing entries is also reduced. This is an advantage in comparison with [5], [19], [20], [21]. This fact is more relevant if handheld devices are considered, for instance in a P2PSIP environment. On the other hand, the cost is the overload suffered by the super-peers [1]. They have to support N/K times more bandwidth and queries.

Along this paper, it is supposed that a mechanism exists for the selection of a super-peer in each cluster [11], [14].

3.1 Generic Routing Performance

The following conventions and assumptions that are used throughout the rest of the paper:

- *Lower overlay*: It is the overlay maintained on each cluster. Only peers belonging to the cluster can be reached directly through this overlay.
- *Interconnection overlay*: This overlay network is formed by all the super-peers at the upper level of the hierarchy. It allows to reach all the clusters.
- N : The total number of peers in the full hierarchical overlay network.
- K : The number of clusters in the hierarchy. It can be also considered as the number of peers in the interconnection overlay because only one super-peer per lower overlay is considered.
- N/K : The number of peers at each cluster. It is *assumed* that all peers are equally distributed along all the different clusters to simplify the analysis.
- $1/K$: The probability of looking up a peer which belongs to the same cluster. It is assumed that all the queries are equally distributed along the clusters.

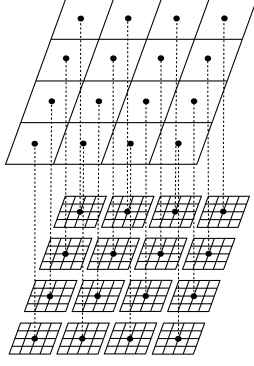


Figure 2. Hierarchical CAN overlay network

- $C(x)$: The number of hops needed to find a super-peer on the interconnection overlay depending on the number of super-peers x . This value depends on the type of overlay used.
- $D(x)$: The number of hops needed to find a peer on a lower overlay depending on the number of peers x belonging to the cluster.
- It is also assumed that all the peers in a lower overlay know their super-peer on the interconnection overlay. This fact implies that only *one hop* is needed to reach the super-peer.

Taking into account the above considerations, the Routing Performance (R.P.) of this DHT-based hierarchical overlay networks is obtained. Let us define the cost of finding a peer on each overlay:

- $D(N/K)$: The cost of find a peer on the sub-overlay of a cluster.
- $C(K)$: The cost of find a super-peer on the interconnection overlay.

If the probability of obtaining information related with the lower overlay from a super-peer is considered negligible, because this value is K/N and $N \gg K$, the routing performance can be written as:

$$\begin{aligned} R.P. &= \frac{1}{K}D\left(\frac{N}{K}\right) + \frac{K-1}{K}\left[1 + D\left(\frac{N}{K}\right) + C(K)\right] = \\ &= D\left(\frac{N}{K}\right) + \frac{K-1}{K}[1 + C(K)] \end{aligned} \quad (1)$$

With the last formula as reference, the performance on a hierarchical CAN overlay network as well as on a Kademlia overlay network is studied on the next sections.

4 Hierarchical CAN Model

In this section we study the performance of a hierarchical CAN overlay with two levels as it is shown in Fig.2. If $C(x) = d_i x^{\frac{1}{d_i}}$ and $D(x) = d_l x^{\frac{1}{d_l}}$ are used on Eq.1, we have that the routing performance (R.P.) is:

$$R.P. = d_l \left(\frac{N}{K}\right)^{\frac{1}{d_l}} + \frac{K-1}{K} \left(1 + d_i K^{\frac{1}{d_i}}\right) \quad (2)$$

Eq.2 represents the routing performance for a hierarchical CAN overlay where d_l is the number of dimensions for the lower overlays and d_i is the number of dimensions for the interconnection overlay. Each value can be optimised independently considering the number of peers at each level. Thus, it can be stated that $d_l = \ln \frac{N}{K}$ and $d_i = \ln K$ (these values can be obtained minimising the Routing Performance of the canonical CAN overlay network [15]). However, if the development of a hierarchical CAN based application is considered, it would be reasonable to have the same value of d in both levels of the hierarchy because just one version of CAN needs to be developed and the code can be easily reused. Taking this fact into account, Eq.3 presents the routing performance when only one value of d is considered:

$$R.P. = d \left(\frac{N}{K}\right)^{\frac{1}{d}} + \frac{K-1}{K} \left(1 + dK^{\frac{1}{d}}\right) \quad (3)$$

Thus, the optimum configuration parameters for the hierarchical CAN overlay network and some general design rules have been obtained. Two parameters can be modified in order to get a satisfactory system performance: the number of lower overlays K and the number of dimensions d of the overlay network.

Theorem 1. *If $K \gg 1$, then the optimum value of K is $K \simeq \sqrt{N}$.*

Proof. If $K \gg 1$, Eq.3 can be rewritten as:

$$R.P. = f(K) = d \left(\frac{N}{K}\right)^{\frac{1}{d}} + 1 + dK^{\frac{1}{d}} \quad (4)$$

In order to obtain the best configuration parameter for K , the first derivate respect to K is:

$$f'(K) = -\frac{1}{K} \left(\frac{N}{K}\right)^{\frac{1}{d}} + K^{\frac{1}{d}-1} \quad (5)$$

$f'(K)$ is equal to 0 when $K = \sqrt{N}$. This point will be a minimum if the second derivate is positive at this point. Thus, the second derivate is obtained:

$$f''(K) = \left(1 + \frac{1}{d}\right) \frac{1}{K^2} \left(\frac{N}{K}\right)^{\frac{1}{d}} + \left(\frac{1}{d} - 1\right) K^{\frac{1}{d}-2} \quad (6)$$

If $f'(K) = 0$ is a minimum, it implies that $f''(K) > 0$ and the following formula can be deduced:

$$\begin{aligned} \left(1 + \frac{1}{d}\right) \frac{1}{K^2} \left(\frac{N}{K}\right)^{\frac{1}{d}} &> \left(1 - \frac{1}{d}\right) K^{\frac{1}{d}-2} \\ \frac{N}{K^2} &> \left(\frac{d-1}{d+1}\right)^d \end{aligned} \quad (7)$$

Considering that $f'(K) = 0$ when $K \simeq \sqrt{N}$ and this value of K is substituted on Eq.7, it is obtained that:

$$1 > \left(\frac{d-1}{d+1}\right)^d \quad (8)$$

So, Eq.8 is valid for $\forall d \in (1, \infty)$ and it can be said that $K = \sqrt{N}$ is a minimum. \square

This result is important not only because it gives a numerical value for the optimal number of lower overlays K in the hierarchical CAN overlay network, but also due to the fact that this value is *independent* of the number of dimensions d .

Theorem 2. *Only one value of d minimises the number of hops on a hierarchical CAN overlay network with N peers and K lower overlays.*

Proof. The first derivate of Eq.3 applied to d is used calculate if a minimum exists:

$$\begin{aligned} f'(d) &= \left(\frac{N}{K}\right)^{\frac{1}{d}} \left(1 - \ln\left(\frac{N}{K}\right)^{\frac{1}{d}}\right) + \\ &+ \frac{K-1}{K} K^{\frac{1}{d}} \left(1 - \ln K^{\frac{1}{d}}\right) \end{aligned} \quad (9)$$

If we take into account that $d \in (1, \infty)$, then:

$$f'(d=1) = \frac{N}{K} \left(1 - \ln \frac{N}{K}\right) + (K-1)(1 - \ln K) \quad (10)$$

and

$$\lim_{d \rightarrow \infty} f'(d) = 1 + \frac{K-1}{K} \quad (11)$$

Thus, if $\frac{N}{K}, K > e$ (this fact occurs always, $K=2$ has no sense on our scenario), then $f'(d=1) < 0$ and

$f'(d=\infty) > 0$ which implies that one or more solutions exist for $f'(d) = 0$ in our range of work.

The next step is to prove the unicity of the solution. Then, the second derivate respect to d is calculated:

$$f''(d) = \frac{1}{d} \left(\frac{N}{K}\right)^{\frac{1}{d}} \left(\ln\left(\frac{N}{K}\right)^{\frac{1}{d}}\right)^2 + \frac{K-1}{K} \frac{K^{\frac{1}{d}}}{d} \left(\ln K^{\frac{1}{d}}\right)^2 \quad (12)$$

Taking into account again that $f''(d) > 0, \forall d \in (1, \infty)$, then $f'(d)$ is a monotonous increasing function and there is only one solution of $f'(d) = 0$ and this value is a minimum. \square

Theorem 3. *If the optimum value of K is used ($K \simeq \sqrt{N}$), then $d = \ln \frac{N}{K} = \ln K$ is the optimum value of d parameter.*

Proof. The following variable substitution can be done:

$$A = (N/K)^{\frac{1}{d}} \quad (13)$$

$$B = K^{\frac{1}{d}} \quad (14)$$

The above substitution when applied to Eq.9 results with:

$$f'(d) = A(1 - \ln A) + B(1 - \ln B) \quad (15)$$

Furthermore, if A and B are transformed to find a relation between them:

$$\frac{1}{d} = \frac{\ln A}{\ln(N/K)} \quad (16)$$

$$\frac{1}{d} = \frac{\ln B}{\ln K} \quad (17)$$

Equalising the last two equations we have:

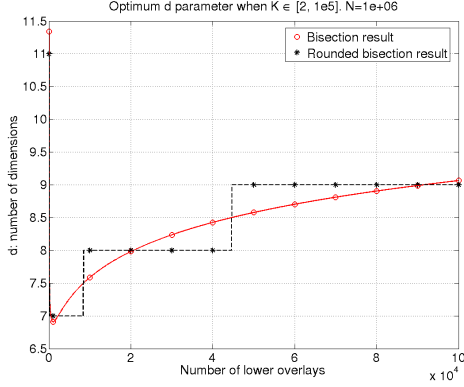
$$\ln B = \frac{\ln K}{\ln(N/K)} \ln A \quad (18)$$

If a variable x is defined as $x = \frac{\ln K}{\ln(N/K)}$, it is obtained that $B = A^x$. With this result, we can rewrite Eq.15 as:

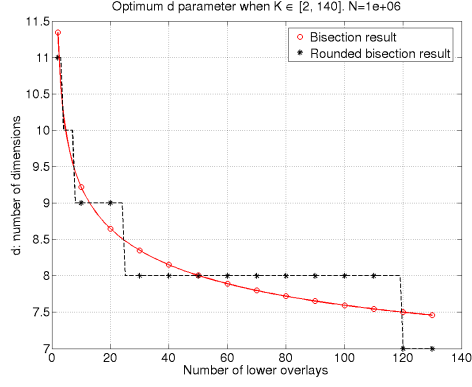
$$f'(d) = A(1 - \ln A) + \frac{K-1}{K} A^x (1 - \ln A^x) \quad (19)$$

$f'(d) = 0$ cannot be solved analytically but can be solved using the bisection method. Nevertheless, if it is considered that the optimum value for K is $K = \sqrt{N}$, then $\frac{N}{K} = K$ and $x = 1$. Thus, Eq.19 can be expressed as:

$$f'(d) = \left(1 + \frac{K-1}{K}\right) A(1 - \ln A) \quad (20)$$



(a) Optimum value of d for a wide range of K



(b) Optimum value of d for small values of K

Figure 3. Optimum number of dimensions depending on the number of lower overlays

For $f'(d) = 0$, it is obtained that $A = e$. If this value is used on Eq.16, then we have that $d = \ln(N/K) = \ln(K)$ and *Theorem 3* is proved. \square

Corollary 1. *The best routing performance that can be found for a hierarchical CAN overlay is*

$$R.P. = \frac{\sqrt{N} - 1}{\sqrt{N}} + \left(1 + \frac{\sqrt{N} - 1}{\sqrt{N}}\right) \ln \sqrt{N} \sqrt{N}^{\frac{1}{\ln \sqrt{N}}}$$

Proof. The best R.P. can be obtained if *Theorem 1* (optimum value of K) and *Theorem 3* (optimum value of d) are applied to Eq.3. \square

Corollary 2. *If a hierarchical CAN overlay is configured with its optimum parameters, the number of dimensions d is a half of the number of dimensions for the optimum flat counterpart.*

Proof. The proof is obtained by mathematical manipulation of the result given on *Theorem 3*:

$$d_{\text{hierarchical}} = \ln K = \ln \sqrt{N} = \frac{1}{2} \ln N = \frac{1}{2} d_{\text{flat}} \quad (21)$$

\square

Corollary 2 means that only half of the maintenance state of the canonical CAN overlay network is needed for its hierarchical counterpart.

Corollary 3. *If $\sqrt{N} \gg 1$, the optimum routing performance of a hierarchical CAN overlay network is one hop greater than the optimum routing performance of a flat CAN overlay network.*

Proof. If we take the best routing performance that can be achieved to hierarchical CAN overlay network from *Corollary 1* and given that $\sqrt{N} \gg 1$:

$$\begin{aligned} R.P. &= 1 + 2 \ln \sqrt{N} \sqrt{N}^{\frac{1}{\ln \sqrt{N}}} = 1 + \ln N N^{\frac{1}{\ln N}} = \\ &= 1 + d_{\text{flat}} N^{\frac{1}{d_{\text{flat}}}} = 1 + R.P._{\text{flat}} \end{aligned} \quad (22)$$

\square

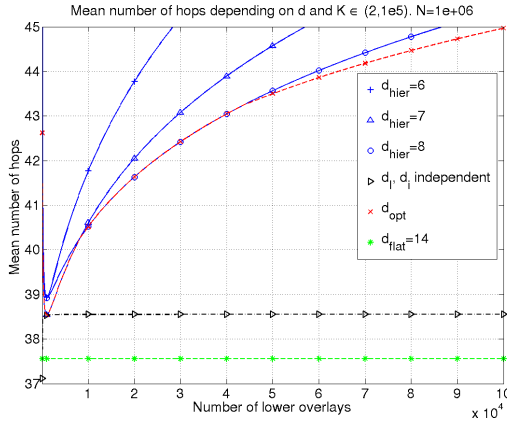
Thus, the routing performance of a hierarchical CAN overlay network cannot be better than the flat routing performance according to *Corollary 3*, although the difference is small.

4.1 Hierarchical CAN Design Rules

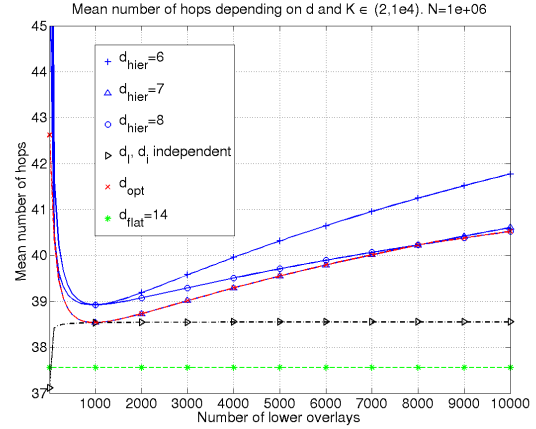
Theorem 3 can be used when a distributed application based on a hierarchical overlay network is able to change the number of clusters dynamically. For instance, in a file sharing network the number of predefined lower overlays could be configured dynamically according to the number of peers attached on the network if some mechanism is provided for this purpose.

Eq.19 is useful when the number of lower overlays cannot be configured dynamically and it is fixed in advance. One example of this scenario could be the P2PSIP scenario. In this scenario, the number of groups is given by the number of domains subscribed to the interconnection overlay. Thus, it would be interesting to know the range that parameter d can adopt around the optimal configuration value.

In order to understand the importance of the number of dimensions d on a hierarchical CAN overlay network Eq.19 has been solved using the numerical method of bisection. The value of $N = 10^6$ has been used for this analysis and a



(a) Routing Performance for a wide range of K



(b) Routing Performance zoom around the minimum

Figure 4. Routing performance for hierarchical CAN overlay

large range of K values has been explored. The results are shown on Fig.3(a) where the solid line is the result of the bisection method and the dashed line is the nearest integer to the bisection results for each value of K . It can be observed how the value of d changes depending on K . The value of $d = 7$, which is the value for the best performance point according to Theorem 3, is reached quickly as it can be seen in Fig.3(b). This value of d is valid for $K \in (120, 10^4)$. This fact implies a wide range of work along K for the optimum value of d .

Furthermore, in order to see the performance of a hierarchical CAN overlay, the routing performance complexity for an optimum flat CAN overlay is plotted in Fig.4 with a dashed line with asterisks ($d_{flat} = 14$), the routing performance for a hierarchical CAN overlay network with the optimum values of d_l and d_i in a dashdot line with triangles and the routing performance of a hierarchical CAN overlay is also plotted when the same value of d is used on both levels of the hierarchy with solid lines, several markers have been used for the different values of d as it is indicated on the legend. In addition to this information, the routing performance for the best configuration for the d parameter according to Eq.19 and Fig.3(a) has been plotted on a dashed line with crosses. The x axis represents the number of K lower overlays and the legend of each figure explains what values of d have been used for each line.

The values used to generate Fig.4(a) are $N = 10^6$ nodes and $K \in [2, 1e5]$. In this figure the behaviour of the routing performance is shown along a wide range of K . The figure shows how an optimum flat CAN overlay ($d = 14$) has a better performance than its hierarchical counterpart but it needs twice the number of dimensions of a hierarchical CAN overlay ($d = 7$). Nevertheless, the difference for this

optimum value is just one hop, the hop needed to send the query from the cluster peer to the super-peer. The cost of routing the query through the interconnection overlay and the foreign cluster overlay is the same that the canonical counterpart.

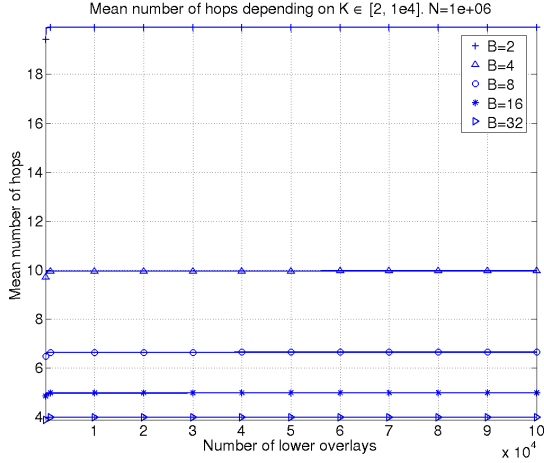
The minimum predicted by Theorem 3 can be seen clearly on Fig.4(b). An important fact is that the dashed line and the dashdot line meet on this point. The dashdot line is the routing performance when each overlay is configured with the optimum value of d independently. This fact means that around the optimum of d we have $d = d_i = d_l$. Thus, $d \simeq \ln \frac{N}{K} \simeq \ln K$ and $N \simeq K^2$ as it has been previously predicted on Theorem 3. This value is around $K = 1000$ as expected. Moreover, it can be observed that for $K \in [100, 10000]$, the routing performance only decreases two hops respect to the optimum value. Thus, the optimum configuration parameter of d has a wide range of work along K .

5 Hierarchical Kademia Model

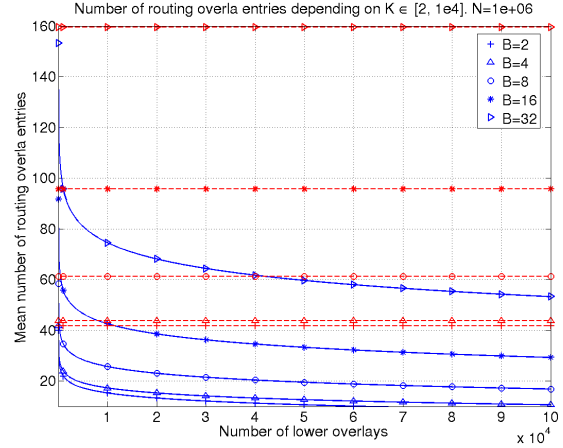
Kademia has a logarithmic dependency in the routing performance and routing state with the number of peers on the overlay network [10]. We are going to apply the same methodology that it has been applied to CAN. Thus, introducing this logarithmic dependency in Eq.1 as $C(x) = D(x) = \log_B x$, the following expression can be derived:

$$R.P. = \log_B \left(\frac{N}{K} \right) + \frac{K-1}{K} (1 + \log_B K) \quad (23)$$

If $K \gg 1$ and considering the properties of the logarithm, it can be written:



(a) Routing Performance



(b) Optimum value of d for small values of K

Figure 5. Routing overlay entries

$$R.P. = \log_B(N/K) + 1 + \log_B K = 1 + \log_B N \quad (24)$$

Thus, the routing performance is almost equal for hierarchical or flat overlays. The routing performance depending on B (the bucket size) has been plotted in Fig.5(a).

However, the routing state for the peers belonging to the lower layer decreases to $\log_B(N/K)$. The evolution of the routing state as function of K can be seen in Fig.5(b). If the number of K lower overlays is large enough, the routing state, in comparison with the flat approach, can be reduced by more than the 40%.

With this hierarchical architecture, routing performance can be improved by increasing the value of the parameter B as on the flat counterpart. However, the peers can be grouped in K different clusters in order to avoid an increment on the maintenance state at the cluster peers for the routing overlay tables. The number of routing entries in a canonical Kademlia network with a bucket size of $B = 2$ is the same that a hierarchical Kademlia network with $B = 16$ and the number of clusters $K = 10^4$. Nevertheless the number of hops for the first case is 19 and for the second one only 5. Only super-peers need to maintain a larger state because they have to store the information of their lower overlay and also the information of the interconnection overlay. Thus, a hierarchical Kademlia overlay network improves the trade-off between the routing performance and the maintenance state on cluster peers.

6 Conclusions and Future Work

In this paper a hierarchical DHT-based overlay network architecture where peers are divided into different clusters is proposed. On each cluster an overlay is maintained and a super-peer is selected representing the cluster. The super-peers maintain a second overlay that interconnects the different clusters. This concept is assessed analytically. A generic expression for the Routing Performance is given on Eq.1. In order to simplify the evaluation the same type of overlay is used on both, the lower overlay networks and the interconnection overlay network. Two case examples are studied, viz. a hierarchical CAN overlay and a hierarchical Kademlia overlay network.

In relation with the hierarchical CAN overlay the optimum configuration parameters to get the best Routing Performance on this type of architecture are analysed. The configuration parameters are the number of clusters K and the number of dimensions d . The optimal value of K is $K = \sqrt{N}$ where N is the number of peers. On the other hand, the optimal value of d depends on K and N (Eq.19).

The Routing Performance for hierarchical CAN and Kademlia obtained by this architecture is quite similar than the canonical counterpart. For the optimal configuration parameters is one more hop due to the fact that a extra hop is needed to send a query to the cluster super-peer for a inter-cluster search. Nevertheless, the cluster peers reduce their routing tables and maintenance bandwidth but super-peers increases the supported bandwidth because inter as well as intra-cluster queries have to be processed by them.

Particularly, the Routing Performance is one hop greater again in hierarchical Kademlia network rather than for the

canonical counterpart because of the extra hop need to send a query to the cluster super-peer. The maintenance state in cluster peers can be reduced if the proposed hierarchical approach is adopted.

Following this initial study we will apply this hierarchical architecture to other types of overlay networks and perform a simulation-based analysis to validate the results. It will also be analysed how hierarchical DHT-based overlay networks behave under churn.

Acknowledgement

We would like to acknowledge Albert Banchs, Antonio de la Oliva, Piotr Pacyna and Manuel Urueña for their insightful comments.

This work has been partially supported by the European Union under the IST Content NoE ² (FP6-2006-IST-038423) and by the BioGridNet project ³ (CAM, S-0505/TIC-0101).

References

- [1] B. Beverly Yang and H. Garcia-Molina. Designing a super-peer network. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 49–60, 2003.
- [2] D. Bryan, P. Matthews, E. Shim, and D. Willis. Concepts and terminology for peer to peer sip, March 2007. Internet Draft draft-willis-p2psip-concepts-04.txt (work in progress).
- [3] R. Cuevas, C. Guerrero, I. Martinez-Yelmo, and C. Navarro. Bittella: A novel content distribution overlay based on bit-torrent and social groups. In *1st International Workshop on Peer to Peer Networks (PPN'07)*, Nov. 2007.
- [4] V. Darlagiannis, A. Mauthe, and R. Steinmetz. Overlay Design Mechanisms for Heterogeneous, Large Scale, Dynamic P2P Systems. *Journal of Network and Systems Management, Special Issue on Distributed Management*, 12(3):371–395, Sept. 2004.
- [5] P. Ganesan, K. Gummadi, and H. Garcia-Molina. Canon in g major: designing dhts with hierarchical structure. In *Distributed Computing Systems, 2004. Proceedings. 24th International Conference on*, pages 263–272, 2004.
- [6] L. Garces-Erice, E. W. Biersack, K. W. Ross, P. A. Felber, and G. Urvoy-Keller. Hierarchical p2p systems. In *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, 2003.
- [7] M. Kwon and S. Fahmy. Toward Cooperative Inter-overlay Networking. *Proceedings of IEEE ICNP*, 2003.
- [8] M. Kwon and S. Fahmy. Synergy: an overlay internetworking architecture. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pages 401–406, 2005.
- [9] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, 7(2):72–93, 2005.
- [10] P. Maymounkov and D. Mazieres. *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, volume 2429/2002 of *Lecture Notes in Computer Science*, chapter Kademia: A peer-to-peer information system based on the XOR metric, pages 53–65. Springer, 2002.
- [11] 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*, volume 2, pages 588–593, 2006.
- [12] J. Mischke and B. Stiller. Peer-to-peer overlay network management through agile. In *Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on*, pages 337–350, 2003.
- [13] J. Mischke and B. Stiller. Rich and scalable peer-to-peer search with shark. In *Autonomic Computing Workshop, 2003*, pages 112–121, 2003.
- [14] 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. The Third IEEE Workshop on*.
- [15] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM Press.
- [16] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–351, 2001.
- [17] H. Simon. The architecture of complexity. In M. Press, editor, *The Sciences of the Artificial*, pages 192–229. 1981.
- [18] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 11(1):17, 2003.
- [19] Z. Xu, R. Min, and Y. Hu. Hieras: a dht based hierarchical p2p routing algorithm. In *Parallel Processing, 2003. Proceedings. 2003 International Conference on*, pages 187–194, 2003.
- [20] Z. Xu and Z. Zhang. Building low-maintenance expressways for p2p systems. Technical report, Internet Systems and Storage Laboratory. HP Laboratories Palo Alto, Mar. 2002.
- [21] X.-M. Zhang, Y.-J. Wang, and Z. Li. *LNCS: Parallel and Distributed Processing and Applications*, chapter Research of Routing Algorithm in Hierarchy-Adaptive P2P Systems, pages 728–739. Springer Berlin / Heidelberg, 2007. 10.1007/978-3-540-74742-0_65.
- [22] S. Zoels, Z. Despotovic, and W. Kellerer. Cost-based analysis of hierarchical dht design. In *Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference on*, pages 233–239, 2006.

²<http://www.ist-content.eu>

³<http://www.biogridnet.org>