Super Nodes Positioning for P2P IP Telephony over Wireless Ad-hoc Networks

Mehdi Mani
GET/INT-Institut National des Télécommunications
9 Rue Charles Fourier
Evry Cedex 91011, France
mehdi.mani@int-edu.eu

Winston Seah
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
winston@i2r.a-star.edu.sg

Noël Crespi
GET/INT-Institut National des Télécommunications
9 Rue Charles Fourier
Evry Cedex 91011, France
noel.crespi@int-edu.eu

ABSTRACT

IP Telephony is a potential killer application among the various multimedia applications and services. It is therefore natural to expect support for these services over new network architectures like wireless ad-hoc networks. Despite the proliferation of IP Telephony services in the Internet, the traditional client/server models that have been used are found to be highly inefficient for wireless ad-hoc networks as compared to peer-to-peer (P2P) models. On the other hand, P2P strategies require some tuning to work well in wireless ad-hoc networks. In this paper, we discuss some undesirable situation that may happen if P2P systems are deployed over wireless ad-hoc networks without adaptation. We then define our strategies for positioning Super Nodes in the physical network underlay as well as P2P ID space according to the constraints of these network technologies. We evaluate the efficiency of our approche in reducing the session establishment time and request failure rate as two important criteria for the performance of IP telephony systems.

Keywords

IP Telephony, P2P, DHT

1. INTRODUCTION

Wireless ad-hoc networking technology is fast becoming the mainstream of the networking industry. Despite the extensive research on routing and lower layer protocols, there has not been enough efforts devoted to the service architecture for these networks.

To better fit the infrastructure-less architecture of these network technologies, services should also be provided in a distributed and self-organized manner. Among the various multimedia services, IP telephony is emerging as a killer application such that its support on wireless ad-hoc technologies brings in some interesting us cases. For instance, in-campus ad-hoc IP telephony could be an attractive service. Inside a campus, each peer is able to call another peer with minimum deployed infrastructure. The most important part of an IP telephony service architecture is the *service control system* which is responsible for the localization of clients, processing of queries and routing them to the correct destinations. Therefore, to provide IP telephony services on ad-hoc networks, the first step is to establish a scalable, dynamic and reliable service control system.

Traditionally, IP telephony systems follow the client/server model (e.g. H323 and SIP). The clients register with certain servers, which are responsible for handling the queries pertaining to these clients and routing the requests to the desired destinations. The client/server model is very inefficient for wireless ad-hoc networks for the following reasons. Firstly, this model requires the configuration of servers and proxies which contradicts with the sought after auto-configuration feature of ad-hoc networks. Secondly, such a centralized model does not fit the dynamic and changing topology of these networks.

On the other hand, P2P IP telephony systems like Skype [1] and P2P-SIP [9] have emerged in the Internet. These systems benefit from a distributed indexing based on DHT for user localization, lookup and request processing. These are interesting P2P solutions with minimum requirement of dedicated infrastructure . However because of inefficient routing of DHT [3]they can not be deployed directly in ad-hoc networks with scarce bandwidth and limited energy resources.

There are some researches to tune P2P DHT based service overlay in ad hoc networks [8][6][4][3]. The major contribution of all of these work is benefiting from network layer information to optimize the route between overlay neighbors which may be physically very far. However, there is another important aspect that from the best of our knowledge on which there is not enough research. That is the importance of the selection of Super Nodes that forms the service overlay in wireless ad-hoc networks.

In this paper, we define new strategies for selection of the Super Nodes in overlay network. We investigate how adaptive selection policies with careful consideration of the location information of the Super Nodes can improve the performance of overlay system. Then, via simulation, we evaluate how the proposed strategies can speed up query processing and lead to shorter session establishment time and less query failure.

2. CHALLENGES OF SUPER NODE SELECTION

P2P SIP and Skype follow *structured multi-layer P2P* model [7] to create their service overlay. In such model, a certain number of nodes termed Super Nodes (SN) form a P2P overlay. Then, each node is either a SN or has registered with a SN.

DHT is used in order to determine the required rules to localize the users. A peer ID may be created by a hash function on the IP address of the nodes or allocated randomly. Moreover, each item (ie. telephone No. or SIP URI) will be hashed to obtain a key value

MUM'07, December 12-14, 2007, Oulu, Finland

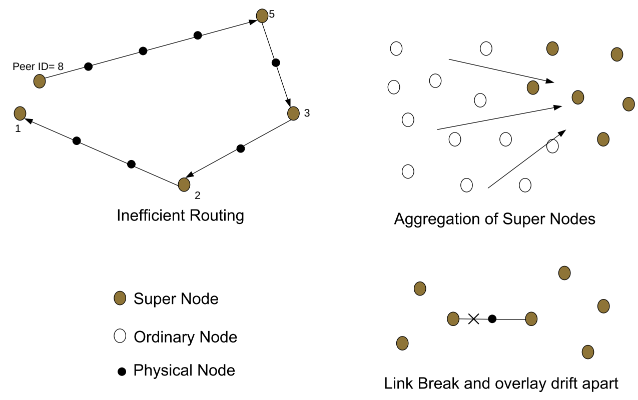

Figure 1: Undesirable Scenarios

(It is also called Resource ID). Then, a pointer to this item will be stored in the peer with the closest peer ID to this item key. The peer ID and item keys are in the same ID space provided by Hash function.

Now, consider that Alice wants to join Bob@int-edu.eu. So it should find the SN with which Bob has registered. So Alice should hash the Bob's ID (Bob@int-edu.eu) to obtain the key and find the SN with nearest ID to this key among all the SNs. If Alice was able to keep all the SNs address in her table, the lookup was only one hop. However, because of the huge number of the SNs in a system it is impossible. This is why DHT is used. Each node keeps the information of maximum log(N) SNs (N=number of SNs in the system). In this case, the lookup isn't carried out in one hop but DHT guaranties the lookup hops in O(logN).

DHT allows the deployment of a self-organized P2P system that is proper for ad-hoc networks. However, all of P2P systems using DHT, suffers from inefficient routing. The main reason is to find and reach a destination, the request should pass through a certain number of SNs over the network; no matter if destination is physically in the vicinity of the source node. This may be tolerable in the internet with high speed routers, but it pushes considerable load on ad-hoc networks with scarce bandwidth and limited energy resources. One solution to cope with this problem is to use a two layer Cluster-DHT overlay [2][6][4]. In this case each item is registered with two SNs. One which is the head of the cluster where this item exists and the second one with nearest ID to the item key. By using this method if the source and destination are in the same cluster, the request won't travel over the network to reach the destination because both of source and destination are registered with the cluster head.

Another important issue for adapting and tuning a P2P systems like Skype to be used in ad-hoc networks is defining the criteria for selecting a SN. In Skype, if a node with sufficient CPU is not behind NAT and has been active for a certain period of time it is very likely to be selected as SN [1]. However there are other important criteria that are important in selection of SNs in ad-hoc networks. Figure 1 shows two other undesirable scenarios that may happen if SNs are selected based on the nodes capabilities only and without considering their locations, their relative distance to other SNs and their accessibility.

In fact, dissemination of SNs over the network should be almost uniform if not SNs may be aggregated in a certain area and this lead to invasion of request toward certain parts of the network and creating congested areas. Moreover, the selected SNs should have redundant connections to the other SNs and the rest of the network. If not, with a link breaking down, the SN may be split apart from the rest of the network and if this SN be an important relay to other SNs this link break cause the whole overlay failure.

3. GENERAL FEATURES OF THE PROPOSED P2P OVERLAY

To cope with inefficient routing we also consider a two-layer cluster/DHT overlay. Then, we define new criteria and strategies to select SNs and admit them to the overlay to react intelligently to the performance metrics like responsiveness delay and query failure. Followings are the general characters of our overlay system.

Two Kinds of Nodes: We assume the existence of Ordinary Nodes (ON) and Super Nodes (SN). Super Nodes participate in forming the Overlay Networks and they process and route the service queries.

Bootstrap Super Nodes: We assume the existence of some bootstrap super nodes to guarantee a minimum of service in the initial phase.

Clustering: The network is divided into the clusters according to the proximity of the nodes. Each cluster is managed by a *Cluster Head* (CH). A cluster head is always a Super Node (but not vice versa). In the bootstrap phase, all the bootstrap super nodes become a cluster head.

Candidate List: Every *cluster head*, among the ONs of its cluster, looks for the legitimate nodes in order to create a Super Node candidate list. Super Node candidates should be the nodes which have enough hardware capabilities like CPU speed and memory capacity. In addition, they should be trustworthy nodes with regards to their availability. The criteria for selecting candidates to join the Super Node overlay in our approach are Hardware Capabilities (CPU speed, Storage capacity, Power Consumption), Active time, Connectivity and Physical Location.

Location Information: Existence of a free range location estimation system [5]is a complementary assumption that we consider.

The overlay system is constructed based on CHORD technology. In CHORD each Super Node keeps a list of its overlay neighbors which is called finger list. Finger list comprises at most $log(N)$ fingers. The *i*'th finger of a Super Node with *ID* = ID_{SN} should have an *ID* greater or equal to $ID_{SN} + 2^i$. Figure 2, shows the creation of a super node finger in CHORD.

The super nodes update their Finger Tables according to the received *advertisement* from other super nodes. In the Internet, the strategy is to choose the super node with the smallest *ID* greater than $ID_{SN} + 2^i$ as the *i*'th finger among all the possibilities (Algorithm 1).

Due to the routing cost in wireless technologies, our strategy is to select, as much as possible, the fingers which are physically near to the Super Node. The distance between a Super Node and its fingers can directly affect the session establishment delay. To have the finger i with minimum possible distance, we have modified the finger selection process as in Algorithm 2.

All the peers (users), are identified by a contact-ID which may be a SIP URI in the format like user-name@domain-name. In order to localize the peers, the overlay system should store the binding between contact-ID and the node IP address.

On the other hand, in order to establish a call, a peer sends a query indicating the destination contact-ID (SIP URI) to find its IP address. In our approach, the query first arrives at the cluster head, if the destination resides in the same cluster, the cluster head returns the requested IP address to the caller and the process is finished. If

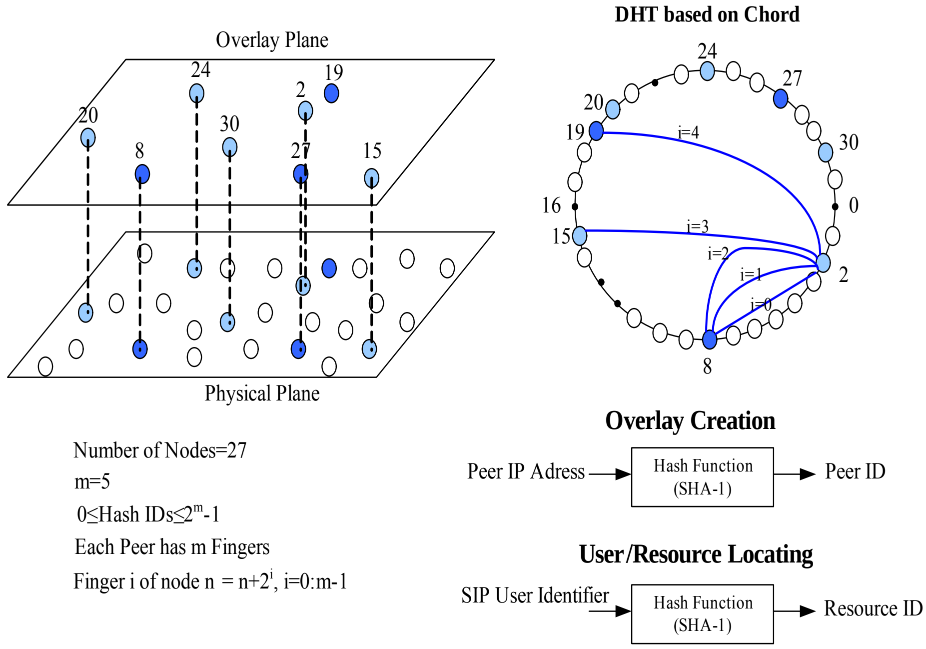

Figure 2: Overlay based on CHORD

Algorithm 1 Finger Selection in CHORD

for i=1:logN **do**
  $finger_i$=$SN_k|ID_{SN_k}$=min$\{ID_{SN_j} \geq ID_{SN} + 2^i, \forall SN_j \in Overlay\}$
end for

Algorithm 2 Modification of CHORD

for i=1:log N **do**
  $h = i$;
  A: $List = \{SN_j | ID_{SN} + 2^h \leq ID_{SN_j} < ID_{SN} + 2^{h+1}\}$
  if List $\neq \oslash$ **then**
    $finger = SN_k \,|distance(SN_k, SN)$
    $= Min\{distance(SN_j, SN), \forall SN_j \in List\}$
  else
    $h = i + 1$
    goto A
  end if
  $finger_i = finger$
end for

not, the cluster head starts the lookup process in the DHT overlay. The destination contact-ID will be hashed to obtain the key. Then the request will be sent to the finger with the closest peer-ID to this key. This process continues until the query arrives at a super node to which this key is assigned.

4. THE DETAIL OF SUPER NODE ADMISSION PROCEDURE

The overlay network should grow up according to the increase in the number of service requests. We admit new Super Nodes in our service overlay system mainly based on three admission strategies:

- The number of members in a cluster controlled by a Super Node.
- Responsiveness of the overlay network: Mean Lookup Delay and Maximum Lookup Delay.
- High density of member IDs in a domain of overlay ID space.

4.1 Number of Local Members in a Cluster

When a cluster becomes over crowded and the number of Ordinary Nodes exceed the predefined limit, a new Super Node among the Super Node candidates of Cluster Head's list will be selected to create a new cluster. The cluster will be split in two clusters upon emergence of the new Cluster Head. The process for this new Cluster Head to join to the DHT overlay depends on the DHT algorithm.

However, this is a common process in all of DHT algorithms. In this process, according to the peer ID of the new Super Node, some of the contacts that are already saved in other Super Nodes should move into the new node. In addition, the neighbors of this new node in the overlay ID space should modify their finger tables.

When a cluster must be split, an important question is if there are any location considerations for the new Cluster Head. In fact, the new cluster should be created in a manner that reduces the existing farthest distance to the cluster head. To reach this end, all the Cluster Heads keep the information of the node with farthest distance.

Then, in the case of cluster splitting, if we consider that the coordination of this farthest node is (x_f, y_f) and the coordination of the Cluster Head is (x_{ch}, y_{ch}), we consider the point with the coordination of $(\frac{x_f+x_{ch}}{3}, \frac{y_f+y_{ch}}{3})$ as the ideal place to locate the new cluster head (Figure 3). With this strategy the distance of the farthest node to the new cluster head will be one third of its distance to the previous cluster head. Indeed, the border line between the new and existing clusters fall in the middle of their cluster heads. Therefore, a considerable load of the existing cluster head will be transferred to the new cluster head.

However, it is very likely that a Super Node candidate will not

Algorithm 3 Admitting New Super Node Because of Overcrowded Cluster

if (Number of Cluster Members $\geq$ Limit) **then**
  (x_f, y_f) = coordination of the farthest cluster member to the cluster head;
  (x_{CH}, y_{CH}) = coordination of the cluster head;
  The ideal coordination for new Cluster Head =$(\frac{x_f + x_{ch}}{3}, \frac{y_f + y_{ch}}{3})$;
  Find the SN candidates with maximum distance d from ideal coordination (ideal area);
  if (No SN candidate is found) **then**
    New Cluster Head = The Best Candidate in SN Candidate List;
  else
    New Cluster Head = The Best Candidate found in ideal area;
  end if
  Start Cluster Splitting Process;
  Start the join process in DHT overlay;
end if

be found exactly in this point. Therefore, we consider an area with radius of d around the ideal point. This area is called ideal area.

If a Super Node candidate was found in the ideal area it would be selected as the new cluster head. If not, the best SN candidate in the list of this cluster head will be selected.The complete admission process is depicted in Algorithm 3.

4.2 Non-Uniform Member IDs Distribution

When the number of assigned ordinary nodes to a super node overflows a certain limit, a super node candidate should be admitted in the overlay to moderate the load on the overloaded super node. If we consider that the stressed super node peer-ID = ID_{SN} and its first predecessor peer-ID = ID_{SNp}, an acceptable choice for the new Super Node, is a Super Node candidate with ID_{SNc} such that:

$$ID_{SNp} < ID_{SNc} < ID_{SN}$$

With such choice, the contact IDs in the range of $[ID_{SNc}, ID_{SN})$ which are already assigned to the stressed super node will be transferred to the new super node. In order to find a super node candidate that meets the above mentioned condition, our strategy of search is as following:

1. Search in the cluster
2. Search in the fingers
3. Search in all Super Nodes

In each of these search phases, if a Super Node candidate met the condition the search process will be stopped.

4.3 Responsiveness Delay

A cluster head receives the query from its cluster members. If the requested ID is not registered in the cluster's head, it triggers the lookup in the DHT overlay to find the requested contact ID. Two records will be stored in each Super Node: Average lookup delay and Max lookup delay. These two records will be interpreted as mean and maximum responsiveness delay.

The Super Node starts the lookup process via its finger-with-nearest-ID to the requested contact ID in the query. Therefore, the responsiveness delay for each query depends on the finger which is the next overlay hop in lookup process.

Consequently, the average responsiveness delay is calculate as follow:
Average Responsiveness Delay = $\overline{RD} = \frac{\Sigma(NQ_i \cdot \overline{RD_i})}{NQ}$
where
NQ = *Total Number of queries received in this Super Node*;

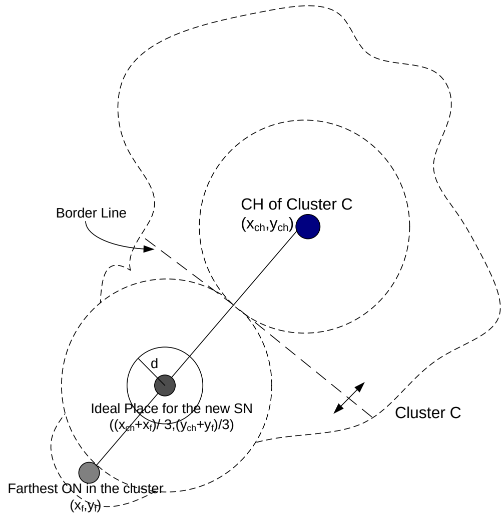

Figure 3: Best Place for New Cluster Head

NQ_i = *Number of queries with* $Finger_i$ *as the next overlay hop*;
$\overline{RD_i}$ = *Average Responsiveness delay related to the queries with* $Finger_i$ *as the next overlay hop.*
If the average lookup delay via this Super Node is high via all of its fingers (or some of them) and the number of overlay member in the cluster has not exceeded the limit, the main reasons may be: i) This Super Node has lost some of its links to the rest of the mesh toward its fingers in DHT overlay according to the departure of the peers or an obstacle in wireless connection. ii) This Super Node is far from all its fingers.

In such cases, the Super Node should be replaced. The detail of this replacement is given in Algorithm 4.

Algorithm 4 Admitting New Super Node Because of Responsiveness Delay

if ($\overline{RD} \geq$ Limit) **then**
  if (This SN is also a CH) **then**
    New-CH = SN candidate-with-farthest-ID to the current CH's ID in the SN candidate list;
    CH = New-CH; %Replacing the current CH
  else
    NF_{far} = Number of Far Fingers $\equiv$ Number of $\overline{RD_i}$s which are more than Limit;
    if $NF_{far} \geq$ Half of Fingers **then**
      Turn Off This SN;
    else
      Add a new SN with nearest ID to the finger corresponding to the maximum $\overline{RD_i}$;
    end if
  end if
end if

5. SIMULATION MODEL AND RESULTS

We consider a 50x50 grid (2500 physical node) as our test-bed to create the physical network. The horizontal and vertical separation between adjacent nodes is 20 meters. Moreover, the wireless range of each node is also considered 20 meters. These nodes don't necessarily participate in overlay.

The link quality between two neighbor nodes is simulated by a parameter k with a random value in the range of $[0, 1]$. If $0 < k <$

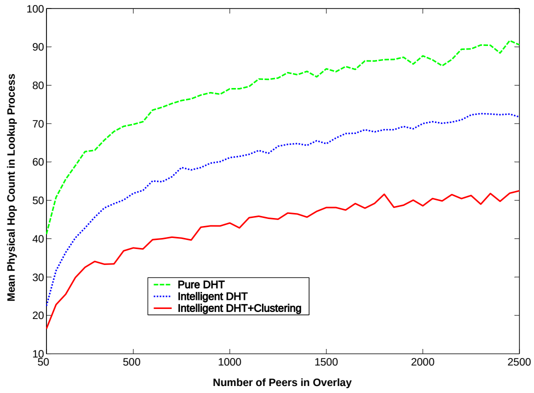

Figure 4: Mean Hop Number

.2 the route is considered broken. This parameter is important to select the Super Node candidate. Moreover, the maximum cluster member is considered 100 in our simulation.

Among these nodes, we consider the existence of 4 bootstrap nodes that creates the overlay network in the beginning. We have fixed their place in the network. Then among the physical nodes a node will be selected randomly to join the overlay network with the *poisson* random process and rate of $\lambda = 2/min$. The duration that a node remain active in overlay, is an exponential random variable with $\mu_3 = 15minutes$.

Indeed, upon arrival of a node in the system, n queries will be created. n is equal to 10 percent of the number of overlay peers. Each query indicates a couple of source and destination peers. According to the destination peer, the route in the overlay will be calculated by using DHT.

Moreover, three strategies of overlay construction are considered for the simulation:
1) *Pure CHORD*: The overlay is constructed based on Chord rules.
2) *Intelligent DHT*: We apply Algorithm 1 for finger selection in CHORD.
3) *Intelligent DHT+Clustering*
We have defined 2 performance metrics such as *Mean Hop-Count* and *Query Failure* to compare the performance of these three strategies.

As Figure 4 shows, regarding the *mean hop-count*, Intelligent DHT+Clustering has shortened the query route for $\%38$ in comparison to the pure DHT. In addition, even without using clustering, Intelligent DHT improves significantly the mean query hop counts for $\%22$. The *query failure* simulation results are presented in Figure 5. In the upper figure, there is no TTL restriction. Therefore, the only reason for query failure is bad link quality. The lower is the query route, the lower will be query failure probability. This is why Intelligent-DHT+Clustering show the best performance among other approaches.

6. CONCLUSIONS

In this paper we have presented new strategies to select Super Nodes for P2P IP-Telephony Services in wireless ad-hoc networks. For this selection, we consider the arrangement of Super Nodes in the overlay, their physical locations, their relative distances and their connectivity redundancy. We defined intelligent admission process of SNs in order to react properly to the requirements of the overlay. To this end we consider mean physical hop count in a query route and query failure as our performance metrics. We have also modified CHORD to add some proximity intelligence in order to set finger table. We demonstrated through simulations how our strategies can improve the performance.

7. REFERENCES

[1] Skype: Free internet telephony that just works. *http://www.skype.com*.

[2] G. Ding and B. Bhargava. Peer-to-peer file-sharing over mobile ad hoc networks. *In proceeding of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (PERCOMW'04)*, 2004.

[3] G. Ding, J. Vicente, S. Rungta, D. Krishnaswamy, W. Chan, and K. Miao. Overlay on wireless mesh networks: Implementation and cross-layer searching. 2006.

[4] R. A. Ferreira, S. Jagannathan, and A. Grama. Enhancing locality in structured peer-to-peer networks. *Proceedings of the Tenth International Conference on Parallel and Distributed Systems (ICPADSŠ04)*, 2004.

[5] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. *In proceeding of 9th annual international conference on Mobile computing and networking*, 2003.

[6] M. Li, W. C. Lee, and A. Sivasubramaniam. Effitient peer-to-peer information sharing over mobile ad hoc networks. *proceeding of IEEE INFOCOM*, 2003.

[7] D. Liben-Nowell, H. Balakrishnan, and D. Karger. Analysis of the evolution of peer-to-peer systems. *In Proceeding of ACM Conf. on Principles of Distributed Computing (PODC)*, July 2002.

[8] M. Waldwogel and R. Rinaldi. Efficient topology-aware overlay network. *ACM Computer Communication Review*, January 2003.

[9] P. S. I. WG. *http://tools.ietf.org/wg/p2psip*.

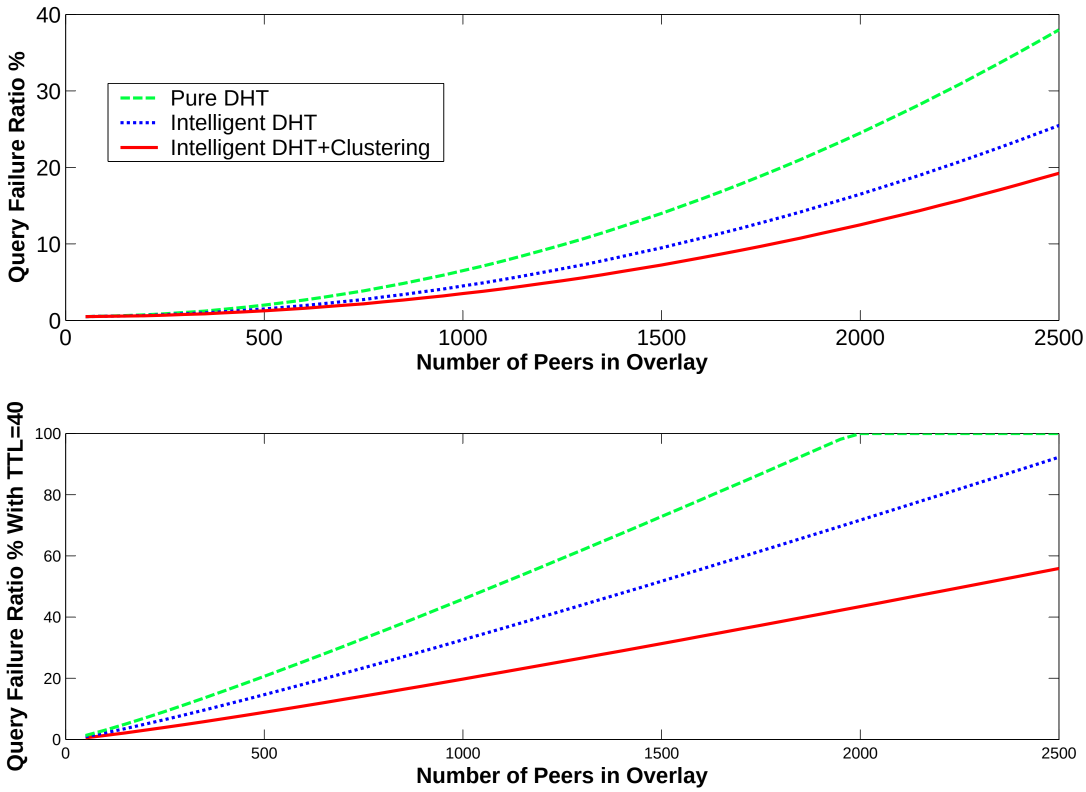

Figure 5: Query Failure Ratio