Delay and Capacity Analysis of Structured P2P Overlay for Lookup Service

Jagadish Ghimire, Mehdi Mani, Noel Crespi, Teerapat Sanguankotchakorn

November 13, 2011

Abstract

In this paper, we provide an analytical model for the performance study of different structured P2P overlay networks used for lookup service in IP telephony systems. The overlay provides an infrastructure for the lookup service required before an actual voice communication is initiated. Our model captures the performance behavior of such overlays including the mean session set-up delay of a call as well as the system capacity. These parameters reflect how good a IP telephony overlay is performing. We formulate the system as a queuing network. We idealize Chord routing semantics to extract useful observations to obtain closed form expressions for the session setup delay and capacity as a function of the number of participating supernodes. The analysis also answers the question of finding an optimum number of supernodes for minimum session setup delay.

Keywords: Structured Overlay, Lookup Services, Delay Analysis

1 Introduction

Peer-to-Peer has been a current service paradigm over Internet for many kinds of applications ranging from file sharing to naming services. P2P finds its application in large-scale systems riding over Internet. Besides file sharing , content distribution, virtual networked environment and gaming [13], IP Telephony service using P2P technology has been one of the most successful P2P applications that have emerged in recent years. The scale of popularity of Skype [3] is an outstanding proof of these applications.

Skype follows a certain kind of P2P architecture known as Supernode based P2P. In such P2P systems few among the peers are selected as supernodes (SN). An overlay is created among these SNs. The other peers (called Client Nodes (CN)) are associated with one or more SNs and rely on them for the lookup. Usually, a P2P overlay in such systems is used for resolving the call destination and the required lookup query routing. When Bob wants to call Alice, the client agent on Bob's device sends a lookup request to the SN with which it is associated. The lookup request contains the identity (e.g. SIP URI) of the destination (Alice). The SN of Bob, upon receiving the lookup request, forwards the message to one of its neighbors in the overlay network. The way lookup travels in the overlay network depends upon the deployed routing scheme. The lookup processes is finished when the message is received by the SN who is serving Alice. Then this SN can reply to Bob with the IP address (and port number) of Alice so that Bob can initiate a voice communication with Alice.

Such SN based P2P are getting popular since they can be used to make P2P network maintenance overhead scalable [8]. The supernode P2P networking lies in the category of unstructured P2P systems. In contrast, there is a different class of P2P called structured P2P systems [25, 22]. They rely on the Distributed Hash Table (DHT) and offer efficient message routing [17]. These systems may deploy the SN/CN hierarchy; however since the routing semantic in the SN overlay is based on the DHT rules, in the literature, these hierarchical structured P2P systems are still referred as structured or DHT based P2P. P2P IP Telephony systems based on DHT technologies have also been developed with the expectation that DHT provides better scalability for these systems [23]. There have been some interest on the study of the lookup performance of such super-peer based systems ([14]). Unfortunately, there is not a clear and deep insight about the performance of these structured P2P IP Telephony systems in terms of their call processing capacity, scalability and the required computing resources.

In this paper, we are focusing on the analytical model of the IP telephony application of such structured P2P systems. Our earlier paper [10] forms the basis of this work where previous analysis is extended as a framework for generic P2P systems. It is important to point out that Session Initiation Protocol (SIP) ([21]) has emerged as a standard open protocol of message exchange from IETF (Internet Engineering Task Force) for establishing the connections before the actual communication begins. P2P-SIP has been proposed as a zero-configuration alternative to the existing SIP protocol (which requires dedicated SIP proxies). In this context, our analysis also fits in with the general idea of P2P-SIP. Despite the popularity of SIP, different IP Telephony services (including Skype) use message protocols different from SIP. Our analysis does not make any specific system assumptions on the exact messaging protocol used for the lookup service.

Great deal of research has been done in the area of performance analysis of P2P file-sharing and streaming [9, 27, 26]. However, a general lack of delay analysis for the performance of P2P IP telephony systems motivates this work. In our work, we propose a generic and simple mathematical approach for deriving and describing the performance behavior of such structured overlay in terms of the session setup delay and capacity which are expressed as a function of the number of SNs in a structured P2P system. Moreover, we illustrate our generic analysis for two structures namely: Chord [25] DHT and Ring DHT. The two performance parameters of interest in our work are: session setup delay and the capacity.

Session Setup Delay (also called the post dialing delay) is an important performance parameter of IP telephony application. It is related to the satisfaction of the users and indicates how good a given IP telephony service is operating. ITU-T [12] recommends that 95 percentile of calls in a local telephony scenario should not exceed a post dialing delay of 3 secs.

Another parameter of interest in such systems is the maximum capacity of such overlay in terms of the maximum number of users supported without exceeding the tolerable call drop threshold. In this regard, this capacity is reflected in terms of the maximum call request rate supported by the system.

The main contribution of this paper can be listed as follows.

- We introduce a key analysis parameter called Absorption probability and propose a generic analysis model that can be used to find out the expected overlay hops of a lookup, the mean delay of a lookup as well as the capacity of the overlay network in terms of maximum call request rate to be processed

- Using the generic model, we express the relationship of the number of SNs to the mean session setup delay as well as maximum lookup processing capacity for two different overlay structures: the Ring structure and the Chord structure. Also, we determine the optimum number of SNs for minimum session setup delay.

- We verify the analytical results with equivalent simulation experiments.

- We perform simulation based study of a general imbalanced network scenario and compare the results with the ideal model of our analysis.

- We formulate and prove some interesting properties of an idealized Chord as well as Ring structure overlay.

Our results show that the quantitative characteristics of the capacity as well as delay parameters as a function of system resources (ie. number of supernodes) exhibit a trade off relationship. As a specific case, we have focused in obtaining the resource size (the number of SNs) to obtain minimum session setup delay. Such a minimum helps to use more efficiently the resources in a P2P IP Telephony system and avoid admitting new SNs which not only do not improve the performance of the system but also affect the session establishment delay and increase the unused/wasted resources. Moreover, we observed that the capacity scaling of overlay network depends upon the structure . As a specific case, we observed a completely different behavior between a Ring structure and Chord Overlay. The former exhibits a limit on the maximum capacity where as Chord has a potential of enhancing capacity arbitrarily by adding more SN resources.

In section 2, we present the generic analysis approach that can be used to perform evaluation of SN based structured overlay networks for IP telephony application. In section 3, we illustrate the analysis of Ring Topology and derive closed form expressions for the mean lookup delay as well as the capacity of such networks. In section 4, we perform evaluation of a load-balanced model of Chord overlay. In section 5 we present the results of equivalent simulations along with the analytical results. Further, we investigate how these performance parameters scale with the number of SNs. In section 6 we present the results of realistic simulations which relax the assumptions made in the analytical model and thus see the performance behavior of non-balanced overlay structures. Finally in section 9, we summarize our work and mention the future directions. Appendices detail proofs of some results established in the main text.

2 Generic Analysis of Overlay Networks

Analysis of overlay networks for lookup latency and service capacity can be very complicated as such. It involves a set of challenges inherent to the overlay network characteristics and the node dynamics. In this work, we intend to propose a simple analytical approach that can model useful performance behavior of overlay networks including the lookup latency in terms of overlay lookup hops, the delay time of the lookup as well as the lookup processing capacity of the network.

2.1 Absorption Probability

The heart of the analysis involves a straightforward overlay characteristics variable called *Absorption Probability* defined as follows.

Definition 1. *Absorption Probability of a given SN is defined as the probability with which an incoming lookup is terminated at that SN. An incoming lookup message can be a new lookup request generated by the CN associated with the given SN; or it could have been forwarded to the SN by its predecessor(s).*

If a lookup message reaches to a SN with whom the destination CN is registered, then the lookup is no more forwarded and this can be viewed as its absorption. In this sense, an absorption probability defines the total probability for any lookup message being absorbed at the given SN. Obviously, this parameter depends upon:

- *SN Degree*: We define the SN degree as the size of the neighbor list (overlay routing table) of a SN. Among two overlay structures with same number of nodes and similarly greedy behavior of routing, we expect a larger absorption probability for the structure with larger average node degree. This is because network with larger degrees tend to reach the required host SN more quickly.
- *Number of SNs*: In the same overlay network, more number of SNs means less chances of a lookup being absorbed in a given SN (as long as the destination is selected uniformly). Thus, we expect in general that absorption probability decreases with an increase in number of SNs.
- *Load distribution among SNs*: Moreover, in a general scenario, different SNs are exposed to different lookup arrival rates and the load of forwarded lookups. Thus, even in a given overlay network, different SNs will have a different absorption probability.

For an overlay network with N SNs, we represent absorption probability of SN i as $P(N, i)$. In our analytical models, for the sane of simplicity we relax the condition in the third following bullet and assume that the overlay is performing under uniform load balancing.

- All SNs have the same number of incoming as well as outgoing neighbors.
- Equal number of CNs are associated with all SNs
- Each CN generates equal traffic per unit time with a lookup destination selected uniform randomly among all participating CNs

With this assumption for a given overlay, each peer will have the same absorption probability in whatever follows. We represent this Absorption Probability simply as $P(N)$. However, Later in section 6, we present the simulation results to see the performance in presence of an arbitrary load distribution and arbitrary node-degree distribution.

2.2 Mean Lookup Hop

In this section , we move on to deduce expression for the mean lookup hops in an overlay network as a function of the absorption probability ($P(N)$). In an overlay of N peers, we represent the expected number of SNs visited as $\overline{S}$. Then the mean overlay lookup hops is simply $\overline{S} - 1$. In the following proposition, we show the relationship between the mean lookup hops and the absorption probability in an ideal load balanced scenario.

Proposition 1. *In a load-balanced overlay network, the average number of SNs visited by a lookup is the inverse of its absorption probability. i.e.*

$$\overline{S} = \frac{1}{P(N)} \tag{1}$$

Proof. $P(N)$ is the total probability for any SN to absorb a lookup message incoming to it. Thus the average number of hops traveled (represented as $\overline{S}$) should satisfy the following relation.

$$\sum_{i=1}^{\overline{S}} P(N) = 1 \tag{2}$$

Considering that all SNs have the same absorption probability ($P(N)$) , we obtain the following.

$$\overline{S} \times P(N) = 1$$
$$\overline{S} = \frac{1}{P(N)} \tag{3}$$

□

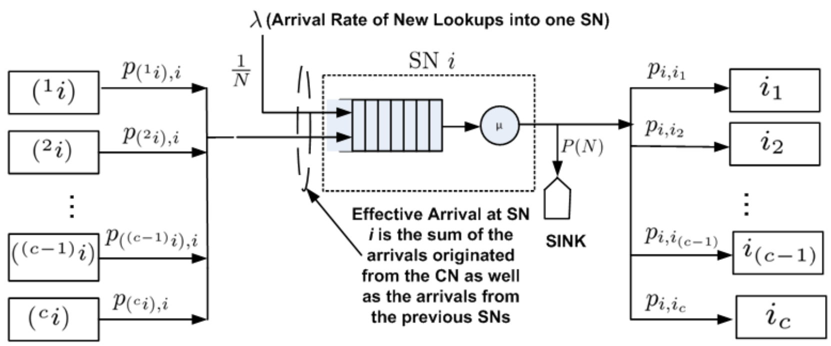

Figure 1: Node Model of a Generic Overlay Network

Table 1: Summary of queuing network parameters

Parameter	Description
λ_{total}	mean total arrival rate
μ	SN service rate
(^ki)	the SN whose k^{th} outgoing neighbor is SN i
i_k	the k^{th} outgoing neighbor of SN i
$p_{(^ki),i}$	the routing probability from the SN (^ki) to SN i
p_{i,i_k}	the routing probability from SN i to SN i_k
$p_{0,i}$	the probability of a lookup message to originate at SN i
$p_{i,0}$	the absorption probability
λ_i	Effective arrival rate in SN_i
e_i	Visit ratio of SN_i
ρ_i	Utilization factor of SN_i

2.3 Queuing Network Modeling for Delay and Capacity Analysis

Our P2P system is modeled as an open queuing network. Each SN is a queue which absorbs any incoming message with a probability $P(N)$. Any queue i receives lookup message either from its CNs (modeled as an outside world) or from its predecessors. Each queue has c incoming neighbors and the same number of outgoing neighbors. The model of a queue (SN) is shown in figure 1. In our model, similar to [5], node 0 represents the outside world.

In following subsections, we define parameters of the queuing network. The summary of the parameters and other related notations are summarized in Table 1.

2.3.1 Traffic Parameters

The assumptions relating to traffic parameters are listed below.

- Total lookup message arrival process in the system is a Poisson process with λ_{total} mean arrival rate.

- All SNs get exactly the same rate of lookup message requests from their hosted CNs (this is the result of the assumptions stated before)
- The destination of a lookup is selected uniformly among all the participating CNs such that for any lookup, the probability of a SN being the destination SN is equiprobable.
- All SNs in the system have an exponentially distributed service time with mean $\frac{1}{\mu}$
- All SNs have an infinite buffer for lookup message

2.3.2 Routing Probabilities

We introduce new notations for routing probabilities. $({}^k i)$ is the SN whose k^{th} out-neighbor is SN i. Moreover, i_k is the k^{th} out-neighbor of SN i. Then, $p_{({}^k i),i}$ is the routing probability from the SN $({}^k i)$ to SN i. Similarly, p_{i,i_k} is the routing probability from SN i to SN i_k. Note that $p_{0,i}$ is the probability of a lookup message to originate at SN i (which is equiprobable among all SNs according to our assumption) and $p_{i,0}$ is equal to the absorption probability.

2.3.3 Calculation of Network Parameters

In this subsection, we define and calculate the network parameters of interest.
Effective arrival rate (λ_i) of a queuing element SN i is the gross arrival rate in SN i including both new lookup arrivals from the associated CNs as well as the forwarded lookups.
Visit ratio per SN (e_i) of a queuing element SN i is defined as the ratio of the effective arrival rate of the lookup messages into this SN to the arrival rate of new jobs entering into the network.i.e. $e_i = \frac{\lambda_i}{\lambda_{total}}$ where, λ_{total} is the total arrival rate of new jobs into the network.
Utilization per SN (ρ_i) of a queuing element SN i is the ratio of effective arrival rate per SN (λ_i) and the service rate per SN (μ).

Proposition 2. *The visit ratio (e_i), effective arrival rate (λ_i) and the utilization (ρ_i) of a given SN i in the load-balanced overlay model is*

$$e_i = \frac{1}{NP(N)} \qquad \lambda_i = \frac{\lambda_{total}}{NP(N)} \qquad \rho_i = \frac{\lambda_{total}}{\mu NP(N)} \tag{4}$$

Proof. The traffic equation for open queuing network is borrowed from [5] as shown below:

$$e_i = p_{0,i} + \sum_{j=1}^{N} p_{j,i}.e_j \tag{5}$$

Expanding it for SN i with the given node model and the routing probabilities, we obtain the following

$$e_i = p_{0,i} + p_{({}^1 i),i} e_{({}^1 i)} + p_{({}^2 i),i} e_{({}^2 i)} + \cdots + p_{({}^c i),i} e_{({}^c i)}$$

Exploiting the symmetricity of the load-balanced overlay model, we know that,

$$e_i = e_j \text{ For all } i, j = 1, 2, 3, \cdots, N$$

This eliminates the set of simultaneous equations in equation 5, to solve e_i

$$e_i = p_{0,i} + e_i \sum_{k=1}^{c} p_{({}^k i),i} \tag{6}$$

Noting that $p_{i,0} = 1 - \sum_{j=1}^{N} p_{i,i_j}$ and since $p_{i,0} = P(N)$, we have $\sum_{j=1}^{N} p_{i,i_j} = 1 - P(N)$. Now, for the c-regular graph model of the overlay network, $p_{({}^k i),i} = p_{i,i_k}$. So, we have

$$\sum_{j=1}^{N} p_{({}^j i),i} = 1 - P(N) \tag{7}$$

Now, using equation 7 in equation 6, we obtain the expression for visit ratio as follows.

$$e_i = p_{0,i} + e_i\,(1 - P(N)) = \frac{1}{N \times P(N)}$$

. Simply substituting this result of visit ratio into definition of effective arrival rate and utilization proves the Proposition. □

2.3.4 Session Setup Delay and Capacity

Having calculated the network parameters, in this subsection we present the expressions for session setup delay and the capacity of our P2P overlay.

Theorem 1. *The total mean lookup delay (session setup delay) ($\overline{D}$) in the load-balanced overlay model is*

$$\overline{D} = \frac{1}{\mu(1-\rho_i)} \frac{1}{P(N)} \tag{8}$$

Proof. All arrivals are considered to be Poisson distributed. The service times are exponentially distributed. In this case, the network is a simple product form Jackson Network [5]. The mean delay per SN ($\overline{D_i}$) is given as [5]

$$\overline{D_i} = \frac{1}{\mu(1-\rho_i)} \tag{9}$$

Then the total lookup delay in the overlay should be the product $D_i \times \overline{S}$. Both terms are already determined. So, we have

$$\overline{D} = \frac{1}{\mu(1-\rho_i)} \overline{S} = \frac{1}{\mu(1-\rho_i)} \frac{1}{P(N)} \tag{10}$$

where ρ_i is obtained from Proposition2. The total session setup delay is composed of the lookup delay (which is derived in this theorem) and the delay of receiving the acknowledgment from the callee. The second component of the delay depends on the underlying network topology and is decoupled from the overlay performance analysis. □

In P2P systems, the capacity of overlay has been defined differently based on the application [9, 27]. In this subsection, we present our definition of capacity for P2P IP telephony and then we calculate the expression for it.

Definition 2. *Capacity of the P2P IP telephony service control overlay is the maximum lookup arrival rate that the overlay can support. Owing to an infinite lookup buffer in each SN, our definition of capacity corresponds to the marginal lookup arrival load after which the delay is no more finite.*

Theorem 2. *The capacity of the load-balanced overlay model (λ_{max}) can be expressed as follows:*

$$\lambda_{\max} = N \times P(N) \times \mu \tag{11}$$

Proof. The capacity corresponds to the total arrival rate (λ_{total}) when the utilization per SN (ρ_i) derived previously approaches 1. This corresponds to a higher bound in utilization per SN for finite delay. Equating $\rho_i = 1$ can be solved to find the value of λ_{total} which represents the capacity of the system in terms of lookup processing. □

3 Ring Topology Overlay Analysis

In this section, we apply our analysis model to a ring overlay topology and see how our generic approach can be used to model such structures.

3.1 Overlay Description

An example of a ring topology overlay is a Chord like DHT with one successor and without further fingers. This is a very simple DHT based system where each peer maintains the routing pointer to only one successor in the ID space (i.e. SN degree is equal to 1). This is same as the structure described in section *IV(B)* of [25] which defines the structure as a *Chord Ring*. In our scenario, SNs are the peers and the CNs associate with one of these SNs and generate traffic to the network . Each peer (SN in our case) hashes a unique identity (example ip address) to a fixed length hash using uniform hashing algorithms like SHA-1. This hashed number is the identity of the peer in the overlay network and is called *Node ID*. Node ID of a certain peer always falls in the interval of $[0, 2^k - 1]$ where k is the number of bits of the generated hash. The keyspace is thought to be organized in a circular space. A peer occupies a certain position based on its location in the circular ID space. A peer with node id i considers a peer with node id j as its neighbor if the circular distance (clockwise) of j from i is less than the distance of any other node id l from i. If $K = 2^k$ is the size of keyspace, then for N peers, if the keyspace is uniformly partitioned into the participating N peers, the inter-peer distance is $\frac{k}{N}$. The keys in between a peer i and its successor are considered to be the keys belonging to the peer i. Any CN (or a resource) with one of these keys will be associated (registered) with peer i. We call this SN or peer as a *host SN* of the given CN. We use this terminology throughout this paper.

The uniform load balancing assumption, can be interpreted as the two following properties in a ring structured overlay..

- Each SN gets an equal key space partition. i.e. there is a perfect uniform keyspace partitioning: In a given instance, we can not expect to achieve this. But, if a uniform hashing scheme is used, in average we can expect to achieve this with high probability. [25]

- The probability that any lookup's destination CN is hosted by the participating SNs is equal: The number of CNs associated with a SN is proportional to the keyspace partition assigned to it. For equal number of CNs associated with the SNs, the host of the destination of each lookup is thus a uniform integer random variable in the range $[1, N]$.

3.2 Overlay Analysis

Consider that there are N SNs forming an overlay network. In our scheme, SNs are numbered from 1 to N. SN $i+1$ is the successor of SN i. And, this wraps around such that for SN N, its successor is SN 1. For any incoming lookup message in SN 1, SN 1 will absorb the lookup message instantly if the destination address of the lookup message is searching for a CN hosted by it. If not, it will forward the lookup message to SN 2. Subsequently, the lookup query travels through S number of SNs before it is absorbed in host SN. For any lookup message originated at SN 1, we define

$$\begin{aligned} p_i = \text{Prob}\{&\text{SN } i \text{ absorbs any lookup message originated in SN 1} \\ &|\text{SNs } (i-1)\&(i-2)\&\cdots\&2\&1 \text{ DO NOT absorb it}\} \end{aligned}$$

Noting the fact that the probability of absorption increases in every subsequent forwards until it becomes 1 when it reaches to SN N, we have

$$p_i = \frac{1}{N-(i-1)} \tag{12}$$

3.3 Absorption Probability and Mean Lookup Hops

We need first to determine $P(N)$ which is the key analysis parameter. We express $P(N)$ as a function of the participating SNs (N) in the following proposition.

Proposition 3. *Under the assumption that nodes are associated with SNs with perfect load-balancing, and also that each source initiates a lookup to a destination uniformly then absorption probability($P(N)$) for each SN in the ring overlay is given as*

$$P(N) = \frac{2}{N+1} \tag{13}$$

Proof. Let

$$\text{Prob } \{ \text{ a lookup message visits SN i}\} = p_{v,i}$$

and

$$\begin{aligned}\text{Prob}\{ & \text{ a lookup message visits SN i} \\ & \mid \text{the lookup was initiated in SN j}\} = p_{v,i}^{j}\end{aligned}$$

From total probability theorem, we have

$$p_{v,i} = \frac{1}{N}\sum_{j=1}^{N} p_{v,i}^{j} \tag{14}$$

Now, by definition we note the following

$$p_{v,N}^{N} = 1 \qquad p_{v,N}^{N-1} = (1-p_1) \qquad p_{v,N}^{N-2} = (1-p_1)(1-p_2)\cdots$$

Generalizing this leads to the following

$$p_{v,N}^{j} = \prod_{k=1}^{N-j}(1-p_k) \tag{15}$$

But following the definition of p_i

$$\prod_{i=1}^{j}(1-p_i) = \frac{N-j}{N} \tag{16}$$

So, from equation 15 and equation 16, we get the following

$$p_{v,N}^{j} = \frac{j}{N} \tag{17}$$

From equation 14 and 17

$$p_{v,N} = \left(\frac{N+1}{2N}\right) \tag{18}$$

We define

$$P\{A_i\} = \text{Prob}\{ \text{ A lookup originates at SN } i\} = \frac{1}{N}$$

and

$$P\{B\} = \text{Prob}\{ \text{ A lookup arrives to SN } N\} = p_{v,N}$$

Note that ,

$$P\{B|\ A_i\} = p^i_{v,N}$$

From the theorem of conditional probability, we observe that

$$P\{A_i|\ B\} = \frac{P\{B|\ A_i\}P\{A_i\}}{P\{B\}} = \frac{p^i_{v,N} \times \frac{1}{N}}{p_{v,N}} = \left(\frac{\frac{i}{N}\frac{1}{N}}{p_{v,N}}\right).$$

Moreover, we can see that p_{N-j+1} represents the probability the SN N will absorb any packet originated in SN j, . From equation 12, we see that $p_{N-j+1} = \frac{1}{j}$

$$\begin{aligned} P(N) =& \sum_{j=1}^{N} p_{N-j+1} P\{A_j|B\} = \sum_{j=1}^{N} \left(\frac{1}{j}\frac{j}{N}\frac{1}{N}\frac{1}{p_{v,N}}\right) \\ =& N\left(\frac{2N}{N+1}\frac{1}{N}\frac{1}{N}\right) = \left(\frac{2}{N+1}\right) \end{aligned} \tag{19}$$

whose simplification proves the result. □

Having derived the absorption probability, it is trivial to find the mean lookup hops of the overlay network. The mean number of SNs visited by a lookup message represented as $\overline{S}$ in the ring overlay topology is given in equation 20

$$\overline{S} = \frac{N+1}{2} \tag{20}$$

3.4 Queuing Model and Network Parameters

We deploy the generic queuing model explained in the previous section to model the ring topology structure with $c = 1$. Each node is a queuing element with 1 successor node and 1 predecessor node. A lookup message can be sourced from the hosted CNs as well as it can be a forwarded lookup message from its successor. A node forward a lookup message to its successor if it is not the host SN of the lookup destination. If the node receiving the lookup is the host SN of the destination CN, the lookup is finished and no more forwarded. As explained before, we consider this as a lookup message entering a state 0 which represents an outside world of the open queuing network. The total arrival rate is divided equally among all participating SNs so that the arrival rate of fresh request from the hosted CNs to their host SN is $\frac{\lambda_{total}}{N}$. In this case, the node model is depicted in figure 1.

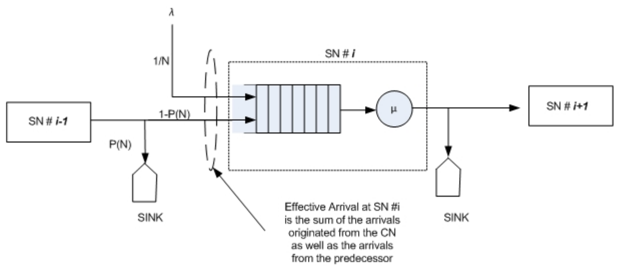

Figure 2: Node Model of a Ring Topology Overlay

We have obtained the routing parameters and these are shown in the node model. We present the equations to calculate the network parameters of interest in Porposition 4.

Proposition 4. *The visit ratio(e_i), the effective arrival rate (λ_i) and the utilization (ρ_i) at any SN i in our ring overlay model are given as follows*

$$e_i = \frac{N+1}{2N} \qquad \lambda_i = \frac{\lambda_{total}(N+1)}{2N} \qquad \rho_i = \frac{\lambda_{total}}{\mu}\frac{(N+1)}{2N} \tag{21}$$

Proof. Replacing the expression of P(N) from equation **??** in the results in the Proposition 2. □

3.5 Delay and Capacity Analysis

In this section we derive the mean lookup delay as well as the capacity of ring topology overlay.

Using the expression of ρ_i from equation 21 into equation 8, we obtain the expression for average delay of lookup in a ring topology as stated in the following theorem.

Theorem 3. *The average delay experienced by a lookup message in the ring overlay model represented as $\overline{D}$ is given as follows:*

$$\overline{D} = \frac{N+1}{2\mu - \lambda_{total}\frac{N+1}{N}} \tag{22}$$

Proof. The general expression of the delay expressed as a function of the number of SNs from equation 8 can be used to derive the delay for ring topology as follows.

$$\begin{aligned}\overline{D} =& \frac{1}{\mu(1 - \frac{\lambda_{total}}{\mu}\frac{N+1}{2N})}\frac{N+1}{2} \\ =& \frac{N+1}{2\mu - \lambda_{total}\frac{N+1}{N}}\end{aligned}$$

□

Delay expression seen as a function of N has an interesting property worth mentioning here. The root of $\frac{d\overline{D}}{dN}$ has a positive root as long as $\lambda_{total} < 2\mu$ and this root corresponds to a minimum of the function $\overline{D}$. This means that there exists an optimum number of N for minimum lookup delay. The expression can easily be used to estimate this minimum. So, in general, the delay first decreases with an increase in number of SNs until a point when the minimum is achieved. After this point of operation, any further increase in number of SNs increases the mean lookup delay.

Now, we derive the expression for capacity of the overlay. The capacity has been defined already in Definition 2. In overlay used for IP telephony service, the capacity of lookup processing might limit the number of maximum customers the system can accommodate.

Theorem 4. *The capacity(λ_{max}) supported by the load balanced ring overlay network is given as*

$$\lambda_{\max} = 2\mu \frac{N}{N+1} \tag{23}$$

Proof. The capacity corresponds to the total arrival rate (λ_{total}) when the utilization per SN (ρ_i) approaches 1. This corresponds to a higher bound in utilization per SN for finite delay. Equating $\rho_i = 1$ leads to the result. □

Equation 23 is an important result in terms of explaining how capacity (in terms of supporting more call lookup rates) scales with increasing number of SNs. For the system to have finite delay, the total arrival rate of lookup should always be less than this capacity. i.e.

$$\lambda_{total} < \lambda_{\max} \tag{24}$$

The functions shows that capacity of ring topology network increases with an increase in number of SNs. This is as expected. But, the capacity characteristics of Ring Topology exhibits an interesting property of an upper bound in achievable maximum capacity. We present this characteristics in the following corollary. This is an interesting result that suggests that the overlay structure has a direct impact on the capacity expansion property of the network.

Corollary 1. *The maximum capacity of a ring topology is limited to 2μ.*

Proof. We know that capacity increases with N. Now, the maximum capacity achievable is when N goes to infinity. We find the limit $\lim_{N\to\infty} \lambda_{max} = \lim_{N\to\infty} 2\mu\frac{N}{N+1} = 2\mu$ □

4 Chord Overlay Analysis

In this section, we consider a more complicated overlay structure called the Chord DHT. In followings we present very brief introduction to few key ideas in Chord DHT that are relevant for the work presented in this paper. For further details about Chord, please refer to [25].

4.1 Overlay Description

A Chord node or a Chord peer is a participant of the overlay network. Each peer hashes its identity (IP address for example) into a k bit long *node ID* using uniform hashing functions like SHA-1. The set of possible IDs form what is called a keyspace. The size of keyspace is $K = 2^k$. There are N peers in the network and usually $N << K$. A resource in a Chord Overlay is also hashed using the same function and thus is identified by a k bit *resource ID* which also falls in the same keyspace as the *node ID*.

The keys are logically organized in a circular space. The corresponding position by virtue of *node ID* in the circular space determines the logical position of the Chord peer in the overlay network. This has nothing to do with the actual position of the peer in the physical network. Imagine keys organized in a circle from key 0 to key $K-1$. N nodes occupy N different keys. The distance between two keys is considered in a clockwise sense. A peer i has a successor peer j which is the first peer encountered in the circular keyspace if we travel clockwise from the peer i. The keys between peer i and its successor are supposed to be the keyspace partition assigned to peer i. Any resource (or a pointer to the resource) with a resource ID in this subset is supposed to be stored by this peer. This is same as the Ring Topology described in the previous section also called a Chord Ring.

Unlike in Ring Topology, Chord has a more sophisticated routing structure. It defines the notion of fingers. A peer with *node ID* l maintains a finger table with k entries. The i^{th} entry of this finger table is populated with a peer (IP address and port number) who is responsible for the key $l + 2^{i-1}$. In effect, a peer has k entries with a maximum of k distinct fingers (or neighbors). But, in general, not all k entries are distinct. There might be repetitions because of the fact that more than 1 finger entry might point to the same peer. In Chord, besides fingers, they have defined the concept of successors. Moreover, a peer can maintain routing information of S consecutive successors (note that the first successor is also the first finger). In our analysis, we assume that a chord peer does not manage routing table for successors (except the first one).

Chord defines a greedy routing behavior. For any resource with resource ID p to be looked up, a chord peer upon receiving the corresponding lookup message first checks whether the key belongs to itself or not. If yes, the lookup does not need to be forwarded and will be terminated with a lookup response to the originator peer. If not, it will forward to the finger peer who is closest (closeness is defined by the circular distance in clockwise direction in the keyspace) to the resource ID p but not exceeding the resource ID. For a detailed understanding of this routing behavior, please refer to [25]. The greedy routing behavior can be proved to eventually result in successful forwarding as long as the successor is an entry of the routing table.[25]

4.2 Overlay Analysis Model

We consider a hierarchically structured P2P model based on Chord. In the model, SNs create the DHT overlay and the CNs rely on them for lookup service.

4.2.1 Assumptions

Here we elaborate the required assumptions of load-balance for finding the absorption probability. We define our load-balanced Chord with the assumption that SNs are distributed so that they occupy exactly same key-space partition of the id space. In a key space of size $K = 2^k$ we place $N = 2^c$ SNs such that each SN takes the responsibility of a key-space size of $2^{(k-c)}$. SNs are numbered from 0 to $(N - 1)$ in clockwise order. We illustrate this for the case of 8 SNs with a key-space size $K = 16$ in Figure 3.

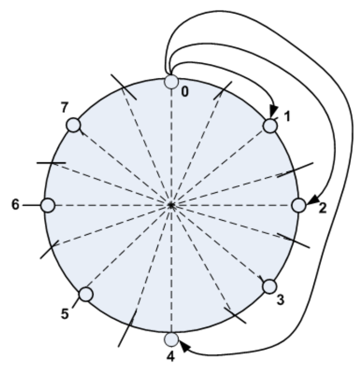

Figure 3: A simplified Chord overlay Model

For the sake of simplicity in our calculation we assume uniform key-space partitioning which leads to balanced load distribution on the SNs. Our assumption is an idealization however the original paper on Chord describes that to occur with "high probability" [25]. Our analysis as such fails to account for the load-imbalance in the DHT.

4.2.2 Characteristics

In the above Chord overlay model, we explore few facts which are stated and proved in the followings.
Peer out-degree
Peer out-degree is the number of distinct fingers of a given peer in the overlay network. Fingers are the neighbors of a peer who are listed in its routing table [25].

Proposition 5. *For $N = 2^c$ and $K = 2^k$ and $N \leq K$, the out-degree of a peer in the Chord overlay is c.*

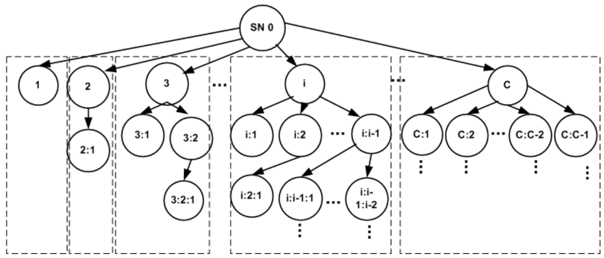

Figure 4: Recursively built Lookup Routing path tree for SN 0

The proof is presented in Appendix A. This Proposition is due to the fact that out of k finger entries of the routing table [25], only c of them are distinct. Note that when $N = K$, then the number of neighbors (distinct fingers) are k and represents the case when all the finger entries are distinct. We express it as

$$d(SN_i) = c \tag{25}$$

Chord Routing path as a rooted-tree
We consider the probable routing path of a lookup message originating at a given SN as a tree. We call this tree the *Lookup Routing Path Tree* (LRPT).

For any SN, if a lookup is originated in it, the lookup process can be terminated instantly if the destination is also being served by the same SN. Else, it needs to be forwarded to one of its c neighbors. The first neighbor is the successor SN. The c^{th} neighbor is the furthest neighbor. A careful observation of the routing semantics leads to the following Proposition.

Proposition 6. *The lookup message forwarded to i^{th} neighbor can either be terminated instantly or can be forwarded to one of the first $(i-1)$ neighbors of the i^{th} neighbor.*

Appendix B details the proof. The Proposition means for example, if a SN forwards a lookup to its 4^{th} neighbor, the lookup can either be terminated here or can be forwarded to one of its first 3 neighbors. Following Proposition 6, a general LRPT for SN 0 is shown in figure 4. A node labeled as $i : j : k$ is the k^{th} neighbor of the j^{th} neighbor of i^{th} neighbor of SN 0.

An example of the tree is shown in Figure 5, for $c = 5$. In the figure, a dot represents a SN in the lookup routing path. The number b labeled for each SN represents that it is the b^{th} neighbor of its immediate predecessor in the tree.

*Effective b-neighbor reachable SN(**E**b-**NRSN**)*
For a given SN i, an effective b-neighbor reachable SN is a SN in the routing path tree of SN i that, apart from absorption possibility, have $(b-1)$ possible forwarding branches. In figure 5, LRPT of SN 0 for the case of $c = 5$ is shown to illustrate this. Each SN falling in the LRPT is labeled with a number $b, 1 \le b \le c$ to mean that it is one of the Eb-NRSNs of SN 0. In the lookup routing path tree, we have the following facts for any SN_i:

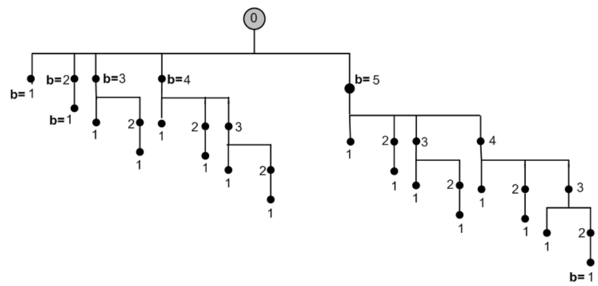

Figure 5: Recursively built Lookup Routing path tree for $c = 5$

1. Each **E**b**-NRSN** is referred by its predecessors for $\frac{2^{b-1}}{2^c}$ fraction of total keys. $(2^{b-c-1}K)$
2. The probability that an **E**b**-NRSN** is eventually referred for a lookup originating at SN_i is represented as π_b. Then we observe the following

$$\pi_b = \frac{1}{2^{c-(b-1)}} \tag{26}$$

. This is the immediate conclusion of the previous fact.

Another observed fact is stated in the following Proposition.

Proposition 7. *Each SN has* $2^{(c-b)}$ ***E****b****-NRSN*** *and is the* ***E****b****-NRSN*** *for* $2^{(c-b)}$ *SNs. This can be formulated as follows.*

$$\textit{Number of Eb-NRSN } (n_b) = 2^{(c-b)}$$
$$\textit{Number of SNs for which a certain SN is Eb-NRSN} = 2^{(c-b)} \tag{27}$$

The proof is given in Appendix C. As an example, in Figure 5 with $c = 5$ SN 0 has 16 E1-NRSN and 8 E2-NRSN and only 1 E5-NRSN.

4.3 Overlay Analysis

In this section, we derive absorption probability and the subsequent properties based on our generic model.

Proposition 8. *In the given Chord Overlay Model, the absorption probability* $(P(N))$ *is given as follows*

$$P(N) = \frac{1}{(1+\frac{c}{2})} \textit{ for } N = 2^c \textit{ and } N <= K \tag{28}$$

Proof. We define the following events.

$$B = \text{a lookup message in the system visits SN } i$$
$$A = \text{a lookup message arrived in SN } i \text{ is absorbed}$$

Then,

$$P(N) = P\{A|B\} = \frac{P\{AB\}}{P\{B\}} = \frac{P\{A\}}{P\{B\}} \tag{29}$$

In the load balanced model, with uniformly selected destinations, a lookup message is equiprobabally absorbed by all SNs. Therefore,

$$P\{A\} = \frac{1}{N} \tag{30}$$

We let, define event B_{ji} as

$$\begin{aligned} B_{ji} = & \text{ a lookup message in the system visits SN } i \\ & |\text{it was originated in SN } j \end{aligned}$$

Then, using the theorem of total probability,

$$P\{B\} = \sum_{j=0}^{N-1} P\{B_{ji}\}.\text{Prob}\{\text{Lookup is originated in SN } j\} \tag{31}$$

Here, load balanced assumption means

$$\text{Prob}\{\text{Lookup is originated in SN } j\} = \frac{1}{N} \tag{32}$$

Also, we have

$$P\{B_{ii}\} = 1$$

In general, $P\{B_{ji}\}$ depends upon the position of SN i in the lookup routing path tree of SN j. For any SN j whose Eb-NRSN is the SN i, we obtained the probability of the lookup generated at SN j to visit SN i in equation 26. We also know the number of Eb-NRSN in equation 27. We can rewrite equation 31 as follows:

$$P\{B\} = \frac{1}{N} \sum_{b=1}^{c} n_b \pi_b \tag{33}$$

Now using equation 26,27 and 33, we expand $P\{B\}$ as follows.

$$\begin{aligned} P\{B\} = & \frac{1}{N}[1 + 2^{c-1}\frac{1}{2^c} + 2^{c-2}\frac{1}{2^{(c-1)}} + \cdots + 2^{c-b}\frac{1}{2^{c-(b-1)}} \\ & + \cdots + 2\frac{1}{2^{(2)}} + \frac{1}{2}] = \frac{1}{N}[1 + \frac{c}{2}] \end{aligned} \tag{34}$$

Now, using equations 29, 30 and 34, we see
$P(N) = \frac{1}{(1+\frac{c}{2})}$ for $N = 2^c$ and $N <= K$. □

Proposition 9. *In the given Chord Overlay Model, the average number of SNs visited per lookup* ($\overline{S}$) *is given as follows*

$$\overline{S} = 1 + \frac{c}{2} \tag{35}$$

Proof. This can be seen in the generic analysis. The mean number of SNs visited per lookup is the inverse of the absorption probability. □

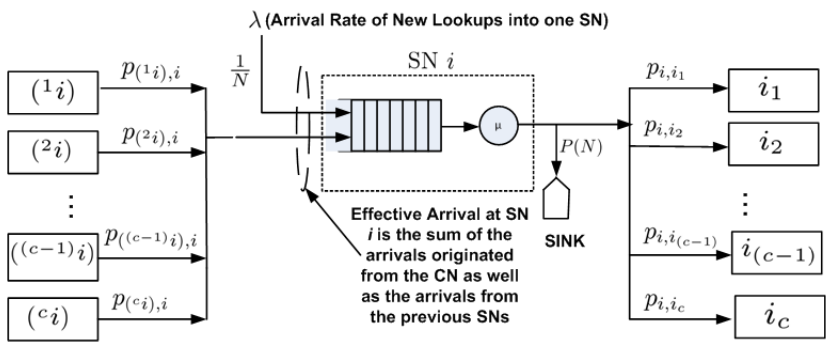

Figure 6: Node Model of Chord

4.4 Queuing Model and Network Parameters

The generic queuing model explained before can also be used to model the Chord topology structure with $c = log_2(N)$. Each node is a queuing element with c successor nodes (loosely called fingers in this paper) and c predecessors. A lookup message can be sourced from the hosted CNs as well as it can be a forwarded lookup message from it successor. A node forwards a lookup message to one of its c fingers if it is not the host SN of the lookup destination. If the node receiving the lookup is the host SN of the destination CN, the lookup is no more forwarded and is considered to enter the state 0 which represents an outside world of the open queuing network. The total arrival rate is divided equally among all participating SNs so that the arrival rate of fresh request from the hosted CNs to their host SN is $\frac{\lambda_{total}}{N}$. The node model is depicted in figure 4.4.

The routing parameters are shown in the node model. Next, we obtain the network parameter .

Proposition 10. *The visit ratio* (e_i)*, effective arrival rate* (λ_i) *and the utilization* (ρ_i) *of a given SN i in the load-balanced chord overlay model is* $e_i = \frac{1+\frac{c}{2}}{N}$ $\quad \lambda_i = \frac{\lambda_{total}}{N}(1 + \frac{c}{2}) \quad \rho_i = \frac{\lambda_{total}}{\mu.N}(1 + \frac{c}{2})$

Proof. The proof of these expressions is straightforward from the expression of $P(N)$ in Proposition 1. □

4.4.1 Delay and Capacity Analysis

Now, we can apply our analysis model to derive expression for the mean delay as well as capacity of the Chord network. The results are explained below.

Theorem 5. *The mean lookup delay (session setup delay)* $(\overline{D})$ *and the capacity* (λ_{max}) *in the load-balanced chord overlay model are*

$$\overline{D} = \frac{\left(1 + \frac{c}{2}\right)}{(\mu - \frac{\lambda_{total}}{N}(1 + \frac{c}{2}))}$$

$$\lambda_{\max} = N \times P(N) \times \mu = \frac{N}{1 + \frac{c}{2}}\mu$$

Proof. The proofs are simple substitution of the expression of $P(N)$ in theorem 1 and theorem 2. $\square$

5 Results and Discussions

In this section, we present the analytical results already derived in previous section along with simulation experiments. Equivalent simulation scenario were developed in *ns-2* [18] (a Discrete Event Simulator). We show the results of simulation along with the results of our mathematical analysis in an attempt to verify our analysis.

The mean lookup processing time at each SN ($\frac{1}{\mu}$) is set to be $\frac{1}{4}$ *sec.* A total of 700 peers are uniformly associated with N SNs. Each peer (CN) is generating Poisson traffic of lookup requests at a mean rate of 1 lookup per 2 Mins. The destination CN of each generated lookup is one of all of these peers selected uniform randomly. An initial overlay construction is made so as to guarantee the assumed load-balanced scenario.

5.1 Mean Lookup Delay

The absorption probability has been derived for the load-balanced chord in Proposition 8 and for Ring Topology in Proposition 10. The analytical result and the simulation results are presented in figure 7 for chord and 8 for Ring Topology.

In chord, $P(N)$ tends to decrease more slowly when number of SNs increases beyond a certain point. The more is the absorption probability , the less is the number of overlay hops. So, this scaling of absorption probability of Chord is definitely desirable from latency point of view. Compared to the Chord topology, the decrease in $P(N)$ in Ring is hyperbolic and the rate of decrease is higher for large values of N.

In Proposition 9 and equation 20, we have derived the average number of SNs visited by a lookup in a load-balanced Chord and Ring P2P overlay. Figure 9 represents the analytical results as well as the results from the equivalent simulation scenario for Chord DHT where as figure 10 is for Ring Topology. In chord overlay, the number of overlay hops per lookup increases logarithmically with an increase in number of SNs. This is in accordance to the order of the routing suggested in [25]. The increase is linear in Ring Topology.

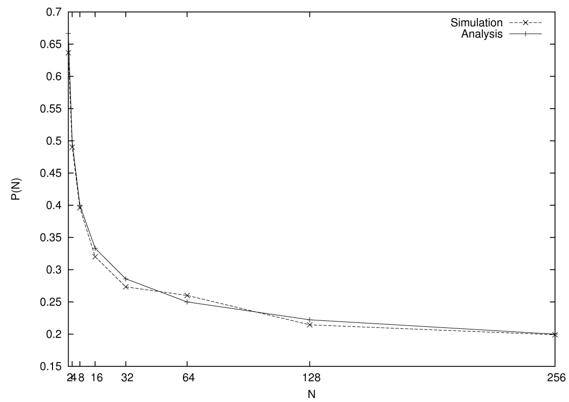

Figure 7: Absorption Probability for Load-balanced Chord

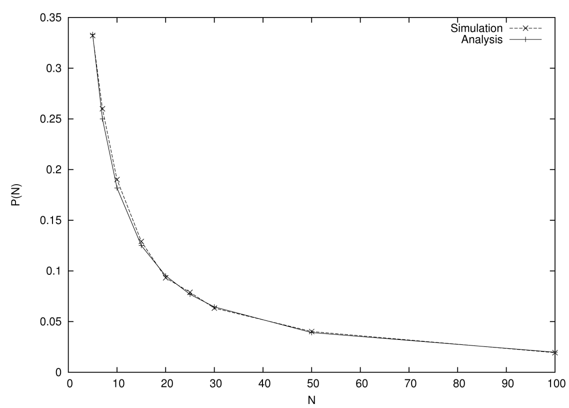

Figure 8: Absorption Probability for a Load-balanced Ring

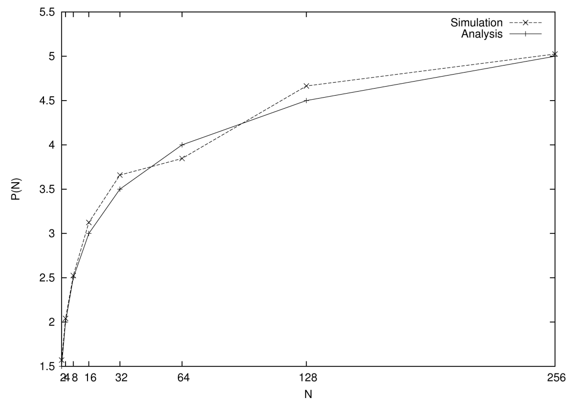

Figure 9: Average Number of SNs visited per Lookup for Load-balanced Chord

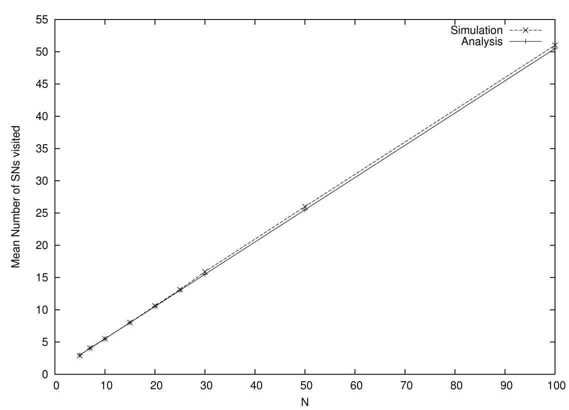

Figure 10: Average Number of SNs visited per Lookup for Load-balanced Ring

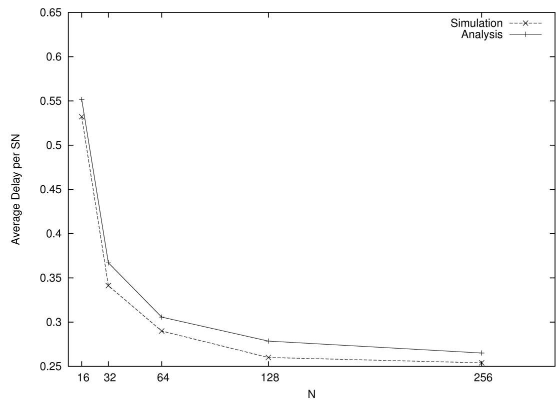

Figure 11: Mean Delay per node per Lookup of a load-balanced Chord

As figure 11 shows, by increasing the number of SNs the delay per SN (D_i) decreases. This is because by increasing N we effectively decrease the load in each SN. The less a SN is utilized, the less is the lookup delay we expect from it.

Figure 12 shows the average delay experienced by a lookup message in our load-balanced chord overlay. An increase in the number of SNs first leads to the reduction in the total mean lookup delay; this reduction is rather sharp at first. Then, after a certain point, the delay starts to rise. The point corresponding to the minimum delay represents the optimum number of SNs for minimum session set-up delay. Similar qualitative behavior is exhibited by the average delay in Ring Topology case (see figure 13). This behavior can be explained as follows.

The total lookup delay (D) is a product of two components ($D = D_i \times \overline{S}$). An increase in the number of SNs on the one had increases the average lookup hops $\overline{S}$) (figure 9) and on the other hand decreases the delay per SN (D_i) (figure 11). When the network is relatively more utilized (i.e. smaller N), the decrease in D_i by increasing N is more significant than the corresponding increase in $\overline{S}$. This results in a net decrease in the total lookup delay. But, as SNs become less utilized (i.e. larger N), then the decrease in D_i is less significant than the corresponding increase in $\overline{S}$. This results in an increase in total delay. Now, *minimum overlay delay is obtained at the point where the lookup delay changes characteristics. This corresponds to the optimum number of SNs for an IP telephony overlay that intends to minimize Session setup delay.*

5.2 Capacity (Maximum Lookup Processing Rate)

The capacity for our scenario of analysis has been defined and derived in the previous section. Here, we present the mathematical results along with equivalent

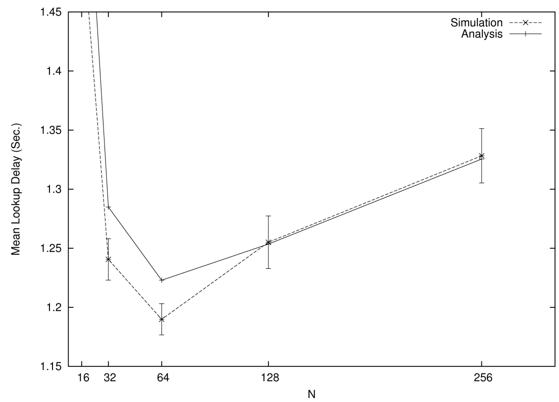

Figure 12: Mean Lookup Delay of a Load-balanced Chord

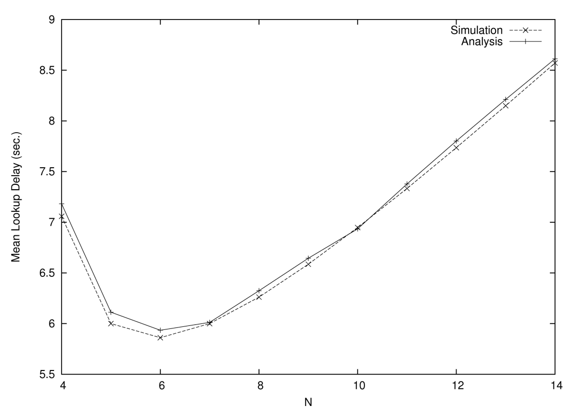

Figure 13: Mean Lookup Delay of a Load-balanced Ring

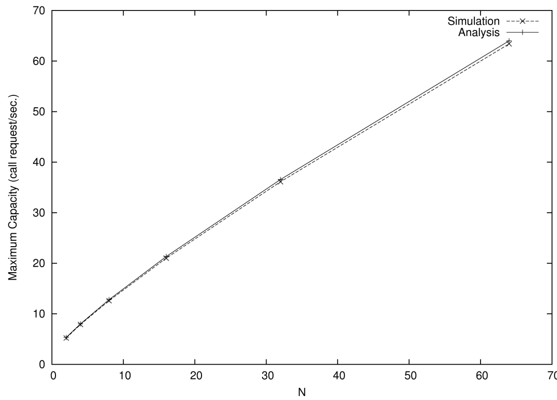

Figure 14: Lookup Processing Capacity of Load-balanced Chord for $\mu = 4 calls/sec$

simulations. The simulation of capacity for the equivalent scenario of analysis is challenging. In our simulation, we set lookup buffer size at 1000 which is large enough to simulate an infinite buffer. We increased the system lookup arrival rates until the point at which we observe packet drops.

Figure 14 and 15 show how capacity in terms of lookup processing of Chord and Ring overlay scales with an increase in number of SNs. We see that as number of SNs increases, the capacity of chord system increases. Similar scaling is existent in the Ring Topology. However, as predicted by our analysis the capacity of the network is quickly saturated to a value near the theoretical maximum of the Ring Topology network.

An interesting comparison between the Chord and Ring overlay is worth noting. We saw that capacity in chord increases arbitrarily with increase in N which means that the capacity of the network can be expanded by adding more SNs into the network. However, in Ring Topology, there is an upper bound of maximum capacity that can be achieved for a given processing capacity of a SN. So, we can not arbitrarily increase the capacity of the network just by adding more SNs. For a given capacity of each SN, there exist a system size (in terms of total lookup rate) that can not be supported. This is an interesting result showing how inefficient topologies can result in a limitation of the capacity of the network. Our analysis can be used to explore and compare such performance characteristics among overlay networks of different structures. Our mathematical results can be used to predict and dimension such systems based on the maximum users the system can support.

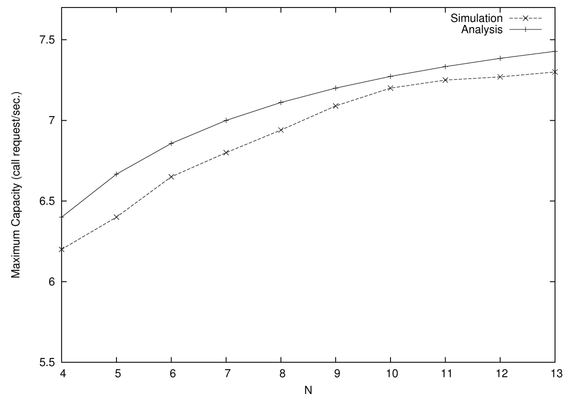

Figure 15: Lookup Processing Capacity of Load-balanced Ring for $\mu = 4 calls/sec.$

5.3 Delay-Capacity Trade-off

There exists a trade-off relationship between the mean lookup delay and capacity of such systems. We see that when the number of SNs is increased beyond the point of minimum session setup delay, capacity continues to increase whereas the delay also starts to increase. IP Telephony systems are required to operate below a certain session setup delay as per the recommendation of ITU [12]. So, we can not arbitrarily add more SNs into the system to increase the capacity. In this regard, there is a trade-off relationship between the two system performance parameters. Finding an optimum balance between capacity and delay is an interesting application of our results. The results can be extended to a more general optimization problem for benchmarking such systems. The results can be used in problems related to dimensioning and design as well as to evaluate the optimality of SN selection algorithms for such IP telephony application scenario. We reiterate the fact that the important performance parameters might be completely different for other applications.

6 Simulation based study of non-balanced general cases

The key limitation of our preceding analysis is the assumption of a load-balanced overlay. In average, such a load-balance is expected to occur in long term because of the uniform hashing functions used to embed resources into keyspace [25]. But, an instance of an overlay network will have non-equal keyspace partitioning and thus load-imbalance. The objective of the simulation study in this section is to see by simulation how the average behavior of real topologies relate to the load-balanced idealization used in this paper. Unlike the simulation before, we do not guarantee an equal keyspace partitioning among the SNs. We create instances of the overlay network by hashing the node addresses with uniform hashing function. The generated instance of the overlay network in general is imbalanced. Our average result is the average of the average of a large number of instances.

6.1 Simulation Parameters and Model

We create a discrete event simulation engine in Matlab and model each SN as a queuing element of service time $\frac{1}{\mu}$. Each SN is assigned a node ID of k bits long obtained by hashing its address by a uniform hashing function. Accordingly, the SNs create the overlay according to the Chord rules of finger selection. In average, λ_{total} lookup messages arrive in the system per second. The arrivals are Poisson distributed. SN i will get in effect an average rate of $\frac{\lambda_{total}}{K}K_i$ lookup messages per second from its hosted CNs where K_i is the size of the keyspace assigned to SN i. Besides this, a SN will also receive forwarded lookup messages. We set that the substream from hosted CN into each SN is also Poisson distributed. It is quite obvious to see that $P(N, i)$ is no more same for all SN i. Note that in the simulation of load-balanced case, we always made sure that K_i is same for all SN i so that the substreams into individual SNs were divided equally.

For a given N, if we generate a Chord overlay by hashing the addresses, we generate different instances of network topology. Unlike the idealized model, a Chord or Ring graph in this case is not a deterministic graph. It is a semi-random type and is dependent upon the node IDs generated. This makes a lot of combination of instances possible. To tackle the issue of finding an average performance parameter for a given value of N, we follow the following approach.

For each value of N, we generate node IDs and thus create a topology. This topology is considered to be an instance of the network for the given value of N. Call this instance an instance i. For instance i, we perform experiment for a time of $\frac{T}{I}$ with a total of L_i lookup messages simulated. Consider that we perform experiments for a total of I instances for a total simulation time of T. If for each lookup message i of the j^{th} instance, the experienced delay is d_{ij}, then the total mean delay is calculated as follows:

$$\overline{D}_N = \sum_{j=1}^{I} \sum_{i=1}^{L_i} d_{ij}$$

Since this approach is a statistical experiment, we also find out the confidence interval of the mean statistics for a confidence level of 95%. The parameters and their values used during simulations are shown in Table II.

Moreover, for comparision between the average results so obtained and the results of the balanced case, we plot the results together in the next section.

Parameter	Descritption	Value
N	Number of SNs	
K	Size of Key space	
k	Number of Bits of node ID= $log_2(K)$	
λ_{total}	Mean Total Arrival rate into the system	
$\frac{1}{\mu}$	Mean service time at each SN	
I	Number of Instances simulated for each N	
	Confidence Level of Results	95%

Table 2: Simulation Parameters

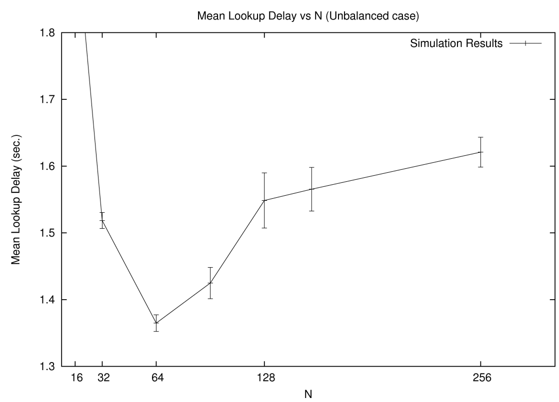

Figure 16: Mean Lookup Delay of the general Imbalanced Chord

6.2 Results and Discussions

In figure 16 we see the mean lookup delay for Chord topology in the case of general imbalance. The simulation results are obtained with a confidence level of 95%. We present the results to show that the mean lookup delay averaged over different instances of a general case is close to but larger than the mean lookup delay of a load balanced instance. Thus the prediction of our analytical model are expected to exceed in non-balanced scenario. The nature of the delay characteristics however are similar in both cases. Within the obtained error interval of the results, our simulation results are an indication that the simple load-balanced idealization can be a tool of estimating performance parameters with a fairly simple analytical model. Such an idealization is expected to reflect average performance behavior of the P2P overlay. We consider that these simulation results emphasize the effectiveness of the analysis method. Obtaining the relation between the average performance of the general case and the performance of a load-balanced idealization analytically however is our next extension of the work.

7 Extending the analysis to other DHT systems

Different DHT based systems have different routing protocols. The topology of the routing graph differs significantly from one DHT to other. The difference in the number of next-hop neighbors, the routing semantics used and many other parameters including the possibility of parallelization make the unified analysis of the diverse DHT systems very difficult. However, we emphasize that our analytical model is naturally decoupled into two parts. The first part of the analysis involves the calculation of the absorption probability ($P(N)$). This will be specific to the specific DHT protocol in use. However, the second part of the analysis requires the knowledge of $P(N)$ only. The rest of the analysis is general enough to be used for a diverse types of DHT routing protocols including Pastry and Kademilla. Thus by abstracting the details of specific routing scheme to a single parameter, the proposed analytical model can be used in estimating the lookup performance and lookup capacity of different and diverse types of DHT systems.

8 Related Works

[17] is an excellent survey of P2P systems for distributed lookup. [25] is the original paper describing Chord DHT. Our work considers this specific DHT based P2P as the lookup infrastructure for the analysis.

Because of the diverse applications of P2P systems, the performance analysis have been done for different kinds of P2P systems characterized by different performance goals. There is a good deal of work in modeling P2P file sharing systems [9, 27, 19, 16]. [9] models a P2P file-sharing system as a network of queues and with the help of model answers useful questions including the effect of freeloaders in the overall performance. In [27], authors study the service capacity of file-sharing systems using age dependent branching processes. They see how the average download delay of such system scales in offered load as well as peer departure rate. There have been work regarding the theoretical performance analysis of P2P streaming systems.[24] and [26] include the research in this direction.

There have been few measurement study of Skype traffic for performance analysis of the IP telephony service [6, 11]. Moreover in [7] authors model Skype to obtain the condition of Supernode load that lead to bottleneck. However to the best of our knowledge no work has been done to model structured P2P IP telephony overlay for obtaining the performance characteristics. Modeling and performance analysis of such system in terms of session-setup delay as well as service capacity is clearly lacking.

[14] is an interesting related work on the study of the lookup performance of super-peer based DHT systems. They propose bidirectional extensions to

existing DHT protocols and present a simulation study on the impact of such changes on the lookup performance as well as the additional cost incurred.

SIP is described in [21]. The IETF activities on P2P-SIP can be tracked at [2]. [15] and many other recent works present interworking architecture between Intelligent Multimedia Subsystems (IMS) and P2P-SIP. A list of P2P-SIP implementations and their performances can be found at [1].

Recently, peer-to-peer system has found its application extended to mobile cellular systems. [4] proposes a novel peer-to-peer architecture for facilitating mobile devices into well established content distribution networks.

9 Further Work and Conclusion

We have performed mathematical analysis of the performance behavior of structured P2P overlay based on Chord routing for IP telephony application scenario as a function of the number of Supernode resources. Our results show that mean session setup delay starts to decrease with an increase in number of SNs until a point when any further increase in number of SNs causes the delay to increase. This point of operation with minimum delay corresponds to the optimum supernodes required for minimum session setup delay. Similarly, the maximum lookup processing capacity (and thus the maximum number of users an IP telephony overlay can accommodate) increases with an increase in the SN resource with a scaling behavior without saturation. This means that capacity and the delay performance of overlay hold a trade-off relationship beyond the point of operation for minimum delay.

There have been countably a large number of Peer-to-Peer structures proposed in literature (e.g. Pastry [22], CAN [20]). We have considered Chord routing and thus there is a need of a more generalized analytical framework that can be used to quantify these parameters for several other P2P structures.

Moreover, the analysis as pointed before fails to account for the load-imbalance. Our future work is geared towards incorporating the general case of load-imbalance as well as network dynamics in our analysis.

References

[1] `http://www.p2psip.org/implementations.php` (Online).

[2] P2psip Status Pages . `http://tools.ietf.org/wg/p2psip/` (Online).

[3] S. A. Baset and H. G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. *INFOCOM 2006*, pages 1–11, 2006.

[4] Andreas Berl and Hermann de Meer. Integrating mobile cellular devices into popular peer-to-peer systems. *Telecommunication Systems*, 48:173–184, 2011. 10.1007/s11235-010-9327-x.

[5] Gunter Bolch, Stefan Greiner, Hermann de Meer, and Kishor S. Trivedi. Queueing networks and markov chains: Modeling and performance evaluation with computer science applications. pages 461–467, New York, 2006. John Wiley & Sons.

[6] D. Bonfiglio, M. Mellia, M. Meo, and D. Rossi. Detailed analysis of skype traffic. *Multimedia, IEEE Transactions on*, 11(1):117–127, Jan. 2009.

[7] G. Caizzone, A. Corghi, P. Giacomazzi, and M. Nonnoi. Analysis of the scalability of the overlay skype system. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 5652–5658, May 2008.

[8] Yatin Chawathe, Sylvia Ratnasamy, Lee Breslau, Nick Lanham, and Scott Shenker. Making gnutella-like p2p systems scalable. In *SIGCOMM '03*, pages 407–418, New York, NY, USA, 2003. ACM.

[9] Z. Ge, D.R. Figueiredo, Sharad Jaiswal, J. Kurose, and D. Towsley. Modeling peer-peer file sharing systems. In *INFOCOM 2003.*, volume 3, pages 2188–2198 vol.3, March-3 April 2003.

[10] J. Ghimire, Mehdi Mani, Noel Crespi, and Teerapat Sanguankotchakorn. Delay and capacity analysis of structured p2p overlay for ip telephony. *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2010.

[11] Te Yuan Huang, Kuan Ta Chen, and Polly Huang. Tuning skypes redundancy control algorithm for user satisfaction. 2009.

[12] ITU-T. Network grade of service parameters and target values for circuit-switched services in evolving isdn. *Receommendations E.721, Telecommunication Standardization Sector of ITU, Geneva, Switerland*, May 1999.

[13] Yoshihiro Kawahara, Tomonori Aoyama, and Hiroyuki Morikawa. A peer-to-peer message exchange scheme for large-scale networked virtual environments. *Telecommunication Systems*, 25:353–370, 2004. 10.1023/B:TELS.0000014789.70171.fd.

[14] Salma Ktari, Artur Hecker, and Houda Labiod. Symmetric routing in dht overlays. *Telecommunication Systems*, 48:163–172, 2011. 10.1007/s11235-010-9326-y.

[15] Hoyeon Lee and Hyun-Woo Lee. Interworking architecture between ims and p2psip for ubiquitous services. In *Information and Communication Technology Convergence (ICTC), 2010 International Conference on*, pages 288 –291, nov. 2010.

[16] F. Lo Piccolo and G. Neglia. The effect of heterogeneous link capacities in bittorrent-like file sharing systems. In *Peer-to-Peer Systems, 2004. International Workshop on Hot Topics in*, pages 40–47, Oct. 2004.

[17] Eng Keong 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, Quarter 2005.

[18] ns 2. `http://www.isi.edu/nsnam/ns/`.

[19] Dongyu Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 367–378, New York, NY, USA, 2004. ACM.

[20] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard M. Karp, and Scott Schenker. A scalable content-addressable network. 2001.

[21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621.

[22] Antony Rowstron and Peter Druschel. Pastry: scalable, decentraized object location and routing for large-scale peer-to-peer systems. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, November 2001.

[23] Gaurav Gupta Salman A. Baset and Henning Schulzrinne. OpenVoIP: An Open Peer-to-Peer VoIP and IM System. In *Proc of SIGCOMM'08 (demo)*, August 2008.

[24] Tara Small, Ben Liang, and Baochun Li. Scaling laws and tradeoffs in peer-to-peer live multimedia streaming. In *IN PROC. ACM MULTIMEDIA 06*, pages 539–548. ACM, 2006.

[25] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *SIGCOMM 2001 Conference*, Aug.

[26] Di Wu, Yong Liu, and K.W. Ross. Queuing network models for multi-channel p2p live streaming systems. In *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, pages 73–81, April 2009.

[27] Xiangying Yang and G. de Veciana. Service capacity of peer to peer networks. In *INFOCOM 2004.*, volume 4, pages 2242–2252 vol.4, March 2004.

Appendix A:
Proof of Proposition 5

Proposition: In the load-balanced Chord Model with $N = 2^c$ and $K = 2^k$, each peer will have c neighbors.

Proof. To prove this result, we assert the following three statements with proofs:

Statement 1: Any SN with SN id $x + 2^{i-1}T$ for $1 \leqslant i \leqslant c$ is a finger entry and a distinct neighbor of a given SN.
Proof of the statement 1: We represent $T = \frac{2^k}{2^c} = 2^{k-c}$, the partition of keyspace belonging to each SN. For a SN X with node id x, all other SNs are occupying the ids: $(x+T)\%K, (x+2T)\%K, (x+3T)\%K \cdots, (x+iT)\%K, \cdots$. % represents the modulo operation. But not all of them fall in the neighbor list of SN X.

We see that $x + 2^{i-1}T = x + 2^{k-c+i}$ is an element of the set $X = \{x + 2^{i-1} : i = 1, 2, 3, \cdots, k\}$ and this S is the set of all fingers anticipated by the Chord routing. (To learn about what are the possible finger entries in Chord, refer to [25]). The maximum entry of the finger list is when $2^{k-c-i} = 2^{k-1}$. i.e. $i = c - 1$. This proves the statement 1.

Statement 2: No other SNs fall in the set $X = \{x + 2^{i-1} : i = 1, 2, 3, \cdots, k\}$.
Proof of the statement 2: First we assume that this statement is false. For the above statement to be false, we require both of the following equations to be satisfied.

$$x + nT = x + 2^{i-1} \tag{36}$$

$$n \neq 2^l \text{ For any } l \in \{1, 2, 3, 4, \cdots c - 1\} \tag{37}$$

But, it is impossible to satisfy both equations together for $T = 2^{k-c}$. Thus statement 2 is valid.

Statement 3: No reference is ever made by the finger pointer to a key that lies between neighbor SNs with node id $x + 2^{j-1}T$ and $x + 2^jT$ for $1 \leqslant j \leqslant c$.
Proof of the statement 3: We determine the entry i_1 in the finger table corresponding to node ID $x + 2^{j-1}T$ from the equation below

$$\begin{aligned} x + 2^{j-1}T &= x + 2^{i_1-1} \\ i_1 &= j + k - c = j + k - c \end{aligned}$$

Also the corresponding entry i_2 in the finger table corresponding to node ID $x + 2^jT$ is solved by using the equation below

$$\begin{aligned} x + 2^{j-1}T &= x + 2^{i_2-1} \\ i_2 &= j + k - c + 1 \end{aligned}$$

The next entry after i_1 in finger entry table is $i_1 + 1$ pointing to the key $x + 2^{i_1+1-1}$. This is the entry corresponding to i_2. It means that the next finger pointer after i_1 is i_2 itself. There are no referrals to any key in between.

Statement 1 means that c SNs having id $x + 2^iT$ for $0 \leqslant i \leqslant c - 1$ fall in the neighbor list of SN X. Statement 2 says that no other SNs fall as the neighbor of SN X. Statement 3 shows that there would be no pointer references to a key between the two fingers as determined in statement 1. The three statements then together mean that there are exactly c neighbors of SN X. $\square$

Appendix B: Proof of Proposition 6

Proposition: Prove that in a load-balanced Chord, a lookup being forwarded to the i^{th} neighbor of SN X is forwarded to one of the first $i-1$ neighbors only.

Proof. In order to prove this result, we first assert the following statement.

Statement 4: The i^{th} neighbor of the i^{th} neighbor of node X is the $(i+1)^{th}$ neighbor of node X.
Proof of the statement 4: We represent the node id of the i^{th} neighbor of node X with node id x as $N_i(x)$. Then from previous argument in statement 1 of Appendix A, we have

$$N_i(x) = x + 2^{i-1}T \tag{38}$$

The i^{th} neighbor of the i^{th} neighbor of node X is represented as $N_i(N_i(x))$. We obtain the following

$$\begin{aligned} N_i(N_i(x)) &= N_i(x + 2^{i-1}T) = x + 2^{i-1}T + 2^{i-1}T \\ &= x + T(2^{i-1} + 2^{i-1}) = x + 2^iT \end{aligned} \tag{39}$$

But, from 38, we have

$$N_{i+1}(x) = x + 2^{i+1-1} \tag{40}$$

From 39 and 40, we conclude that statement 4 is true.

Using this statement 4, for any lookup forwarded by node X to its i^{th} neighbor Y, we can say that Y will never forward the lookup to its i^{th} neighbor Z. This is because if it had to, X would have directly forwarded to its $i+1^{th}$ neighbor (node Z) following the greedy algorithm. The greedy behavior of routing also prohibits any subsequent forward by node Y beyond its i^{th} neighbor because if any such nodes existed, Z would be chosen by X at first place. $\square$

Appendix C: Proof of Proposition 7

Proposition: Prove that each SN in the load-balanced Chord has 2^{c-b} Eb-NRSN.

Proof. In the Lookup Routing Path Tree (LRPT) of any SN i, a SN in the j^{th} branch of LRPT can have one more SN if and only if $j \geq b$.

We define $\Lambda_b(i)$ as the number of Eb-NRSN in the i^{th} branch of the LRPT of the SN under consideration. Then, we have the following:

$$\begin{aligned}
&\Lambda_b(i) = 0 \text{ For } 0 \leqslant i < b \\
&\Lambda_b(b) = 1
\end{aligned} \tag{41}$$

The LRPT has a general recurrence as described earlier. The recursion leads to the fact that the number of Eb-NRSN in the $i+1^{th}$ branch is the sum of the numbers in the previous branches. i.e.

$$\begin{aligned}
\Lambda_b(i+1) &= \sum_{j=0}^{i} \Lambda_b(j) = \sum_{j=b}^{i} \Lambda_b(j) \\
&= \Lambda_b(b) + \sum_{j=b+1}^{i} \Lambda_b(j) \text{ For all } i > b
\end{aligned} \tag{42}$$

This recurrence and the initial condition of $\Lambda_b(b) = 1$ means that the expression for $\Lambda_b(i+1)$ is expressed as:

$$\begin{aligned}
\Lambda_b(i+1) &= 1 + 1 + 2 + 2^2 + 2^3 + \cdots + 2^{i-b} = 1 + \frac{2^{i-b} - 1}{2 - 1} \\
&= 2^{i-b}
\end{aligned} \tag{43}$$

Then we have

$$\Lambda_b(c) = 1 + \sum_{i=b+1}^{c-1} \Lambda_b(i) = 2^{c-b-1} \tag{44}$$

Now, the total number of Eb-NRSN (i.e. n_b) is the sum of all these. i.e

$$\begin{aligned}
n_b &= \sum_{i=b}^{c} \Lambda_b(i) = 1 + \sum_{i=b+1}^{c-1} \Lambda_b(i) + \Lambda_b(c) \\
&= 2\Lambda_b(c) = 2 \times 2^{c-b-1} = 2^{c-b}
\end{aligned} \tag{45}$$

□

Appendix D: Derivation of Lookup Hop Distribution of Load-balanced Chord DHT

Let, random variable S denote the number of SNs visited by a lookup originating at one of the SNs.
we define,

$$P\{S = i\} = \text{Prob}\{ \text{ A Lookup visits } i \text{ SNs before absorption}\}$$

By observing the *lookup routing path tree* in figure 5, and defining $X(i)$ as the number of nodes in the tree in figure 5 that are $(i-1)$ hop away from the originating SN,
we note the followings:

$$X(1) = 1 \quad X(2) = c \quad X(3) = \sum_{i=1}^{c-1} i \quad X(4) = \sum_{i=1}^{c-2}\sum_{j=1}^{i} j$$

$$\ldots$$

$$X(\kappa) = \underbrace{\sum_{a=1}^{c-(\kappa-2)} \sum_{b=1}^{a} \cdots \sum_{i=1}^{h}\sum_{j=1}^{i}}_{(\kappa-2) \text{ sums of sums}} j \;\; for 1 \leq \kappa \leq (c+1) \tag{46}$$

For any lookup originating at one of the SNs, the probability of a lookup being absorbed after visiting i SNs is proportional to $X(i)$ as all the nodes in the root are equiprobabally absorbing a lookup.We get,

$$P\{S = i\} = \frac{X(i)}{N} \tag{47}$$

Now, the expected number of SNs visited before a lookup is absorbed is:

$$\begin{aligned} E\{S\} = \overline{S} &= \sum_{i=1}^{c+1} iP\{S = i\} \\ &= 1 + \frac{c}{2} \end{aligned} \tag{48}$$