Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Modeling, Optimization and Performance Evaluation of Distributed Hash Table Overlays

Ph.D. Theses
by
Péter Kersch

Research Supervisor:

Róbert Szabó, Ph.D.
Department of Telecommunications and Media Informatics

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUDAPEST, HUNGARY

2008

To my parents.

Table of Contents

List of Tables

List of Figures

Aknowledgments

I would like to thank my supervisor Róbert Szabó for his help, fruitful discussions and good ideas during my research work and special thanks go for his optimism and continuous encouragement. I would like to thank also the High-Speed Networks Laboratory for supporting my work and pushing me to start the submission process.

I'm grateful to Zoltán Lajos Kis for reviewing the manuscript and providing me valuable comments.

Special thanks go to my friends Emese Vágó and Barna Reskó who - being in the same boat of work towards PhD - provided me help, encouragement and made these years entertaining.

Finally, I'm most grateful to my family for their love and continuous support.

Chapter 1

Introduction

Originally emerging as file sharing applications, the peer-to-peer principle have proven to be very successful in a number of different areas of application ranging from Internet telephony (e.g., Skype [1]) via video on demand and Internet television (e.g., PPLive, SopCast [2]) to grid computing. Nowadays, peer-to-peer traffic accounts for more than half of the overall Internet traffic [3]. Among many advantages of the peer-to-peer principle, the elimination of bandwidth, processing and storage bottlenecks, savings on deployment and operation costs and the difficult traceability of peer-to-peer users all contributed to this success.

Generally speaking, a peer-to-peer network can be defined as a self-organizing system of equal, autonomous entities (peers), which aims for the shared usage of distributed resources in a networked environment avoiding central services [4]. One of the most basic functionalities of a peer-to-peer system is the lookup of shared resources in the network. This functionality can be implemented in two fundamentally different ways:

Unstructured peer-to-peer systems (e.g., Gnutella [5]) take a reactive approach. Peers do not maintain information about the location of resources at other nodes; resources in the network can be looked up using either flooding or random walk search [6] over an overlay network. No specific overlay structure needs to be maintained; the only overlay requirements are low diameter and resilience against partitioning. This reactive approach implies very low maintenance overhead and flooding-based or random walk search have the advantage of allowing also complex queries. The main disadvantage of unstructured peer-to-peer systems is the large communication overhead of queries. Flooding requires between $O(n)$ - $O(n^2)$ messages per query (where n is the number of nodes in the system). Random walk search outperforms flooding in communication overhead per query, especially for high replication ratios; however, it only provides probabilistic guarantees on successful lookups.

Structured peer-to-peer systems take a proactive approach. Peers maintain collectively a distributed indexing structure which allows fast and directed lookup of resources in the network. The main advantage of this architecture is the good scalability of the lookup process: the communication overhead of a query is $O(\log n)$ for most implementations. However, structured P2P systems require smart algorithms to minimize maintenance overhead and

does not intrinsically support complex queries. Distributed indexing in structured P2P systems is most often implemented in the form of *Distributed Hash Tables* (DHT) – although there are a few exceptions, e.g., skip graphs [7].

Architecturally, a DHT can be decomposed into two major components: a routing subsystem and a storage subsystem. The storages subsystem manages mapping of stored data items to DHT nodes while the routing subsystem maintains a (virtual) overlay network which provides efficient routing between these nodes.

A huge number of Distributed Hash Tables variants have been proposed [8, 9, 10, 11, 12, 13, 14, 15, 16, 17] since the publication of the very first structured peer-to-peer systems in 2001. Although lookup in most DHT overlays is based on the same foundations – being structurally similar to the "small-world" navigation model of Kleinberg [18] – architectural and algorithmic details of these different DHT variants differ significantly. Furthermore, their performance depends on a set of different and often incompatible parameters which makes analytical comparison rather difficult. Some aspects of DHT routing are covered by generic analytical models providing valuable help in comparing the myriad of different DHT implementations. E.g., the reachable component method (RCM) proposed in [19] allows comparison of static resilience of routing overlays while the authors in [20] developed an analytical framework to investigate the impact of lookup strategy, lookup parallelism and replication on lookup latencies.

My first research goal was to develop a generic model that allows analysis and comparison of a different aspect of DHT routing: the relationship between overlay structure and routing performance. In Chapter 4, I propose a model that can be used to describe and compare static routing performance for a large subclass of DHTs. The proposed model model uses logarithmic distance transformation and stochastic methods and provides upper bounds on the expected number of routing hops as a function of network size and a set of (uniform) overlay parameters.

P2P systems are inherently dynamic; peers join, leave and rejoin the network frequently. This process of permanent joining and leaving of peers is called *churn* in peer-to-peer terminology. In order to keep the distributed indexing structure consistent and guarantee lookup performance under churn, DHTs require resilient overlay structure, efficient maintenance mechanisms and churn-tolerant lookup strategies. Bamboo [17] and eQuus [16] are one of the few DHT implementations where churn tolerance has been an important explicit design criterion. The creators of Bamboo advocate for proactive maintenance to avoid maintenance avalanche effects and improve lookup performance under churn by fine-tuning timeout settings based on round trip time statistics. eQuus achieves resilience and low maintenance overhead by clustering DHT nodes based on physical proximity and ensuring that most maintenance traffic flows between physically closest nodes of the same cluster. In Chapter 5, I propose an alternative approach and present a stochastic overlay maintenance mechanism which reduces maintenance overhead asymptotically to the theoretical lower bound.

The rest of my dissertation is structured as follows: first, Chapter 2 provides a general

overview of the architecture and functionality of distributed hash tables. Then Chapter 3 introduces the overlay models and modeling assumptions used in the forthcoming chapters. My research work is presented in Chapter 4 and 5: Chapter 4 describes the generic model analyzing relationship between overlay structure and static routing performance while Chapter 5 presents the proposed stochastic maintenance mechanism. Finally, Chapter 6 concludes my dissertation.

Chapter 2

An Overview of Distributed Hash Tables

From the point of view of an application, Distributed Hash Tables provide similar functionality than ordinary "in memory" hash tables. An application can insert and remove key-value mappings, and given a key, it can retrieve the associated value (in the context of a peer-to-peer system, a key is an identifier used to refer to a shared resource while the associated value is the resource itself or the locator of the resource). All of these operations are performed quickly and efficiently and scale well for large amounts of data in both "in memory" and distributed hash tables. However, as opposed to ordinary hash tables, storage of key-value pairs is distributed over all nodes of the DHT and all hash table methods can be issued from any of these nodes (see Figure 2.1). Consequently, internal operation of a DHT differs significantly from the operation of ordinary "in memory" hash tables. To present the architecture and operation of distributed hash tables, I have used the terminology and formalism proposed in [21].

One of the key conceptual components of a DHT is the common metric space into which nodes and resources are mapped to. All distributed hash tables use a virtual identifier space $\mathcal{I}$ which possesses a closeness metric $d : \mathcal{I} \times \mathcal{I} \rightarrow \mathbf{R}$ so that $(\mathcal{I}, d)$ is a metric space or a quasi-metric space[1]. Both the group of peers forming the DHT and the set of all shared resources are mapped to this ID space $\mathcal{I}$ (see Figure 2.1). Mapping of peers can be described by a function $F_P : \mathcal{P} \rightarrow \mathcal{I}$ where $\mathcal{P}$ is the set of peers forming the DHT. F_P is usually implemented by either drawing a random identifier according to uniform distribution over $\mathcal{I}$ or by applying a hash function to the public key of the peer. Resources are mapped to $\mathcal{I}$ using a function $F_K : \mathcal{K} \rightarrow \mathcal{I}$ where K is the set of keys used to refer to shared resources. F_K is most often implemented by applying a hash function to the keys.

Peers responsible for a given resource are determined based on the above mappings to the common metric space $(\mathcal{I}, d)$. A key-value pair describing a resource is usually stored by the peer (or the set of peers) whose image in $(\mathcal{I}, d)$ is the closest to the image of the

[1] A quasi-metric space does not satisfy the symmetry requirement of metric spaces.

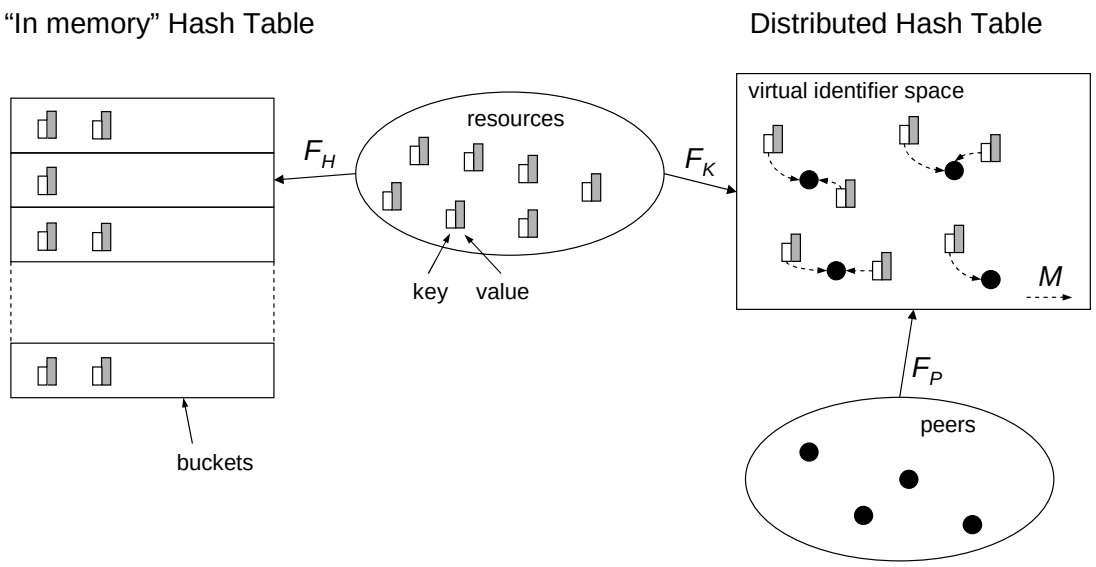

Figure 2.1: Comparison of "in memory" hash tables and DHTs

given resource in $(\mathcal{I}, d)$. Formally, this can be described using a function $\mathcal{M} : \mathcal{I} \rightarrow 2^{\mathcal{P}}$ and a constraint $\forall i \in \mathcal{I} : \forall p \in \mathcal{M}(i), \forall q \notin \mathcal{M}(i) : d(F_P(p), i) \leq d(F_P(q), i)$ on this function. As a result, locating a key-value pair (which describes a shared resource in a DHT) corresponds to finding one of the closest peers to the image of the resource in $(\mathcal{I}, d)$.

The function $\mathcal{M}$ is usually complete, which means that each identifier of $\mathcal{I}$ is under the responsibility of at least one peer. To provide fault tolerance $\mathcal{M}$ typically contains more than one element and the cardinality of $\mathcal{M}$ is typically constant, which means that each key-value pair is replicated to the same number of peers (Figure 2.1 shows the simplest case when each key-value pair is stored by only one peer).

Comparing distributed hash tables to ordinary hash tables, peers correspond to buckets and the function $\mathcal{M}$ corresponds to the hash function F_H in ordinary hash tables. Changing the number of buckets in an ordinary hash table implies changing the hash function F_H and this usually requires relocation of most key-value pairs. For "in memory" hash tables, bucket size is usually constant (or changes only rarely when reaching a capacity threshold), hence this is not a problem. In contrast, a peer-to-peer network is inherently dynamic and the set of peers in a DHT might change continuously, implying changes in the mapping of key-value pairs to nodes ($\mathcal{M}$). Continuous relocation of key-value pairs in a DHT would generate a huge communication overhead, hence changes in these mappings should be minimized when peers join and leave the network. DHTs address this problem by selecting responsible peers based on proximity in the metric space $(\mathcal{I}, d)$ as mentioned above. Consequently, changes in key-value pair $\rightarrow$ responsible peers mappings are restricted to the neighborhood of the joining or leaving peer in $(\mathcal{I}, d)$. (This concept is also called *consistent hashing* [22] and had been proposed for distributed web caching before the era of distributed hash tables.)

Another benefit of selecting responsible peers by proximity in $(\mathcal{I}, d)$ is that all DHT

operations can be easily implemented on top of a routing algorithm which locates the closest peers to a given point in $\mathcal{I}$. To realize this routing process, DHTs create and maintain an overlay network. An overlay network can be modeled by a directed graph $G = (\mathcal{P}, \mathcal{E})$ where $\mathcal{P}$ denotes the set of vertices (peers) while $\mathcal{E}$ denotes the set of edges (overlay connections[2]). Routing in the overlay is typically based on a simple greedy algorithm: a request for a given point in $\mathcal{I}$ is forwarded via the connection pointing to the peer which is the closest to this given point in $(\mathcal{I}, d)$.

Overlay topology depends heavily on distances between the images of peers in the metric space $(\mathcal{I}, d)$. In most DHT overlays, connections can be categorized into short-range (local) and long-range connections. Each node has short-range connections to some specific subset of the closest peers in $(\mathcal{I}, d)$. Additionally, they have long-range connections to some distant nodes so that the distribution of these connections is structurally similar to the family of small-worlds graphs introduced by Kleinberg in [18]. In this small-worlds graph family, the probability of having a long-range connection between to nodes is inversely proportional to the D^{th} power of their distance (where D is the dimension of the metric space), and Kleinberg has shown that this is necessary to provide efficient distributed search based solely on local information.

The role of short-range and long-range connections in the overlay is complementary. Short-range connections guarantee success of greedy forwarding: since each node is connected to its closest neighbors in $(\mathcal{I}, d)$, it is always possible to forward requests at least a small step closer to the target. In contrast, long-range connections are not critical for successful routing but they expedite the lookup process and usually provide $O(\log n)$ bounds on the average number of lookup hops. This is achieved by ensuring that the distance from the target decreases by a constant factor in expected value after each routing step.

[2] A connection from node v_1 to a peer node v_2 means that node v_1 knows the address of node v_2 (this is usually in the form of a pair of ID + IP address / port number). Different algorithms use different names for connections, e.g., Chord [8] calls them successors, predecessors and finger pointers, while in Pastry [9], they are called leaf set and routing table entries, etc.

Chapter 3

Modeling Assumptions and Notations

To describe the routing overlay of distributed hash tables, I have used the terminology and reference model defined in [21] (see also Chapter 2). I have considered a one dimensional Euclidean metric space within the interval $[0, 1)$ that wraps around (this can be represented as a ring, see Figure 3.1). Distance between two nodes in this metric space is defined as their distance along the ring in clockwise direction[1], formally: $d(x, y) = y - x + I_{x>y}$

Each node has two different types of connections to other nodes: short-range connections (called "local" connection in [21]) to a fixed number (N_S) of closest nodes (in clockwise direction) and long-range connections to some distant nodes. These nodes are called short-range and long-range peers of the node, respectively. Figure 3.1 shows short-range connections (S_1, S_2, S_3) and long-range connections (L_0, L_1, L_2, L_3) of node (A). The distance of the node and its farthest short-range peer is denoted by d_S.

Routing is assumed to be greedy: a node forwards a lookup request to its peer being the closest to the target node in the metric space of the DHT (without overshooting it). This greedy routing process can be described by the following pseudo-code algorithm:

```
ROUTE-TO-NODE(node, target)
1  while node ≠ target
2        do
3           proxy ← GET-CLOSEST-PEER(target, peers)
4           if d(proxy, target) < d(node, target)
5              then node ← proxy
6              else  error
```

Routing overlay of many DHT implementations (Chord [8], Pastry [9], Symphony [14], Accordion [15] etc.) can be described (or approximated) using the above system model (e.g. routing in Pastry is more complex but is based on the same greedy algorithm). However,

[1]This definition implies that the metric space is in fact only a quasi-metric space, since it does not satisfy the symmetry requirements. Extending the model to bidirectional routing where distance is defined as the shortest path along the ring (in any of the two directions), a real metric space can be obtained.

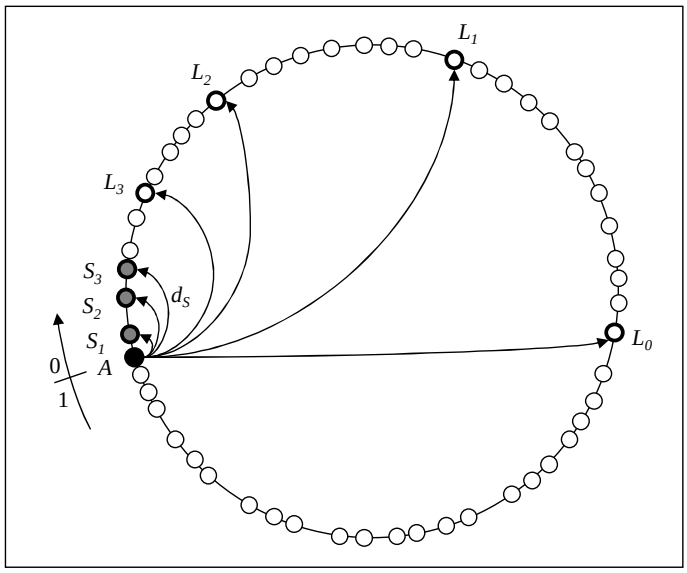

Figure 3.1: Model of unidirectional DHT overlays (example)

there are a few exceptions, for example DHTs using multidimensional metric spaces (e.g., CAN [10]) or non-Euclidean metric spaces (e.g., Kademlia [12]).

Degree of randomness

Since randomness and flexibility in the choice of long-range connections plays an important role in both analysis and maintenance of overlays, I have also defined the following two extreme DHT overlay categories:

Definition 3.1 (Probabilistic power-law routing overlay (PPLRO))**.** A routing overlay is called probabilistic power-law routing overlay when the choice of long-range connections is not deterministic and they only have to satisfy the following requirements: the probability of having a long-range connection to another overlay node is inversely proportional to the D^{th} power of the distance between the two nodes in the D dimensional metric space $(\mathcal{I}, d)$ where the DHT maps node identifiers [18]. Join algorithm of probabilistic power-law routing overlays create initial long-range connections of joining nodes according to this distance distribution and the choice of long-range connections is mutually independent of each other.

Definition 3.2 (Deterministic power-law routing overlay (DPLRO))**.** A routing overlay over a one-dimensional metric space[2] is called deterministic power-law routing overlay if long-range connections are determined by the power series of the distances $d_i = \frac{q}{c^i}$ where c and q are constant so that $c > 1$ and $0 < q \leq 1$. For unidirectional overlays, the i^{th} long-range connection is chosen as the first node whose distance exceeds d_i while for bidirectional

[2]Extending this definition to multidimensional metric spaces on the analogy of probabilistic power-law routing overlays is not trivial.

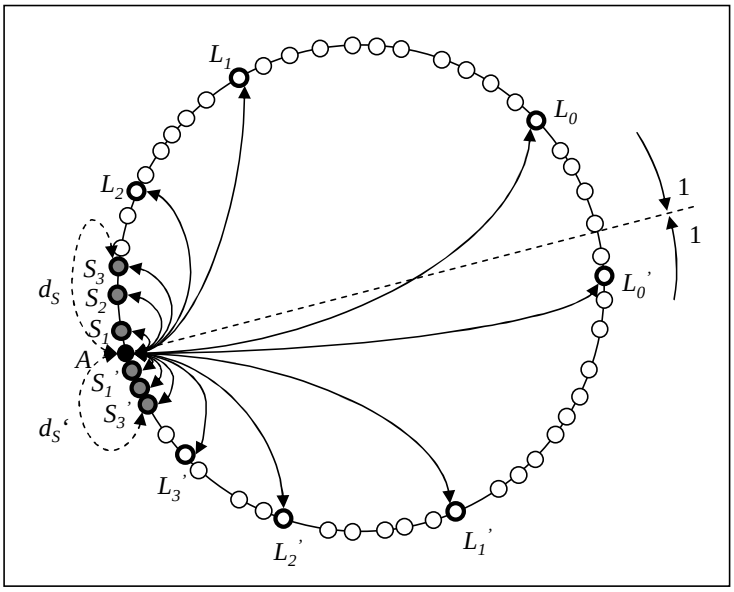

Figure 3.2: Model of bidirectional DHT overlays (example)

overlays, the i^{th} connection is the node closest to the point at distance d_i.

Symphony [14], Accordion [15] and the routing scheme proposed in [23] are using probabilistic routing overlays while a deterministic power-law routing overlay can be thought of as a generalization of the Chord [8] overlay (for Chord, $c = 2$).

It is important to note that the term "power-law" is also used to denote the overlay of unstructured P2P systems structurally similar to scale-free random graphs [24]. In that context, it refers to distribution of node degree. However, in Definition 3.1 and 3.2, the term "power-law" refers to distance distribution of long-range connections.

Bidirectional overlay model

The unidirectional overlay and routing model presented above can be easily extended to bidirectional overlays with bidirectional routing. In a bidirectional overlay, both short-range and long-range connections are bidirectional. The other important difference is the distance metric of the space $(\mathcal{I}, d)$. Using the ring representation, distance of two nodes is defined as their shortest distance along the ring, formally: $d_b(x, y) = min\,[d(x, y), d(y, x)]$. Hence in contrast to unidirectional overlays, this is a real metric space also satisfying the symmetry requirements. As a result, from the point of view of distances, each node can split the DHT metric space $(\mathcal{I}, d)$ into two symmetrical partitions. Connections of a node are created independently in both of these partitions. Figure 3.2 shows short-range and long-range connections of node A in both partitions of the metric space for an example bidirectional overlay.

Setting the circumference of the ring to 2 units for bidirectional overlays, it is easy to

derive the bidirectional equivalent of any unidirectional overlay (where the circumference of the ring is set to 1 unit). Connections of the bidirectional overlay obey the distance distribution of the corresponding unidirectional overlay separately in both partitions of the metric space $(\mathcal{I}, d)$. Hence the definition of probabilistic and deterministic power-law routing overlay can be extended for bidirectional connections by applying either Definition 3.1 or Definition 3.2 separately for both partitions of the metric space.

As a consequence of the definition of distance metric for bidirectional overlays, greedy routing also becomes bidirectional; requests can be forwarded in both directions depending on the position of the peer node being closest to the target.

Chapter 4 presents results for unidirectional routing overlays while Chapter 5 extends theses results and applies them for bidirectional routing overlays.

Node ID distribution

The distribution of node identifiers in the metric space $(\mathcal{I}, d)$ also affects mathematical analysis of routing in the overlay. The ID space of node identifiers is discrete and finite for most real systems, hence nodes can only be mapped to a finite subset of points in the Euclidean metric space $(\mathcal{I}, d)$. However, the granularity of this finite ID space is so fine (the size of the ID space varies between $2^{128} - 2^{256}$), that node ID mapping can be considered continuous in $(\mathcal{I}, d)$.

Furthermore, I assumed that given a network of n nodes, node identifiers are drawn independently at random according to a uniform distribution over the range $[0, 1)$ in $(\mathcal{I}, d)$ (this is a reasonable assumption in most cases). This implies that distances between adjacent IDs on the ring will be exponentially distributed. In a few cases (explicitly noted), I assumed that peer identifiers partition the metric space $(\mathcal{I}, d)$ deterministically in equal partitions. This is not a realistic scenario, but simplifies considerably mathematical analysis. In these cases, I've always compared analytical results using deterministic identifier assignment to simulation results using random uniform distribution of peer identifiers.

Finally, for long-range connection selection in probabilistic power-law routing overlays, I've assumed that it is possible to find a peer node at any given distance (drawn according to a given distribution) in the metric space. This is not realistic in a real system composed of a finite number of nodes. In practice, the closest existing node to the given point is used instead. However, the resulting error between these theoretical and real distances is inversely proportional to the size of the network, hence this is negligible for large networks (which are the main scope of this dissertation). Section 4.2.4 addresses this problem in more details.

Chapter 4

Mathematical Modeling of Routing in DHTs

A plethora of Distributed Hash Table concepts have been proposed and analyzed in the past 6-7 years to provide scalable and robust distributed storage and lookup systems [8, 9, 10, 11, 12, 14, 15, 16, 17], etc. Although architectural and algorithmic details of these DHT proposals can differ significantly, the foundations of lookup mechanisms are very similar for most of them. There are several empirical studies (based on simulations) comparing static and dynamic performance of different DHT routing mechanisms using various parameter settings [25, 26]. There exist also detailed analytical models for some DHTs, however these models are usually restricted to one specific DHT implementation. Finally, some aspects of DHT routing are covered by generic models, e.g., static resilience of DHT routing against failure [19] or the impact of lookup strategy, lookup parallelism and replication on DHT routing performance under churn [20]. However, to the best of my knowledge, there exist no generic analytical models capturing the relationship between overlay structure and routing performance of DHTs in static networks. In this chapter, I try to fill this gap proposing a generic stochastic model of DHT overlays and overlay routing covering a large family of DHTs.

The proposed analytical model builds on the fact that most DHT overlays are structurally similar to the "small-world" model of Kleinberg [18] and the sequence of long-range connections of a DHT node becomes linear after logarithmic transformation of distances in the DHT metric space. More specifically, I have identified a large subclass of DHT overlays (regular power-law routing overlays) where this transformed sequence can be described for each node as independently selected random samples from an infinite renewal process. Using this renewal process model, I analyze the distribution of the per-hop routing progress in general and also for the special cases of the deterministic and probabilistic power-law routing overlays. Furthermore, I introduce the λ long-range connection density and the c_v long-range connection density coefficient of variation parameters to characterize long-range connection distribution of an overlay. Finally, using renewal theory, I derive upper bounds

on the expected number of routing hops as a function of network size and the above overlay parameters.

The rest of this chapter is structured as follows: In Section 4.1, I present related works for performance analysis and comparison of different DHT routing mechanisms and provide a brief introduction to renewal theory – widely used in the upcoming sections. In Section 4.2, I introduce the idea of logarithmically transformed view and analyze long-range connection distribution in this transformed view. Finally, in Section 4.3, I analyze routing via long-range connections in the transformed view and derive upper bounds on the expected number of routing hops.

4.1 Related works

Defining models and metrics to describe performance of different DHT routing architectures is not a trivial task. An application using the DHT lookup service is mostly interested in lookup latencies and in the ratio of successful lookups. A user running DHT implementations might also be concerned by resource usage (CPU, memory, storage, bandwidth, etc...) while a network operator is only interested in the overall traffic (lookup + control) generated in the network. Since most of these describe conflicting objectives, comparison only makes sense if conflicting performance metrics are analyzed together describing fundamental trade-offs.

In the following subsection, I review existing models describing performance of DHT routing in static networks (modeling of performance tradeoffs in dynamic network environments will be tackled in Section 5.1). Then, I provide a brief introduction to renewal theory, which is the main mathematical tool used in the rest of this chapter.

4.1.1 Modeling DHT routing

Some of the commonly used performance metrics (e.g., overlay network diameter, node state) are not directly relevant for neither applications nor users nor network operators. In [27], the author investigates the trade off between node state and overlay network diameter. Loguinov et al. also use network diameter as the primary metric for routing in [28].

Node state affects primarily memory usage at nodes. However, the amount of memory required to keep track of connections is typically far from being a bottleneck in current systems. Node state can also influence the maintenance bandwidth (e.g., in DHTs using per connection periodic keep-alive messages to detect connection failures). However, it cannot be used as a general metric to characterize maintenance traffic.

Overlay network diameter can be used to derive only lower bounds on the worst-case number of routing hops for a lookup in a given overlay structure. Short paths between nodes do not guarantee that a distributed routing algorithm is also able to find them [18]. Hence, the distribution or the average number of routing hops is a more informative performance metric which also allows to derive [20] lookup latency – a key performance metric from a user perspective.

Analytical comparison of a performance metric (e.g., the number of routing hops) of different DHTs is usually described by asymptotic notation, commonly used to characterize algorithm complexity. E.g., CAN [10] with a D dimensional identifier space provides lookups in $O(\frac{D}{2}n^{\frac{1}{D}})$ hops in a network of n nodes. Although this is a useful and simple way to determine scalability of a particular algorithm, it has its limitations. Due to potentially different unknown constants hidden within the notation, it is not possible to compare two different algorithms with the same asymptotic behavior (e.g., $O(\log n)$ hop count is typical for many DHTs). Furthermore, it is also possible that an algorithm with better asymptotic behavior performs worse for practical network sizes.

Asymptotic notation may even be misleading when not used carefully. The paper presenting Koorde [11] (a DHT based on de Bruijn graphs) is a good example of such a misuse. Using a base-k de Bruijn graph, Koorde completes routing in $O(log_k\ n)$ hops. Based on this, the authors claim that choosing $k = \log n$, routing cost is $O(\log n/\log k) = O(\log n/\log\log n)$. However, the base of the underlying de Bruijn graph cannot be changed on the fly as the network grows since this would require rebuilding the whole DHT from the scratch. Therefore the parameter k should not be treated as a function of network size. (Similarly, the dimension D of a CAN [10] network is not expressed as a function of network size because this is also a parameter that cannot be changed without rebuilding the whole system.) As a consequence, the number of routing hops for Koorde using base-k de Bruijn graphs is in fact $O(\log n/\log k)$.

For a few DHT architectures, there are some exact analytical results: e.g., the average number of routing hops for Chord [8] is $\frac{1}{2}log_2 n$. [20] is one of the few papers which provide a generic analytical framework for the performance comparison of different DHTs. Given the average number of routing hops in static networks, the authors analyze the influence of three key factors on routing performance under churn: lookup strategy, lookup parallelism and replication. My results on the expected number of routing hops in static networks can be potentially used as an input for this analytical framework to derive these additional performance metrics.

Finally – although not directly related to distributed hash tables – the "small-world" navigation model of Kleinberg [18] is a fundamental contribution to theory of routing in distributed systems. A network is said to be "small-world" when there exists a short path between any two nodes, although most nodes are not directly connected. This low network diameter is a necessary but not sufficient property for efficient distributed routing. In [18], Kleinberg investigates requirements on overlay topology for efficient distributed routing based solely on local information in small-world networks. Similarly to DHT overlays, he defines a graph (embedded into a metric space) with short-range connections to the closest nodes and long-range connection(s) to some distant nodes. As in DHTs, Kleinberg's routing is greedy: requests are forwarded via the peer node being the closest to the target node in the metric space. As the main finding of the paper, Kleinberg shows that distributed routing will achieve the best asymptotical performance when the probability of having a long-range

connection to another node is inversely proportional to the D^{th} power of distance of the two nodes (where D is the dimension of the metric space embedding the small-world graph).

Most DHT routing architectures – although not inspired by Kleinberg's work – can be related to the one dimensional Kleinberg small-world model.

4.1.2 Renewal processes revisited

Renewal processes are a special class of stochastic processes used to model independent identically distributed occurrences. Let X_1, X_2, X_3, ... be independent identically distributed ($i.i.d$) and positive random variables defined by the distribution function $P(X < x) = F(x)$. Furthermore, let T_n be defined as $T_n = \sum_{i=1}^{n} X_i$. Then the counting process $Y(t) = \max\{n : T_n \leq t\}$ is a renewal process ($t \geq 0$).

Renewal processes are usually defined in the time domain. In the time domain, $Y(t)$ denotes the number of events until time t, T_n corresponds to the occurrence time of the n^{th} event and the random variables X_i correspond to inter-arrival times between subsequent events. The name renewal process is motivated by the fact that every time there is an occurrence, the process "starts all over again"; it renews itself (since the variables X_i are $i.i.d$).

In contrast to the general usage, renewal processes in my dissertation are not defined in the time domain but in the distance domain of a one dimensional metric space. Furthermore occurrences are not events but the images of long-range connections in this metric space and the random variables X_i correspond to distances between the images of subsequent long-range connections.

In the followings, I briefly list the results of renewal theory that I use in the upcoming sections (for further reading, see [29], [30] and [31]). Note that the vocabulary of renewal theory traditionally assumes a time domain for renewal processes. However, all results are equally valid for the distance domain too.

Renewal function The expected value of the number of arrivals in function of the elapsed time is called renewal function: $m(t) = E[Y(t)]$.

Residual life Picking a random point in time (t), the random variable corresponding to the time from this point until the next event (at time $T_{Y(t)+1}$) in a renewal process is called residual life:

$$V(t) = T_{Y(t)+1} - t \tag{4.1}$$

Residual life is also called *residual lifetime*, *residual time* or *forward recurrence time*.

Expected value of asymptotic residual life The expected value of asymptotic residual life in a renewal process can be expressed as

$$\lim_{t \to \infty} E[v] = \frac{\mu_2}{2\mu}. \tag{4.2}$$

where $\mu = E[x]$ is the expected value of inter-arrival times and $\mu_2 = E[x^2]$ is the second moment of inter-arrival times.

Distribution of asymptotic residual life Considering a renewal process with an inter-arrival time distribution $F(x)$, the probability density function of asymptotic residual life can be expressed as

$$\lim_{t\to\infty} g(v) = \frac{1 - F(v)}{\mu}, \tag{4.3}$$

where $\mu = E[x]$ is the expected value of inter-arrival times.

Length of a randomly selected renewal period Picking a random point in time (t) in a renewal process, the *pdf* of the length of the renewal period marked by this point ($T_{Y(t)+1}$ - $T_{Y(t)}$) is asymptotically:

$$\lim_{t\to\infty} h(x') = \frac{f(x')x'}{\mu}, \tag{4.4}$$

where $f(x)$ is the *pdf* of inter-arrival times and $\mu = E[x]$ is the expected value of inter-arrival times in the renewal process. It is important to note that the distribution of x' and x are not the same since a random point in time will select longer periods at higher probability than shorter periods.

Note that considering a random sample from a renewal process, the above formulas are also valid in general, not only for the asymptotic case.

Lorden bound The renewal function of a renewal process is upper bounded by

$$m(t) \leq \frac{t}{\mu} + \frac{\mu_2}{\mu^2} + 1, \tag{4.5}$$

where μ is the expected value of inter-arrival times and μ_2 is the second moment of inter-arrival times in the renewal process (see [29], page 110.)

Poisson processes are a special class of renewal processes. Inter-arrival times in a Poisson process are exponentially distributed. A Poisson process can be characterized by the λ parameter of this exponential distribution which is also called the intensity of the process.

A Poisson process of intensity λ can also be defined as a pure birth process: the probability that an arrival occurs during an infinitesimally small interval dt is λdt (independent of arrivals outside this interval) and the probability that more than one arrival occurs is $o(dt)$. This definition is equivalent with the renewal process definition.

Random sampling Random and independent sampling of events with probability p from a Poisson process of rate λ results into a Poisson process of rate $p\lambda$.

Superposition Superposition of two Poisson process of rate λ_1 and λ_2 respectively results into a Poisson process of rate $\lambda_1 + \lambda_2$.

4.2 Transformed view of long-range connections

Transformation is a widely used mathematical concept in many disciplines to reveal, analyze and exploit hidden system characteristics. One of the best known examples of the application of a transformation method is JPEG encoding where discrete cosine transform maps a 8x8 pixel area into spatial frequency components [32]. In this example, transformation is used to exploit "hidden characteristic" of human vision being much more sensitive to small variations in color and in brightness for lower spatial frequencies than for higher frequencies. Hence higher spatial frequency components can be encoded at smaller resolutions. In my thesis work, I apply a logarithmic transformation to distances between node identifiers in the metric space of a DHT to reveal "hidden characteristics" of DHT routing.

Definition 4.1 (Logarithmically transformed view)**.** Let $(\mathcal{I}, d)$ be the metric space of a DHT (see Chapter 2) where the distance between the image $x_0 = F_P(p_0)$ of a node p_0 and the image $x_i = F_P(p_i)$ of another node p_i is defined as $d(x_0, x_i)$. Then, using the transformation function $f_t(u) = -\ln u$, the distance of p_0 and p_i in the logarithmically transformed view of p_0 is defined as:

$$d'(x_0, x_i) = f_t\left[d(x_0, x_i)\right] = -\ln\left[d(x_0, x_i)\right]. \tag{4.6}$$

It is important to note that $d'(x_0, x_1)$ is not a distance metric since it does not obey the three metric space properties. However, in my dissertation, $d'(x_0, x_1)$ is not used as a distance metric; transformation of distances is only a mathematical tool within the concept of logarithmically transformed view.

The transformed view of a base node p_0 can be used to characterize distances between p_0 and a set of other nodes in a DHT. This transformed view can be represented along a half-line as follows: the base node p_0 itself is at the end of the half-line while other DHT nodes p_i (e.g. peers of the base node, or the target node of a lookup process) are represented along this half-line at distance $d'(x_0, x_i)$ from p_0.

4.2.1 Long-range connection density

Figure 4.1 represents long-range connections of a Chord [8], Pastry [9] and Kademlia [12] node as well as long-range connections of a node in a probabilistic power-law routing overlay (e.g., Symphony [14] or Accordion [15]). For Pastry, the parameter b is the bit length of numbers in the routing table, for Kademlia, the parameter k is the maximum size of buckets and for Chord, the parameter c is the parameter used in the definition of deterministic power-law routing overlays (see Definition 3.2). For each of these DHTs, the upper line shows long-range peers of the node in the real metric space[1] (to ease graphical representation, the ring geometry of the metric space has been straightened) while the lower line shows these peers in the logarithmically transformed view of the node. In the real metric space, the

[1]Since Kademlia uses a XOR metric, long-range peers of the Kademlia node are represented based on their XOR distance from the node. Note that this is different from ID-based placement along the ring.

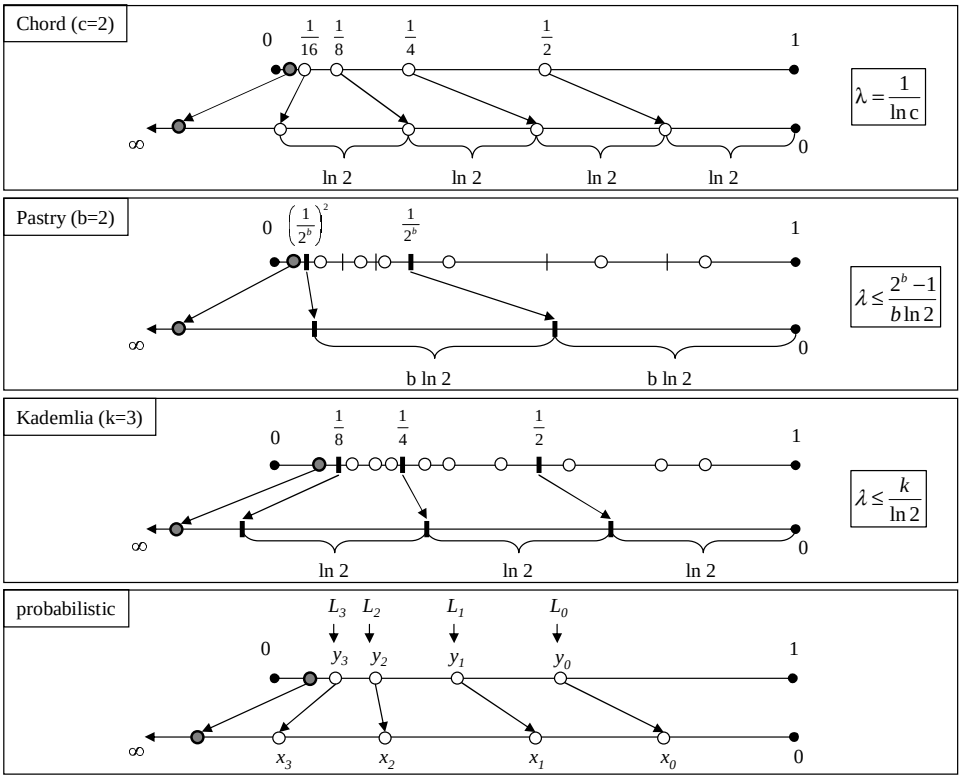

Figure 4.1: Comparison of the routing table of some well known DHTs

represented node is in point 0. In the transformed view, this point corresponds to $+\infty$. Finally, long-range connections in the transformed view span within the range $[0, -\ln d_S)$, where d_S is the distance from the farthest short-range peer of the node (represented by grey circle in Figure 4.1).

Consider now the segment $(d_S, 1)$ covered by long-range connections in the real metric space which corresponds to the segment $(0, -\ln d_S)$ in the transformed view of the node. Although long-range connection distribution differs for each of the above DHT implementations, Figure 4.1 shows that it is possible to partition this segment in the transformed view into equally sized partitions of length Δx so that the number of long-range connections $N_L(\Delta x)$ be the same inside each of these partitions (either deterministically or in expected value). Based on this observation, I have defined a $\lambda_{\Delta x}$ long-range connection density parameter as

$$\lambda_{\Delta x} = \frac{E\left[N_L(\Delta x)\right]}{\Delta x}. \tag{4.7}$$

In general, the choice of Δx is not arbitrary. In order to obtain constant long-range connection density in the entire long-range connection domain of the transformed view, Δx might need to be set to a DHT specific value (see again Figure 4.1). However, for a large subclass of DHTs (including regular power-law routing overlays defined in the next subsection), long-range connection density can be defined independent of the size Δx of

partitions as:

$$\lambda = \lim_{\Delta x \to 0} \frac{E\left[N_L(\Delta x)\right]}{\Delta x}. \tag{4.8}$$

The main advantage of the proposed λ (or $\lambda_{\Delta x}$) parameter is that it provides a simple and generic way to characterize long-range connection distribution. Furthermore, in $O(\log n)$ node state DHTs, λ characterizes the overlay independent of network size. For DHTs with constant node degree (e.g., Symphony [14]), λ depends on the size of the network ($\lambda \sim \frac{1}{\ln n}$).

Many DHT implementations have one or more tunable system parameter which affects long-range connection distribution, e.g., the bucket size k for Kademlia or the bit length b of numbers in the routing table for Pastry. The λ long-range connection density parameter allows easy comparison of overlay structure for these DHT implementations despite their mutually incompatible sets of system parameters. Figure 4.1 shows the λ parameter for each DHT as a function of their tunable system parameters. Note that in many cases (e.g., Chord, Pastry or Kademlia), the theoretical long-range connection values are only upper bounds of the actual long-range connection density since some routing table entries may be empty (especially for shorter distances).

4.2.2 Regular power-law routing overlays

Definition 4.2 (Regular power-law routing overlays (RPLRO))**.** A power-law routing overlay is called regular if the sequence of long-range connections of a node in its transformed view correspond to a randomly chosen sample of length $-\ln d_S$ from an infinite renewal process and these random samples are chosen independently for each node.

Hence, from the definition of renewal processes, distances between subsequent long-range connections of a node in its transformed view are *i.i.d* in a RPLRO. Furthermore, the distribution function $F(x)$ of these *i.i.d* random variables identifies unambiguously a regular power-law routing overlay. Random sampling from an infinite process (instead of defining long-range connections as a renewal process starting from point 0 in the transformed view) ensures uniformity of long-range connection density in the whole long-range connection range (also including the first part of this range).

Theorem 4.1. *Long-range connection density of a regular power-law routing overlay is uniformly $\lambda = \frac{1}{\mu}$, where μ is the mean distance between subsequent long-range connections of a node in its transformed view.*

Proof. Let $f(x)$ be the *pdf* and $F(x)$ be the *cdf* of renewal intervals (corresponding to the distances between subsequent long-range connections of a node in its transformed view). As a result of the random sampling property of RPLROs, the starting point of each partition of length $\Delta x \to 0$ can be considered as a randomly chosen point in the corresponding infinite renewal process. Hence this interval of length $\Delta x \to 0$ contains at least one renewal point (long-range connection) either when the corresponding renewal interval (selected by this randomly chosen point) is smaller than Δx or when this interval is larger than Δx but

the distance between the randomly selected point and the next renewal is less than Δx. Since the *pdf* of the length of the renewal period selected by a random point is $\frac{f(x)x}{\mu}$, the probability that such an interval contains at least one renewal (long-range connection) can be written as:

$$p_1 = \int_0^{\Delta x} \frac{f(x)x}{\mu} dx + \int_{\Delta x}^{\infty} \frac{f(x)x}{\mu} \frac{\Delta x}{x} dx. \tag{4.9}$$

The probability that this interval contains k or more arrivals (where $k > 1$) can be upper bounded as follows:

$$p_k \leq p_1 F^{k-1}(\Delta x). \tag{4.10}$$

The expected number of renewals (long-range connections) within a randomly selected period of length Δx can be written as:

$$E[N_L(\Delta x)] = \sum_{i=1}^{\infty} p_i. \tag{4.11}$$

Hence, combining 4.10 and 4.11:

$$p_1 \quad \leq \quad E[N_L(\Delta x)] \quad \leq \quad p_1 \left[1 + \sum_{i=1}^{\infty} F^i(\Delta x)\right]. \tag{4.12}$$

Dividing by Δx and applying Equation 4.7:

$$\frac{p_1}{\Delta x} \quad \leq \quad \lambda_{\Delta x} \quad \leq \quad \frac{p_1}{\Delta x} \left[1 + \sum_{i=1}^{\infty} F^i(\Delta x)\right]. \tag{4.13}$$

Applying $\Delta x \to 0$ to Equation 4.9 and the (reasonable) assumptions[2] that $\lim_{\Delta x \to 0} F(\Delta x) = 0$ and $\lim_{\Delta x \to 0} f(\Delta x) < \infty$:

$$\lim_{\Delta x \to 0} \frac{p_1}{\Delta x} = \frac{f(0)\Delta x}{2\mu} + \frac{1 - F(\Delta x)}{\mu} = 0 + \frac{1}{\mu}. \tag{4.14}$$

Finally, substituting Equation 4.14 into the Inequality 4.13:

$$\frac{1}{\mu} \leq \lambda \leq \frac{1}{\mu} \quad \to \quad \lambda = \frac{1}{\mu}. \tag{4.15}$$

□

Probabilistic power-law routing overlays are regular (see Section 4.2.3). Pastry and Kademlia are not regular but are close to being regular with only small distortions. Finally, Chord and deterministic power-law routing overlays in general are not regular, but, they can be made regular: Considering the transformed view of a node in a DPLRO, its first long-range peer is always located at $\ln c$. Substituting the constant q in Definition 3.2 by a random variable so that this first long-range peer be evenly distributed in the range $[0, \ln c]$, the overlay becomes regular.

[2]These assumptions can be made because 0 distance between subsequent long-range connections does not make sense in an overlay.

To characterize regular power-law routing overlays, I also introduced a c_v long-range connection coefficient of variance parameter describing the relative variance of distances between long-range connections in the transformed view. $c_v = \frac{\sigma}{\mu}$, where σ is the standard deviation while μ is the mean of distances between consecutive long-range connections in the transformed view of a node. In Section 4.3.1, I show that using the λ and c_v parameters it is possible to derive a lower bound on routing performance.

4.2.3 Probabilistic power-law routing overlays

It is interesting to compare the degree of randomness in the choice of long-range connection for different DHT implementations in Figure 4.1. In Chord, each connection is deterministic. Pastry is somewhat more flexible, each routing table entry may contain any node of the network from a given ID range, increasing the degree of randomness. Kademlia goes one small step further in flexibility and randomness and allows the choice of any nodes (up to a maximum number of k) from a given range (Chapter 5 further analyzes randomness of long-range connection from a maintenance point of view).

However, the choice of long-range connections can be made "even more random" within the family of routing overlays for which long-range connection density can be defined. For the "most random" routing overlays out of this family, long-range connections of a node in its transformed view correspond to a random and independent placement of points in the range $(0, -\ln d_S)$ according to a uniform distribution, which is equivalent to a truncated (spatial) Poisson process. In the following, I show that this family of "most random" routing overlays is equivalent to the family of probabilistic power-law routing overlays over a one dimensional metric space (see Definition 3.1).

Theorem 4.2. *Consider a truncated Poisson process of rate λ in the range $(0, -\ln d_S)$. Furthermore consider a routing overlay where the sequence of long-range connections in the transformed view of each node is defined as a random realization of this Poisson process. Then this routing overlay is a probabilistic power-law routing overlay of long-range connection density λ.*

Proof. Consider a small range $[x, x + \Delta x]$ in the transformed view. Inverse transforming this range back to the real metric space using $y = f_t^{-1}(x) = e^{-x}$ results into the range $[e^{-x-\Delta x}, e^{-x}] = [y - \Delta y, y]$ in the real metric space.

Using the birth process definition of Poisson processes, the probability of having an arrival (long-range connection) in the range $[x, x+\Delta x]$ of the transformed view is $\lambda \Delta x$ when $\Delta x \to 0$. Since the inverse transformation function $f_t^{-1}(x)$ is strictly monotone decreasing, the probability of having a long-range connection in the corresponding range $[y - \Delta y, y]$ of the real metric space is the same.

Using the derivative of the transformation function $f_t(y) = -\ln y$, it is possible to express

the relationship between the length of these ranges when they are infinitesimally small:

$$\lim_{\Delta y \to 0} \Delta x = -f_t'(y)\Delta y = \frac{\Delta y}{y}.$$

Hence the probability of having a long-range connection in an infinitesimally small range of length $\Delta y \to 0$ at a distance y from this node is $\lambda\frac{\Delta y}{y}$. This is equivalent to the long-range connection distribution requirement of Definition 3.1 for one dimensional metric spaces. Being generated from a Poisson process, long-range connections also satisfy the independence requirement of Definition 3.1, hence the generated overlay is a probabilistic power-law routing overlay.

Finally, the expected number of arrivals in a Poisson process of rate λ for an interval of length Δx is $\lambda\Delta x$, hence substituting into Equation 4.7 any positive value of Δx interval length, the obtained long-range connection density value for this overlay equals to the λ rate of the generating Poisson process.

□

Theorem 4.3. *Consider a probabilistic power-law routing overlay in a one dimensional metric space with a long-range connection density λ. Then the sequence of long-range connections of any node in its transformed view correspond to a random realization of a truncated Poisson of rate λ in the range $(0, -\ln d_S)$.*

Proof. According to Definition 3.1, the probability of having a long-range connection in an small range $[y-\Delta y, y]$ of a one dimensional metric space is $c\frac{\Delta y}{y}$ when $\Delta y \to 0$ (c is a positive constant).

Transforming now this range into the transformed view using $x = f_t(y) = -\ln y$ results into the range $[-\ln y, -\ln(y-\Delta y)] = [x, x+\Delta x]$. Since the transformation $f_t(y)$ is strictly monotone decreasing, the probability of having a long-range connection in the corresponding range $[x, x+\Delta x]$ of the transformed view is the same.

Using the derivative of the inverse transformation function $f_t^{-1}(x) = e^{-x}$, it is possible to express the relationship between the length of these ranges when they are infinitesimally small:

$$\lim_{\Delta x \to 0} \Delta y = -f_t^{-1'}(x)\Delta x = e^{-x}\Delta x = y\Delta x.$$

Hence the probability of having a long-range connection in an infinitesimally small range of length $\Delta x \to 0$ in the transformed view is $c\frac{\Delta y}{y} = c\Delta x$, independent of the value of x within the range $(0, -\ln d_S)$. Since long-range connections of a probabilistic power law routing overlay are also independent of each other according to Definition 3.1, the sequence of long-range connections of any node in its transformed view correspond to a random realization of a truncated Poisson of rate c in the range $(0, -\ln d_S)$.

Finally, using the definition of long-range connection density and the assumption that the long-range connection density of the given overlay is λ, it is deducible that $c = \lambda$.

□

Since a Poisson process is a special renewal process, from Theorem 4.2, it follows that probabilistic power-law routing overlays belong to the subclass of regular power-law routing overlays.

4.2.4 Distortions in the transformed view

In Section 4.2.3, I've assumed that a node can find (and create a connection to) a peer node at any given point of the metric space $(\mathcal{I}, d)$. In reality, given a network of n nodes, a connection can be created only to $n-1$ points in $(\mathcal{I}, d)$ corresponding to the images of all the other nodes in $(\mathcal{I}, d)$. Hence in real systems, a connection is established to the peer node whose images is the closest to the "theoretical" point in $(\mathcal{I}, d)$ drawn according to the required distribution. This introduces small distortions to theoretical distance distribution.

Similar distortions exist for deterministic power-law routing overlays. E.g., in Chord, long-range connections (fingers) point to the first node whose distance is not smaller than $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, ...$ This results into a sequence of distances $\frac{1}{2}+\epsilon_1, \frac{1}{4}+\epsilon_2, \frac{1}{8}+\epsilon_3, ...$ In a network of n nodes uniformly distributed in the range $[0,1)$ of the metric space $(\mathcal{I}, d)$, ϵ will be a random variable with exponential distribution of parameter n.

While the distribution of this small offset is the same for all distances in the real metric space, it depends strongly on the distance in the transformed view of a node. For large real distances, this offset is negligible, however, it increases exponentially and can be considerable for small real distances in the transformed view.

4.3 Stochastic analysis of routing

The role of short-range and long-range connections in the routing process is complementary. While short-range contacts ensure the success of greedy forwarding, long-range contacts expedite routing and provide $O(\log n)$ bounds on the number of routing hops. For deterministic routing geometries, the routing process can be clearly separated into a first phase using only long-range contacts and a second phase using only short-range contacts. For non-deterministic routing geometries, the first routing hops usually take place via long-range connections while the last hops usually take place via short-range connections and the probability of routing via a short-range peer increases monotonously approaching to the target. Nevertheless, for non-deterministic routing geometries, it is not possible to separate routing process into distinct long-range and short-range routing phases.

Analytical study of this dual routing process is rather complicated. Analysis becomes much easier if forwarding is restricted to either only short-range or only long-range connections.

Restricting forwarding to short-range connections, progress toward the target becomes linear. Assuming that node identifiers are drawn independently and at random according to a uniform distribution from the interval $[0,1)$ of the metric space $(\mathcal{I}, d)$ (see Chapter 3), each routing hop has the same length in expected value independent of the current distance

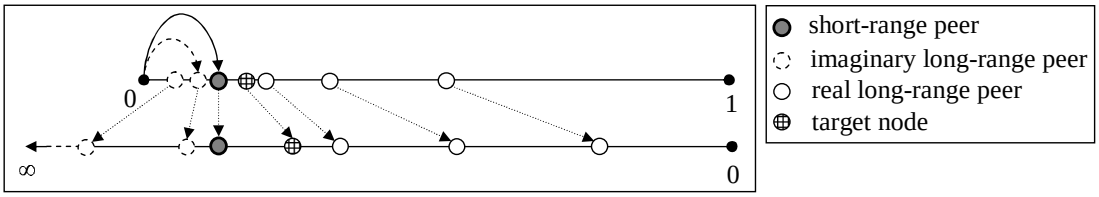

Figure 4.2: Forwarding through imaginary long-range connections

from the target[3]. The length of consecutive routing hops can be described by a series of independent Erlang distributed random variable with rate n (number of nodes in the network) and shape parameter N_S (number of short-range connections per node). Obviously, routing via only short-range contacts degrades routing performance from $O(\log n)$ to $O(n)$.

In Section 4.3.1, I show that using logarithmic transformation, analysis is also possible when restricting forwarding to long-range connections; progress toward the target will be linear in the transformed view of the target. However, simply forbidding forwarding via short-range contacts may cause routing failures. Therefore, to analyze long-range only forwarding, I use an imaginary routing overlay where the sequence of long-range connections in the transformed view of a node is infinite instead of being truncated after reaching the short-range connection domain. For regular power-law routing overlays, this means that long-range connections correspond to infinite random samples from a renewal process instead of random samples of length $-\ln d_S$.

In the real routing overlay, forwarding takes place via a short-range peer only when the target is closer than the closest long-range peer of the forwarding node. When the real routing overlay is forwarding via a short-range peer, the modified long-range only model is forwarding through an imaginary long-range peer at a smaller distance

Figure 4.2 demonstrates the difference between real forwarding (via a short-range connection) and long-range only forwarding via an imaginary long-range connection. The upper line in the figure represent the real metric space while the lower line shows the transformed view of the forwarding node. For a real overlay, forwarding occurs via a short-range peer. However, when restricting forwarding to long-range connections in order to simplify analysis, forwarding takes place via the imaginary long-range peer being the closest to the target node.

Forwarding via an imaginary long-range connection always results in less progress than the real forwarding would result in via a short-range connection. Therefore results on routing progress obtained from analysis restricted to long-range forwarding can be used as a lower bound on real routing progress.

Modeling long-range only forwarding, a routing process is terminated when the distance of the current forwarding node from the target decreases below d_S (the distance between the

[3]. If $N_S > 1$, the expected value of the last hop reaching the target is smaller.

target node and the node whose farthest short-range peer is the target node[4]). Let M_l be the number of routing hops for long-range only routing until the termination and let M be the number of routing hops for the real routing process. Since routing progress of long-range only forwarding is always equal to or less than routing progress of real forwarding, the real routing process will always reach a peer with direct short-range connection to the target node in M_l or less hops. Hence $M_l + 1$ can be used as an upper bound on the real number of routing hops:

$$M \leq M_l + 1. \tag{4.16}$$

The rest of this section is structured as follows: Section 4.3.1 analyzes progress of the routing process in the transformed view of the target using this long-range only forwarding model. Then Section 4.3.2 uses the obtained long-range only results to derive upper bounds on the number of routing hops for the real routing process (using both short-range and long-range connections).

4.3.1 Analysis of routing in the transformed view

Analysis of routing in the transformed view can be best introduced through an example. Figure 4.3/a shows one hop of an example routing process: a request reaches forwarding node F_k in step k and node F_k forwards this request to its long-range peer F_{k+1} being the closest to the target node T without overshooting it. Figure 4.3/b shows distances in the real metric space (upper line) and the transformed view (lower line) of node F_k while Figure 4.3/c shows the same distances as seen in the real and transformed view of the target. Note that the default direction of the ring is reversed in Figure 4.3/c in order to represent remaining distances from the perspective of the target node.

d_k and d_{k+1} is the distance from the target in step k and $k+1$ respectively, while d'_k and d'_{k+1} are the same distances in the transformed view of the target. To analyze per hop routing progress in the transformed view of the target node, I express the progress $u_k = d'_{k+1} - d'_k$ toward the target after step k as a function of the distance v_k from the next-hop node in the transformed view of forwarding node F_k. Applying transformation to distances in Figure 4.3/c:

$$u_k = -\ln d_{k+1} - d'_k. \tag{4.17}$$

Inverse transforming distances in Figure 4.3/b:

$$d_{k+1} = e^{-d'_k} - e^{-d'_k - v_k} = e^{-d'_k}(1 - e^{-v_k}). \tag{4.18}$$

Finally, substituting Equation 4.18 into 4.17:

$$u_k = -\ln\left[e^{-d'_k}\left(1 - e^{-v_k}\right)\right] - d'_k = -\ln\left(1 - e^{-v_k}\right). \tag{4.19}$$

[4]Assuming uniform distribution of node identifers in the DHT metric space, d_S will be a random variable with Erlang distribution of rate n and shape parameter N_S (where n is the number of nodes in the DHT and N_S is the number of short-range connections per node.)

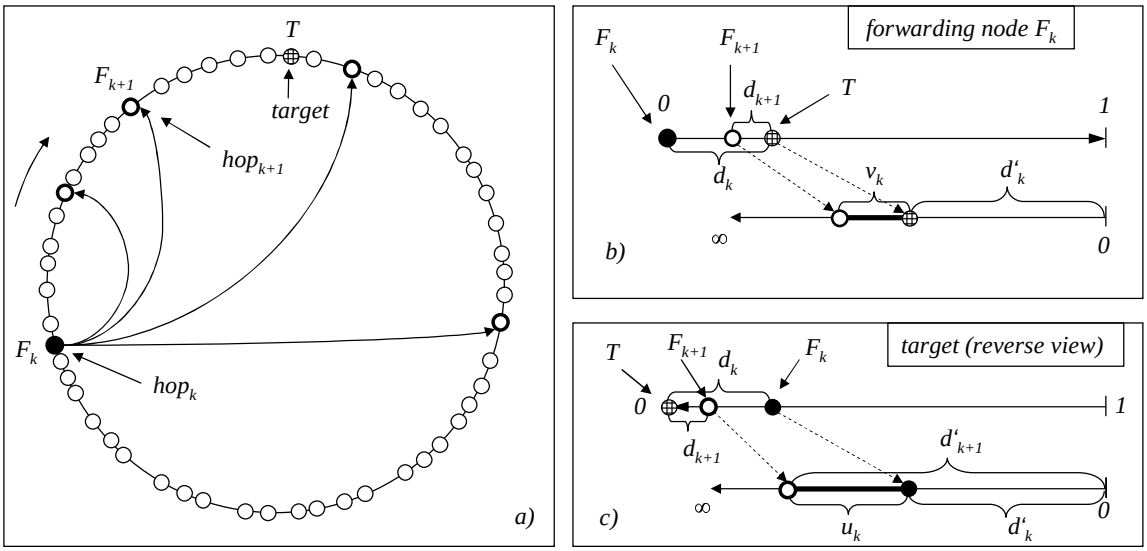

Figure 4.3: Routing from hop k to hop $k+1$

Hence the progress u_k after routing hop k in the transformed view of the target can be expressed as a function of the distance v_k from the next-hop node in the transformed view of forwarding node F_k:

$$u = h(v) = -\ln\left(1 - e^{-v}\right). \tag{4.20}$$

Note: as Chapter 3 mentions, routing in DHTs with non-Euclidean metric spaces cannot be analyzed using this model. The reason is that Equation 4.18 uses the assumption that $d(x,z) = d(x,y) + d(y,z)$ which holds only for the one dimensional Euclidean metric space. For any other metric spaces: $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality).

In the followings, I analyze routing in regular power-law routing overlays in general. Using the above transformation results, I derive a lower bound on routing performance as a function of λ and c_v. Then I analyze two special cases of regular power-law routing overlays in more details: the probabilistic and "regularized" deterministic overlays. I derive exact analytical results on their routing performance and compare these results to the generic lower bound.

Regular power-law routing overlays

Theorem 4.4. *Considering the routing process via long-range connections in a regular power law routing overlay, the length of the per-hop progress u_k in the transformed view of the target is* i.i.d *for subsequent hops. Furthermore, the expected value of this per-hop routing progress is lower bounded by*

$$E[u_k] \geq -\ln\left[1 - e^{-\frac{1+c_v^2}{2\lambda}}\right]. \tag{4.21}$$

where λ is the long-range connection density and c_v is the long-range connections density coefficient of variation in the overlay.

Proof. In regular power-law routing overlays, the sequence of long-range connections of different nodes in their transformed view correspond to independently selected random samples from an infinite renewal process. Therefore, the target node can be considered as a uniformly distributed random point in the transformed view of a forwarding node (see Figure 4.3/b). As a consequence, the random variable v_k corresponds to the distance of a random point from the next renewal (long-range connection) in the renewal process of long-range connection (residual life). Hence, the series of the random variables v_k will be *i.i.d*, and applying Equation 4.20, the series of the random variables u_k will be also *i.i.d*.

Let $\mu = E[x]$ be the expected value and $\mu_2 = E[x^2]$ be the second moment of the length of renewal periods (corresponding to distances between subsequent long-range connection in the transformed view of a node). Then, from renewal theory, the mean residual life in this renewal process (corresponding to $E[v]$) can be expressed as:

$$E[v] = \frac{E[x^2]}{2E[x]} = \frac{Var(x) + E^2[x]}{2E[x]} = \frac{E[x]}{2}(1 + c_v^2). \tag{4.22}$$

Using Theorem 4.1, $E[x] = \frac{1}{\lambda}$ for any RPLRO, hence:

$$E[v] = \frac{1 + c_v^2}{2\lambda}. \tag{4.23}$$

Using Equation 4.20, the distribution of the per-hop routing progress in the transformed view of the target (u) can be expressed as a convex function $h(v)$ of the random variable v. Therefore the Jensen inequality can be applied as follows:

$$E[u] = E[h(v)] \geq h(E[v]) = -\ln\left[1 - e^{-\frac{1+c_v^2}{2\lambda}}\right]. \tag{4.24}$$

□

Probabilistic power-law routing overlays

According to Theorem 4.3 on PPLROs, the sequence of long-range connections in the transformed view of a node can be described as a realization of a stationary Poisson process of rate λ, where λ is the long-range connection density of this overlay. Hence, in the transformed view of the forwarding node F_k, the target of the lookup process corresponds to an arbitrary point while the image of the next-hop node F_{k+1} corresponds to the next arrival in this Poisson process. The random variable v_k describes the distance between these two points in the transformed view of F_k (see Figure 4.3/b).

As a consequence of the memoryless property of Poisson processes, picking an arbitrary point in the process, the distance to the next arrival will always be exponentially distributed with parameter λ. Hence the distribution of v_k is the same for each step of the routing process (via long-range connections) and the *pdf* of v v_k is:

$$g_{prob}(v) = \lambda e^{-\lambda v}. \tag{4.25}$$

Another consequence of the memoryless property of Poisson processes is that the random variables v_k and v_{k+1} are independent, hence the series v_k are *i.i.d* random variables. Since u_k (the progress toward the target in the k^{th} routing hop) can be derived from v_k using Equation 4.20, u_k is also a series of *i.i.d* random variables (since PPLROs are regular, this could be derived also applying Theorem 4.4). The *pdf* of u can be obtain by transforming the *pdf* of v using the function $h(v)$:

$$f_{prob}(u) = g_{prob}\left(h^{-1}(u)\right)\left|\frac{dh^{-1}(u)}{du}\right| = \lambda(1-e^{-u})^{(\lambda-1)}e^{-u}. \tag{4.26}$$

Hence:

$$F_{prob}(u) = \int_0^u f_{prob}(t)dt = (1-e^{-u})^{\lambda}. \tag{4.27}$$

Finally, the expected value of the length of one routing hop in the transformed view of the target[5]:

$$E_{prob}[u] = \int_0^{\infty} f_{prob}(u)u\ du = H_{\lambda}, \tag{4.28}$$

where H_x is the harmonic number [33] (generalized for real numbers) of x. For practical λ values, the following approximation can be used[6]:

$$H_{\lambda} \approx \ln\left[(e-1)\lambda + 1\right]. \tag{4.29}$$

The above results can be transformed back from the transformed view of the target node to the real metric space of the DHT as follows.

Theorem 4.5. *Consider the routing process in a probabilistic power-law routing overlay of long-range connection density λ. Then the series of random variables $w_k = \frac{d_{k+1}}{d_k}$ describing the ratio of distances from the target after and before a routing hop via a long-range connection are* i.i.d *and the* pdf *and expected value of w_k are:*

$$f^w_{prob}(w) = \lambda(1-w)^{(\lambda-1)} \quad \textit{if} \quad 0 < w < 1 \quad \textit{and 0 otherwise} \tag{4.30}$$

and

$$E[w] = \frac{1}{1+\lambda}. \tag{4.31}$$

Proof. Since $u_k = d'_{k+1} - d'_k$ in the transformed view of the target and since transformed distances can be obtained as $d'_k = -\ln d_k$ and $d'_{k+1} = -\ln d_{k+1}$ from distances in the real metric space, u_k can be expressed as:

$$u_k = -\ln d_{k+1} - (-\ln d_k) = -\ln\frac{d_{k+1}}{d_k}.$$

Hence defining, the random variable $w_k = \frac{d_{k+1}}{d_k}$ as the ratio of distances after and before a routing hop via a long-range connection, this random variable w_k can be expressed as a

[5]Calculated using the Mathematica software from Wolfram Research Inc. (http://www.wolfram.com)

[6]This approximation provides less than $\pm 1\%$ relative error if $\lambda > 0.5$ and less than $+5\%$ relative error if $0 < \lambda < 0.5$. Note that λ is typically larger than $\frac{1}{\ln 2} \approx 1.41$ for most DHTs (see Section 4.2.1)

function $w_k = \Phi(u_k) = e^{-u_k}$ of the random variable u_k. According to Theorem 4.4, u_k is a series of *i.i.d* random variables, therefore w_k will be also a series of *i.i.d* random variables (to simplify notation, u_k and w_k are denoted simply by u and v hereafter). $\Phi(u) = e^{-u}$ is a strictly monotone decreasing function. Hence the *cdf* of w can be expressed from the *cdf* of u as:

$$F^w_{prob}(w) = 1 - F_{prob}\left(\Phi^{-1}(w)\right) = 1 - \left[1 - e^{-(-\ln w)}\right]^{\lambda} = 1 - (1-w)^{\lambda}. \tag{4.32}$$

The *pdf* of w can be obtained as the derivative of $F^w_{prob}(w)$:

$$f^w_{prob}(w) = \frac{dF^w_{prob}}{dw} = \lambda(1-w)^{(\lambda-1)}. \tag{4.33}$$

As a result of greedy routing, $d_{k+1} < d_k$ holds for each routing step, hence $0 < w_k < 1$ and the expected value of the random variable w is:

$$E[w] = \int_0^1 f^w_{prob}(w) w dw = \left[\frac{(1-w)^{\lambda}(1+\lambda w)}{1+\lambda}\right]_0^1 = \frac{1}{1+\lambda}. \tag{4.34}$$

□

Hence the distance to the target decreases in expected value by a factor of $1+\lambda$ after each routing hop via a long-range connection. Since these distance decrease ratios in subsequent routing hops are independent, the expected value of distance decrease after i routing hops via long-range connections can be expressed as:

$$E\left[\frac{d_{k+i}}{d_k}\right] = \left(\frac{1}{1+\lambda}\right)^i. \tag{4.35}$$

Deterministic power-law routing overlays

Definition 3.2 introduces deterministic power-law overlays which can be considered as a generalization of the Chord overlay. These overlays are not regular because the sequence of long-range connections in the transformed view of nodes correspond to the same renewal process for each node and lacks the random sampling property of regular power law routing overlays. However, DPLROs can be made regular substituting the constant q in Definition 3.2 with a random variable so that the first long-range peer is evenly distributed over the range $[0, \ln c]$ in the transformed view of the node. In this subsection, I analyze these "regularized" deterministic power-law routing overlays[7].

To ease comparison with PPLROs and regular power-law routing overlays in general, the generic λ long-range connection density is used during the analysis instead of the parameter c in the definition of deterministic power law routing overlays. The relationship between these two parameters can easily be determined from Figure 4.1:

$$\lambda = \frac{1}{\ln c} \quad \Leftrightarrow \quad c = e^{\frac{1}{\lambda}}. \tag{4.36}$$

[7] [**J2**] analyzes DPLROs without the above regularization. In my dissertation I investigate regularized DPLROs in order to be able to compare results with the lower bound of Theorem 4.4 (not yet included into [**J2**])

As for any regular power-law routing overlay, target nodes in the transformed view of forwarding nodes can be considered as uniformly distributed random points. Hence the *pdf* of the random variable v_k for DPLRO:

$$g_{det}(v) = \begin{cases} \lambda & \text{if} \quad 0 < v < \dfrac{1}{\lambda} \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

Transforming this distribution using Equation 4.20, the *pdf* of the random variable u_k:

$$f_{det}(u) = g_{det}\left(h^{-1}(u)\right)\left|\frac{dh^{-1}(u)}{du}\right| = \begin{cases} \lambda\dfrac{e^{-u}}{1-e^{-u}} & \text{if} \quad u > -\ln\left[1-e^{\frac{1}{\lambda}}\right] \\ 0 & \text{otherwise} \end{cases} \tag{4.38}$$

Hence the expected value of the per-hop routing progress in the transformed view of the target node can be obtained as[8]:

$$E_{det}[u] = \int_{-\ln\left[1-e^{-\frac{1}{\lambda}}\right]}^{\infty} \lambda\frac{e^{-u}}{1-e^{-u}}u du. \tag{4.39}$$

Comparison of per-hop routing progress in the transformed view

In the previous subsections, I have analyzed routing via long-range connections in regular power-law routing overlays. In Theorem 4.4, I have shown that the length of the per-hop progress in the transformed view of the target (distance between the images of subsequent forwarding nodes in this transformed view) is *i.i.d.* In other words, lookup approaches toward the target in its transformed view at constant "speed" in expected value for any regular power-law routing overlay.

Furthermore, knowing the λ long-range connection density and the c_v long-range connection density coefficient of variation parameters of the overlay, it is possible to derive a lower bound on the expected value of the length of this per-hop progress in the transformed view of the target (see Equation 4.21).

I have also derived analytically the expected value of this per hops progress for two special cases, namely the probabilistic and the "regularized" deterministic power-law routing overlays. Figure 4.4 compares these expected values as well as their generic lower bounds (derived using Inequality 4.21) as a function of the long-range connection density. The result shows that deterministic overlays provide better per-hop progress than probabilistic overlays for all values of λ. This is a consequence of different coefficients of variation (c_v) for distances between subsequent long-range connections. According to Equation 4.21, the lower bound on $E[u]$ decreases monotonically with increasing c_v values. For DPLROs, where the distance between subsequent long-range connections is constant in the transformed view of a node:

$$c_v^{det} = 0, \tag{4.40}$$

[8]The analytical form of this integral is too complicated, Figure 4.4 represents the results numerically

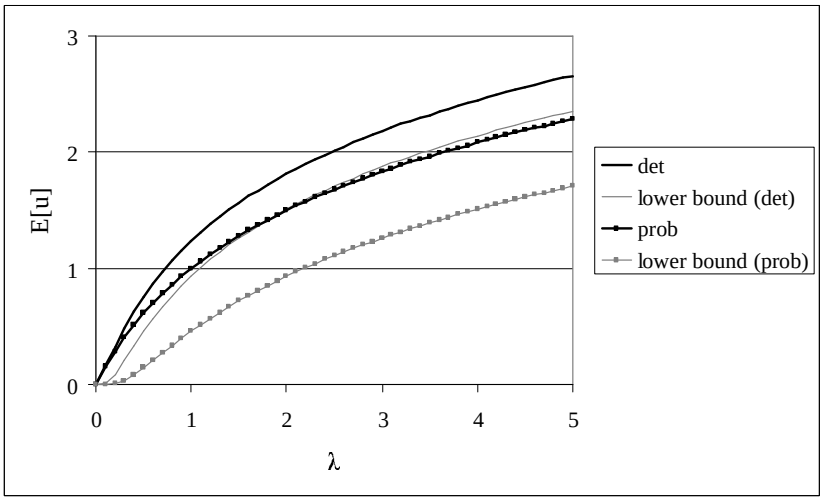

Figure 4.4: Expected value of u and its estimated lower bound as a function of λ

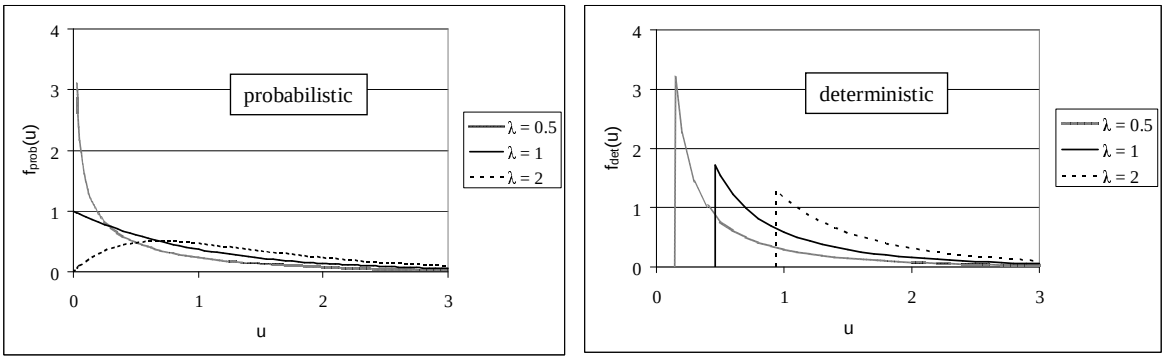

Figure 4.5: Pdf of u for different λ values

while for PPLROs, where these distances are exponentially distributed:

$$c_v^{prob} = \frac{\sigma}{\mu} = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda}} = 1. \tag{4.41}$$

Although Equation 4.21 can be used to express lower bounds on the expected value for any regular power-law routing overlay, the distribution of per-hop routing progress can differ significantly for different overlays. Figure 4.5 compares the *pdf* $f(u)$ for different long-range connection density values both for probabilistic and deterministic power-law routing overlays. As it can be expected, the deterministic routing overlay guarantees a minimum progress for each routing hop. In probabilistic routing overlays, there is no such lower bound on the length of one single hop.

Equation 4.27 reveals another interesting property of the distribution of u for probabilistic power-law routing overlays: $F_{prob}(u)$ can be obtained by raising the *cdf* of an exponential distribution to the power λ. As a result, $\lambda = 1$ is a very special long-range connections density value where the length of one routing hop in the transformed view of the target node

is exponentially distributed with parameter 1. This means that the sequence of routing hops in the transformed view of the target node corresponds to the same stochastic process as the sequence of long-range connections in the transformed view of any nodes; both can be described by a Poisson process of rate 1. For any other λ values, the length of routing hops is not exponentially distributed. Figure 4.5 shows well the difference in the shape of the probability density function for long-range connection density values $\lambda = 1$, $\lambda < 1$ and $\lambda > 1$.

The sequence of long-range connections in the transformed view of a node can also be approximated by a renewal process for many "non-regular" DHTs (see Section 4.2.1). However, irregularities in the distribution of distances between successive long-range connections induces distortions to the "constant speed" progress in the transformed view of the target and make mathematical analysis difficult.

For example, in Pastry, long-range connection density have a slight periodic variation. The length of this period is $b \ \ln 2$ in the transformed view (where b is the bit length of numbers in the routing table). This slight periodic fluctuation is visible on the graph showing the expected number of routing hops as a function of network size (see Figure 4. in paper [9]).

4.3.2 Upper bound on the expected number of routing hops

In the previous subsections, I have analyzed routing via long-range connections in the transformed view of target nodes. I have shown that the per-hop routing progress u_k is $i.i.d$ for regular power-law routing overlays, hence the images of forwarding nodes in the transformed view of the target can be described as a renewal process. Furthermore, I proposed a lower bound on the expected value of this per-hop progress as a function of the λ and c_v overlay parameters (see Theorem 4.4).

Although these results cannot be used directly to characterize overlay performance, the proposed renewal process model allows analytical derivation of an important overlay performance metric: an upper bound on the expected number of routing hops as a function of network size and the overlay parameters (λ, c_v and N_S). The computation of this overlay performance metric is based on the upper bound of Lorden for renewal processes (see [29], page 110):

$$U_L(t) \leq \frac{t}{\mu} + \frac{\mu_2}{\mu^2} - 1, \tag{4.42}$$

where $U_L(t)$ is the renewal function (the expected value of renewals until time t), μ is the mean of renewal periods and μ_2 is the second moment of renewal periods. Applying this bound to the renewal process corresponding to the sequence of forwarding nodes in the transformed view of the target node, the variables in Inequality 4.42 correspond to the followings:

- $U_L(t)$ corresponds to the expected value of the number of routing hops for long-range only routing

- t corresponds to the length of the long-range routing path in the transformed view of the target node (from the image of the initiator node to the image of the first node having a direct short-range connection to the target node)
- μ corresponds to $E[u]$, for which, Theorem 4.4 gives a lower bound as a function of the overlay parameters λ and c_v
- μ_2 corresponds to $E[u^2] = \int_0^\infty u^2 f(u)du$

Lemma 4.6. *Considering the routing process via long-range connections in a regular power law routing overlay, the second moment of the length of the per-hop progress u in the transformed view of the target node is upper bounded by*

$$E[u^2] \leq 2.41\lambda, \tag{4.43}$$

where λ is the long-range connection density of the overlay

Proof. The random variable u can be obtained from the random variable v using Equation 4.20 while the variable v itself corresponds to the residual life in the renewal process corresponding to the sequence of long-range connections in the transformed view of a node (see proof of Theorem 4.4). From renewal theory, the *pdf* of the residual life v can be expressed as follows:

$$g(v) = \frac{1 - F(v)}{E(x)} = \lambda(1 - F(v)), \tag{4.44}$$

where $F(x)$ is the *cdf* of renewal intervals (corresponding to the distances between subsequent long-range connections of a node in its transformed view), $E[x]$ is the expected length of these intervals and I have used $\frac{1}{E[x]} = \lambda$ from Theorem 4.1.

Since a *cdf* is always a non-decreasing function and $0 \leq F(x) \leq 1$, the *pdf* $g(v)$ is a non-increasing function upper bounded by $g(v) \leq g(0) = \lambda$.

This upper bound can be used to derive an upper bound on $f(u)$. Using Equation 4.20 to transform v to u:

$$f(u) = g(-\ln(1 - e^{-u}))\frac{e^{-u}}{1 - e^{-u}} \quad \leq \quad \lambda\frac{e^{-u}}{1 - e^{-u}}. \tag{4.45}$$

Hence

$$E[u^2] = \int_0^\infty u^2 f(u)du \quad \leq \quad \int_0^\infty \lambda\frac{e^{-u}}{1 - e^{-u}}u^2 du \quad \leq \quad 2.41\lambda. \tag{4.46}$$

□

Lemma 4.7. *Assuming uniform distribution of node identifiers in the metric space of the DHT and uniform selection of target nodes for the routing process, the expected length of the routing path in the transformed view of the target is:*

$$E[t] = \ln n + \gamma - 1 - H_{N_S - 1} + \epsilon, \tag{4.47}$$

where N_S is the number of short-range connections per node, n is the number of nodes in the overlay, H_k is the k^{th} harmonic number, γ is the Euler-Mascheroni constant [9] *and ϵ is a small positive error term*

$$\epsilon \in o\left(\frac{n^{N_S}}{e^n}\right)$$

negligible except for very small network sizes.

Proof. When using the long-range only forwarding model, a routing process (as seen in the transformed view of the target) starts from the image of the initiator node and ends at the image of the first node having a direct short-range connection to this target node. The expected length $E[t]$ of this routing path can be obtained as the difference between the expected location of these start and end points.

Assuming that nodes of the overlay are uniformly distributed in the range $[0, 1)$ of the DHT metric space and target nodes are selected also uniformly by initiator nodes, the distance between initiator and target nodes will be uniformly distributed over the interval $(0, 1)$. Applying logarithmic transformation to this distribution according to Equation 4.6 results into exponentially distributed distances in the transformed view of the target with the *pdf*: $f'(x) = e^{-x}$. Hence the expected value of the distance between the target and initiator nodes in the transformed view of the target will be

$$E[L_{start}] = \int_0^\infty e^{-x} x dx = 1. \tag{4.48}$$

Assuming again that nodes of the overlay are uniformly distributed in the metric space of the DHT, the distance between a node and its farthest short-range peer will be Erlang distributed with rate n and shape parameter N_S, where n is the number of nodes in the overlay and N_S is the number of short-range connections per node. Hence the *pdf* of this distance distribution will be:

$$f_{Erl}(y) = \frac{n^{N_S} y^{N_S-1} e^{-ny}}{(N_S-1)!}. \tag{4.49}$$

Transforming the *pdf* of this Erlang distribution according to Equation 4.6, the *pdf* in the transformed view will be:

$$f'_{Erl}(x) = f_{Erl}\left(f_t^{-1}(x)\right) \frac{df_{Erl}^{-1}}{dx} = \frac{e^{-\left[ne^{-x}+x(N_S-1)\right]} n^{N_S}}{(N_S-1)!}. \tag{4.50}$$

Hence the expected value of the distance between a node and its farthest short-range peer in its transformed view[10]:

$$E[L_{end}] = \int_0^\infty f'_{Erl}(x) x dx = \ln n + \gamma - H_{N_S-1} + \epsilon, \tag{4.51}$$

[9] *The Euler-Mascheroni constant is defined as* $\gamma = \lim_{k\to\infty}(H_k - \ln k) \approx 0.5772$

[10] Calculated using the Mathematica software from Wolfram Research Inc. (http://www.wolfram.com)

where H_k is the k^{th} harmonic number, γ is the Euler-Mascheroni constant and ϵ is a small positive error term upper bounded by

$$\epsilon < e^{-n}\left[1 + \frac{(n+N_S)^{N_S-2}}{(N_S-1)!}\right]. \tag{4.52}$$

Hence ϵ can be expressed using the following asymptotic bound:

$$\epsilon \in o\left(\frac{n^{N_S}}{e^n}\right). \tag{4.53}$$

Typically, N_S is a small number, hence – except for very small network sizes – ϵ is negligibly small. □

Theorem 4.8. *The expected number of routing hops U in a regular power-law routing overlay is upper bounded by:*

$$U(n,\lambda,c_v,N_S) \quad \leq \quad \frac{\ln n - H_{N_S-1} - 0.42}{-\ln\left[1-e^{-\frac{1+c_v^2}{2\lambda}}\right]} + \frac{2.41\lambda}{\ln^2\left[1-e^{-\frac{1+c_v^2}{2\lambda}}\right]} + \epsilon, \tag{4.54}$$

where n is the number of nodes in the overlay, λ is the long-range connection density, c_v is the long-range connection density coefficient of variation, N_S is the number of short-range connections per node and ϵ is a small positive error term

$$\epsilon \in o\left(\frac{n^{N_S}}{e^n}\right)$$

negligible except for very small network sizes.

Proof. Substituting the results of Lemma 4.6, Lemma 4.7 and Theorem 4.4 into the Lorden bound (Inequality 4.42), an upper bound can be obtained on the expected number of routing hops for long-range only forwarding:

$$U_L(t) \quad \leq \quad \frac{\ln n - H_{N_S-1} - 0.42 + \epsilon}{-\ln\left[1-e^{-\frac{1+c_v^2}{2\lambda}}\right]} + \frac{2.41\lambda}{\ln^2\left[1-e^{-\frac{1+c_v^2}{2\lambda}}\right]} - 1. \tag{4.55}$$

Then, the upper bound on the number of routing hops for real routing (via both short-range and long-range connections) can be obtained using $U(t) < U_L(t)+1$ from Inequality 4.16. □

When the first and second moments of the per-hop progress u in the transformed view of the target are known, the upper bound of Theorem 4.8 can be further tightened:

Theorem 4.9. *The expected number of routing hops U in a probabilistic power-law routing overlay is upper bounded by:*

$$U(n,\lambda,N_S) \quad \leq \quad \frac{\ln n - H_{N_S-1} - 0.42}{H_\lambda} + \frac{1.645 - \psi'(1+\lambda)}{H_\lambda^2} + 1 + \epsilon, \tag{4.56}$$

where n is the number of nodes in the overlay, λ is the long-range connection density, and N_S is the number of short-range connections per node, $\psi'(x)$ is the first derivative of the digamma function and ϵ is a small positive error term

$$\epsilon \in o\left(\frac{n^{N_S}}{e^n}\right)$$

negligible except for very small network sizes.

Proof. Using Equation 4.28, the first moment of u is $\mu = H_\lambda$, where H_x is the harmonic number generalized for real numbers. The second moment of u can be derived from the *pdf* of u given by Equation 4.26:

$$\mu_2 = \int_0^\infty f_{prob}(u)u^2 du = \frac{\pi^2}{6} + H_\lambda^2 - \psi'(1+\lambda). \tag{4.57}$$

Substituting μ, μ_2 and the result of Lemma 4.7 into the Lorden bound (Inequality 4.42) an upper bound can be obtained on the expected number of routing hops for long-range only forwarding:

$$U_L(t) \quad \leq \quad \frac{\ln n - H_{N_S-1} - 0.42 + \epsilon}{H_\lambda} + \frac{\frac{\pi^2}{6} + H_\lambda^2 - \psi'(1+\lambda)}{H_\lambda^2} - 1. \tag{4.58}$$

Performing simplifications, the upper bound on the number of routing hops for real routing (using both short-range and long-range connection) can be obtained using $U(t) < U_L(t) + 1$ from Inequality 4.16. □

4.4 Summary

Although most DHT overlays are structurally similar to the "small-world" navigation model of Kleinberg [18] – architectural and algorithmic details of different DHT variants differ significantly. Furthermore, lookup performance depends on a sets of different and often incompatible parameters which makes analytical comparison rather difficult. The objective of this chapter was to propose a general analytical model that can be used to investigate and compare static routing performance performance of most DHT implementations as a function of their overlay structure.

To capture the above mentioned common foundations of overlay structure, I have introduced the concept of logarithmically transformed view, where distances between a reference node and other nodes are represented after a logarithmic transformation. I have shown that long-range peers of a node form a linear sequence in this transformed view for most DHTs. Furthermore, I have identified an important subclass of DHT overlays – regular power-law routing overlays – where this sequence can be described as a random sample from an infinite renewal process. Based on this stochastic model, I have introduced the λ long-range connection density and c_v long-range connection density coefficient of variation parameters.

For $O(\log n)$ node state, these parameters characterize long-range connection distribution independent of network size.

Using the renewal process model of long-connections, I have analyzed stochastically the progress of lookup process via long-range connections. I have shown that the sequence of intermediate forwarding nodes in the transformed view of the target node can be also described as a renewal process. Additionally, I have derived (i) the distribution of this per-hop routing progress for the spacial cases of probabilistic and "'regularized" deterministic power-law routing overlays (ii) a generic upper bound on the per-hop routing progress in the transformed view of the target as a function of the λ and c_v long-range connection distribution parameters.

Finally, using the renewal process model of the routing process, I have derived closed form upper bounds on the expected number of routing hops as a function of network size and the overlay parameters λ, c_v and N_S.

The above model and results can be applied directly to any DHT using probabilistic power-law routing overlays (e.g., Symphony [14], Accordion [15], etc.). Additionally, overlay structure and static routing performance of any DHT using a one-dimensional metric space and being structurally similar to the "small-world" navigation model of Kleinberg can be approximated applying this model (e.g., Chord [8] and its variants, Pastry [9], Bamboo [17], Kademlia [12], etc.)

Chapter 5

An Asymptotically Optimal Overlay Maintenance in DHTs

P2P systems are inherently dynamic; peers join, leave and rejoin the network frequently. This process of permanent joining and leaving of peers is called *churn* in peer-to-peer terminology. Unstructured peer-to-peer systems do not need to maintain a specific overlay structure hence they can easily cope with this dynamism. In contrast, structured peer-to-peer systems need to update a distributed indexing structure, therefore efficient maintenance mechanisms are required in highly dynamic network environments.

For DHTs, the distributed maintenance process has two major components: overlay maintenance and storage maintenance. Overlay maintenance is responsible for maintaining overlay structure while storage maintenance handles changes in the mapping of data items (and their possible replicas) onto nodes. In this chapter, I focus on optimization of overlay maintenance in distributed hash tables and propose an architecture which minimizes asymptotically overlay maintenance overhead.

Routing overlays in most DHTs consist of two different types of connections: a predefined number of short-range connections to the closest nodes in the DHT metric space and $O(\log n)$ long-range connections to some distant nodes[1]. The role of short-range and long-range connections is complementary: while short-range connections guarantee the success of routing, long-range connections expedite the routing process to $O(\log n)$ hops. These different roles imply different requirements on the maintenance process. A query in the routing overlay might fail because of any single missing short-range connection. Therefore, to guarantee successful routing, short-range connection maintenance has to be strict, self-stabilizing and proactive. In contrast, to guarantee $O(\log n)$ routing performance, long-range connections only have to meet probabilistic requirements. Degradation of routing performance as a result of some missing or misplaced long-range connections is marginal.

The key concept of the proposed solution is to fully exploit these relaxed requirements

[1]There are a few exceptions: e.g., Kademlia [12] nodes only have long-range connections while CAN [10] nodes only have short-range connections.

applying only stochastic maintenance to long-range connections. The proposed stochastic maintenance algorithm ignores individual long-range connections and only considers the (estimated) distance distribution of the set of all long-range connections of a DHT node. This distance distribution is kept within predefined bounds using two architectural components: (i) probabilistic power-law routing overlays with bidirectional connections preserving the power-law nature of distance distribution without any explicit maintenance and (ii) a stochastic long-range connection maintenance algorithm which keeps the (estimated) λ long-range connection density parameter of the distance distribution within a predefined range.

In order to further reduce the amount of control traffic in a network under churn, I also propose a range-based connection establishment algorithm allowing to create a new long-range connection with $O(1)$ communication overhead.

Finally, I show that combining probabilistic power-law routing overlays with bidirectional routing, the proposed long-range connection maintenance algorithm and the proposed range-based long-range connection establishment algorithm, maintenance overhead in the resulting system is asymptotically optimal.

The rest of this chapter is structured as follows: Section 5.1 provides a general overview of various proposals to handle churn in structured peer-to-peer networks. Then, Section 5.2 describes all components of the architecture providing together asymptotically optimal overlay maintenance overhead. Finally, Section 5.3 investigates both analytically and by simulations the performance of the proposed system and compares it to existing churn-tolerant DHT proposals.

5.1 Related work

DHT performance under churn can be characterized as a tradeoff between conflicting performance metrics: lookup performance (latencies and lookup failure ratio) vs. maintenance overhead. To cope with churn and achieve low lookup latencies while not wasting too much maintenance bandwidth, DHT design has to consider the following key architectural issues: static resilience of the overlay, efficient maintenance mechanisms and churn-tolerant lookup strategies.

5.1.1 Static resilience

Static resilience refers to fault tolerance of the network without maintenance mechanisms or before completion of maintenance. In [19], the authors propose a comparative framework called reachable component method (RCM) to characterize and compare resilience of DHT overlays against random failures. RCM uses a routability metric defined as the ratio of survived routing paths in the system as a function of node failure probability (p) and the number of nodes in the network (n). RCM allows to filter out overlay designs that fail to scale in dynamic network environments: if routability converges to zero as $n \to \infty$ for

any non-zero failure probability, then the DHT is unscalable in dynamic networks. Using a Markov chain model and assuming a fully populated identifiers space, the authors calculate this routability for 5 different DHT. Their analysis shows that Chord [8], Kademlia [12] and CAN [10] overlays are scalable while the tree topology proposed by Plaxton [34] and Symphony [14] with its constant node state are unscalable.

An interesting aspect of resilience against network partitioning is presented in [28]: Loguinov et al. propose ODRI (Optimal Diameter Routing Infrastructure) – based on de Bruijn graphs and focusing on edge bisection width – and demonstrate that de Bruijn graphs are significantly more difficult to disconnect than other structures. However, preventing partitioning is not sufficient to provide fault tolerant routing; provisioning of alternative routes is also required and the paper does not tackle this issue.

The authors in [25] examines static resilience of DHT overlays from a routing geometry point of view. They argue that flexibility in peer selection and route selection is a key issue that overlay geometries should allow in order to provide resilience. Comparing the tree, ring, hypercube, butterfly and XOR topologies, the ring geometry is selected as the most flexible one from both points of view. The paper also draws interesting conclusions on constant node degree overlays: the authors conjecture that the inflexibility of the constant node degree butterfly topology of Viceroy is not a "flaw" in the particular Viceroy design but a fundamental limitation of constant state overlays. This is in accordance with the results of [19] regarding the scalability problems of another constant node degree DHT: Symphony [14].

Chord [8] was one of the very first DHT routing algorithms inspiring a dozen of other P2P routing protocols based on the ring geometry. Although neighbor selection in the original Chord algorithm is deterministic, several variations have been published allowing flexible (probabilistic) neighbor selection. Most of these probabilistic variations are based on the "small-world" network model proposed by Kleinberg in [18].

In [23], Aspnes et al. present a P2P routing mechanism based on the one dimensional version of the Kleinberg model. They provide analytical upper and lower bounds on delivery time in static networks and examine the effect of node and connection failures on expected delivery time. Their results are based on the assumption that connections to the nearest neighbors are always present and long-range connections follow an inverse power-law distance distribution. However, their model does not include any maintenance algorithm to restore validity of these assumptions after failures in the network. This explains the mismatch between analytical findings and simulation results presented in Section 4.3.2 of their paper. To reduce the number of failed searches caused by missing short-range connections, a backtracking algorithm is incorporated into the greedy routing mechanism.

Symphony [14] is another P2P routing protocol based on the Kleinberg model proposed by Manku et al. The authors identified that short-range connections are critical for successful lookups and propose additional backup short-range connections to increase resilience against failures. However, similarly to [23] this only provides static resilience since the authors do

not implement any maintenance algorithm to restore short-range connections after failures.

eQuus presented in [16] is a DHT explicitly designed for highly dynamic environments. The key idea of this paper is to incorporate network locality into overlay structure in order to achieve high resilience and low maintenance overhead. In eQuus, an ID is assigned to a group of nodes that are physically closest to each other – in contrast to most other systems where each node holds an own ID. Nodes having the same ID are all connected to each other and therefore these groups are called cliques. Cliques form an overlay network very similar to that of Pastry [9]. Connections between cliques are implemented such that each node of the clique maintains a list with the addresses of k randomly selected nodes from the other clique (k is a constant independent of network size). As a consequence, the overlay network of cliques is extremely robust and resilient against failures.

5.1.2 Maintenance

There are two major challenges in overlay maintenance: (i) self-stabilization, which means ensuring the recovery of overlay structure from arbitrary combinations of failures and (ii) the tradeoff between lookup performance and maintenance overhead.

Many maintenance algorithms assume that the system starts from an ideal state and returns to that ideal state after each failure [35]. Accumulation of failures in the network may eventually result in states that the protocol is unable to recover from. Liben-Nowell et al. take an important step towards adaptation to dynamic network environments by considering a P2P network as a continuously evolving system as compared to the above quasi-static approach. In [35], they introduce a new metric to describe performance of maintenance algorithms in P2P systems: the rate at which each node must expand resources in the maintenance protocol in terms of the half-life of the system. Network half life is defined as the period during which half of the nodes in the network are replaced by new arrivals. Using this metric, they give a lower bound on maintenance protocol bandwidth needed to keep the network connected as nodes join and leave. They also present a modification of maintenance algorithm in Chord approaching this lower bound asymptotically within a logarithmical factor.

In [36], Jelasity et al. propose T-Man, a gossip based protocol to create a large class of different overlay topologies. Starting from a random network, T-Man gradually converges to the desired topology by organizing peers according to a ranking function. T-Man requires a synchronization point when starting the protocol and the initial topology has to be close to a random graph. An interesting application of T-Man can be found in [37], where the gossip-based protocol is customized to create a Chord topology out of any random graph topology.

The Ring Network protocol [38] goes one step further for the special case of ring topology. RN is self-stabilizing because it converges to a Chord-like ring topology starting from any weakly connected bootstrap system and the protocol maintains this structure in face of peers joining, leaving or failing. The protocol is completely asynchronous; it does not require

any synchronization between peers. However, RN does not exploit the inherent neighbor selection flexibility of the ring geometry [25]: Not only short-range, long-range connections are also deterministic resulting in several times higher maintenance traffic.

Besides self-stabilization, the other important challenge in DHT maintenance is the inherent tradeoff between maintenance overhead and lookup performance. In [26], Li et al. introduce PVC, a performance vs. cost framework to describe and compare the performance of DHTs under churn. PVC views a protocol as consuming a certain amount of network bandwidth in order to achieve a certain lookup latency, and helps to reveal the efficiency with which protocols use additional network resources to improve latency. PVC is a useful metric not only to compare different DHTs but also to fine-tune protocol parameters to achieve the best latency for a given maintenance traffic or vice versa. The authors applied the PVC metric for simulations, however it could also be used as a tool for analytical comparison.

Most DHTs aim at providing a predefined lookup performance (e.g., $O(\log n)$ lookup latencies) which degrades only slowly and gracefully with increasing churn rates. Hence increasing churn rate implies increasing maintenance overhead. Accordion [15] takes a different approach. Accordion nodes are assigned predefined (per node) bandwidth budgets and each node uses its available bandwidth budget (not consumed by lookups) to further increase node state, creating additional connections. This allows achieving $O(1)$ lookup latencies in static networks while still providing $O(\log n)$ latencies under churn.

The approach taken by eQuus [16] to tackle the lookup performance vs. maintenance overhead tradeoff is also completely different. Instead of aiming at reducing maintenance traffic, eQuus ensures that most of this maintenance traffic flows within cliques, whose members are physically closest to each other. E.g., a joining node can create all of its initial connections based on information queried from its clique peers. In eQuus, significant inter-clique maintenance traffic is required only when cliques need to be merged or split as a result of a large number of joining or departing nodes. However, these merge and split events occur seldom: in expected value, $\Omega(n)$ join/leave events are required before either a merge or a split operation has to be performed. Nevertheless, ensuring that clique members are in fact the physically closest closest nodes is not for free: joining nodes need to send out $O(\log^2 n)$ ping messages to find the closest clique to join.

Maintenance timing is also an important issue. In some protocols, nodes periodically ping their peers to detect node failures and/or periodically run a maintenance process to restore consistency of the network. For instance in Chord [8], nodes periodically invoke *stabilize()* and *fix_fingers()* procedures to maintain consistency of successor and finger pointers in the network. Bamboo [17] also advocates for proactive periodic maintenance. Other protocols like Kademlia [12] take a reactive approach and detect failures only when a peer is unreachable during regular communication. In these cases a maintenance mechanism is triggered by these failures instead of a periodic execution. The reactive approach has the advantage of saving unnecessary maintenance traffic but in case of very low lookup traffic and high churn it may lead to the maintenance avalanche effect (when attempts to replace

missing connections recursively result in detecting additional connection failures).

5.1.3 Churn-tolerant lookup strategies

Failure or ungraceful departure of peer nodes can only be detected via timeouts. As a consequence – although maintenance algorithms can significantly reduce the ratio of failed connections in the overlay – these failed connection cannot be completely eliminated using finite maintenance bandwidth. Detecting failures when routing lookup requests, timeouts can significantly increase lookup latencies.

Kademlia [12] addresses this problem via parallelization of lookups. Kademlia defines a system-wide concurrency parameter α and the initiator node keeps α outstanding requests in parallel at the same time. This ensures that the lookup process halts for a timeout period only if all the α outstanding requests are sent to failed nodes. Although communication cost increases with α (nearly linearly), the probability of timeout penalties decreases exponentially with α. Hence lookup latencies can be significantly improved incurring only moderate communication overhead.

Bamboo [17] takes a different approach: instead of parallelizing lookups, Bamboo nodes collect round-trip time (RTT) statistics from all of their peer nodes, which allows fine-tuning of timeout settings. These RTT statistics are collected based on keep-alive messages sent out periodically by overlay maintenance to all peer nodes.

The iterative or recursive nature of lookup algorithms also affects significantly lookup latencies under churn. In recursive routing (RR), intermediate nodes on the routing path forward the query directly and the originator node has no control over the routing process. In iterative routing (IR), intermediate nodes only return the address of candidate next hop node(s) and these nodes will be queried directly by the initiator node itself. Recursive routing is faster in static networks, since an intermediate node can directly forward the request without going back first to the initiator node. However, recursive routing aggravates the effect of timeouts: if a single failed connection is encountered in the routing path, the initiator node has to restart the whole lookup process from the beginning after timeout.

The set of available lookup optimization mechanisms also depends on the iterative or recursive nature of lookup. E.g., parallelization of lookups requires the iterative mechanism in order to have control over the degree of parallelism. In contrast, fine-tuning of timeout settings requires recursive lookup, since this restricts communication to only peer nodes, which makes collection of RTT statistics feasible.

The authors in [20] provide a detailed analysis of lookup performance under churn for both the iterative and the recursive routing strategies. Additionally, they propose a new strategy called RR+ACK which combines advantages of both lookup strategies. RR+ACK is derived from recursive routing, however, each intermediate node also sends an ACK to the initiator node containing the address of the next-hop node. In case of failure, the initiator node can re-initiate another lookup using the next-hop address from the latest ACK.

5.2 Architecture for asymptotically optimal maintenance

The key concept of the proposed architecture is to fully exploit the relaxed requirements on long-range connections in order to minimize overlay maintenance overhead. In probabilistic power-law routing overlays, the only requirement on long-range connections is the power-law distance distribution; hence they are a natural candidate for this architecture. In addition, I show that the bidirectional version of probabilistic power-law routing overlays exhibits a self-healing property: in steady states (when node arrival and departure rates are equal) missing long-range connections caused by node departures/failures are automatically replaced (statistically) by new connections of joining nodes – without any explicit maintenance.

The only disadvantage of bidirectional overlays – as compared to their unidirectional equivalent – is that they double node state while the improvement in routing performance is less than twofold. However, the $O(\log n)$ node state of distributed hash tables is typically not a memory bottleneck. Furthermore, this doubling of node state does not increase communication cost of establishing new connections since establishing a new bidirectional connection means a new connection for the peer node too. Hence, the doubling of new connections per connection establishment compensates the doubling in the number of connections.

Exploiting the self-healing property of probabilistic power-law routing overlays with bidirectional connections, the proposed long-range connection maintenance algorithm basically "lets the system maintain itself". Explicit maintenance action is taken only to enforce balanced node degree and to create missing long-range connections when node arrival rate is smaller than departure/failure rate.

The proposed maintenance algorithm for long-range connections is reactive and stochastic. Maintenance does not periodically check the status of long-range peers; failed long-range connections are detected by timeouts during regular communication. Additionally, maintenance is not performed at a per-connection basis. Maintenances algorithm ignores individual long-range connections and only considers the distance distribution of the set of all long-range connections of a DHT node.

5.2.1 Bidirectional PPLRO model

To describe bidirectional routing, I'have used the bidirectional overlay model presented in Section 3 and illustrated by Figure 3.2. In this model, a node separates the DHT metric space $(\mathcal{I}, d)$ into two symmetrical partitions. Furthermore, connections of a node in both of these partitions can be derived separately from connections of a node in the corresponding unidirectional overlay. Hence results of Section 4.2.3 on modeling long-range connections distribution in PPLROs with unidirectional connections can be applied separately to both of these partitions.

According to Theorem 4.3, considering a probabilistic power-law routing overlay (with

unidirectional connections), long-range connections of a node in its logarithmically transformed view correspond to a random realization of a truncated Poisson process of rate λ, where λ is the long-range connection density of the overlay. Hence for the bidirectional equivalent of this overlay, long-range connections of a node in its transformed view correspond to two independent realizations of the same Poisson process of rate λ (one for both metric space partition).

5.2.2 Self-healing property of long-range connections

I show that long-range connections in probabilistic power-law routing overlays with bidirectional connections have a self-healing property under churn. Assuming a steady state, where node arrival and departure rates are equal, a PPLRO with λ long-range connection density and bidirectional connections remains a PPLRO of the same λ long-range connection density without any explicit maintenance – solely as a side-effect of new connections established by joining nodes. Since a PPLRO has only probabilistic requirements on long-range connections, this is equivalent to the statement that distance distribution of long-range connections remains unchanged under churn assuming equal node arrival and departure rates.

Theorem 5.1. *Assuming a steady state where node arrival and departure rates are equal, the distance distribution of long-range connections in a probabilistic power-law routing overlay with bidirectional connections remains unchanged. This statement holds under the following (reasonable) assumptions: node arrivals and departures are independent of each other, independent of existing connections in the overlay and finally, node identifiers of both joining and departing nodes are uniformly distributed in the metric space $(\mathcal{I}, d)$ of the DHT.*

Proof. Consider a steady state churn scenario where node arrival and departure rates are equal ($r_d = r_a = r$ and all of these rates are normalized by the number of nodes in the overlay).

During an infinitesimally small time period dt, the probability that an arbitrary node leaves the overlay is $p_{leave} = rdt$. Consider now the logarithmically transformed view of an arbitrary node. According to Theorem 4.3, the series of long-range connections in the transformed view of this node correspond to a random realization of a truncated Poisson process of rate λ, where λ is the long-range connection density of this overlay[2]. As a consequence – applying the assumptions of the theorem on uniform distribution and independence – any of the existing long-range connections of this node will be deleted with probability $p_{leave} = rdt$ during the time period dt. Hence node departures can be considered as a random removal with probability p_{leave} from this Poisson process. Using the random selection property of Poisson processes, long-range connections of this node after removal of connections to departed nodes can be considered as a random realization of a Poisson process of rate $\lambda(1 - p_{leave})$.

[2]Separately for the sequences of long-range connections at both the left and right sides.

Examining now the arrival of new nodes, the ratio of new nodes joining the overlay during the same infinitesimally small time period dt will be $r_{join} = rdt$. These new nodes create their long-range connections during the join process according to the λ long-range connection density value of the overlay. Since all connections are bidirectional, each of the original nodes at the other endpoint of these new connections will also have a new long-range connection created to the corresponding new nodes. Furthermore – as a result of the symmetry properties of the DHT metric space $(\mathcal{I}, d)$ – the probability of creating a new bidirectional long-range connection between a given original node and a given new node is the same even if the connection establishment would have been initiated by the original node.

Therefore, long-range connections of an original node can be considered as if new nodes were already part of the overlay and all connection establishments would have been originated by this original node. This would result into the same long-range connection distance distribution as the real join scenario. Using again the assumptions of the theorem on independence and uniform distribution, the probability that a long-range connection of this original node points to a new node will be $r_{join} = rdt$ – independent of the distance of the two nodes. Using the random selection property of Poisson processes, the series of new long-range connections in the transformed view of an original node can thus be considered as a random realization of a Poisson process of rate λr_{join}.

Finally, using the superposition property of Poisson processes, long-range connections of any original node in its transformed view after departure and arrival of new nodes can be described as a random realization of a Poisson process of rate $\lambda(1 - p_{leave}) + \lambda r_{join} = \lambda(1 - rdt) + \lambda rdt = \lambda$. Hence the distance distribution of long-range connection does not change under churn if node arrival and departure rates are equal. □

5.2.3 Stochastic maintenance of long-range connections

Although the self-healing property of probabilistic power-law routing overlays with bidirectional connections provides solid foundations to minimize maintenance overhead, it does not entirely replace maintenance mechanisms. In real P2P networks, the number of users – as well as node arrival and departure rates – fluctuate over multiple time scales (e.g., follow a daily profile). As a consequence, in some periods of time, node departure rate can be significantly higher than node arrival rate and vice versa. When the departure rate of nodes is higher than the arrival rate of new nodes, missing long-range connections need to be created by the maintenance process in a bidirectional PPLRO. The contrary is required for growing networks where arrival rate of new nodes is higher than departure rate: without removing some connections, long-range connection density of nodes would grow continuously resulting in extreme node degrees for older nodes.

The key idea of the proposed maintenance mechanism is to exploit the self-healing property of bidirectional PPLROs so that active maintenance only has to cope with the "difference" between node arrival and departure rates. This is achieved using stochastic methods.

Stochastic maintenance means that maintenance process does not explicitly maintain individual connections, it only ensures a specific distance distribution of long-range connections. In Section 4.2.3, I have shown that distance distribution in a PPLRO can be described by one single parameter, the λ long-range connection density. Additionally, in Section 4.3.1, I have proved that this single parameter characterizes unequivocally routing performance of a PPLRO for routing hops via long-range connections.

Therefore, the proposed stochastic long-range connection maintenance algorithm keeps the the $\widehat{\lambda}$ estimated long-range connection density of each node within a predefined range $[\lambda_{min}, \lambda_{max}]$, where $\lambda_{min} = \lambda_{opt} - \Delta\lambda$ and $\lambda_{max} = \lambda_{opt} + \Delta\lambda$ (both λ_{opt} and $\Delta\lambda$ are tunable system parameters). Applying Theorem 4.3, $\widehat{\lambda}$ is calculated for each node by maximum likelihood (ML) estimation of the rate of the Poisson process corresponding to the sequence of long-range connections in the transformed view of the given node. Given an interval of length Δx of a Poisson process with N arrivals, the ML estimation of the rate of this process can be obtained as $\widehat{\lambda} = \frac{N}{\Delta x}$. In the transformed view of a node, the number of arrivals corresponds to the number of long-range connections (N_L) while the length of the interval is $-\ln d_S$, where d_S is the distance to the farthest short-range peer of the node. Given the bidirectional nature of the overlay, the proposed maintenance algorithm uses separate $\widehat{\lambda_r}$ and $\widehat{\lambda_l}$ estimations for long-range connections in the right side and left side partitions of the DHT metric space:

$$\widehat{\lambda_r} = -\frac{N_L^r}{\ln d_S^r} \quad \text{and} \quad \widehat{\lambda_l} = -\frac{N_L^l}{\ln d_S^l}, \tag{5.1}$$

where d_S^l and d_S^r denotes the distance from the farthest short-range connection at the left and right sides respectively while N_L^l and N_L^r denotes the number of long-range connections at the left and right sides respectively.

The proposed maintenance algorithm is reactive and is triggered either when another node creates a new (bidirectional) connection to this node or when a connection failure is detected (via timeouts). Upon these events, a node executes Algorithm 5.1 separately for long-range connections at both the left and right sides. The parameter *direction* indicates whether maintenance refers to the left or right partition of the DHT metric space (hereafter, the l and r indices are omitted to simplify notation). First – using Equation 5.1 – maintenance algorithm recalculates the values of the estimated connection densities. If $\widehat{\lambda}$ is within the range $[\lambda_{min}, \lambda_{max}]$, then no maintenance actions are taken. Otherwise, if λ falls below λ_{min} at either the left or right side, then new connections are created at this side until $\lambda > \lambda_{opt}$. Similarly, if the estimated connection density exceeds λ_{max} then randomly selected connections are deleted until $\lambda < \lambda_{opt}$.

According to the definition of probabilistic power-law routing overlays (Definition 3.1), new long-range connections have to be created using a probability distribution inversely proportional to the distance. Hence the distance at which the procedure CREATE-NEW-LR-CONNECTION() creates a new long-range connection is generated using the *pdf* $f(x) = \frac{c}{x}$. The constant c depends on the range $(d_S, 1]$ of long-range connections (where d_S is

Algorithm 5.1 Stochastic maintenance

```
STOCHASTIC-MAINTENANCE(direction)
1  λ̂ ← ESTIMATE-LR-CONNECTION-DENSITY(direction)
2  if λ̂ < λ_min
3     then while ESTIMATE-LR-CONNECTION-DENSITY(direction) < λ_opt
4               do CREATE-LR-CONNECTION(direction)
5  elseif λ̂ > λ_max
6     then while ESTIMATE-LR-CONNECTION-DENSITY(direction) > λ_opt
7               do DELETE-RANDOM-LR-CONNECTION(direction)
```

the distance to the farthest short-range peer of this node) and it can be derived from the distribution function criteria:

$$\int_{d_S}^{1} \frac{c}{x} dx = 1 \quad \Rightarrow \quad c = -\frac{1}{\ln d_S}. \tag{5.2}$$

Hence, the *pdf* of long-range connection distance distribution is:

$$f(x) = -\frac{1}{\ln d_S} \frac{1}{x} \tag{5.3}$$

and its *cdf* is

$$F(x) = \int_{d_S}^{x} f(u) du = 1 - \frac{\ln x}{\ln d_S}. \tag{5.4}$$

It is important to note that distances drawn to create multiple long-range connections also need to be independent from each other (see Definition 3.1), hence they are generated applying a series of *i.i.d* random variables obeying the *cdf* defined by Equation 5.4.

Similarly, when deleting a long-range connection in the procedure DELETE-RANDOM-LR-CONNECTION(), it is important to ensure that selection of deleted connections be independent from each other and the probability of selecting a long-range connection for deletion be the same for each existing connection independent of its distance from the deleting node.

Keeping long-range connection density within a predefined range, the proposed stochastic maintenance algorithm also provides balanced node degree. Explicit enforcement of balanced node degree is required because some optimization mechanisms introduce a small bias into the selection of new long-range connections. E.g., the algorithm reducing long-range connection establishment overhead of joining nodes (presented in Section 5.2.5) increases the probability of selecting higher degree nodes for long-range connections (The probability of acting as a forwarding node increases with the number of long-range connections, hence higher degree nodes are more likely to be found as the first hit in a given node ID range). Without maintenance, this preferential attachment would result into scale-free networks[3]

[3]Scale-free networks are also called power-law networks. Note that the term "power-law" refers to node degree distribution in this context. For power-law routing overlays (see Definitions 3.1 and 3.2), "power-law" refers to distance distribution of long-range connections

[24], where a few nodes have very large node degree and would cause strongly unbalanced load distribution in the DHT.

5.2.4 Self-stabilizing maintenance of short-range connections

Having connections to the closest peers in both directions is critical to guarantee the success of routing process. Maintaining a small N_S number of short-range connections instead of only one increases static resilience but cannot replace maintenance mechanisms.

For short-range connection maintenance, I have used a combinations of the leaf-set maintenance algorithm of Bamboo [17] (Bamboo is DHT derived from Pastry, and leaf set corresponds to the set of short-range connections) and the self-stabilizing maintenance algorithm of the Ring Network protocol [38].

Bamboo uses an epidemic algorithm to maintain short-range connections: each node periodically sends the list of its short-range connections to one of its randomly selected short-range peers (push) and this peer node sends back the list of its own short-range connections (pull). Through this mechanism, nodes can detect when a short-range peer becomes unreachable. Additionally, comparing the list of own short-range connections to the list of short-range connections received from peer nodes, nodes can detect missing short-range connections and using a closest peer search, create these missing connections as in the Ring Network protocol.

Since the number of short-range connections is fixed, per node short-range connection maintenance overhead does not increase as system's size grows. However, it is challenging to adapt dynamically the period of echo messages to churn rate.

5.2.5 Range-based long-range connection establishment

In order to further minimize overlay control traffic in DHTs, I have also proposed an algorithm to reduce the cost of establishing long-range connections (see Algorithm 5.2). The proposed algorithm can be used to establish initial long-range connections of joining nodes and also to create additional long-range connections as part of the maintenance process.

The key idea of the proposed algorithm is as follows: Instead of creating a connection to the node being the closest to the point drawn at random according to the required distance distribution, I define a small range around the selected point, and connection is created to the first node found in this range during the lookup process. Hence, the first (fastest) lookup hit (node ID) matching the determined range will be used. The selection of this range is illustrated in Figure 5.1, d is the distance drawn at random to create a new long-range connection. I define a range $[\frac{d}{c}, dc]$ where $c = 1 + \epsilon$, $0 < \epsilon \ll 1$ and ϵ is a constant system parameter. If there are no nodes within this range (the probability of this event increases with smaller d and ϵ values), than the lookup process will terminate at the closest node and connection will be created to this node (outside the range).

The parameters *initiator*, *proxy*, *target*, *rangeStart* and *rangeEnd* in Algorithm 5.2 are

Algorithm 5.2 Range-based long-range connection establishment

```
CREATE-LR-CONNECTION(direction)
 1  ▷ According to the cdf in Equation 5.4
 2  distance ← DRAW-RANDOM-DISTANCE()
 3  rangeOffsetMin ← distance/(1+ε)
 4  rangeOffsetMax ← (1 + ε) distance
 5  if rangeOffsetMax > 1
 6     then rangeOffsetMax ← 1
 7
 8  if direction == RIGHT
 9     then
10          target ← self + distance
11          rangeStart ← self + rangeOffsetMin
12          rangeEnd ← self + rangeOffsetMax
13  elseif direction == LEFT
14     then
15          target ← self − distance
16          rangeStart ← self − rangeOffsetMax
17          rangeEnd ← self − rangeOffsetMin
18
19  CREATE-LR-CONNECTION(self, self, target, rangeStart, rangeEnd)

CREATE-LR-CONNECTION(initiator, proxy, target, rangeStart, rangeEnd)
 1  if proxy ∈ [rangeStart, rangeEnd]
 2     then
 3          CREATE-CONNECTION(initiator, proxy)
 4     else
 5          newProxy ← GET-CLOSEST-NODE(proxy, target)
 6          if d(newProxy, target) < d(proxy, target)
 7            then
 8                 CREATE-LR-CONNECTION(initiator, newProxy, target, rangeStart, rangeEnd)
 9            else
10                 ▷ if no nodes found within the range, fall back to the closest node
11                 CREATE-CONNECTION(initiator, proxy)
```

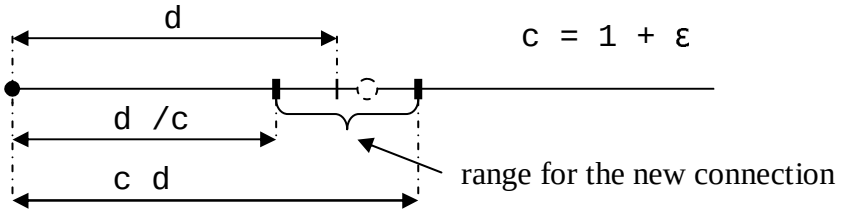

Figure 5.1: Defining range for a new long-range connection

all node identifiers in the metric space of the DHT. The variable *self* denotes the ID of the node creating the new connection. Finally, the parameter *direction* can take the values LEFT and RIGHT, depending on which metric space partition is selected to create the new connection. Note that addition and substraction in lines 10-12 and 15-17 denote shifting in clockwise (+) or counterclockwise (-) direction along the ring geometry of the DHT metric space.

It can be shown that the expected number of routing hops required to reach an arbitrary node within this range is upper bounded by a constant independent of network size. Using Theorem 4.5, the distance to the target decreases by a factor of $\frac{1}{1+\lambda}$ in expected value after each routing hop via long-range connections (where λ is the long-range connection density of the overlay). Theorem 4.5 applies to unidirectional overlays where routing is restricted to one direction and it is not possible to "overshoot" the target even if this would further decrease the absolute distance to the target. Hence, per-hop routing progress for bidirectional PPLROs is lower bounded by per-hop routing progress of unidirectional PPLROs, and therefore:

$$E\left[\frac{d_{i+1}}{d_i}\right] \leq \frac{1}{1+\lambda}, \tag{5.5}$$

where d_i and d_{i+1} are the distances from the target in routing step i and $i+1$, respectively. Considering the routing process towards a point at distance d, the range defined in Figure 5.1 is reached (in worst case) when the distance to this target decreases below $d - \frac{d}{c}$. This corresponds to a distance decrease factor

$$\frac{d - \frac{d}{c}}{d} = \frac{c-1}{c} = \frac{\epsilon}{1+\epsilon}. \tag{5.6}$$

Since this factor is a constant (independent of network size) and since according to Equation 5.5, distance to the target decreases at each routing step at least by a constant factor in expected value, finding a node in the given range takes only $O(1)$ steps. Hence the average overall communication overhead of creating a new long-range connection within the range is $O(1)$ (creating a connection to the closest node to a given point would require $O(\log n)$ hops).

Another advantage of range-based long-range connection establishment is that it allows taking into account network locality by selecting the physically closest node from the first few nodes found in the given range.

The only inconvenience of range-based connection establishment is that it introduces slight distortions in the required distance distribution of long-range connections. However, simulation results show that the small bias caused by picking up an arbitrary node from the above defined range instead of the closest node to the drawn point does not affect the distance distribution of long-range connections considerably. Figure 5.2 shows the distance distribution of long-range connections in a network of 128k nodes using $\epsilon = 0.2$ (other overlay parameters for the simulation correspond to the default values in Table 5.2). The dotted line represents simulation results while the solid line corresponds to the theoretical *cdf* according to Equation 5.4.

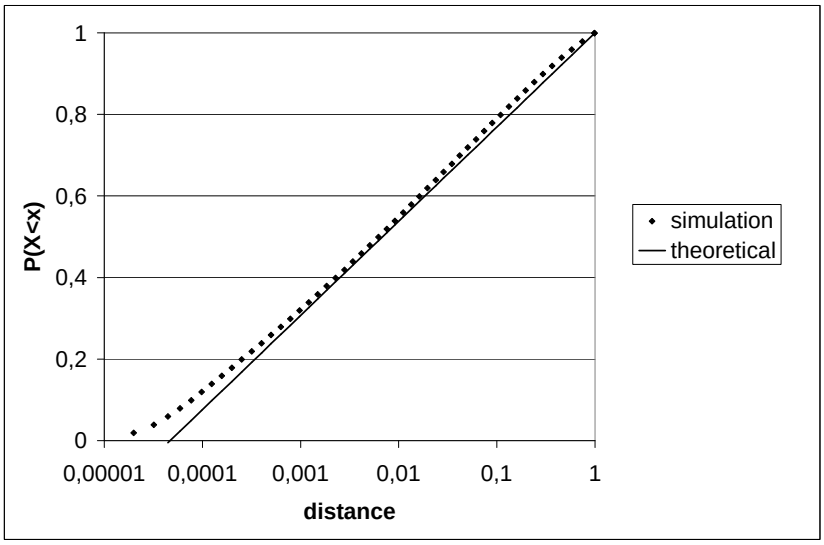

Figure 5.2: Long-range connection distance distribution (theoretical vs. simulation)

Besides range-based connection establishment, other factors also contribute to slight deviations from the theoretical values in the smaller distance range. For example, node identifiers are drawn at random (according to a uniform distribution), hence d_S values are not the same for all nodes but follow an Erlang distribution with rate n (number of nodes in the network) and shape parameter N_S (number of short-range connections).

5.2.6 Join algorithm

Although the join process of new nodes is not strictly part of maintenance, a proper implementation of this algorithm can further decrease the overall control traffic of a DHT under churn.

The applied join procedure is described in Algorithm 5.3. Join process starts by locating one or more bootstrap nodes, which can be arbitrary nodes of the DHT overlay. Then, using these bootstrap node(s), the joining node executes a query for its own node ID and creates a short-range connection to the closest node (closest does not mean physical distance but refers to the distance of node IDs in the metric space of the DHT). Querying short-range connections of this closest node, the joining node creates connections to all of its short-range peers (the N_S closest nodes from both partitions of the DHT metric space, where N_S is an adjustable system parameter). After short-range connection establishment, the joining node completes the join process creating its long-range connections in both partitions of the DHT metric space.

Locating the first (closest) short-range peer corresponds to a regular lookup operation, hence involves $O(\log n)$ routing hops. Other short-range connections can be directly queried from this first short-range peer (and its short-range peers), requiring $O(N_S)$ messages. Finally, using the range-based connection establishment algorithm presented in Section 5.2.5,

one long-range connection establishment costs $O(1)$ messages.

Algorithm 5.3 Join process

```
JOIN-NETWORK()
 1  bootstrapNode ← FIND-BOOTSTRAP-NODE()
 2  closestPeer ← LOOKUP-CLOSEST-NODE(bootstrapNode, self)
 3  CREATE-SR-CONNECTION(closestPeer)
 4
 5  ▷ Create short-range connections
 6  repeat
 7          leftCandidate ← GET-CLOSEST-CANDIDATE(candidates, self, LEFT)
 8          rightCandidate ← GET-CLOSEST-CANDIDATE(candidates, self, RIGHT)
 9          leftPeerAdded ← CREATE-SR-CONNECTION(leftCandidate)
10          rightPeerAdded ← CREATE-SR-CONNECTION(rightCandidate)
11    until leftPeerAdded or rightPeerAdded
12
13  ▷ Create long-range connections in the right partition of the metric space
14  while ESTIMATE-LR-CONNECTION-DENSITY(RIGHT) < λ_opt
15        do CREATE-LR-CONNECTION(RIGHT)
16
17  ▷ Create long-range connections in the left partition of the metric space
18  while ESTIMATE-LR-CONNECTION-DENSITY(LEFT) < λ_opt
19        do CREATE-LR-CONNECTION(LEFT)

CREATE-SR-CONNECTION(candidate)
1  REMOVE(candidates, candidate)
2  if IS-SHORT-RANGE(candidate, N_S)
3     then
4          CREATE-CONNECTION(self, candidate)
5          newCandidates ← QUERY-SHORT-RANGE-CONNECTIONS(candidate)
6          ADD(candidates, newCandidates)
7          return TRUE
8     else
9          return FALSE
```

The order of short-range and long-range connection establishment is important from several aspects. Without establishing short-range connections first, joining nodes might not be able to forward requests (received through already established long-range contacts) during the transient period of the join process. Short-range connections are also required to guarantee the $O(1)$ bound on communication overhead per connections when using the range-based long-range connection establishment algorithm (see Section 5.2.5) to create initial long-range connections of joining nodes. Without its short-range connections, a joining node intending to create a new long-range connection to a node at a distance d might have to forward this request through an existing long-range peer at a larger distance from the

target point – hence invalidating the assumption on the required distance decrease ratio in Equation 5.6.

There are several alternatives to create initial long-range connections during the join process. One way is to pick up random points within the range $[0, 1)$ of the DHT metric space according to the distance distribution given by Equation 5.4 and create connections to nodes being the closest to these points (this is the approach taken in Symphony [14]). Similarly to long-range connection maintenance, this process is repeated until estimated long-range connection density ($\widehat{\lambda_l}$ and $\widehat{\lambda_r}$) reaches λ_{opt} in both partitions of the DHT metric space.

Another alternative is to create long-range connections sequentially in a decreasing distance order (due to the bidirectional nature of the overlay, this process is executed separately in both partitions of the DHT metric space). In the followings, I show that choosing distance ratios for subsequent connections so that their logarithm follows an exponential distribution with parameter λ_{opt} results into the same power-law long-range connection distance distribution.

Theorem 5.2. *Let $x_i \sim Exp(\lambda)$ be a series of* i.i.d. *random variables with an exponential distribution of parameter λ. Furthermore, consider the sequential long-range connection establishment process of a joining node where d_i is the distance of the joining node from its connection created in step i. Then choosing distances to subsequent long-range connections according to the recursive formula of Equation 5.7 results into the long-range connection distance distribution of a probabilistic power law routing overlay of long-range connection density λ.*

$$d_0 = 1 \quad \textit{and} \quad \ln \frac{d_i}{d_{i+1}} = x_i(\lambda) \tag{5.7}$$

Proof. Let d'_i and d'_{i+1} be the distances from two subsequent long-range connections in the logarithmically transformed view of the joining node. According to the definition of this transformation (see Definition 4.1), $d'_i = -\ln d_i \;\Rightarrow\; d_i = e^{-d'_i}$. Hence

$$\ln\left(\frac{d_i}{d_{i+1}}\right) = \ln\left(\frac{e^{-d'_i}}{e^{-d'_{i+1}}}\right) = d'_{i+1} - d'_i,$$

which means that the logarithm of the ratio of distances to two subsequent long-range connections correspond to their distance in the transformed view of the joining node. According to Theorem 4.3, long-range connections in this transformed view of a node correspond to a random realization of a truncated Poisson process of rate λ, where λ is the long-range connection density of the overlay. Hence distances between long-range connections in this transformed view are independent and exponentially distributed with parameter λ.

□

5.2.7 Lookup strategy

Although this chapter focuses on overlay maintenance mechanisms, lookup strategy is also a key architectural component of a DHT designed for dynamic network environments. Therefore, I briefly review the implications of the proposed maintenance mechanism on lookup strategy.

In the proposed maintenance strategy, period keep-alive message are only used for short-range connections; long-range connection failures are detected only via timeouts during regular lookup traffic. Therefore a recursive lookup strategy is a prerequisite to enable this failure detection. Additionally, recursive lookup allows DHT nodes to fine-tune timeout settings separately for each of their peers based on collection of round-trip time statistics[4] similarly to the Bamboo DHT [17].

Performance of recursive lookup degrades quickly when the ratio of undetected failed links increases. However, this performance can be significantly improved using the RR-ACK strategy proposed in [20]. RR+ACK is derived from recursive routing, however, each intermediate node also sends an ACK to the initiator node containing the address of the next-hop node. Although additional ACK messages slightly increase lookup traffic, in case of a failure, the initiator node can re-initiate another lookup using the next-hop address from the latest ACK without having to restart the whole lookup process.

5.3 Evaluation

To evaluate performance of the proposed maintenance strategy, I proposed a linear equation system and a Markov chain model describing evolution of long-range connections in a network under churn. Using this model, I have derived analytically the long-range connection maintenance overhead of the overlay as a function of network size and churn rate.

I have also implemented the proposed system in a cycle-based simulation environment and performed extensive simulations for a wide range of protocol parameters and networks sizes. Finally, based on these simulation results, I have determined optimal protocol parameter ranges for the proposed system.

5.3.1 Analysis of maintenance traffic in a network under churn

As described in Section 5.2.3, stochastic long-range connection maintenance is based on keeping long-range connection density within a lower and upper threshold. Let $\lambda_{min} = \lambda_{opt} - \Delta\lambda$ and $\lambda_{max} = \lambda_{opt} + \Delta\lambda$, thus long-range connection density can vary within the range $[\lambda_{min}, \lambda_{max}]$.

As a first approximation, I assumed that node identifiers partition the DHT metric space into equal partition hence the d_S distance from the farthest short-range peer is the same for

[4]These statistics are based solely on regular lookup traffic, in contrast to Bamboo where keep-alive messages are sent to each long-range peer periodically.

all nodes. Under this assumption, the number of long-range connections[5] can be expressed in terms of estimated long-range connection density uniformly for each node of the overlay as $N_L = -ln(d_S)\widehat{\lambda}$. Hence the λ_{min} and λ_{max} thresholds can be converted to thresholds on the number of long-range connections: $N_{opt} = -ln(d_S)\lambda_{opt}$, $\Delta N = -ln(d_S)\Delta\lambda$, $N_{min} = N_{opt} - \Delta N$ and $N_{max} = N_{opt} + \Delta N$. If N_L drops below N_{min}, then new connections are created until reaching N_{opt}. If N_L exceeds N_{max}, then connections are deleted until reaching again N_{opt}.

Furthermore, I assumed that the arrival of new nodes as well as the departure or failure of nodes can be described by a Poisson process. Let r_{in} be the arrival rate and r_{out} be the departure rate of nodes in a system under churn (I assumed a worst case scenario where departing nodes either fail or depart ungracefully). Let R_c be the creation rate of new long-range connections and R_d the deletion rate of existing long-range connections in the system (hereafter, I refer to long-range connections simply as connections in this subsection). I defined all of the above rates normalized to the size of the network; for example R_c denotes the number of new connections created per nodes and per time units.

Both R_c and R_d can be decomposed into two components. New connections are created on the one hand by new nodes joining the network and on the other hand by nodes whose connection density drops below the lower threshold λ_{min}. Let R_{cm} denote the rate of connection establishment resulting from exceeding the lower threshold λ_{min}. Hence:

$$R_c = N_{opt} r_{in} + R_{cm}. \tag{5.8}$$

Similarly, connection deletion occurs when a node detects failure of a connection or when a node deletes connections after exceeding the upper threshold λ_{max} of its connection density. The rate of connection deletion resulting from failure of nodes cannot be expressed directly in terms of failure rate. This is a consequence of the applied "lazy" failure detection mechanism: a failed connection is detected only by timeouts during regular communication. Assuming that the probability of selecting a failed connection for message forwarding equals to the ratio of failed connections in the network, failure detection rate $R_f = c_f R_{all}$, where c_f is the ratio of failed connections in the network and R_{all} is the overall communication rate over long-range connections. Let R_{dm} denote the other component corresponding to the rate of connection deletion resulting from exceeding the upper threshold λ_{max}. Then R_d can be expressed as

$$R_d = c_f R_{all} + R_{dm}. \tag{5.9}$$

If a node has N_{max} connection and another node creates a new connection to it, then maintenance will delete $\Delta N + 1$ connections to reach the N_{opt} connection number. Similarly, if a node has N_{min} connection and one of these connections is deleted, then maintenance will create $\Delta N + 1$ new connections to reach the N_{opt} connection number. Hence, both R_{cm}

[5] To simplify formalism, I do not explicitly note right side and left side connections numbers and long-range connection densities; they can be calculated exactly in the same way.

	N_{min}	$N_{min}+1$	...	N_{opt}	...	$N_{max}-1$	N_{max}
N_{min}	$-R_c-R_d$	R_c	...	R_d	...	0	0
$N_{min}+1$	R_d	$-R_c-R_d$	...	0	...	0	0
...	...	...	...	...	...	...	...
N_{opt}	0	0	...	$-R_c-R_d$	...	0	0
...	...	...	...	...	...	...	...
$N_{max}-1$	0	0	...	0	...	$-R_c-R_d$	R_c
N_{max}	0	0	...	R_c	...	R_d	$-R_c-R_d$

Figure 5.3: Transition rate matrix of the number of long-range connections

and R_{dm} can be expressed in terms of R_c, R_d and the probability of having a connection number corresponding to the upper and lower level connection density thresholds:

$$R_{dm} = (\Delta N + 1)P(N_L = N_{max})R_c \tag{5.10}$$

and

$$R_{cm} = (\Delta N + 1)P(N_L = N_{min})R_d. \tag{5.11}$$

Substituting (5.8) and (5.9) into (5.10) and (5.11):

$$R_{dm} = (\Delta N + 1)P(N_L = N_{max})(N_{opt}r_{in} + R_{cm}) \tag{5.12}$$

and

$$R_{cm} = (\Delta N + 1)P(N_L = N_{min})(c_f R_{all} + R_{dm}). \tag{5.13}$$

c_f and the two probabilities in the above equation system depend on the history of the network. Furthermore, besides R_{cm} and R_{dm}, there is a third independent variable (R_{all}) depending on many factors in a complex way. Hence it is difficult to provide a general solution. However, in the followings, I show that the above equation system can be solved for steady states when the arrival rate and failure rate of nodes is the same ($r_{in} = r_{out} = r$).

Theorem 5.3. *Assuming a steady state where the arrival rate r_{in} and failure rate r_{out} of nodes is the same, the following upper bounds hold on connection creation and deletion rate of long-range connection maintenance process:*

$$R^a_{dm} \leq 2\frac{\lambda_{opt}}{\Delta\lambda}r \quad \textit{and} \quad R^a_{cm} \leq 2\frac{\lambda_{opt}}{\Delta\lambda}r. \tag{5.14}$$

Proof. First, I calculate the $P(N_L = N_{max})$ and $P(N_L = N_{min})$ probabilities then I derive the ratio of failed connections in the network for this steady state.

Considering the number of long-range connections as a random variable, the evolution of this random variable can be described by a stationary continuous time Markov chain. The transition rate matrix $\mathbf{Q}$ of the resulting Markov chain is shown in Figure 5.3. According to the definition of transition rate matrix, an element in row i and column j represents the probability per time unit that the system makes a transition from state i to state j if $i \neq j$ and the rate at which the probability of state i decreases if $i = j$.

Let π be the vector describing stationary distribution of the random variable corresponding to the number of long-range connections. Then in steady state, π should satisfy the following equation: $\pi\mathbf{Q} = 0$. As a consequence, π can be obtained as the left eigenvector of the transition matrix associated with the eigenvalue 0. The above transition matrix has only one left eigenvector associated to the eigenvalue 0; independent of the size of the matrix, this eigenvector can be expressed as:

$$\pi = \left(\frac{1}{(\Delta N+1)^2}, \frac{2}{(\Delta N+1)^2}, ..., \frac{\Delta N+1}{(\Delta N+1)^2}, ..., \frac{2}{(\Delta N+1)^2}, \frac{1}{(\Delta N+1)^2} \right). \tag{5.15}$$

Hence

$$P(N_L = N_{min}) = P(N_L = N_{max}) = \frac{1}{(\Delta N+1)^2}. \tag{5.16}$$

A steady state also implies that c_f is constant, hence the rate at which failed connections are created and deleted are equal. A failed connection is either deleted when a node detects it via timeouts or simply disappears from the system when a node fails or leaves the network. Hence $N_{opt} r_{out} = c_f R_{all} + N_{opt} r_{out} c_f$ which gives:

$$R_f = c_f R_{all} = N_{opt} r (1 - c_f). \tag{5.17}$$

Substituting these results into (5.12) and (5.13), gives the following equations:

$$R_{dm} = \frac{(\Delta N+1)}{(\Delta N+1)^2}(N_{opt} r + R_{cm}) \tag{5.18}$$

and

$$R_{cm} = \frac{(\Delta N+1)}{(\Delta N+1)^2}\left(N_{opt} r (1 - c_f) + R_{dm}\right). \tag{5.19}$$

Solving the equation system, the following upper bounds can be derived for R_{dm} and R_{cm}:

$$R_{dm} = \frac{N_{opt}}{\Delta N} \frac{\Delta N + 2 - c_f}{\Delta N + 2} r \leq \frac{N_{opt}}{\Delta N} r = \frac{\lambda_{opt}}{\Delta \lambda} r \tag{5.20}$$

and

$$R_{cm} = \frac{N_{opt}}{\Delta N} \frac{\Delta N + 2 - c_f(\Delta N + 1)}{\Delta N + 2} r \leq \frac{N_{opt}}{\Delta N} r = \frac{\lambda_{opt}}{\Delta \lambda} r. \tag{5.21}$$

The obtained results correspond to maintenance rates of long-range connections in either the left side or right side of the DHT metric space separately for each node. Hence overall maintenance rates can be obtained by doubling these rates.

□

These upper bounds do not depend on network size, hence connection creation and deletion rate of long-range connection maintenance process per node scale as $O(1)$ with network size.

To derive overall maintenance overhead in steady state, I first recapitulate the notion and background of all maintenance components used in the above analysis. All maintenance rates referred to long-range connections in either the left side or the right side partitions

	Conn. creation/deletion rate		Communication overhead
New nodes joining	$N^a_{opt}r$:	$O(\log n)r$	$O(\log n)r$
Create by maintenance	R^a_{cm}:	$O(1)r$	$O(1)r$
Delete by maintenance	R^a_{dm}:	$O(1)r$	$O(1)r$
Detect failed connections	R^a_f:	$O(\log n)r$	$O(\log n)r$

Table 5.1: Overview of maintenance overhead components

of the DHT metric space for each node. Overall rates are denoted in the followings by the upper index a and can be obtained by doubling these rates. All maintenance components are all related to either connection creation or connection deletion rates (R^a_c and R^a_d). Table 5.1 summarizes connection creation and deletion rates and the associated communication overhead for each of these components.

I have defined R^a_c as the number of new long-range connections created per node and per time unit. R^a_c has two components (see Equation 5.8): the first one is the rate of connection establishment due to connections created by new nodes when they join the network ($N^a_{opt}r$) while the second one corresponds to the rate at which long-range connection maintenance creates additional connections (R^a_{cm}). N^a_{opt} increases logarithmically with network size, hence the per node connection establishment rate of joining nodes scales as $O(\log n)$ with network size. $R^a_{cm} \leq 2\frac{\lambda_{opt}}{\Delta\lambda}r$ according to Theorem 5.3, hence the per node connection creation rate of maintenance process scales as $O(1)$ with network size. Using the range-based long-range connection establishment algorithm presented in Section 5.2.5, communication overhead per new connections scales as $O(1)$ in both cases, therefore the associated communication overhead is also $O(\log n)$ and $O(1)$, respectively.

I have defined R^a_d as the number of long-range connections deleted per node and per time unit. R^a_d also has two components: the rate at which maintenance process deletes existing connections (R^a_{dm}) and the rate at which failed connections are discovered during regular operation (R^a_f). From Theorem 5.3, $R^a_{dm} \leq 2\frac{\lambda_{opt}}{\Delta\lambda}r$, hence per node connection deletion rate by maintenance scales as $O(1)$ with network size. Since a connection deletion involves sending only one single disconnect message, the associated communication overhead is also $O(1)$. From Equation 5.17, $R^a_f = N^a_{opt}r(1-c_f)$, hence the detection rate of failed connections scales as $O(\log n)$ with network size. Strictly speaking, this component does not contribute to overall maintenance overhead because the detection of a failed connection does not directly trigger any maintenance action[6]. However, I also included it into the maintenance overhead components table, because detecting a failed connection requires resending one message via another connection, hence each detected failed connection increases overall communication overhead by one message.

Let the time unit for churn rate be defined as the system's half-life defined in [35] (system half-life measures the time during which half of the nodes in the network are replaced by new arrivals). Using this definition, the overall per node maintenance traffic of long-range

[6]Except when $N_L = N_{min}$, but the associated maintenance overhead has already been accounted for the connection creation by the maintenance component.

connections per half-life is $O(\log n)$ in our system. This result is very important because the authors in [35] show that this is in fact the theoretical lower bound of maintenance traffic to ensure that the network remains connected.

Another interesting observation related to Table 5.1 is that most of the missing long-range connections are replaced by new connections established by joining nodes and only a small fraction need to be created explicitly by maintenance. In other words, most of the maintenance is achieved automatically as a side effect of the unavoidable control traffic related to the creation of new connections of joining nodes.

5.3.2 Simulation results

Performance of a DHT under churn can be described as a tradeoff between three main performance factors: lookup latencies, lookup error ratio and maintenance overhead. In Section 5.3.1, I have shown that the proposed system is asymptotically optimal in terms of maintenance overhead. However, different protocol parameters of maintenance also affect lookup latencies and lookup error rates. Therefore I have performed extensive simulations for a wide range of parameter settings in order to determine optimal protocol parameter ranges for the best tradeoff between these three performance factors.

To be able to focus on protocol behavior, I ignored transport layer details and implemented the proposed system in an own cycle-based simulator written in Java. In contrast to an event-based simulator, a cycle-based simulator partitions simulation time into short cycles, and passes control periodically to each node in each cycle. The main advantage of this high level, cycle-based approach is scalability; I could run simulations up to network sizes of 128k nodes on a medium class PC. The only disadvantage is that cycle-based simulation does not reflect temporal ordering of events within a cycle. However, the effect of such temporal disorders can be arbitrarily reduced choosing shorter cycle periods. To find an optimal tradeoff between simulation performance and correctness, I have performed simulation trials with high temporal resolution (200 simulation cycles per average node inter-arrival time), than performed the same simulations again gradually reducing temporal resolution. At 2 simulation cycles per average node inter-arrival time, simulation results still did not show statistically significant difference compared to higher temporal resolutions, therefore further simulations have been performed using this setting.

Similarly to the analytical evaluation, I used network half-life as a metric of churn rate. I focused on steady state behavior where arrival and departure rates were equal. Since simulation time in a cycle-based simulator is measured in simulation cycles, network half-life is defined in this case as the number of simulation cycles during which half of the nodes are replaced by new arrivals in the DHT. Furthermore, I've measured maintenance overhead in terms of the number of maintenance message exchanges per node and per network half-life. This maintenance overhead metric has the advantage of being independent of churn rate. The actual maintenance bandwidth can be calculated from this maintenance metric for any churn rates by deriving the amount of maintenance traffic from the number of messages

parameter	range	default value
M	$[1, 100]$	10
λ	$[0.6, 6.0]$	$\frac{1}{ln2} \approx 1.44$ (Chord)
$\frac{\Delta\lambda}{\lambda}$	$[0.1, 0.3]$	0.2
N_S	$[2, 5]$	3
L	$[1, 100]$	10

Table 5.2: Summary of simulation parameters

and dividing this by network half-life.

Table 5.2 summarizes the set of protocol parameters varied during simulations. N_S is the number of short-range connections per node in the overlay (separately at both the left and right sides), M is the number of short-range connection maintenance cycles per network half-life and λ_{opt} and $\Delta\lambda$ are the long-range connection maintenance parameters[7]. Finally, L characterizes user activity; it corresponds to the average number of lookups initiated by one node in a network half-life period. During simulations, one or two of these parameters have been varied within the given range, setting other parameters to their default value. Unless otherwise noted, a parameter in a simulation is set to its default value. Most of these simulations have been run for all of the following network sizes: $1k$, $2k$, $4k$, $8k$, $16k$, $32k$, $64k$, $128k$.

Modeling churn

To generate churn during simulations, I modeled both the arrival and departure of nodes as a Poisson process. In the simulator, this was approximated by adding a new node and removing a random existing node with a predefined (small) probability in each simulation cycle.

Churn characteristics in real P2P networks can differ significantly from the above Poisson process model. The authors in [39] analyzed traces from real P2P networks and fitted distributions on node inter-arrival times and session lengths. However, I have shown that the distribution of node inter-arrival times and session lengths does not significantly affect the performance of the proposed maintenance mechanism.

To analyze the impact of different churn models, I have repeated the same simulation with different node inter-arrival time and session length distributions. Two of them were based on BitTorrent measurements from [39] (a FlatOut and a Red Hat torrent), while two others were artificial churn models based on simple distributions. Node inter-arrival time and session length distributions of these churn models are summarized in Table 5.3 (for Weibull distributions, k denotes the shape parameter). Each of theses simulations were performed for a network of $16k$ nodes using the default system parameters from Table 5.2. The intensity parameter of inter-arrival time and session length distributions was normalized

[7]The default λ value is set equivalent to the long-range connection density of Chord (see Section 4.2.1)

	BT FlatOut	BT Red Hat	Exp	Uniform
inter-arrival time distribution	Weibull ($k = 0.62$)	Weibull ($k = 0.53$)	exp	uniform
session length distribution	Weibull ($k = 0.59$)	Weibull ($k = 0.34$)	exp	uniform

Table 5.3: Summary of simulated churn models

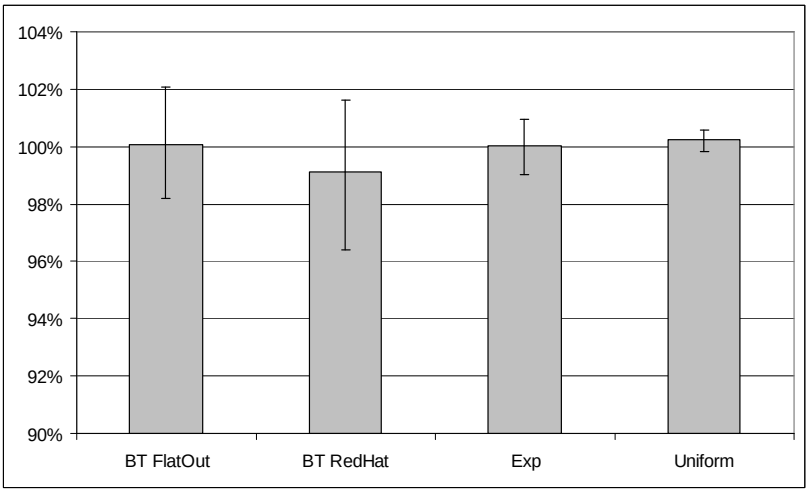

Figure 5.4: Relative maintenance overhead for different churn models

to obtain an average network size of $16k$ nodes in steady state and to have the same network half life for each churn model. In order to obtain accurate results and minimize the effect of temporal inconsistencies, the number of simulation cycles per average node inter-arrival time was increased from the default value of 2 to 20.

Figure 5.4 represents relative maintenance overhead per network half life for each of the simulated churn models (maintenance overhead for the Poisson process churn model used in all other simulations corresponds to 100%). In addition to the average value, the 10^{th} and 90^{th} percentiles are shown for each measurements (based on the maintenance overhead from 20 subsequent network half-lives after reaching steady state).

Figure 5.4 shows that the performance of the proposed maintenance mechanism does not significantly depend on churn characteristic. Differences between average per network half-life maintenance overheads are smaller than 1% for all the four simulated churn models. As a result – from the point of view of stochastic maintenance – network dynamism can be characterized by one single parameter: network half-life.

Evaluation of overlay properties under churn

First, I present simulation results related to overlay properties, namely the distance distribution of long-range connections, node degree distribution and the number of routing hops as a function of network size. Note that all of these properties have been measured in a

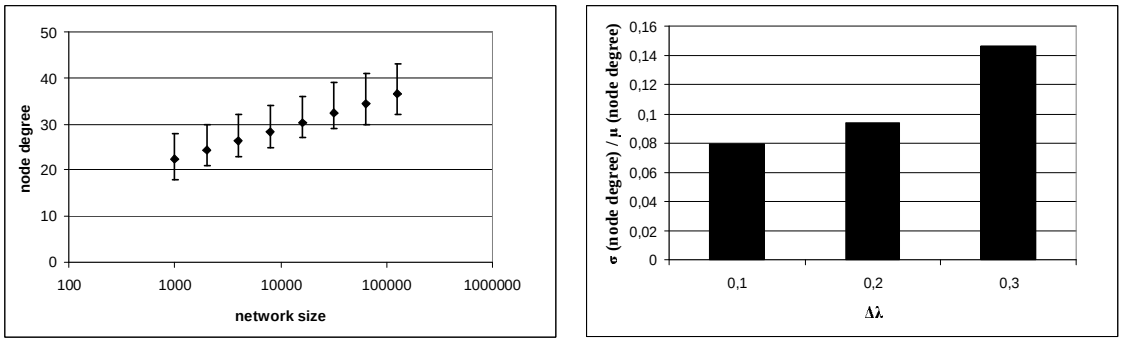

Figure 5.5: Node degree properties

system under churn (steady state).

Figure 5.5 represents node degree properties. The left graph represents the average value as well as the 5^{th} and 95^{th} percentiles of node degree as a function of network size. In accordance with analytical results, node degree clearly increases logarithmically with network size. The right graph shows dependence of relative standard deviation of node degree from the value of the $\Delta\lambda$ parameter. As expected, standard deviation of node degree increases with the value of $\Delta\lambda$ and relative standard deviation is independent of network size (except for very small network sizes, where nodes only have a few long-range connections and quantization increases additionally standard deviation)

Figure 5.6 analyzes the number of routing hops required to find a peer node in the overlay under churn. The left graph shows the average as well as the 5^{th} and 95^{th} percentiles of the number of routing hops as a function of network size while the right graph shows dependence of the average number of routing hops on the λ long-range connection density parameter. In accordance with analytical results, the number of routing hops clearly increases logarithmically with network size.

The right graph in Figure 5.6 represents the average number of routing hops as a function of the inverse of the λ long-range connection density for a network of $16k$ nodes. Additionally, it compares simulation results with the upper bound of Theorem 4.9. This upper bound refers to unidirectional probabilistic power-law routing overlays. Per-hop routing progress for the bidirectional case is at least as much as per-hop routing routing progress of an unidirectional overlay with the same parameters, hence the average number of routing hops in the unidirectional case can be used as an upper bound on the average number of routing hops in the bidirectional case. Since here, the upper bounded values are already upper bounds themselves, the resulting upper bound is rather conservative. Nevertheless, the graph shows that the average number of routing hops increases nearly linearly with the inverse of λ in the range practically used in DHTs. This holds for any other network sizes in the simulated range (between $1k$ and $128k$).

The influence of the short-range connection number parameter N_S on the number of routing hops is marginal since forwarding via short-range connections usually takes place

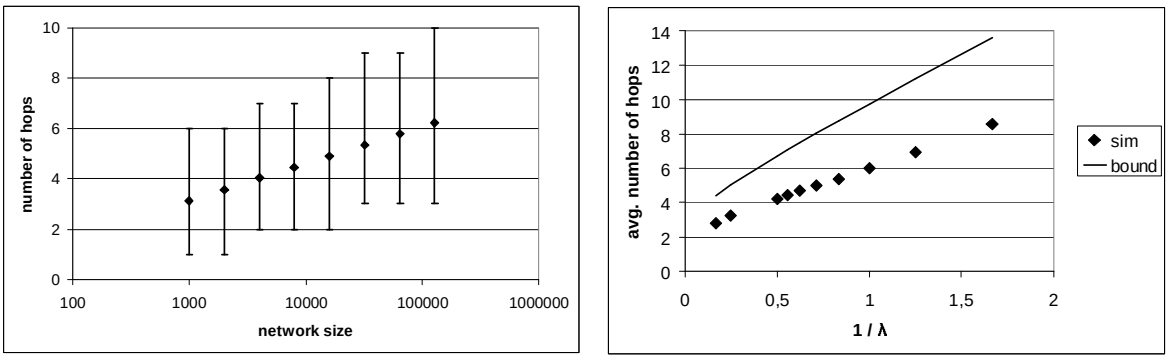

Figure 5.6: Number of routing hops in a network under churn

only at the last few steps. E.g., increasing the number of short-range connections from 2 to 5 decreases the average number of routing hops by only ≈ 0.35 (independent of network size).

Evaluation of maintenance performance

To evaluate maintenance performance, I have used two complementary performance metrics: the availability and the maintenance overhead ensuring this availability. Availability has been measured as the lookup failure ratio in the overlay under churn while maintenance overhead was characterized by the number of maintenance messages (request/reply pairs) per network half-life and per node.

First, I examined how the various overlay properties (N_S, λ, $\Delta\lambda$) influence these two metrics, then I analyzed the effect of maintenance properties and network dynamism (M, L) on performance. Finally, I evaluated stability of the proposed maintenance strategy based on these results.

Figure 5.7 shows the effect of the number of short-range connections (N_S) on maintenance performance. The left graph represents the number of maintenance messages per network half-life as a function of network size for three different values of the N_S parameter while the right graph shows failure ratio for the same three parameter values.

First of all, the left graph supports the major contribution: the $O(\log n)$ bound on the number of maintenance messages per node and per network half-life. It also shows that the N_S parameter accounts for maintenance traffic by a constant factor because the number of short-range connections only influences short-range connection maintenance traffic, which is independent from the size of the network. This constant factor increases with N_S faster than linearly. This is a consequence of a failure dissemination mechanism (as part of the short-range connection maintenance) which notifies each (possibly affected) short-range neighbor about detected short-range connection failures.

Simulations have shown that there is no significant dependence between failure ratio and network size: the right graph in Figure 5.7 representing lookup failure ratio for different

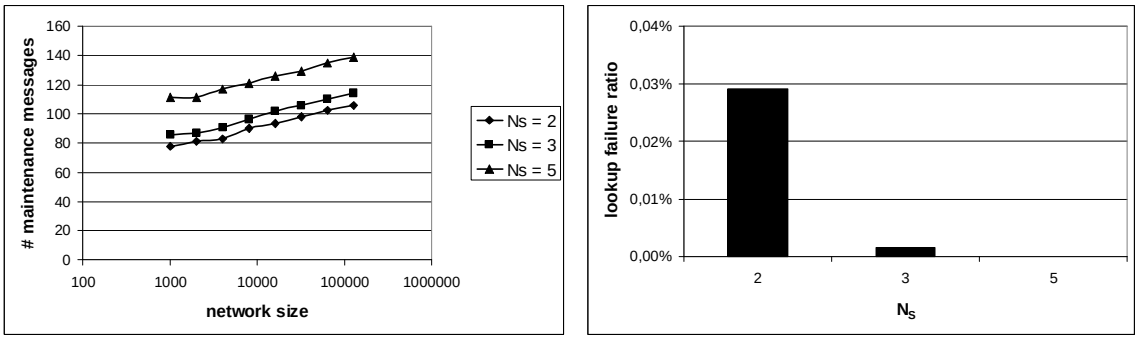

Figure 5.7: Effect of N_S on maintenance performance

N_S values is nearly the same for simulations of all network sizes between $1k$ and $128k$. This is in accordance with the fact that success of the routing process is ensured by short-range connections, hence failure ratio only depends on short-range connection maintenance parameters. The number of short-range connections per node is one of these parameters. Increasing N_S increases resilience of short-range routing and via failure notifications, it also decreases the average time to detect failure of a short-range peer. As a result, lookup failure ratio decreases very fast with increasing N_S values; for $N_S = 5$, it is smaller than 10^{-6}.

Figure 5.8 summarizes the impact of the λ long-range connection density and the $\Delta\lambda$ long-range connection density range width on maintenance overhead. The left graph shows the number of maintenance messages per network half-life and per node as a function of network size for different $\Delta\lambda/\lambda$ ratios. Smaller $\Delta\lambda/\lambda$ ratio means more restrictive maintenance and hence larger maintenance overhead. In fact, the R_{cm} and R_{dm} components of maintenance overhead are inversely proportional with the $\Delta\lambda/\lambda$ ratio (see equations 5.20 and 5.21). In extreme cases, this can even result in losing the loose stochastic nature of long-range connection maintenance. In small networks with small average node degree, a small $\Delta\lambda/\lambda$ ratio might mean that every new or lost connection will move estimated long-range connection density out of the required range and hence requires maintenance action. This explains that maintenance traffic for the smallest $\Delta\lambda/\lambda$ ratio can even be slightly higher in smaller networks. On the other hand, a too large $\Delta\lambda/\lambda$ ratio will result in unfair load distribution due to high variance of node degree (see Figure 5.5) and increasing this ratio above 0.2 results only in small saving in maintenance traffic.

The right graph in Figure 5.8 shows the dependence of the number of maintenance messages per network half-life and per node from the value of the λ long-range connection density parameter (the graph shows simulation results for $16k$ nodes). Messages sent by joining nodes to create new connections are also included into maintenance traffic in our analysis. This explains the nearly linear increase in maintenance traffic for higher λ values (the higher λ is, the more connections need to be created during the join process). The slight increase in maintenance traffic for smaller λ values can be explained by the same phenomenon as the increase for small $\Delta\lambda$ values: long-range connection maintenance loses

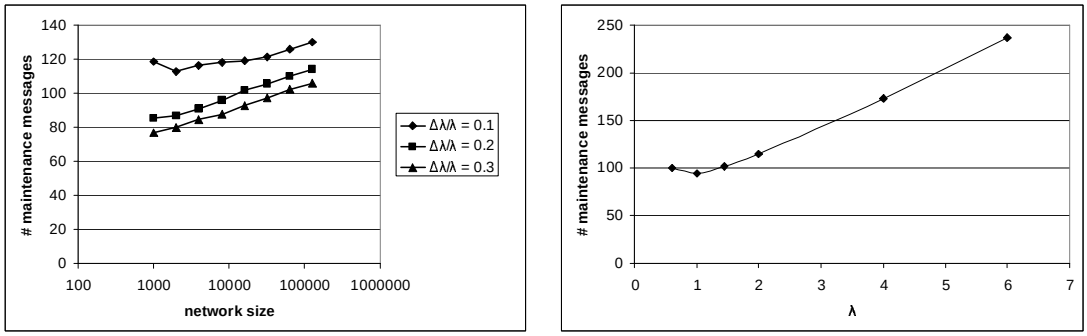

Figure 5.8: Effect of $\Delta\lambda$ and λ on maintenance overhead

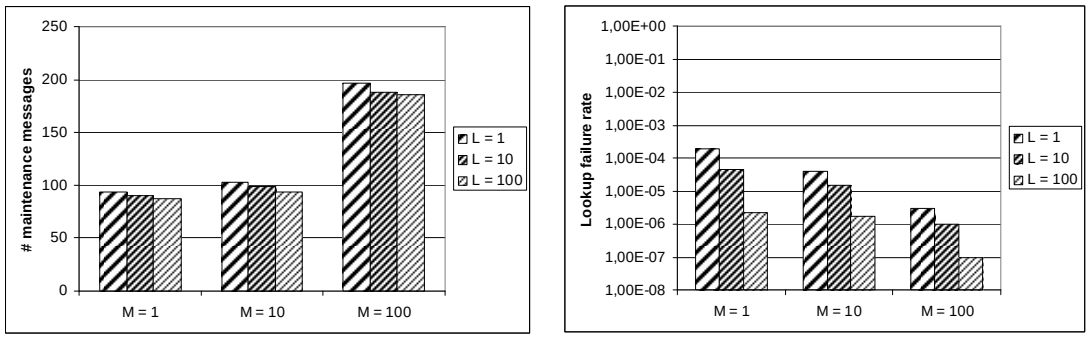

Figure 5.9: Effect of network and maintenance dynamism on maintenance performance

its stochastic nature if the allowed long-range connection density range is too small.

Figure 5.9 depicts the effect of maintenance and network dynamism on performance. The left graph shows the number of maintenance messages per network half-life and per node for various combinations of short-range connection maintenance cycles per network half-life (M) and the average number of lookup request initiated per network half-life by one node (L) parameters. The right graph shows the lookup failure ratio for the same parameter combinations. Maintenance overhead in the left graph is shown for a network of $16k$ nodes, while lookup failure ratio represented in the left graph suites to any network sizes since it does not significantly depend on network size.

Increasing the frequency of short-range connection maintenance obviously increases the amount of short-range connection maintenance traffic. For $M = 10$ (10 maintenance cycles per network half-life), short-range connection maintenance overhead is still negligible compared to long-range connection maintenance traffic while for $M = 100$, it already accounts for more than half of the maintenance traffic. Increasing frequency of maintenance decreases lookup failure ratio significantly, since it allows faster detection of connection failures. Higher values of L – corresponding to more intensive user traffic in the overlay as compared to churn rate – also decrease failure ratio for the same reason: user traffic also

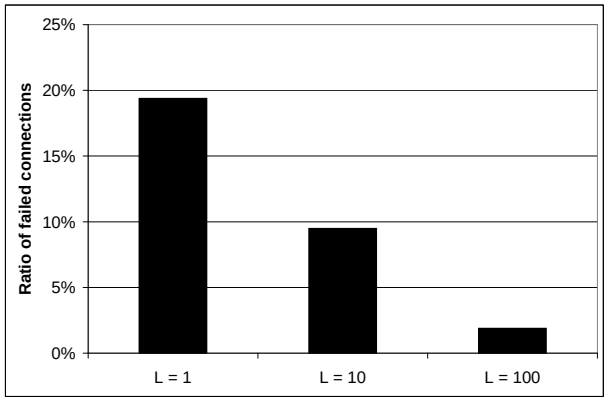

Figure 5.10: Ratio of (undetected) failed connections in the overlay

helps detection of failed connections and decreases the ratio of (undetected) failed connections in the network. Figure 5.10 shows this ratio (denoted by c_f in Section 5.3.1) as a function of the relative lookup rate. As a side effect of the smaller ratio of failed connections, maintenance traffic also decreases slightly (as L increases), since this will result in less message retransmissions for control traffic (see left graph of Figure 5.9).

Using the analytical results of Wu et al. in [20], it is possible to derive mean lookup latencies from the ratio of undetected failed connections in the network, average lookup hop count and average round-trip times in the network.

Summary of simulation results

Based on the above simulation results, it is possible to compile a set of optimal parameter ranges providing the best performance. Table 5.4 provides a summary of these parameter ranges.

Long-range connection density values smaller than 1.0 as well $\Delta\lambda/\lambda$ values smaller than 0.2 decrease the stochastic nature of maintenance and thus increase maintenance traffic. $\Delta\lambda/\lambda$ values greater than 0.3 do not improve maintenance performance considerably but decrease fairness in the network. λ values greater than 3.0 significantly increase maintenance traffic but do not considerably improve lookup performance any more.

The number of short-range connections should be higher than 2 at both sides in order to achieve negligible lookup failure rates. However, increasing this number above 5 does not make sense, since it provides already less than 10^{-6} lookup failure ratio even at high churn rates.

Finally, short-range connection maintenance frequencies smaller than 10 per network half-life result in high lookup failure ratio. On the other hand, values larger than 100 increase significantly maintenance traffic.

The relative lookup rate L depends on applications using the lookup service of the DHT, thus it cannot be set to "optimal" values. Simulations have shown very good performance

parameter	optimal range
λ	$[1.4, 3.0]$
$\frac{\Delta\lambda}{\lambda}$	$[0.2, 0.3]$
N_S	$[3, 5]$
M	$[10, 50]$

Table 5.4: Optimal protocol parameter ranges

for relative lookup rates equal to or higher than 10 and fair performance for $L = 1$. This means that the proposed maintenance strategy provides a good platform for applications where one node performs at least 10 lookups on average during its lifetime. However, for extreme cases where one node performs around or less than one lookup on average during its lifetime in the network, it probably does not make sense to maintain routing state and unstructured P2P networks might be more suitable.

For all the above simulations, I ignored transport layer details and assumed that all link capacities are infinite. Under these assumptions, simulations have shown that choosing the right protocol parameters for the proposed stochastic maintenance strategy, it is possible to ensure very high availability, independent of network size and churn rate.

Obviously, in real networks, we need to consider finite link capacities and congestions in the network. Instead of implementing complicated link capacity and congestion models in the simulator, I take a different approach: First, I calculate the maintenance bandwidth per node assuming infinite link capacities. Then I compare this bandwidth to the actual link capacity. If it is orders of magnitudes smaller than link capacities, the system will be stable with high probability assuming maintenance mechanisms are designed so as to avoid avalanche effects. The best way to avoid this avalanche effect is not to use reactive protocols [17]. Short-range connection maintenance is proactive while long-range connection maintenance is reactive in the proposed maintenance strategy. However, the reactive nature of long-range connection maintenance is significantly attenuated by its loose and stochastic properties, since maintenance is only initiated when reaching a lower or upper threshold long-range connection density. Additionally, this reactive property can be completely eliminated by further restricting long-range connection maintenance so that maintenance has to wait until the beginning of the next periodical maintenance cycle.

To calculate churn level critical for stability, I first determine the average amount of maintenance traffic per node and per network half-life from the average number of maintenance messages during this same period. Taking the largest simulated network size of $128k$ nodes and default protocol parameters according to Table 5.2, the number of maintenance messages per network half-life and per node is 115 * 2 = 230 (see Figure 5.7 or Figure 5.8,

note that graphs represent the number of request/reply pairs). Taking an average maintenance message size of 200 bytes [8] maintenance traffic per network half-life will be 230 * 200 = 46 000 bytes. This means that assuming a churn rate characterized by a network half-life of one minute, $1kB/s$ is a conservative upper bound on the average maintenance bandwidth per node, being orders of magnitudes smaller than a typical link capacity.

5.3.3 Comparison with existing overlay maintenance mechanisms

The key idea behind my proposed maintenance mechanism is to let long-range connections created by joining nodes automatically replace missing long-range connections of existing nodes (caused by failure /departure of other nodes). This can be achieved exploiting (i) the flexibility of probabilistic power-law routing overlays in the selection of long-range connections and (ii) the symmetry of outbound and inbound connection distributions as a result of bidirectional connections. Kademlia [12] applies a similar opportunistic maintenance mechanism to create new connections: upon receiving any message (request or reply) from another node, a Kademlia node updates its corresponding k-bucket with the sender's ID and address. The enablers of this maintenance mechanism are also very similar to the enablers of my proposal: symmetric DHT metric space (as a result of the applied XOR metric) and flexibility of long-range connection selection.

Both mechanisms are opportunistic in the sense that most maintenance action is performed as a "side effect" of regular operations in the DHT. However - being related to lookup mechanisms - the impact of Kademlia's opportunistic maintenance depends significantly on lookup strategy. The two main lookup strategies used in DHTs are recursive and iterative lookup. Using recursive lookup, intermediate nodes on the routing path forward the query directly and the originator node has no control over the routing process. Using iterative routing, intermediates nodes only return the address of candidate next hop node(s) and these nodes will be queried directly by the initiator node itself. As a result, the IDs of sender nodes for incoming messages will be uniformly distributed for the iterative lookup strategy and inversely proportional to the distance for the recursive strategy. Since Kademlia uses an iterative lookup strategy, this implies that updates will concentrate on the topmost buckets and the probability of an update decreases exponentially for lower buckets.

Recursive lookups remedy this problem and ensure uniform distribution of updates for all buckets. However, in this case, new routing table entries are created only when new nodes join the network – since the first message sent to a peer node immediately makes the connection "bidirectional". Therefore, my maintenance proposal uses separate opportunistic mechanisms to detect connection failures and to create new connections. Applying recursive lookup strategies, regular lookup traffic helps detecting connection failures while initial long-range connections created by joining nodes help replacing missing long-range connections of

[8] This is a conservative upper bound. A maintenance message contains at most $2N_S$ identifier/locator pairs. With a typical ID size of 128 bit, IPv6 addresses as locators and a typical $N_S = 3$ settings, this is 192 bytes. However, most messages contain only one ID/locator pair or only a message type field.

other nodes.

Accordion [15] also exploits the recursive lookup strategy to acquire (at low cost) additional routing table entries. When an Accordion node forwards a lookup request, the immediate next-hop node returns an acknowledgment which includes a set of entries from its own routing table. These entries are chosen from between its own ID and the target ID of the lookup request. As a consequence of recursive lookup, this provides new long-range connections for the forwarding node with an inverse power-law distance distribution. In contrast to Kademlia and my proposal, this mechanism is not "fully opportunistic", since it requires an additional acknowledge message whose size depends on the number of returned routing table entries. Furthermore, there are no guarantees that the acquired new long-range connections point to live nodes, since they are simply copied from the routing table of another node – which might contain outdated entries (though Accordion also includes timestamps into returned routing table entries, providing heuristics to filter out these items). These differences are a consequence of different goals: Accordion does not aim to minimize maintenance overhead under high churn but to provide optimal lookup performance given a per node bandwidth budget.

5.4 Summary

P2P systems are inherently dynamic; peers join, leave and rejoin the network frequently. To cope with this dynamism, DHT design has to consider the following key architectural issues: static resilience of the overlay, efficient overlay maintenance mechanisms and churn-tolerant lookup strategies. In this chapter I focused on overlay maintenance mechanisms and proposed an overlay maintenance strategy that combines bidirectional probabilistic power-law routing overlays and a novel stochastic long-range connection maintenance algorithm.

The key idea of the proposed strategy is opportunistic maintenance which exploits initial long-range connections created by joining nodes: most missing long-range connections (caused by departure/failure of nodes) can be replaced statistically as a "side effect" of new long-range connections created by joining nodes. This is achieved by exploiting (i) the flexibility of probabilistic power-law routing overlays in the selection of long-range connections and (ii) the symmetry of outbound and inbound connection distributions as a result of bidirectional connections. I have referred to this phenomenon as a *"self-healing" property of probabilistic power-law routing overlays with bidirectional connection*: I have shown analytically, that in steady states, when node arrival and departure rates are equal, distance distribution of long-range connections remains unchanged without any explicit maintenance solely as a side-effect of new connections established by joining nodes.

The other key idea of the proposed maintenance strategy is *stochastic long-range connection maintenance*. As opposed to per-connection maintenance, stochastic long-range connection maintenance only considers the distance distribution of long-range connections. Stochastic maintenance is triggered whenever a new (incoming) long-range connection or a

failed connection is detected. However, maintenance action is taken only when the estimated long-range connection density parameter of this distribution exceeds a predefined lower or upper threshold at the given node.

Exploiting the self-healing property of probabilistic power-law routing overlays with bidirectional connections, the proposed stochastic long-range connection maintenance algorithm basically "lets the system maintain itself". The scope of stochastic maintenance is limited to avoid extreme node degrees and to create missing long-range connections when node arrival rate is smaller than departure/failure rate.

Finally, I have shown both analytically and by extensive simulations that the maintenance overhead of the proposed maintenance strategy is asymptotically optimal.

Chapter 6

Conclusions

The success of the peer-to-peer networking concept reshaped completely traffic mix of the Internet in the past 10 years. However, due to the inherent complexity of distributed algorithms and the inherent dynamism of P2P networks, there are still several open research issues in P2P networking. In my dissertation, I tackled two of these issues, both of them are related to the routing overlay of distributed hash tables.

Although most DHT overlays are structurally similar to the "small-world" navigation model of Kleinberg – architectural and algorithmic details of different DHT variants differ significantly. Furthermore, lookup performance depends on a sets of different and often incompatible parameters which makes analytical comparison rather difficult. In Chapter 4, I proposed a general analytical model that can be used to investigate and compare static routing performance performance of most DHT implementations as a function of their overlay structure.

To capture the above mentioned common foundations of overlay structure, I have introduced the concept of logarithmically transformed view, where distances between a reference node and other nodes are represented after a logarithmic transformation. I have shown that long-range peers of a node form a linear sequence in this transformed view for most DHTs. Furthermore, I have identified an important subclass of DHT overlays – regular power-law routing overlays – where this sequence can be described as a random sample from an infinite renewal process. Based on this stochastic model, I have introduced the λ long-range connection density and c_v long-range connection density coefficient of variation parameters. For $O(\log n)$ node state DHTs, these parameters characterize long-range connection distribution independent of network size.

Using the renewal process model of long-connections, I have analyzed stochastically the progress of the lookup process via long-range connections. I have shown that the sequence of intermediate forwarding nodes in the transformed view of the target node can be also described as a renewal process. Finally, from this renewal process model of the routing process, I have derived closed form upper bounds on the expected number of routing hops as a function of network size and the overlay parameters λ, c_v and N_S.

The above model and results can be applied directly to any DHT using probabilistic power-law routing overlays. Additionally, overlay structure and static routing performance of any DHT using a one-dimensional metric space and being structurally similar to the "small-world" navigation model of Kleinberg can be approximated applying this model.

P2P systems are inherently dynamic; peers join, leave and rejoin the network frequently. To cope with this dynamism, DHT design has to consider the following key architectural issues: static resilience of the overlay, efficient overlay maintenance mechanisms and churn-tolerant lookup strategies. In Chapter 5, I focused on overlay maintenance mechanisms and proposed an overlay maintenance strategy that combines bidirectional probabilistic power-law routing overlays and a novel stochastic long-range connection maintenance algorithm.

The key idea of the proposed strategy is opportunistic maintenance which exploits initial long-range connections created by joining nodes: most missing long-range connections (caused by departure/failure of nodes) can be replaced statistically as a "side effect" of new long-range connections created by joining nodes. This is achieved by exploiting (i) the flexibility of probabilistic power-law routing overlays in the selection of long-range connections and (ii) the symmetry of outbound and inbound connection distributions as a result of bidirectional connections. I have referred to this phenomenon as a "self-healing" property of probabilistic power-law routing overlays with bidirectional connection: I have shown analytically, that in steady states, when node arrival and departure rates are equal, distance distribution of long-range connections remains unchanged without any explicit maintenance solely as a side-effect of new connections established by joining nodes.

The other key idea of the proposed maintenance strategy is stochastic long-range connection maintenance. As opposed to per-connection maintenance, stochastic long-range connection maintenance only considers the distance distribution of long-range connections. Stochastic maintenance is triggered whenever a new (incoming) long-range connection or a failed connection is detected. However, maintenance action is taken only when the estimated long-range connection density parameter of this distribution exceeds a predefined lower or upper threshold at the given node.

Exploiting the self-healing property of probabilistic power-law routing overlays with bidirectional connections, the proposed stochastic long-range connection maintenance algorithm basically "lets the system maintain itself". The scope of stochastic maintenance is limited to avoid extreme node degrees and to create missing long-range connections when node arrival rate is smaller than departure/failure rate.

Finally, I have shown both analytically and by extensive simulations that the maintenance overhead of the proposed maintenance strategy is asymptotically optimal.

Applicability of new results

New results in Chapter 4 are mainly theoretical, helping to understand better the common foundations of routing in distributed hash tables. In addition, the proposed generic stochastic model can also be used to compare analytically the routing performance of different DHT

implementations. Routing performance depends on different overlay parameters which are usually incompatible across different DHT families. In the proposed stochastic model, these incompatible parameters can be translated into a common set of overlay parameters $\{\lambda, c_v, N_S\}$. Using this common parameter set, static routing performance of different DHTs can be compared analytically applying the proposed upper bounds on the expected number of routing hops. Finally, these closed form upper bounds can also be used as an input to derive bounds on other DHT performance metrics; e.g., lookup latencies (in both static networks and under a given level of churn) applying the analytical framework presented in [20].

New results in Chapter 5 are more application oriented. Since dynamism and churn are inherent properties of most peer-to-peer system, efficient maintenance mechanisms are critical to provide good lookup performance in DHTs. Combining the proposed novel stochastic maintenance mechanism with other churn-tolerant architectural components (e.g., churn tolerant lookup strategies, fine tuning of timeout handling), it is possible to further decrease overlay maintenance overhead in DHTs while preserving high availability and good lookup performance. To demonstrate the applicability of stochastic maintenance, I have implemented a DHT (in a simulation environment) which uses the proposed stochastic maintenance component. Based on analysis of extensive simulations, I have also proposed protocol parameter settings for this DHT implementation in order to achieve the best tradeoff between maintenance overhead, lookup performance and availability.

References

[1] S. Guha, N. Daswani, and R. Jain, "An experimental study of the Skype peer-to-peer VoIP system," in *Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, (Santa Barbara, CA, USA), 2006.

[2] A. Sentinelli, G. Marfia, M. Gerla, S. Tewari, and L. Kleinrock, "Will IPTV ride the peer-to-peer stream?," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 86–93, 2007.

[3] H. Schulze and K. Mochalski, "Internet study 2007," tech. rep., Ipoque, 2007.

[4] K. Wehrle and R. Steinmetz, *P2P Systems and Applications.* LNCS 3485, Springer, 2005.

[5] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proceedings of the 1st International Conference on Peer-to-Peer Computing (P2P'01)*, (Linköping, Sweden), pp. 99–100, 2001.

[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th International Conference on Supercomputing (ICS '02)*, (New York, NY, USA), pp. 84–95, 2002.

[7] J. Aspnes and G. Shah, "Skip graphs," *ACM Transactions on Algorithms*, vol. 3, no. 4, 2007.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: Scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM SIGCOMM*, (San Diego, CA, USA), pp. 149–160, 2001.

[9] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms*, LNCS 2218, (Heidelberg, Germany), pp. 329 – 350, Springer, 2001.

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of ACM SIGCOMM*, (San Diego, CA, USA), pp. 161–172, 2001.

[11] F. Kaashoek and D. Karger, "Koorde: A simple degree-optimal distributed hash table," in *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, LNCS 2735, (Berkeley CA, USA), pp. 98–107, 2003.

[12] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, LNCS 2429, (Cambridge, MA, USA), pp. 53–65, 2002.

[13] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: a scalable and dynamic emulation of the butterfly," in *Proceedings of the 21th Annual Symposium on Principles of Distributed Computing (PODC '02)*, (Monterey, CA, USA), pp. 183–192, 2002.

[14] G. S. Manku, M. Bawa, and P. Raghavan, "Symphony: Distributed hashing in a small world," in *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems*, (Seattle, WA, USA), pp. 127–140, 2003.

[15] J. Li, J. Stribling, R. Morris, and M. F. Kaashoek, "Bandwidth-efficient management of DHT routing tables," in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, (Boston, MA, USA), pp. 99–114, 2005.

[16] T. Locher, S. Schmid, and R. Watternhofer, "eQuus: A provably robust and locality-aware peer-to-peer system," in *Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing*, (Cambridge, UK), pp. 3–11, 2006.

[17] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling churn in a DHT," Tech. Rep. UCB/CSD-3-1299, UC Berkeley, Computer Science Division, UC Berkeley, USA, Dec. 2003.

[18] J. M. Kleinberg, "The small-world phenomenon: an algorithmic perspective," in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, (Portland, OR, USA), pp. 163–170, 2000.

[19] J. S. Kong, J. S. A. Bridgewater, and V. P. Roychowdhury, "A general framework for scalability and performance analysis of DHT routing systems," in *Proceedings of the International Conference on Dependable Systems and Networks (DSN'06)*, (Philadelphia, PA, USA), pp. 343–354, 2006.

[20] D. Wu, Y. Tian, and K.-W. Ng, "Analytical study on improving DHT lookup performance under churn," in *Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing*, (Cambridge, UK), pp. 249–258, IEEE, 2006.

[21] K. Aberer, L. O. Alima, A. Ghodsi, S. Girdzijauskas, S. Haridi, and M. Hauswirth, "The essence of P2P: A reference architecture for overlay networks," in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing*, (Konstanz, Germany), pp. 11–20, 2005.

[22] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web," in *Proceedings of the 29th annual ACM symposium on theory of computing (STOC '97)*, (El Paso, TX, USA), pp. 654–663, 1997.

[23] J. Aspnes, Z. Diamadi, and G. Shah., "Fault tolerant routing in peer-to-peer systems," in *Proceedings of the 21st Annual Symposium on Principles of Distributed Computing (PODC '02)*, (Monterey, CA, USA), pp. 223–232, 2002.

[24] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, Oct. 1999.

[25] K. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proceedings of ACM Sigcomm*, (Karlsruhe, Germany), pp. 381–394, 2003.

[26] J. Li, J. Stribling, R. Morris, M. F. Kaashoek, and T. M. Gil, "A performance vs. cost framework for evaluating DHT design tradeoffs under churn," in *Proceedings of INFOCOM '05*, (Cambridge, MA, USA), pp. 225–236, IEEE, Mar. 2005.

[27] J. Xu, A. Kumar, and X. Yu, "On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 151–163, Jan. 2004.

[28] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," in *Proceedings of ACM SIGCOMM'03*, (Karlsruhe, Germany), pp. 395–406, 2003.

[29] F. Beichelt and L. P. Fatti, *Stochastic Processes and Their Applications*. CRC Press, 2001.

[30] G. F. Lawler, *Introduction to Stochastic Processes*. CRC Press, 2006.

[31] L. Kleinrock, *Queueing Systems*, vol. 1. Wiley, 1975.

[32] K. R. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications*. San Diego, CA, USA: Academic Press, 1990.

[33] "Harmonic number definition." `http://mathworld.wolfram.com/HarmonicNumber.html`.

[34] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," *Theory of Computing Systems*, vol. 32, Feb. 1999.

[35] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of peer-to-peer networks," in *Proceedings of the 21st Annual Symposium on Principles of Distributed Computing (PODC '02)*, (Monterey, CA, USA), pp. 233–242, 2002.

[36] M. Jelasity and O. Babaoglu, "T-Man: Fast gossip-based construction of large-scale overlay topologies," Tech. Rep. UBLCS-2004-7, University of Bologna, Department of Computer Science, Bologna, Italy, May 2004.

[37] A. Montresor, M. Jelasity, and O. Babaoglu, "Chord on demand," in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing*, (Konstanz, Germany), pp. 87–94, 2005.

[38] A. Shaker and D. S. Reeves, "Self-stabilizing structured ring topology P2P systems," in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing*, (Konstanz, Germany), pp. 39–46, 2005.

[39] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC '06)*, (Rio de Janeiro, Brazil), pp. 189–202, 2006.

Publications

[B] **BOOK CHAPTERS**

[B1] **Péter Kersch**, Róbert Szabó. Analysis of DHT routing. Invited chapter in the *Handbook of Peer-to-Peer Networking*, Springer, 2008. (submitted)

[J] **JOURNALS**

[J1] **Péter Kersch**, Róbert Szabó. A graph theoretical lower bound on maintenance overhead of structured P2P overlays. *PIK, Vol. 31. No. 1, pp 24–28, special issue on Modeling of Self-Organizing Systems*, March. 2008. DOI: 10.1515/piko.2008.005

[J2] **Péter Kersch**, Róbert Szabó. DHT routing analysis in a logarithmically transformed space. *Peer-to-Peer Networking and Applications, Vol 1. No. 1, pp 64–74*, Springer, March 2008. DOI: 10.1007/s12083-007-0002-2

[J3] **Péter Kersch**, Róbert Szabó, Lawrence Cheng, Kerry Jean, Alex Galis. Stochastic maintenance of overlays in structured P2P systems. *Elsevier Computer Communications, Vol 31/3 pp 603–619, special issue on Disruptive Networking with Peer-to-peer Systems*, Febr. 2008. DOI: 10.1016/j.comcom.2007.08.017

[J4] Kis Zoltán Lajos, Kovácsházi Zsolt, **Kersch Péter**, Simon Csaba. Mobil többesadás protokollok vizsgálata IPv6 hálózatokban. *Híradástechnika*, Vol. LIX. pp. 20–25, March 2004.

[J5] **Kersch Péter**, Vajda Lóránt, Török Attila. IP mikromobilitási protokollok ad hoc kiterjesztése. *Híradástechnika*, Vol. LVIII. pp. 14–19, Apr. 2003.

[J6] **Kersch Péter**, Kürthy Lóránt, Simon Csaba, Vajda Lóránt. IP mikromobilitási protokollok ad hoc kiterjesztésének tesztelése. *Híradástechnika*, Vol. LVIII. pp. 20–28, Apr. 2003.

[C] **CONFERENCES**

[C1] Lawrence Cheng, Roel Ocampo, Kerry Jean, Alex Galis, Zhaohong Lai, Csaba Simon, Robert Szabo, **Peter Kersch**, Raffaele Giaffreda. Distributed Hash Tables Composition in Ambient Networks. In *Proceedings of IEEE DSOM'06,* 17^{th} *IFIP/IEEE*

International Workshop on Distributed Systems, pp.258–268, Dublin, Ireland, October 23–25, 2006 DOI: 10.1007/11907466

[C2] **Péter Kersch**, Zoltán Lajos Kis, Róbert Szabó. Self Organizing Ambient Control Space - An Ambient Network Architecture for Dynamic Network Interconnection. In *Proceedings of the* 1^{st} *International ACM Workshop on Dynamic Interconnection of Networks*, pp.17–21, Cologne, Germany, September 2^{nd} 2005. DOI: 10.1145/1080776.1080782

[C3] Róbert Szabó, **Péter Kersch**, Balázs Kovács, Csaba Simon, Márk Erdei, Ambrus Wagner. Dynamic Network Composition for Ambient Networks: a Management View. In *Proceedings of Eurescom Summit 2005*, pp.35–42, Heidelberg, Germany, April 27–29 2005.

[C4] Csaba Simon, Rolland Vida, **Péter Kersch**, Christophe Janneteau, Gösta Leijonhufvud. Seamless IP Multicast Handovers in OverDRiVE. In *Proceedings of IST Mobile and Wireless Communications Summit*, Lyon, France, June 27–30 2004.

[C5] Zoltán Lajos Kis, Zsolt Kovácsházi, **Péter Kersch**, Csaba Simon. Adaptation of IPv6 multicast protocols to heterogeneous mobile networks. In *Proceedings of 10th Eunice Summer School and IFIP WG 6.3 Workshop on Advances in fixed and mobile networks*, pp.99-103, Tampere, Finland, June 14–16 2004.

[C6] **Péter Kersch**, Csaba Simon, Lóránt Vajda. Ad Hoc Extension for IP Radio Access Networks. In *Proceedings of TRANSCOM 2003 (*5^{th} *European Conference Of Young Research and Science Workers in Transport and Telecommunications*, pp.191–196, Zilina, Slovakia, June 23-25 2003.