Computer Communications 85 (2016) 41–57

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Evaluation of hierarchical DHTs to mitigate churn effects in mobile networks

Josyl Mariela B. Rocamora*, Jhoanna Rhodette I. Pedrasa

Electrical and Electronics Engineering Institute (EEEI), University of the Philippines, Diliman, Quezon City 1101, Metro Manila, Philippines

ARTICLE INFO

Article history:
Received 28 June 2015
Revised 3 January 2016
Accepted 13 February 2016
Available online 23 February 2016

Keywords:
P2P
HDHT
Churn
Mobile network
OMNeT++

ABSTRACT

Existing flat peer-to-peer (P2P) systems based on distributed hash tables (DHTs) perform unsatisfactorily under churn due to their non-hierarchical topology. These flat DHTs (FDHTs) experience low lookup success ratio, high lookup latency and high bandwidth usage as a consequence of the presence of churn. With this, we explore the use of hierarchical DHT (HDHT), specifically the superpeer design, in mitigating the effects of churn. To the best of our knowledge, we are the first to intensively examine HDHTs with and without high churn through simulations.

Using the OMNeT++ simulator and the OverSim framework, we analyze flat and hierarchical DHTs with and without churn. Results show that the implemented HDHTs perform more satisfactorily than a flat DHT because of better fault isolation and smaller cluster sizes at the cost of higher superpeer traffic. HDHTs are more stable as they have better lookup success ratios. They are more efficient as evidenced by lower lookup latencies and lower average node bandwidth usage. They are more scalable since their performance do not degrade significantly even at high population. With this, the implemented HDHTs can be utilized to alleviate the effects of churn in mobile networks.

1. Introduction

Structured peer-to-peer (P2P) network is one type of overlay network that organizes its nodes in a tightly controlled topology [1,2]. It acts as the routing and location service for P2P systems. Because it uses deterministic content placement based on distributed hash tables (DHTs), this type of network is also known as DHTs. Its advantage over the unstructured P2P network, another type of overlay network, is its efficient item location. However, these DHTs require overlay maintenance traffic to constantly reorganize the topology.

In this research, we focus on the analysis of structured P2P overlays or DHTs under churn, the process of node arrival and departure. This phenomenon occurs when nodes recurrently connect and disconnect from the network causing transient behavior in the system. When present in DHTs, churn induces routing performance degradation such as low lookup success, high latency and high bandwidth consumption (see Sections 2, 4 and 5).

Now that mobile and wireless devices become part of everyday user experience, accessibility to all applications using any machine has been significant. Users expect availability of different applications such as P2P in their devices. Some P2P applications are file sharing, multimedia streaming, Voice over IP (VoIP) and multicasting. In addition, more devices are now and will be connected to the Internet; according to Cisco Systems, “37 billion new things will be connected by 2020” [3].

With the growing mobile population and the increasing user expectation, there exists a need to integrate these mobile devices into the P2P community. These devices will be able to properly utilize P2P applications if the routing and location service, or the overlay network, can withstand the dynamic membership and routing performance degradation caused by churn in mobile networks. Thus, the overlay should be stable, efficient and scalable in order for the mobile users to experience seamless network connectivity for their applications.

Current DHTs employing a flat design have poor performance when faced with the presence of churn. These overlays experience low lookup success, high latency and high bandwidth consumption. Thus, a flat DHT is not designed for dynamic networks. Some DHTs have a hierarchical topology which has fault isolation due to multiple independent overlays. Such hierarchical design offers good stability, efficiency and scalability. However, to the best of our knowledge, hierarchical DHTs have not been examined intensively under churn through simulations.

* Corresponding author. Tel.: +639065681224.
E-mail addresses: josylmariela@gmail.com, josylmariela.rocamora.ph@ieee.org (J.M.B. Rocamora), jipedrasa@upd.edu.ph (J.R.I. Pedrasa).

http://dx.doi.org/10.1016/j.comcom.2016.02.003
0140-3664/© 2016 Elsevier B.V. All rights reserved.

With this, our research aims to investigate the performance of different DHTs, flat and hierarchical, against the effects of churn. Scrutinizing the impact of churn would assist in determining whether these DHTs would withstand churn effects in mobile networks. Through simulations, the contributions of our paper are as follows:

- We have evaluated a flat DHT (FDHT or simply DHT) and a hierarchical DHT (HDHT) with or without the presence of churn under several simulation scenarios. This evaluation determines which design offers better performance when dynamic membership is prevalent.
- We have evaluated two different hierarchical DHTs with or without churn under several simulation scenarios. This determines whether two different HDHTs would perform similarly and where they can be implemented.

In the succeeding sections, we present the problem of churn in P2P overlays over wireless or mobile networks. Section 2 expounds on relevant works in peer-to-peer and the challenges of churn. This section shows how FDHTs perform poorly under churn and how HDHTs can be the better alternative in mobile networks. Section 3 describes the simulation setup and procedures for evaluating the effects of churn on the DHT performance and for investigating HDHTs. Sections 4 and 5 assess the performance of the different DHTs with and without churn. Lastly, Section 6 summarizes all the assessments and recommendations with regards to the implemented DHTs.

2. Related work

This section discusses the concept of churn and the current trend in FDHTs under churn to show the infeasibility of flat topologies under dynamic environments. Recent works on hierarchical P2P systems are also presented to establish their merits over flat P2P systems.

2.1. Overview of peer-to-peer

A P2P network is an overlay network formed on top of the Internet Protocol (IP) network and is independent of the underlying physical network. The overlay network topology and its routing and location mechanism are crucial to the operation of a P2P content distribution system [1], since items must first be located before any file exchange occurs. These two may affect several system attributes including fault tolerance, performance, availability and scalability.

P2P overlay networks can be categorized according to overlay network organization (i.e. how the overlay is constructed) [1,2]. These overlays can be categorized either as structured or unstructured. On one hand, structured networks have tightly controlled topology and deterministic content placement based on DHT. The node identifier space in a structured network defines the node topology, that is where each node is located relative to another node. On the other hand, unstructured overlays have no prior knowledge on network topology and use flooding mechanisms. Some examples of structured overlays are Chord, Pastry, Kelips, Tapestry and Kademlia; while some unstructured overlays are Freenet, Gnutella, FastTrack and BitTorrent.

An advantage of structured over unstructured P2P networks is its efficient item location. Due to the controlled topology mapping of structured networks, peers can easily and efficiently locate items based on file pointers. Unstructured systems, on the other hand, use inherently wasteful flooding to route queries and to trace data location. However, a disadvantage of structured overlays is the need for maintenance of the controlled topology [1].

Table 1
Churn scenarios.

	High churn	Low churn
Physical failure	Fluctuating wireless signal Battery saving mechanisms	Machine reboot Power shortage Power surge
User behavior	Downloading short files Streaming short video clips	Downloading/Uploading torrent Facebook-ing on computers Chatting for hours
Mobility	Cell phone on a moving car Cell phone on a train VANETs	MANETs Cell phone on a moving person

Table 2
P2P file-sharing traces.

Traces	Approximate mean session time
Gnutella trace [20]	2 h
Microsoft trace [21]	6 d
Overnet trace [22]	2 h
Skype trace [23]	1 h

Structured overlays, or sometimes called as DHTs, can be further classified based on overlay node hierarchy, namely FDHTs and HDHTs. FDHTs are simple in terms of construction since nodes are organized only in one level or layer. Examples of FDHTs are Chord [4], Pastry [5], Kelips [6], Tapestry [7] and Kademlia [8]. HDHTs allow peer heterogeneity and localization because nodes may be clustered into different levels or layers. Examples of HDHTs are Chordella [9], Brocade [10], Garces-Erice design [11], HIERAS [12], C-Chord [13], Canon [14], Cyclone [15], and Coral [16].

2.2. Overview of churn

Churn is the process of node arrival and departure. Common causes of this phenomenon are physical failure, user behavior and node mobility. Physical properties, such as channel unreliability and machine failure, trigger network disconnection. Behavioral properties (e.g. how long a user is online or offline, how a user acts) can also trigger disconnection and reconnection. Lastly, nodes moving away or towards radio coverage can be disconnected or connected to the network. Examples on mobility may be related to either physical failure due to limitation of radio coverage or user behavior due to node movement. Table 1 provides a list of scenarios based on the causes of churn.

Depending on the cause, nodes may leave the network gracefully or not. Graceful departure occurs when a peer performs the necessary leave procedure by announcing departure and transferring data to neighbor peers. In contrast, ungraceful departure occurs when a peer leaves the network abruptly. Data and queries for the departed peer are lost. Neighbor peers are only updated about the leave during maintenance.

Churn is typically modeled in terms of session times (i.e. connection duration) or in terms of churn rates (i.e. node join/leave frequency) [17–19]. Low churn rate, or long session time, is in the order of hours or days. High churn rate, or short session time, is in the order of seconds or minutes.

Session times in wireless networks are considered to be high churn. As opposed to the 1–2-h churn rates in wired networks, wireless networks have 120–600-s churn rates [18]. Furthermore, churn rate for mobile devices is modelled as short online/session times [17,19].

Table 2 lists different real world file-sharing applications and the measured session times when users are connected via wired

Table 3
Wireless network traces.

Traces	Session time	Connection type
Kim et al. [24]	≈ 8–9 min	WiMAX
McNett and Voelker [25]	6.25 min (median)	Campus network
Papadopouli et al. [26]	9, 18, 34 min (median)	Campus network

Table 4
Performance degradation of FDHTs under high churn.

Reference	Decreased lookup success	Increased lookup latency	Increased bandwidth usage
Rhea et al. [30]	Yes	Yes	Yes
Herrera and Znati [34]	Yes	Yes	Yes
Chan et al. [17]	Yes	Yes	N/A
Hsiao and King [32]	N/A	Yes	Yes
Ou et al. [31]	Yes	N/A	No
Price et al. [35]	Yes	N/A	N/A
Hoang et al. [18]	Yes	No	N/A

media. This table shows that these file-sharing systems experience churn despite being connected via a stable connection.

Table 3 shows various datasets collected from laptops and wireless clients over mobile and wireless networks. These traces show that wireless networks are in fact high churn environments. WiMAX links are inherently unstable due to signal strength variation and handovers [24]. While in wireless LANs comprised of multiple access points (AP), session times are often short due to mobility-induced handoffs. With these, P2P throughput and routing performance suffer due to instability in mobile networks.

With the problem of high churn in mobile networks, the question of whether resource-constrained mobile devices may be able to participate in P2P overlays arises. According to [19], prototype measurements show that mobile nodes in a P2P overlay incur reasonable CPU processing (around 20%–25%) and network traffic (around 600 bytes/s). Mobile devices connected to either UMTS or WLANs can operate up to approximately 20 h without any running P2P application [27]. However, when connected to a DHT via WLANs, mobile device batteries can only support a 4–6-h lifetime [28], which is much longer than typical session times in high churn. Despite the limited resources, mobile devices can participate in a P2P overlay based on battery life, CPU processing and bandwidth usage [29].

2.3. Churn analysis in FDHTs

In this subsection, we discuss the performance of FDHTs and their infeasibility at highly dynamic scenarios. Since DHTs are mainly involved in routing and location, performance is typically analyzed using three metrics: lookup success ratio, lookup latency, and bandwidth consumption. Success ratio indicates the percentage of the total lookup requests that are completed successfully (i.e. data are found). Lookup latency is the delay in time units incurred from request to response. Bandwidth consumption is the amount of traffic consumed either by data or maintenance overhead.

Table 4 displays recent analysis on the effects of high churn to flat DHTs such as Chord, Kademlia, Kelips, Bamboo, Pastry and Tapestry. In each reference, the considered overlay showed a form of performance degradation whether in terms of lower lookup success, longer lookup latency, or greater bandwidth usage when the churn rate is increased. Some entries are labeled *N/A* (not applicable) for cases where no explicit tests are performed.

Table 5
Performance improvement of FDHTs under high churn due to utilized strategies (R for redundancy, TP for topology maintenance, and LS for lookup strategy).

Reference	Utilized strategy	Increases lookup success	Decreases lookup latency	Decreases bandwidth usage
Ou et al. [36]	R and LS	Yes	No	No
Herrera-Ruiz and Znati [37]	R and TM	Yes	No	No
Hsiao and King [32]	TM	Yes	Yes	No
Hoang et al. [18]	TM	Yes	No	No
Kim et al. [38]	TM	No	Yes	No
Price et al. [35]	TM	Yes	No	No

Success ratio decreases to around 20% at short session times due to relatively long waiting time [30]. On the other hand, success ratio decreases by 20% when high churn is employed due to data item out of place [31]. Due to network partitions, success ratios significantly drop down to about 10% when the session time is about 120 s [18].

Latency increases from about 1–5 s when churn rate is increased [30]. Similar to latency, the percentage of hop counts due to timeouts increase when shorter session time is employed [32]. Bandwidth usage intensifies depending on the maintenance algorithm [30] and on traffic caused by timeouts [32].

In some references, the considered overlay showed no decline in a certain performance metric. According to [31], bandwidth usage does not increase even when peer session times are shortened. At low churn rates, bandwidth is consumed by the maintenance traffic of the large overlay. At high churn rates, bandwidth is used mostly by node join/leave events. So for both cases, the bandwidth consumptions are almost the same. Lastly in [18], latency in terms of hop counts slightly decreases at high churn rates. However, hop counts measured in this work are only of successful lookups. Neglecting failed lookups may not properly represent the system's performance with respect to lookup latency.

The research works in Table 4 indicate that FDHTs perform poorly under high churn, leading to decreased lookup success ratio, increased lookup latency, and increased bandwidth usage. Since high churn is problematic, other research works focus on modifying DHT parameters to mitigate the churn effects.

Typical modifications to FDHTs are redundancy, topology maintenance, and lookup strategy. In redundancy, data and other information are replicated in multiple nodes for faster searching and better availability. A Kademlia implementation with data replication has improved success rates due to better availability of data [33]. In topology maintenance, routing tables and Keep Alive messages are exchanged more frequently to update or stabilize the network topology. In lookup strategy, overlay may use different routing styles such as the use of iterative/recursive routing, lookup parallelism or acknowledgements to ensure delivery. These techniques can be combined and are used to modify existing parameters of flat DHTs. For example, a redundancy scheme may be combined with a maintenance strategy.

Table 5 displays recent analysis on the effects of high churn to modified flat DHTs. In each reference, the considered overlay showed some performance improvement but at additional costs such as maintenance overhead and storage. In redundancy, success ratio or lookup latency improves since data become more likely available and lookup requests are less likely to timeout due to replication. However, more storage space and more bandwidth are required for data replication. In topology maintenance, success ratio or lookup latency can improve since routing tables are more likely updated and lookup requests are more likely to be routed correctly and quickly. But additional traffic is required to shoulder frequent topology maintenance.

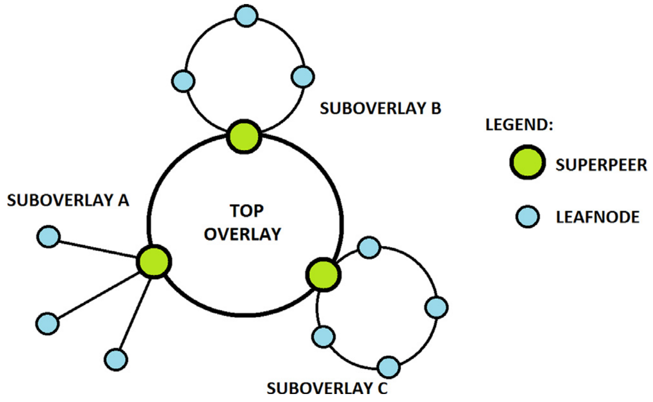


Fig. 1. An example of a superpeer HDHT design.

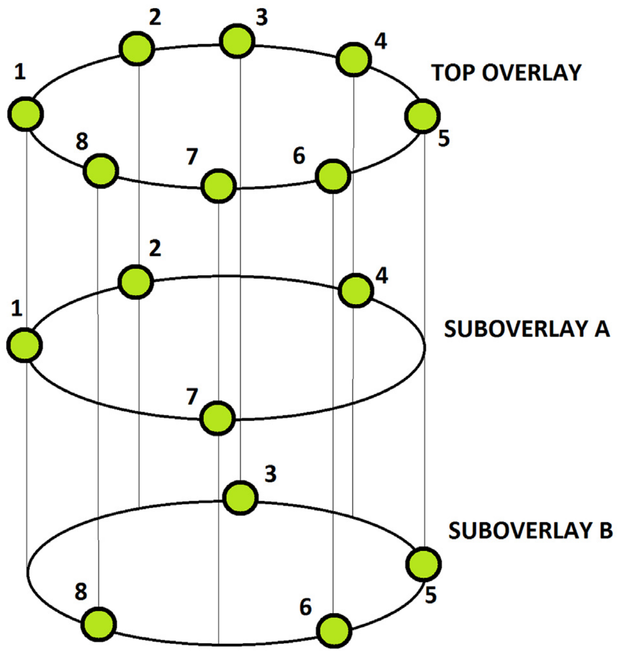


Fig. 2. An example of a homogeneous HDHT design.

Table 6
Existing hierarchical DHTs.

Reference	Type	Number of layers	Compared with FDHT	Considers intensive churn analysis
Chordella [47]	Superpeer	2	Yes	No
Brocade [10]	Superpeer	2	Yes	No
Garces-Erice system [11]	Superpeer	2	Yes	No
Cellular-Chord [13]	Superpeer	2	Yes	No
PAIS [41]	Superpeer	2	Yes	No
MobiStore [42]	Superpeer	2	Yes	No
Tiny Bubbles [43]	Superpeer	>= 2	Yes	No
Pandey system [44]	Superpeer	2	Yes	No
Mizrak system [45]	Superpeer	2	No	No
HIERAS [12]	Homogeneous	>= 2	Yes	No
Canon [14]	Homogeneous	>= 2	Yes	No
Cyclone [15]	Homogeneous	>= 2	No	No
Coral [16]	Homogeneous	3	No	No
PCMS [46]	Homogeneous	2	Yes	No

Considering high churn and graceful/ungraceful departures, the combination of lookup strategy and redundancy increases the success ratios (from about 70% to 80%) at the price of doubling the network traffic (from 2 to 3.5 messages/node/s) [36]. In another work [37], an adaptive redundancy maintenance strategy improves success ratio from 10% to 60% also at the expense of significantly higher bandwidth (from a few kbps to more than 1000 kbps).

With the modifications, performance improvements are only evident in one or two metrics at a certain cost. In some cases, improvement is not even significant. At times, FDHTs still experience poor success ratio, longer lookup latency or higher bandwidth consumption. With this, we infer that FDHTs are intrinsically bad at high churn and that we suggest to investigate HDHTs as an alternative at high churn. By using a hierarchical system, there should be a positive effect on the system performance at no extra costs on the mobile or regular peers but rather on the superpeers.

2.4. Hierarchical DHTs

Hierarchical structured overlays organize their nodes into multiple levels. A hierarchical DHT can be classified depending on the inter-cluster connections and independence either as a *superpeer* or a *homogeneous* design [39].

Fig. 1 shows the superpeer design, which uses gateways to allow inter-cluster communications. One cluster is an almost disjoint set of the whole system and only the gateways are connected to two overlays. These gateways, usually called superpeers, on the top P2P overlay are given greater responsibilities. Other peers are location on sub overlays that may employ different DHTs.

Fig. 2 shows the homogeneous design that has no gateway peers. One peer can communicate to any other peer via the most optimal overlay. Two peers may belong to the same overlay in one level but may belong to different sub-overlays in another level. However, this design may not be practical; if mobile devices maintain several sub-overlays, battery may be consumed more quickly and contention of network might exist. With this, we have chosen the superpeer design in our analysis for P2P over mobile networks.

The hierarchical structure imposes several advantages due to the use of independent clusters. Considering that flat and hierarchical DHTs over mobile networks have the same total population, HDHTs are better in terms of stability, efficiency and scalability because of several localized sub-overlays.

Mathematical computations show that cluster-based structured overlays require O(N) join/leave events before any significant topological changes to occur (i.e. before a *split* or *merge* operation is prompted) [40]. HDHTs are stable because of robustness to dynamic events and are scalable because they can accommodate population growth due to small independent overlays.

A smaller network outperforms a larger overlay due to reduced complexity. A less populated network achieves higher lookup success [17,18], and lower lookup latency [17] or less number of hop counts [4]. With this, HDHTs are efficient because they have small independent overlays that have good performance.

Table 6 lists hierarchical DHT design with two or more levels. Chordella, Brocade, Garces-Erice system, Cellular-Chord, PAIS, MobiStore, Tiny Bubbles, Pandey system and Mizrak system are all superpeer HDHT designs. HIERAS, Canon, PCMS, Cyclone, and Coral are all homogeneous HDHT designs, wherein nodes participate in two or more overlays and a flat structure is configured at the top level.

Chordella is a superpeer design that focuses on optimizing the traffic in the whole overlay [9], wherein the top overlay is Chord and the sub overlays are point-to-multipoint networks. This design exhibits lower bandwidth consumption and lower latency compared to Chord, at varying total population and at no churn [9]. Also, it has better success ratios than Chord at varying node lifetime. However, analysis only considered the effects of churn on Chordella and Chord in terms of success ratios.

Another superpeer design is Brocade [10], which uses independent Tapestry systems for both the top overlay and the sub overlays. Brocade has slightly better latency and bandwidth consumption than Tapestry, a flat DHT system. Unlike Chordella, the analysis on Brocade does not consider churn.

The superpeer design proposed by Garces-Erice et al. [11] shows the framework for constructing hierarchical DHTs using Chord as the top overlay. The hierarchical system based on this design has better performance in terms of latency and hop count compared to Chord. However, this hierarchical system is not analyzed under churn.

Cellular-Chord is also a superpeer HDHT that focuses on integrating nodes in the cellular network using a topology-aware mechanism [13]. It uses Chord systems for both the top overlay and the sub overlays. It is compared to Chord based on hop counts at varying total population and traffic consumption at long simulation time. Similar to other HDHTs, Cellular-Chord is not analyzed under churn.

Some superpeer systems are specifically designed for certain applications. Proximity-aware and interest-clustered P2P file sharing system (PAIS) groups peers according to interests and proximity [41] and is based on the Cycloid flat DHT. MobiStore is a data store P2P system that utilizes clustering and redundancy [42]. Tiny Bubbles is an overlay network for military scenarios experiencing block churn [43]. Despite its aim to address block churn, the research work do not show any data analyzing the system under churn. Pandey system is a content-based publish/subscribe over a two-tier DHT that utilizes domain ontology of Association for Computing Machinery (ACM) [44].

Mizrak system considers two-tier superpeer structure which focuses on load balancing [45]. All peers are connected in one single Chord overlay while the superpeers are on another Chord tier. Each superpeer is then assigned to a subset of the larger overlay and must know how to directly contact each peer on that subset. All messages, even if the destination is intended on the same subset, must be forwarded to the superpeer. This setup is highly dependent on the operation of the superpeer. The simulation results focused on load balancing, such as how many peers were assigned to each superpeer or how much traffic is handled by each superpeer.

HIERAS is a homogeneous HDHT using multiple ring layers in order to improve scalability and routing performance [12]. In this system, routing requests are initially executed at lower rings before escalated to the upper rings. HIERAS is compared to Chord at varying total population and at varying underlying topology model (e.g. TS, BRITE, INET) but is not examined under varying node lifetimes.

Another homogeneous design is Canon, which constructs a hierarchical structure from almost any flat DHT overlay while maintaining flat design functionality and load homogeneity [14]. Similar to HIERAS, Canon is compared to Chord at varying total population but is not analyzed under churn.

Two other homogeneous designs are Cyclone [15] and Coral [16]. Cyclone is used for creating a system with optimal logarithmic routing hops but without establishing unnecessary links. Coral allows peers to store pointers pertaining to other peers and to form various clusters using round trip times. However, the two systems are not analyzed intensively since both are not studied under churn and are not even compared to any flat DHT.

P2P community management system (PCMS) uses autonomous DHT overlay for each community based on existing flat DHT overlays [46]. It proves to be better compared to Chord and Kademlia but churn analysis of the system only involved varying of the node lifetime rather than consider other variables of the system. In addition, the system utilized a homogeneous design which would not be practical for mobile devices.

Based on the table, the effects of churn in a HDHT have not been analyzed intensively. Some studies did not consider churn in their performance evaluation, while some have limited their analysis only on one or two performance metrics and only on few variables. Three research works did not even compare their simulation results with respect to an FDHT. With this, we propose to intensively analyze the effects of high churn in HDHTs with respect to an FDHT by evaluating three performance metrics (lookup success, lookup latency and bandwidth usage) at various operating conditions. These three performance metrics are utilized by several researches as shown in Tables 4 and 5. Section 3.3 discusses these three performance metrics and the evaluation method.

2.5. Summary

Existing flat DHTs do not perform well in highly dynamic scenarios. They suffer increased latency, increased bandwidth consumption and decreased lookup success ratio, which should be avoided in an efficient routing setup. Even with some modifications, they still perform badly or need extra resources for meager improvement. Therefore, we reason that FDHTs are not suitable at high churn, as a consequence of their design. A better alternative is an HDHT, due to several advantages and better performance. HDHTs, specifically the superpeer design, can be used to mitigate the effects of churn in wireless networks.

3. Research design

In this section, we introduce the different simulation configurations and evaluation methods utilized. This research aims to investigate the performance of different DHTs, flat and hierarchical, when subjected to high churn. Evaluating the impact of churn to these DHTs would determine whether HDHTs perform better than FDHTs. We have installed OMNeT++ version 4.2.2, OverSim-20121206 and inet version 20111118 on an Ubuntu Linux 12.04 machine.

3.1. Overlay configurations

In order to analyze DHT performance, we have implemented the DHTs in a discrete event simulator called OMNeT++ [48]. OMNeT++ is chosen for this research specifically because of OverSim, an open-source P2P overlay simulation that has support for various P2P overlays and churn.

The OverSim API follows the Structured P2P API proposed by Dabek et al. [49]. It has four layers/tiers (Tier 2-0, Underlay) to separate different P2P overlay functions. Tier 2 refers to different applications that use DHT functions such as file sharing, name services, gaming and streaming. Tier 1 pertains to DHT functions such as indexing, caching and replication. Tier 0 represents the different key-based routing (KBR) overlays such as Chord, Kademlia and Pastry. Underlay describes the underlying network properties such as physical topology and churn generators.

In the Underlay tier, the *SimpleUnderlay* module of OverSim is utilized as the abstracted underlying physical network composed of two kinds of nodes, namely, mobile nodes, and reliable nodes.

Table 7
Simulation variables (when a variable is fixed, it is set to the value emphasized in bold typeface.)

Simulation variables	Values
Total population (N)	500, 1000, **2000**, 4000, 8000 (nodes)
Maintenance interval (t_m)	**36**, 60, 300, 600 (s)
Reliable population ratio (R_{Reliable})	10%, 25%, **50%**, 75%, 90%
Number of clusters (C)	**10**, 25, 50 (clusters)
Node lifetime ($E[t_{\text{on}}]$)	100,200,**400**,800,1600 (s)

Table 8
Fixed simulation parameters.

Fixed parameters	Values
Repetition	10 runs per scenario
Transition time	1200 s
Measurement time	3600 s
Mean lookup request interval	60 s
Redundancy	4 redundant nodes
Graceful leave delay	5 s
Graceful leave probability	0.5
UDP timeout	5 s

The *NoChurn* and *LifetimeChurn* modules of OverSim model the reliable and mobile nodes, respectively. *NoChurn* generator creates nodes that do not leave the network; while *LifetimeChurn* generator creates nodes that have limited lifetimes.

We have configured three different DHTs for simulation: Chord DHT, Chord–Chord HDHT and Chord–Kademlia HDHT. Chord DHT is chosen to represent FDHTs since it has been widely studied and researched. It is the flat overlay system typically used when comparing flat and hierarchical systems. It is also typically used as the top overlay protocol in several superpeer design.

For HDHTs, the superpeer design is selected because its independent sub overlays are suitable for localizing churn. A framework for hierarchical DHTs identifies clusters to have independently maintained intra-group overlays and the top overlay to have inter-cluster routing [11]. The top overlay used in this study is Chord, similar to several superpeer designs. With this, we define HDHTs with Chord as the top overlay and two different topologies for the sub overlays. We explore Chord and Kademlia as the sub overlays; both of which are supported in OverSim and represent two different topologies, ring-like and mesh-like. Based on our preliminary results, Kademlia shows better performance than Chord.

There are two types of nodes in HDHT, namely, regular peers and superpeers. Regular peers are located in the sub overlays and can either be reliable or mobile. Superpeers are always reliable and have two overlay instances, one for the top overlay and one for the sub overlay. The superpeer functions as a regular peer in the sub overlay instance and it also functions as a gateway for routing inter-cluster communications in the top overlay.

Lookup requests in HDHTs are either routed locally or globally depending on the destination node's location. A peer determines whether the destination is within the same overlay or not by looking into the overlay ID of the destination, which is specified by the Tier 2 application along with the desired key. If the request is local, then the request is routed using the implemented KBR overlay. If the request is global, the request is forwarded to the superpeer of the source node. Then the superpeer locates the superpeer of the destination node, which will determine the address of the destination node. The address is returned back to the source node using the reverse path. Finally, the source node directly communicates with the destination node.

3.2. Experimental setup

Two sets of experiments are simulated in this work. For each set, we test the effects of varying different parameters on the DHT performance under several conditions. The first set evaluates the Chord–Chord HDHT performance compared to Chord FDHT, with and without churn. This set of experiments aims to determine which system has a better performance under churn; a flat system or a hierarchical system. The second set of experiments measures the Chord–Kademlia performance compared to Chord–Chord, both with and without churn. This set discovers whether different HDHTs have similar performance under various scenarios.

Table 7 lists the values of the variables used in the simulation. For each set, a variable is varied one at a time while keeping the other variables fixed. Fixed values are those values in the table emphasized in bold typeface.

Total population (N) indicates the total number of nodes in the simulation. In the case of hierarchical DHTs, the total population is evenly distributed across all the sub overlays. For instance, in a 2000-node 10-cluster system, there are 10 superpeers, 99 reliable nodes per cluster and 100 mobile nodes per cluster.

Maintenance interval (t_m) pertains to the duration between two overlay maintenance procedures of Chord. Maintenance interval values indicated are arbitrarily chosen since values typically used in recent studies for Chord under high churn range from less than 20–120 s [17,18,50–53]. However, these values are not applied to Kademlia as it does not result to any significant performance improvement but results to very high bandwidth and very long simulation time.

Reliable population ratio ($R_{Reliable}$) defines the percentage of reliable nodes in the system. *Number of clusters* (C) sets the number of sub overlays available in HDHT only. Since the total system size is evenly distributed to all the clusters, a higher C means lower cluster size (i.e. less reliable nodes and less mobile nodes perxbrk cluster).

Node lifetime ($E[t_{\text{on}}]$) specifies the mean value of the randomly distributed session times of mobile peers. Session times are based on the Weibull distribution with shape factor equal to 0.40. The distribution and shape factor are selected by comparing the histogram of the generated session times to that of some P2P applications [54] and P2P IPTV [55].

The two variables, R_{Reliable} and $E[t_{\text{on}}]$, only affect DHTs under churn since stable DHTs do not have mobile peers.

Table 8 lists the fixed simulation parameters. The values are selected based on preliminary simulations.

Repetition pertains to the number of runs simulated per scenario.[1] *Transition time* is the duration provided as lead time for the network to converge; while the *Measurement time* is the duration provided for gathering data. When the transition time is equal to 1200 s, most peers have joined the overlay after the time lapses, hence the network is no longer in transient. We have selected measurement time to 3600 s, similar to [30,56,57].

Mean lookup request interval dictates the average value of the randomly distributed DHT GET request intervals. Modifying this value does not affect either lookup latency or lookup success ratio but only varies the bandwidth usage due to application traffic. *Redundancy* specifies the number of nodes that is assigned to store a particular DHT entry and the number of nodes that is queried for lookup requests. Varying this value does not affect the success ratios and bandwidth consumptions but only affects the latencies.

Graceful leave delay is the duration between the instant a node performs a leave procedure and the instant that node departs from the network. *Graceful leave probability* dictates the likelihood of a

[1] In one scenario, we consider only one value per variable.

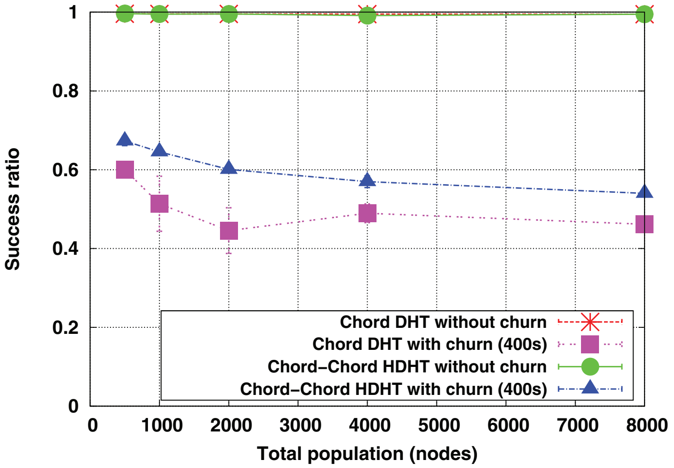


Fig. 3. *Success Ratio* at varying *Total Population* ($t_m = 36$ s, $R_{\text{Reliable}} = 50\%$, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s). *Note:* Chord DHT and Chord–Chord HDHT without churn plots overlap at 100% success ratio.

node departing from the network with or without proper leave procedure. Varying either of the two has no significant effect on the success ratios and latencies.

UDP Timeout defines the time-to-leave of all UDP Remote Procedural Calls (RPC) such as overlay messages, DHT requests. When a response to an RPC is not received, the message is assumed to be lost and the RPC timeouts. This value affects the success ratio in the sense that a request is automatically set to fail if it exceeds the timeout. However, changing this value by a few seconds does not significantly affect the simulation results. Even if we increase the timeout, destination nodes which have left the system will no longer be reachable, thus requests to these nodes would result to failure. Changing this value has no bandwidth consumptions but affects the average lookup latencies.

3.3. Evaluation method

For each experiment, a DHT system is analyzed based on three performance characteristics, namely stability, efficiency, and scalability. To quantify the characteristics even further, we measure three metrics which are lookup success, lookup latency and bandwidth consumption.[2]

Stability is the resilience of the system in the presence of churn and can be measured by the lookup success metric. A stable system does not fail catastrophically in the face of dynamic membership. Each node in a DHT computes for its lookup success ratio which is equal to the percent of total completed lookups that is performed correctly and successfully.[3] Eq. (1) shows the formula for lookup success ratio.

$$\text{Lookup success} = \frac{\text{Number of of correct and complete lookups}}{\text{Total number of completed lookups}} \tag{1}$$

Based on the *DHTTestApp* module in OverSim, lookup requests are successful if the returned value is the same as the value stored in the destination node. However, lookup requests may fail due to a number of reasons such as explicit failure caused by timeouts, incorrect or no value is returned, supposedly expired entry is returned, or unexpected key is requested. Based on simulations, the last failure scenario is a constant simulation artifact that is present in all results and has a very small but negligible value.

Efficiency focuses on resource utilization, an essential requirement in engineering systems, which can be measured by lookup latency and bandwidth usage. An efficient system incurs low average delay and low average consumption. Each node in a DHT calculates for its average lookup delay and average bandwidth consumption based on Eqs. (2) and (3), respectively. In this research, a node's average lookup latency is calculated based on the average of all the delays incurred in completing all the lookup requests of that node. A node's bandwidth consumption is computed as the ratio of the total bytes sent by that node over the total simulation time.

$$E[\text{Lookup Latency}]_n = \frac{\sum_{l=1}^{L} \text{Latency of lookup } l}{L} \ (\text{s}) \tag{2}$$

$$E[\text{BW Consumption}]_n = \frac{\text{Total bytes sent}}{\text{Simulation time}} \left(\frac{\text{Bytes}}{\text{s}}\right) \tag{3}$$

Where:
L = Total number of lookups
N = Total number of nodes
n = A certain node

Bandwidth usage can be further broken down into three components, namely application data bandwidth, application lookup bandwidth, and maintenance bandwidth. Application data bandwidth pertains to all traffic originating from the DHT application. This includes both the messages for DHT requests and the maintenance traffic for DHT updates. For the HDHTs, this bandwidth also includes the global lookup messages for inter-cluster communications. On the other hand, the maintenance bandwidth pertains to overlay transmitted for maintaining the overlays such as for updating routing tables, for checking alive neighbors and for joining overlays.

Scalability entails that the system performance does not degrade upon an increase in population size. We can analyze system scalability when the values of the performance metrics do not significantly degrade as we introduce more nodes.

In the two sections for evaluation, performance metrics are analyzed based on the average values of all the nodes in the system, rather than based on a single node. These averages are computed by getting the means of all the nodes' performance metrics.

4. Evaluation of Chord–Chord HDHT

In this section, Chord–Chord HDHT is compared to Chord DHT at various operating conditions in order to determine which system is better in the presence of churn.

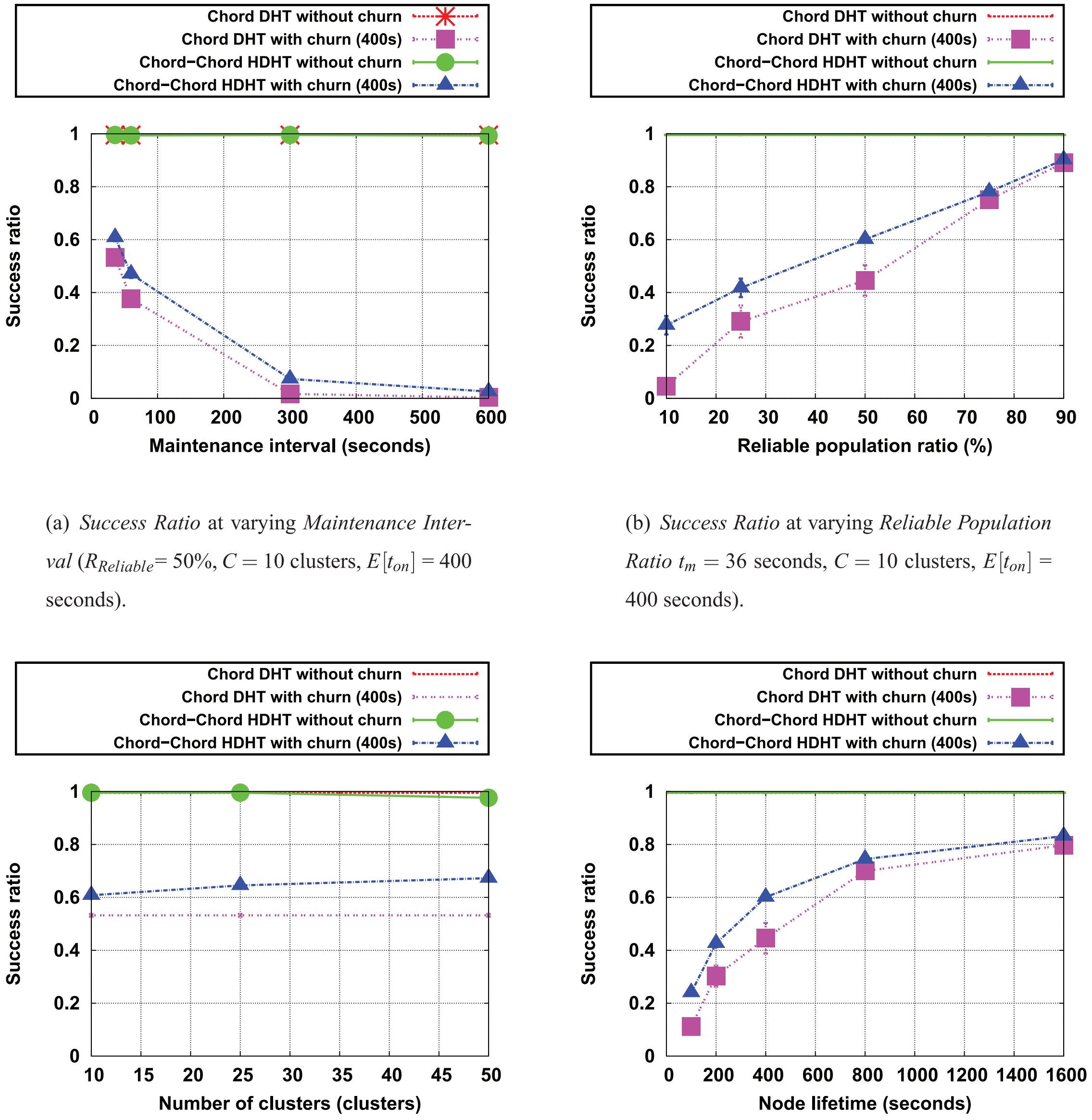
4.1. Success Ratio

Fig. 3 shows the *Success Ratio* with respect to the *Total Population* of four DHT systems. The figure shows that both Chord DHT and Chord–Chord HDHT under churn display decreasing success ratios when the total population increases, similar to [18]. At higher population, resolving a node's location becomes slower since the request is forwarded to more hops and thus more prone to error since the request is more likely to pass unreliable nodes.

We can also see from the figure that Chord–Chord HDHT has higher success ratios than that of the Chord DHT, despite the negative effects of churn. The HDHT has several independent overlays that are much smaller than the Chord DHT overlay. Since a smaller overlay has better performance, Chord DHT has lower success ratios due to having effectively higher population all in one overlay.

[2] In succeeding discussions, lookup success and lookup latency are more commonly referred to as *Success Ratio* and *Latency*, respectively.

[3] A completed lookup pertains to a DHT request that has returned a DHT reply to the source node. However, the reply may or may not be the correct value.



(a) *Success Ratio* at varying *Maintenance Interval* ($R_{Reliable}$= 50%, $C = 10$ clusters, $E[t_{on}] = 400$ seconds).

(b) *Success Ratio* at varying *Reliable Population Ratio* $t_m = 36$ seconds, $C = 10$ clusters, $E[t_{on}] = 400$ seconds).

(c) *Success Ratio* at varying *Number of Clusters* ($t_m = 36$ seconds, $R_{Reliable}$= 50%, $E[t_{on}] = 400$ seconds).

(d) *Success Ratio* at varying *Node Lifetime* ($t_m = 36$ seconds, $R_{Reliable}$= 50%, $C = 10$ clusters).

Fig. 4. *Success Ratio* at different scenarios with $N = 2000$ nodes. Note that Chord DHT and Chord–Chord HDHT without churn plots overlap at 100% success ratio.

Both Chord DHT and Chord–Chord HDHT without churn do not encounter lookup failures since nodes do not churn, similar to the results presented in [31,52]. However, the systems with churn experience the effects of system instability as the success ratios drop significantly relative to their no-churn counterparts. The drop in the success ratios is mainly attributed to failed lookup requests due to timeouts. Also because of un-updated DHT records and topology, lookup requests fail as the returned value is incorrect. Based on all the simulations of DHTs under churn, low success ratios are experienced by all regular peers, whether mobile or reliable.

Fig. 4 shows the *Success Ratio* of four DHT systems when other variables are changed one at a time. In this set of images, Chord DHT and Chord–Chord HDHT without churn plots overlap at 100% success ratio.

Fig. 4(a) shows the *Success Ratio* with respect to the *Maintenance Interval* of four DHT systems. In this figure, we can see that when the *Maintenance Interval* increases, both Chord DHT and Chord–Chord HDHT under churn display decreasing success ratios. The P2P system must update its topology faster than the churn rate for it to be stable. Otherwise, updates are useless and lookup failures are more likely to prevail. Lookup replies may return incorrect values since the DHT records are not up to date. Lookup requests may not reach the correct destination node, since the routing tables contain incorrect peers. Worse, lookup requests can timeout since replies do not return at all.[4]

[4] Leading cause of lookup failures.

Similar to [35], when the maintenance interval is greater than half the node session time,[5] lookup success significantly drops. Success ratio is almost zero since routing tables and DHT records no longer contain correct and updated information. Furthermore, systems with no churn are unaffected by changes in the maintenance durations based on the figure.

Fig. 4(b) shows the *Success Ratio* with respect to the *Reliable Population Ratio* of the two churning DHT systems.[6] This figure shows that as more reliable peers participate in the overlay, it is less likely that requests timeout or return wrong values and DHT records go stale. Success ratio improves due to better node availability. This increasing success ratio slowly converges to the no-churn success ratio.

Fig. 4(c) shows the *Success Ratio* with respect to the *Number of Clusters* of the two HDHT systems.[7] We can see from this figure that as the number of clusters increases, the success ratio of the HDHT under churn slightly increases.

Cluster size depends on the total population and the number of clusters. When the number of clusters in the network is increased, the cluster size is effectively reduced. Since the size is minimized, there are fewer nodes in each sub overlay. We can expect that the success ratio increases because there are fewer peers in each sub overlay. With this, as the number of clusters increases, the success ratio also increases.

Fig. 4(d) shows the *Success Ratio* with respect to the *Node Lifetime* of the two churning DHT systems. We can see from this figure that as the node lifetime increases, success ratio increases. The node join/leave rate decreases as the node lifetime increases. Because fewer nodes join and leave the overlay, nodes are more likely present, along with the keys stored in them. These nodes can participate in the overlay and exchange maintenance or DHT updates. Lookup requests are less likely to timeout because the routing table entries and DHT records are more likely updated. Because of better stability, the system experiences better success ratio.

4.2. Latency

Fig. 5 shows the *Latency* with respect to the *Total Population* of four DHT systems. This figure shows that lookup latency increases when the total population increases. When there are more nodes present, lookup requests incur more hops to complete and take longer to resolve, since some nodes are unreliable. Corollary to this, a smaller network should resolve a lookup faster because of fewer hops.

Based on the figure, Chord–Chord HDHT with churn has lower latencies than that of the Chord DHT with churn. HDHT uses superpeers to pass global requests. Results show that the global lookup latency[8] is roughly 0.8–1.3 s. This denotes that global lookup resolution is very fast as the top overlay acts as high-speed backbone network. A source node can immediately contact desired peers, because of the fast resolution at the top overlay and the small population at the sub overlays. Hence, Chord–Chord HDHT with churn has lower latency than Chord DHT with churn due to the hierarchical design.

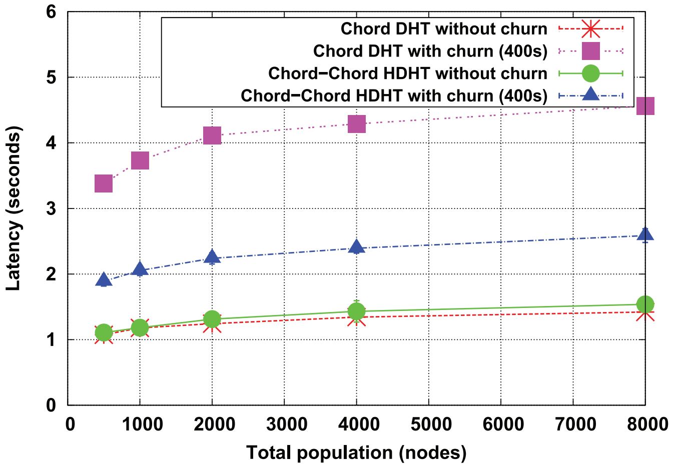


Fig. 5. *Latency* at varying *Total Population* ($t_m = 36$ s, $R_{\text{Reliable}} = 50\%$, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s).

Both Chord DHT and Chord–Chord HDHT have similar lookup latencies at no churn, regardless of any increase in the total population. When churn is not prevalent in a DHT, nodes and the key-value pairs associated to them are always available. Lookup requests are completed quickly without exceeding the defined time-to-live (TTL). But when churn is prevalent in a DHT, the key-value pairs are sometimes missing or outdated. Lookup requests often timeout, exacerbating the average lookup latency. Long lookup latency caused merely by the presence of churn is experienced regardless of the network variables. Thus, churn degrades system performance in terms of both lookup success and lookup latencies. Based on all simulations of DHTs under churn, high latencies are experienced by all regular peers, whether mobile or reliable, in either Chord DHT or Chord–Chord HDHT.

Fig. 6 shows the *Latency* of four DHT systems when other variables are changed one at a time.

Fig. 6(a) shows the *Latency* with respect to the *Maintenance Interval* of the four DHT systems. This figure shows that both Chord DHT and Chord-Chord HDHT under churn display increasing latencies when the maintenance interval increases. When topological and application changes are not updated quickly, lookup query timeouts are pervasive. These requests are completed by default but are not successful due to exceeded TTL. More timeouts occur when the maintenance interval increases, hence the average lookup latency also increases. Similar to Fig. 4(a), average lookup latency is very high for intervals 300 and 600 s since lookup timeouts are more common.

Fig. 6(b) shows the *Latency* with respect to the *Reliable Population Ratio* of the two churning DHT systems. There are less request timeouts at higher reliable population ratios due to better node availability. So lookups are resolved more quickly reducing the average lookup latency. This decreasing latency slowly converges to the no-churn latency values.

Fig. 6(c) shows the *Latency* with respect to the *Number of Clusters* of the two HDHT systems. This figure conveys that as the number of clusters increases, latency decreases to some degree. Cluster size becomes smaller as the number of clusters increases. We can expect that the latency in each sub overlay is decreased due to smaller population.

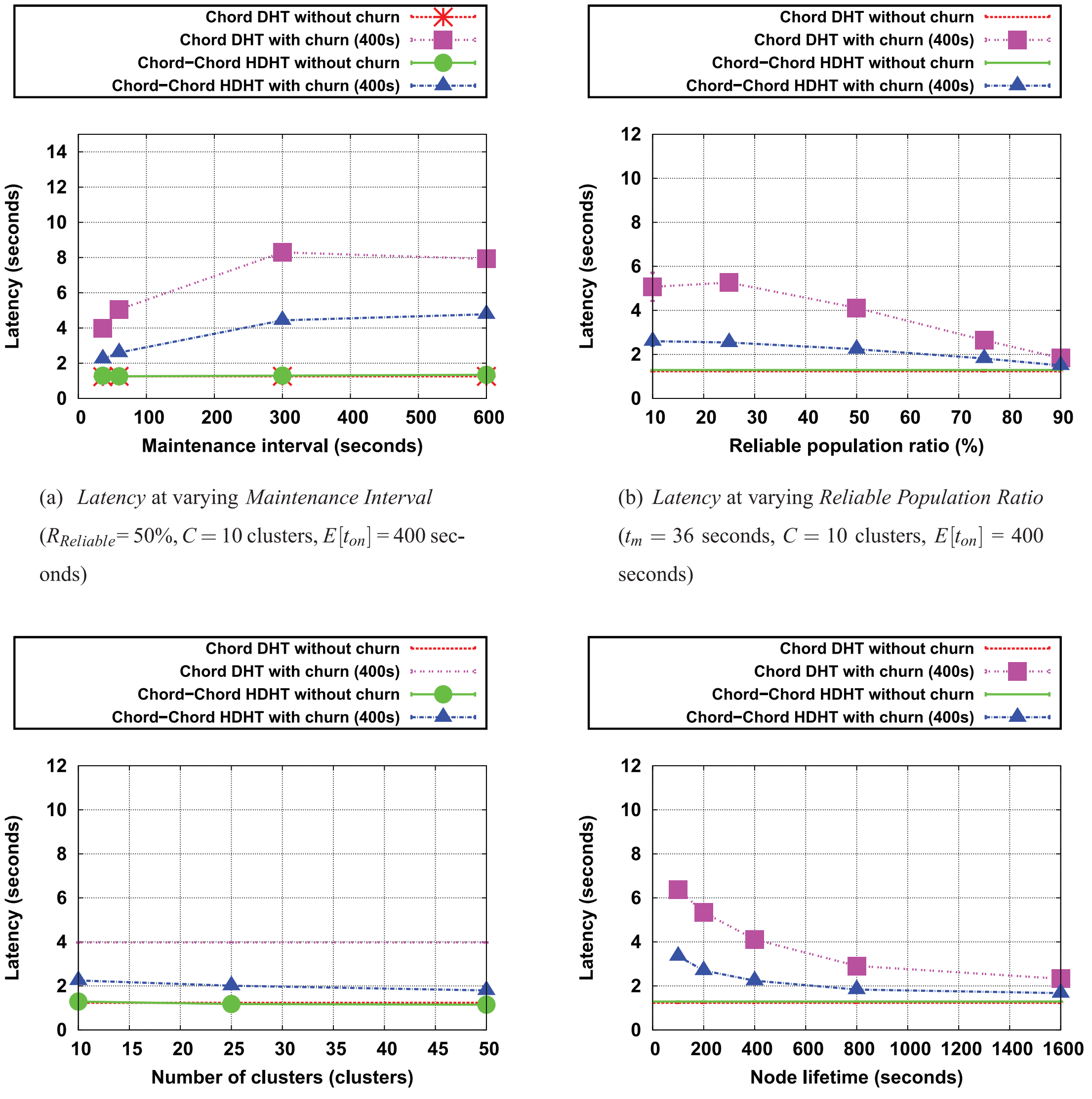
Fig. 6(d) shows the *Latency* with respect to the *Node Lifetime* of the two churning systems. We can see in this figure that a churning system becomes more stable when the node lifetime increases. Nodes are able to engage in overlay activities, such as exchanging topology messages and updating DHT records. Lookup requests are less likely to timeout and are resolved more quickly

[5] In Fig. 4(a), node session time is equal to 200 s. Maintenance interval greater than half the session time are 300 and 600 s. While maintenance interval less than half the session time are 36 and 60 s.

[6] No-churn systems have no mobile peers. So performance metrics of these systems are not affected when we vary R_{Reliable} or $E[t_{\text{on}}]$, which rely on the presence of mobile peers. For Figs. 4(b) and (d), 6(b) and (d), 9, 11, 13(b) and (d), the performance metric of no-churn systems are held at fixed values.

[7] An FDHT has a single overlay. So performance metrics of an FDHT are not affected when we vary the number of clusters. For Figs. 4(c), 6(c), 10 and 13(c), the performance metric of flat systems are held at fixed values.

[8] Time needed to find the desired nodes for the global requests.



(a) *Latency* at varying *Maintenance Interval* ($R_{Reliable}$= 50%, $C = 10$ clusters, $E[t_{on}] = 400$ seconds)

(b) *Latency* at varying *Reliable Population Ratio* ($t_m = 36$ seconds, $C = 10$ clusters, $E[t_{on}] = 400$ seconds)

(c) *Latency* at varying *Number of Clusters* ($t_m =$ 36 seconds, $R_{Reliable}$= 50%, $E[t_{on}] = 400$ seconds)

(d) *Latency* at varying *Node Lifetime* ($t_m = 36$ seconds, $R_{Reliable}$= 50%, $C = 10$ clusters)

Fig. 6. *Latency* at different scenarios with $N = 2000$ nodes.

due to updated topology and DHT entries causing a lower average lookup latency.

4.3. Bandwidth consumption

Fig. 7 shows the *Bandwidth Consumption* with respect to the *Total Population* of the four DHT systems. In this figure, we can see that when the total population increases, bandwidth slightly increases.

Chord nodes periodically send out topology maintenance updates.[9] If the population is increased, more nodes are able to participate in the overlay then more network updates are transmitted resulting to higher maintenance traffic. With this, bandwidth consumption increases as the total population increases, as a consequence of increased maintenance load.

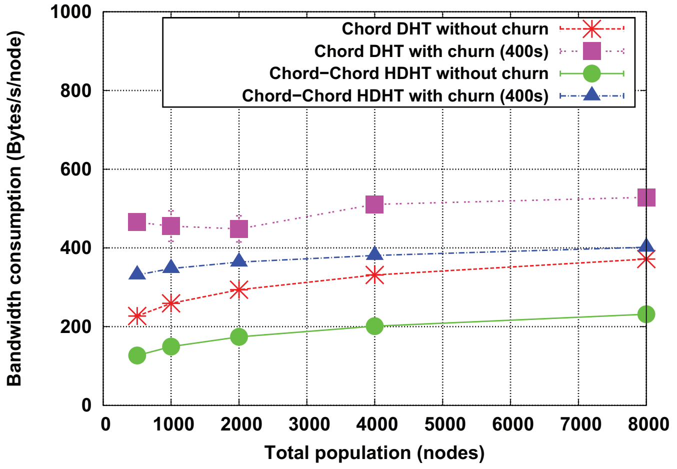


Fig. 7. *Bandwidth Consumption* at varying *Total Population* ($t_m = 36$ s, R_{Reliable}= 50%, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s).

[9] These are messages for updating the overlay network topology (i.e. nodes' routing tables). They are different from the updates for the DHT records, which are essentially application maintenance.

Table 9
Bandwidth consumption of superpeers and regular peers in Chord–Chord HDHT.

	Superpeers	Regular peers
Total traffic	≈590 Bps/node	≈350 Bps/node
Application traffic	≈550 Bps/node	≈185 Bps/node
Maintenance traffic	≈40 Bps/node	≈170 Bps/node

Table 10
Bandwidth consumption of mobile and reliable regular peers in Chord–Chord HDHT. (Regular peers pertain to non-superpeers that are either mobile or reliable.)

	Mobile regular peers	Reliable regular peers
Total traffic	≈390 Bps/node	≈150 Bps/node
Application traffic	≈200 Bps/node	≈30 Bps/node
Maintenance traffic	≈180 Bps/node	≈110 Bps/node

Chord–Chord HDHT (with or without churn) has lower bandwidth consumption than that of its Chord DHT counterpart. Since the former has smaller sub overlay population, fewer nodes transmit maintenance traffic resulting to lower bandwidth usage.

Finally, churning systems experience the effects of churn as the bandwidth consumption significantly increases relative to the no-churn systems. Both Chord DHT and Chord–Chord HDHT have higher bandwidth consumption in order to maintain the overlay topology and update DHT records.

Based on the three performance metrics, Chord–Chord HDHT shows promising improvements over the Chord DHT. The trade-off is evident however when we delve into the superpeers' traffic.

Tables 9 and 10 display the average node bandwidth utilization breakdowns of superpeer against regular peer and of mobile peer against reliable peer, respectively. Table contents are based on a Chord–Chord HDHT under churn scenario with the following fixed parameters: $N = 2000$ nodes, $t_m = 36$ s, $R_{\text{Reliable}} = 50\%$, $C = 10$ clusters, and $E[t_{\text{on}}] = 400$ s.

Table 9 shows that superpeers have higher total traffic compared to regular peers because of higher application traffic. About 90% of all requests are global in a 10-cluster HDHT, so superpeers are burdened with forwarding these global messages resulting to higher application traffic. When the system has more nodes, more nodes send out global requests and the superpeers forward even more of these requests. Basically, superpeer bandwidth consumption increases greatly with an increase in the total population. Comparing the maintenance bandwidth consumptions, superpeers have low topology maintenance traffic than regular peers, since top overlay is stable. Despite the significant bandwidth utilization of the superpeers, this does not reflect in the average node bandwidth, since there are only 10 superpeers in the network. All the non-superpeers' bandwidth usage dilute the superpeers' high bandwidth consumption.

Table 10 shows that among regular peers, the bandwidth consumptions differ depending on their mobility. Mobile peers use more bandwidth than regular peers due to higher application and maintenance traffic.[10] They transmit more messages for joining/leaving the overlay, filling the routing tables and acquiring DHT records. Although churn effects in terms of success ratio and latency are similar whether a regular peer is mobile or reliable, churn effect in terms of bandwidth consumption depends on the node type.

Fig. 8 shows the *Bandwidth Consumption* with respect to the *Maintenance Interval*. From this figure, we can infer that all systems display diminishing bandwidth consumption as the value of the maintenance interval is increasing.

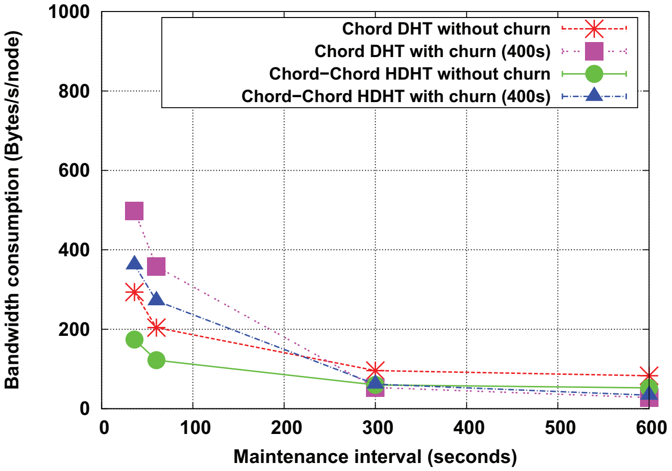


Fig. 8. *Bandwidth Consumption* at varying *Maintenance Interval* ($N = 2000$ nodes, $R_{\text{Reliable}} = 50\%$, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s).

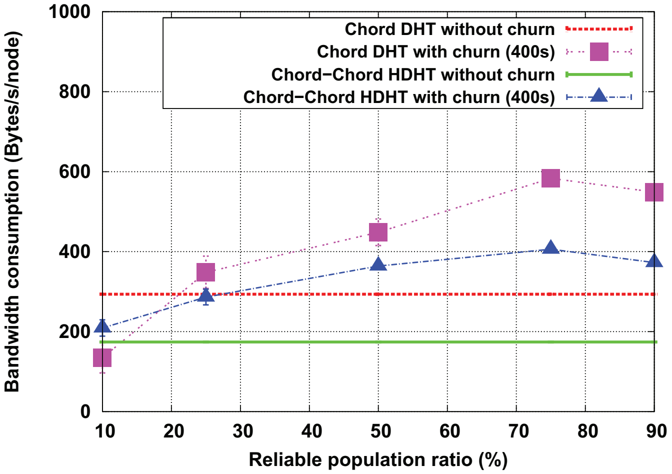


Fig. 9. *Bandwidth Consumption* at varying *Reliable Population Ratio* ($N = 2000$ nodes, $t_m = 36$ s, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s).

Maintenance interval is the reciprocal of the maintenance rate, or the speed at which topological updates or messages are sent. Higher maintenance interval equates to lower maintenance rate. If the maintenance rate is lower, nodes send out less maintenance bytes and use up less bandwidth. When we increase the maintenance interval, we decrease the maintenance rate and we decrease the node bandwidth consumption.

Looking at the 36- and 60-s scenarios, we infer the following. First, a DHT with churn has higher traffic consumption than the same DHT without churn. A churning DHT has to execute more maintenance traffic due to overlay join and leave procedures and due to DHT record refreshes. Second, the hierarchical system has lower traffic consumption than the flat system, since the former has smaller cluster sizes.

However, the 300- and 600-s settings show different results. Nodes do not remain in the network long enough to exchange relevant updates. Since the maintenance rates are too low, nodes exchange messages rarely, thus traffic consumption is too low for all DHTs.

Fig. 9 shows the *Bandwidth Consumption* with respect to the *Reliable Population Ratio* of the two churning DHT systems. This figure conveys that as the reliable population ratio increases,

[10] Both true for either flat or hierarchical DHT.

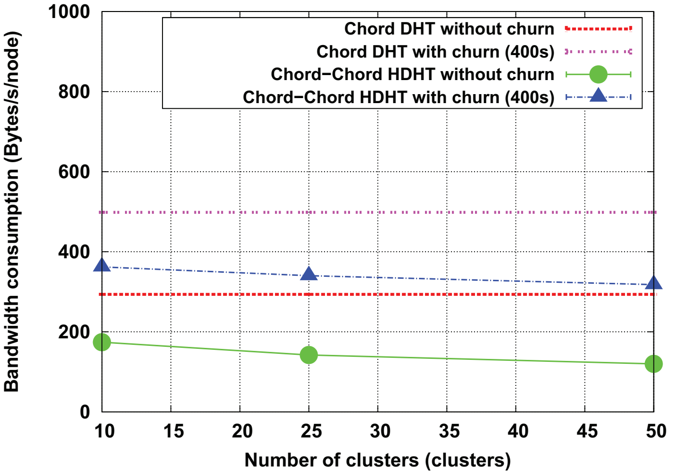


Fig. 10. *Bandwidth Consumption* at varying *Number of Clusters* ($N = 2000$ nodes, $t_m = 36$ s, $R_{\text{Reliable}} = 50\%$, $E[t_{\text{on}}] = 400$ s).

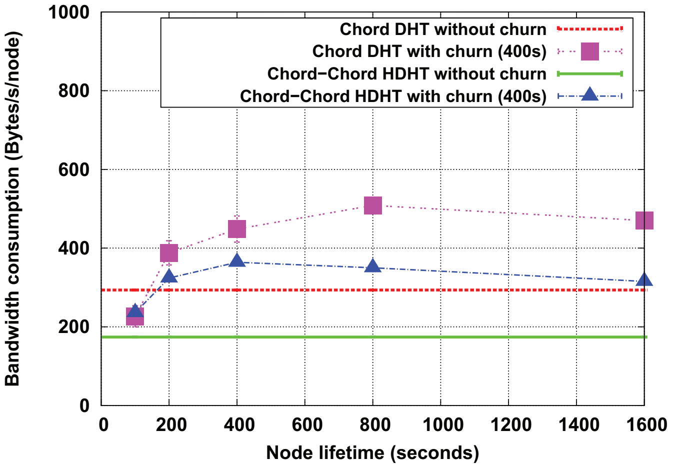


Fig. 11. *Bandwidth Consumption* at varying *Node Lifetime* ($N = 2000$ nodes, $t_m = 36$ s, $R_{\text{Reliable}} = 50\%$, $C = 10$ clusters).

Table 11
Bandwidth consumption of superpeers as C increases in Chord–Chord HDHT.

	Superpeers (no-churn)	Superpeers (churn)
Total traffic at $C = 10$	≈715 Bps/node	≈585 Bps/node
Total traffic at $C = 25$	≈340 Bps/node	≈300 Bps/node
Total traffic at $C = 50$	≈230 Bps/node	≈210 Bps/node

Table 12
Chord-–Chord HDHT performance during bootstrap.

	Disabled DHT requests		Enabled DHT requests	
	No churn	With churn	No churn	With churn
Success ratio	0	0	0.997	0.445
Latency (s)	0	0	1.29	2.68
Bandwidth Cons. (Bps/node)	≈129	≈139	≈174	≈243

systems become more stable; bandwidth utilization increases, then decreases to converge back to the no-churn values.

At a highly mobile environment (e.g. 10% reliable population ratio), nodes are unable to properly communicate with each other resulting to low bandwidth usage. Success ratios are also very poor at this environment based on Fig. 4(b). Only during this case is Chord DHT bandwidth consumption lower than that of Chord–Chord HDHT, because the former is no longer operating properly (i.e. success ratio is almost zero). At 50% reliable population ratio, bandwidth usage has increased. More nodes are available in the network and are able to transmit messages for topology and DHT maintenance. At the 90% case, nodes consume less bandwidth than at 50%. More nodes are stable and require less maintenance traffic.

Fig. 10 shows the *Bandwidth Consumption* with respect to the *Number of Clusters* of the two HDHT systems. This figure shows that if we keep the total population constant and increase the number of clusters, the bandwidth utilization slightly decreases, since fewer nodes transmit maintenance messages in each cluster. Although increasing the number of clusters only shows little performance enhancement overall, it has significant impact in reducing the top overlay traffic.

Table 11 shows the bandwidth utilization of the superpeers as C increases. When the number of clusters increases, the bandwidth consumption of superpeers are reduced. Whether Chord–Chord HDHT is under churn or not, superpeers have to route less messages when there are several clusters since there are fewer nodes in each sub overlay. We can construct an HDHT with more number of clusters to reduce the superpeer load. We expect that Chord–Chord HDHT under churn has lower superpeer traffic since fewer nodes are available to send global requests.

Fig. 11 shows the *Bandwidth Consumption* with respect to the *Node Lifetime* of the two systems under churn. In this figure, we can see that bandwidth consumption initially increases to a certain point as the node lifetime increases. However, it decreases if the node lifetime increases even further. This behavior is caused by better system stability similar to the discussion on Fig. 9.

At short node session time (e.g. $E[t_{\text{on}}] = 100$ s), bandwidth consumption is very low. During this scenario, a great quantity of nodes are joining and leaving the network, causing the DHT to be faulty. Nodes are not able to communicate properly, and there is low traffic in the overlay. If the average node lifetime is increased, more nodes are able to exchange messages in the overlay. Traffic increases due to higher maintenance traffic.

At long node session time (e.g. $E[t_{\text{on}}] > 400$ s), bandwidth consumption decreases as the node lifetime increases and is slowly to converge to the no-churn states. Nodes are more often available and the system requires less maintenance. So if the average node lifetime is increased, the overlay utilizes less bandwidth.

4.4. What happens to Chord–Chord HDHT during transient phase?

Previous discussions concentrate on the analysis of DHT performance during transient phase. However, performance differs when analyzed during network bootstrap, wherein both the top overlay and sub overlays are constructed. The top overlay is quickly formed because it usually consists of 10 superpeers only. The sub overlays are slowly formed because they have larger population sizes. All regular peers in the network perform join procedures in order to self-organize the sub overlays.

Table 12 lists the Chord–Chord HDHT performance during bootstrap. In this table, we show Chord–Chord HDHT with and without churn, and with and without DHT requests with the following parameters: $N = 2000$ nodes, $t_m = 36$ s, $R_{\text{Reliable}} = 50\%$ for churning HDHT, $C = 10$ clusters, and $E[t_{\text{on}}] = 400$ s for churning HDHT. During transient phase, regular peers do not perform any lookup requests. However for this discussion, we enabled the DHT requests for two scenarios to examine the HDHT operation during bootstrap.

When DHT requests are disabled during the transient period, Chord–Chord HDHT has 0% success ratios and 0-s latencies because peers do not perform lookup requests. Similar to Tables 9 and 10, Chord–Chord HDHT experiences small bandwidth consumptions due to overlay maintenance traffic. Chord–Chord HDHT with churn experiences slightly higher traffic because of topology maintenance updates.

When DHT requests are enabled during the transient period, peers can perform lookup requests even when the hierarchical system is not completely constructed.

Chord–Chord HDHT without churn still has 100% success ratio, since peers can only search for keys that are available in the whole network. Because of node availability, Chord–Chord HDHT without churn has low lookup latency and low bandwidth usage.

Chord–Chord HDHT with churn has 44.5% success ratio, which is significantly lower than the expected 60% success ratio (see Fig. 3). Additional lookup failures are caused by the partitions within sub overlays. Even if peers have not yet joined any sub overlay, they can still perform lookup requests. Such queries would fail due to timeouts since peers cannot communicate with their neighbors. Because of the lookup failures and node timeouts, Chord–Chord HDHT under churn experiences higher lookup latency compared to Chord–Chord HDHT without churn. Chord–Chord HDHT under churn encounters higher bandwidth utilization compared to Chord–Chord HDHT without churn due to higher maintenance and application traffic.

Compared to Chord–Chord HDHT after transient phase (see Figs. 5 and 7), Chord–Chord HDHT during transient has similar latency and bandwidth usage values. Chord–Chord HDHT during transient phase still operates properly even at a slightly poorer performance compared to Chord–Chord HDHT during steady-state.

4.5. Summary of the evaluation of Chord–Chord HDHT

Our results show the adverse effects of churn on the performance of DHTs. An unstable system has lower lookup success, higher lookup latency and higher bandwidth consumption compared to the same system that is relatively stable. The effects on the success and latency are distributed among all the peers; while the effect on the bandwidth consumption is concentrated on the mobile peers.

Increasing the total population degrades the system performance of a DHT, since resolving a node's location require more hops. Lookup requests take longer to resolve and in some cases, they fail because the routing tables and DHT records contain erroneous entries.

Increasing the maintenance interval also degrades the DHT performance primarily because nodes have stale routing tables and DHT records. Although higher maintenance interval introduces less bandwidth consumption, the success ratios and latency fail considerably. Lookup requests often timeout, causing failures and increasing the delays.

Increasing the reliable population enhances the DHT performance as a result of better node availability. When there are more reliable nodes in the network, keys are more likely available and lookup requests are quickly resolved.

Increasing the number of clusters at constant total population slightly enhances the HDHT performance as a result of smaller cluster size. At higher number of clusters, there is a slight improvement in the HDHT performance and a significant reduction in superpeer traffic.

Increasing the node lifetime improves the HDHT performance, because of better node availability. Similar to increasing the reliable population ratio, better node availability causes an increase in the success ratio, decrease in the latency and a unique effect in the bandwidth utilization.

Each performance metric improves or degrades depending on the varied network property. Success Ratio (for churn scenario only), Latency and Bandwidth consumption improve when the network is highly populated with reliable peers, when the HDHT has many sub overlays, or when the churn rate is low. However these metrics degrade when the network is highly populated, or when the maintenance updates are not frequent. We can minimize performance degradation by choosing a small N, a short t_m, a high R_{Reliable}, a large C, and a long $E[t_{\text{on}}]$.

In all scenarios with churn, the Chord–Chord HDHT performs better than the Chord DHT. The main cause of such enhancement is the use of multiple independent clusters. When the nodes are distributed to different sub overlays, the network size is effectively reduced. We have inferred that a small network is much better than a large network, so if we have reduced the network size then we can expect better performance. These benefits do not come free; such improvements are shouldered by the superpeers, which handle forwarding of global lookup messages. These messages require additional CPU processing and memory on the superpeers. However, superpeer traffic is only in the order of a few kilobytes/second/node, which is still practical and feasible.

In no-churn scenarios, the two DHTs are comparable in terms of latency and success ratios. They operate similarly especially when both are stable and differ only in the amount of bandwidth consumed. With this, we can deduce that the enhancement of Chord–Chord HDHT over the Chord DHT is only suitable if the nodes experience churn and there are multiple autonomous systems in the whole network.

5. Evaluation of Chord–Kademlia HDHT

In this section, Chord–Chord HDHT is compared to Chord–Kademlia HDHT at various operating conditions in order to determine whether they perform similarly.

5.1. Success ratio

Fig. 12(a) shows the *Success Ratio* with respect to *Total Population* of the four HDHT systems. Both Chord–Chord and Chord–Kademlia have similar success ratio trends. When the total population is increased, the success ratio decreases.

We can see from this figure that Chord–Kademlia HDHT under churn has higher success ratios than Chord–Chord HDHT under churn. This is because of the routing table update policy and size of our Kademlia overlay implementation. In the simulation configuration, we have enabled the *Secure maintenance* parameter of Kademlia, which allows update only if it comes from an authenticated peer. Authentication occurs by pinging the source of the received message. If the source is still alive, the Kademlia node updates its bucket accordingly. In addition, Kademlia peers have more nodes in its tables than the Chord peers and are able to quickly locate the desired nodes because tables are more populated.

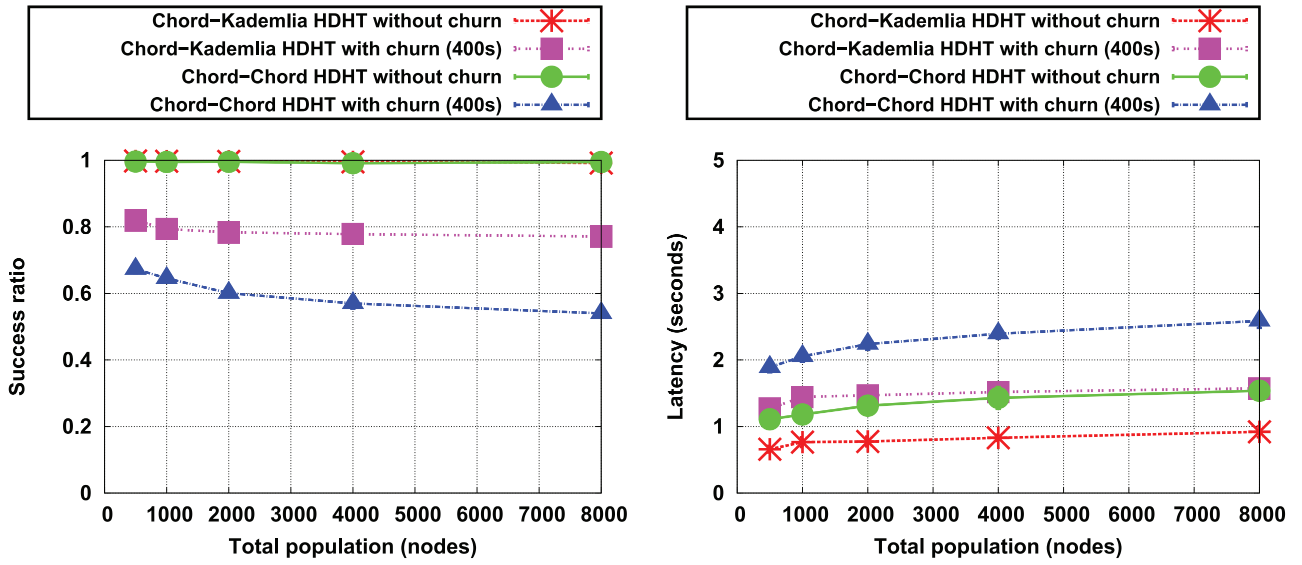
For other Success Ratio scenarios,[11] Chord–Kademlia HDHT performs similarly to Chord–Chord HDHT. The only difference is that the former has better fault tolerance due to its update policy and big routing table.

5.2. Latency

Fig. 12(b) shows the *Latency* with respect to *Total Population* of the four HDHT systems. Both Chord–Chord and Chord–Kademlia have similar behavior. When the total population is increased, the latency decreases.

We can infer from this figure that Chord–Kademlia HDHT has lower latencies than Chord–Chord HDHT. Chord–Kademlia HDHT

[11] Success Ratios at varying Reliable Population Ratio, at varying Number of Clusters and at varying Node Lifetime.



(a) *Success Ratio* (b) *Latency*

Fig. 12. *Success Ratio* and *Latency* at varying *Total Population* with $t_m = 36$ s, R_{Reliable}= 50%, $C = 10$ clusters, $E[t_{on}] = 400$ s. Note that Chord–Chord HDHT and Chord–Kademlia HDHT without churn plots overlap at 100% success ratio.

Table 13
Bandwidth consumption of superpeers and regular peers in Chord-Kademlia HDHT.

	Superpeers	Regular peers
Total traffic	≈700 Bps/node	≈800 Bps/node
Application traffic	≈600 Bps/node	≈600 Bps/node
Maintenance traffic	≈50 Bps/node	≈190 Bps/node

Table 14
Bandwidth consumption of mobile and reliable regular peers in Chord–Kademlia HDHT. (Regular peers pertain to non-superpeers that are either mobile or reliable.)

	Mobile regular peers	Reliable regular peers
Total traffic	≈900 Bps/node	≈120 Bps/node
Application traffic	≈680 Bps/node	≈40 Bps/node
Maintenance traffic	≈200 Bps/node	≈40 Bps/node

has better average lookup latencies than Chord–Chord HDHT because of faster lookup resolution for both local and global requests. Since Chord–Kademlia has larger and more updated routing tables, correctly locating the desired nodes locally is then easier. Local lookup requests are therefore resolved much quickly. Chord–Kademlia HDHT also provides shorter global lookup latency since the superpeer at the destination cluster is able to locate the destination nodes much faster[12] than Chord–Chord HDHT, because of the better freshness and larger size of the routing table.

For other Latency scenarios,[13] Chord–Kademlia HDHT performs similarly to Chord–Chord HDHT. The only difference is that the former resolves lookup requests much faster due to its update policy and big routing table.

5.3. Bandwidth consumption

Fig. 13 shows the *Bandwidth Consumption* of four HDHT systems when other variables are changed one at a time.

Fig. 13(a) shows the *Bandwidth Consumption* with respect to *Total Population* of the four HDHT systems. Although not evident in the figure, bandwidth consumption increases as the total population increases for all HDHTs. For Chord–Kademlia HDHT, the bandwidth usage increases exponentially as the total population increases.

Tables 13 and 14 display the average node bandwidth utilization breakdowns of superpeer against regular peer and mobile peer against reliable peer, respectively, for one particular scenario. Table contents are based on one Chord-Kademlia HDHT under churn scenario with the following fixed parameters: $N = 2000$ nodes, R_{Reliable}= 50%, $C = 10$ clusters, and $E[t_{\text{on}}] = 400$ s.

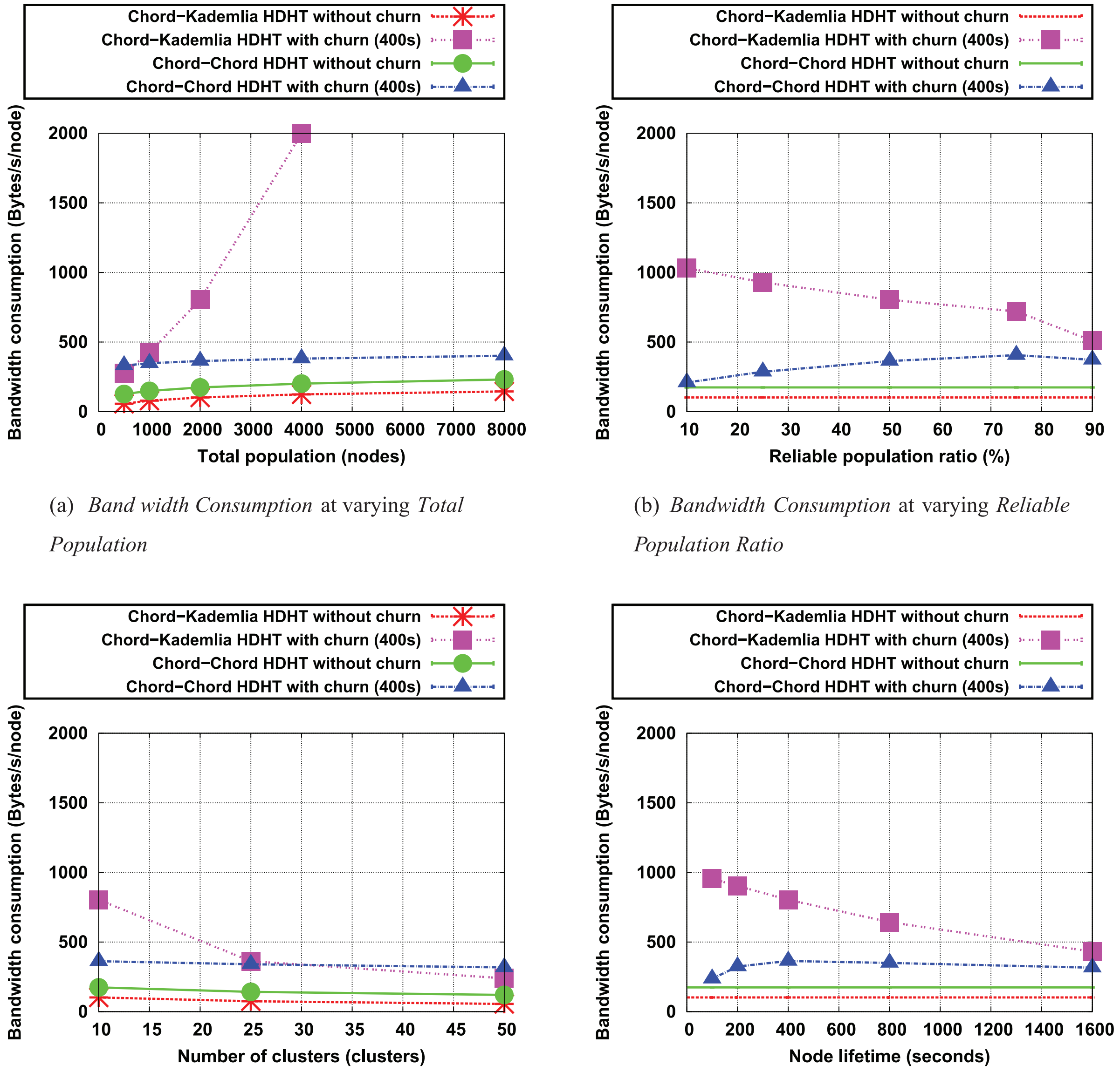
Table 13 shows that regular peers have higher total traffic compared to superpeers in Chord-Kademlia HDHT. This higher regular peer traffic is caused by higher application and maintenance traffic of mobile peers as seen in Table 14. Superpeers are still burdened with forwarding global messages resulting to high application bandwidth. But the regular Kademlia peers, specifically the mobile peers, experience high traffic because of refreshing the large Kademlia buckets and updating their DHT records. Because of the mobile peers, the average node bandwidth consumption of all the regular peers is affected. Although churn effects to success ratio and latency are distributed among peers in all the DHTs, churn effect in terms of bandwidth consumption still depends on the node type similar to the previous chapter.

Because of higher bandwidth usage for regular peers, Chord–Kademlia HDHT under churn has noticeably higher bandwidth utilization than Chord–Chord HDHT under churn. In Chord–Chord HDHT, the regular peers consume lower bandwidth usage than the superpeers for both maintenance and application traffic. But in Chord–Kademlia HDHT, the regular peers consume more bandwidth usage than the superpeers due to higher maintenance and application bytes per second.

However, Chord–Kademlia HDHT with no churn has lower bandwidth consumption than Chord–Chord HDHT with no churn, because the former has very low maintenance traffic ($<$70 Bps). Since all nodes are reliable and are always available, Kademlia buckets of the regular peers rarely need to be refreshed.

[12] Twice as fast because Chord–Kademlia HDHT's global lookup latency is half a second, while Chord–Chord HDHT's is 1 s.

[13] Latency at varying Reliable Population Ratio, at varying Number of Clusters and at varying Node Lifetime.



(a) *Band width Consumption* at varying *Total Population*

(b) *Bandwidth Consumption* at varying *Reliable Population Ratio*

(c) *Bandwidth Consumption* at varying *Number of Clusters*

(d) *Bandwidth Consumption* at varying *Node Lifetime*

Fig. 13. *Bandwidth Consumption* at different scenarios (unless considered as a variable, simulation values are $N = 2000$ nodes, $t_m = 36$ s, $R_{\text{Reliable}}= 50\%$, $C = 10$ clusters, $E[t_{\text{on}}] = 400$ s).

Fig. 13(b) shows the *Bandwidth Consumption* with respect to *Reliable Population Ratio* of the two churning HDHT systems. The figure shows that for Chord–Kademlia HDHT, unlike Chord–Chord HDHT, the bandwidth utilization decreases as the reliable population increases. Mobile peers have high bandwidth usage for application traffic in Kademlia overlays due to the update policy and routing table size. Because of this, the Chord–Kademlia HDHT experiences high bandwidth consumption when there are more mobile nodes than reliable nodes.

Fig. 13(c) shows the *Bandwidth Consumption* with respect to *Number of Clusters* of the HDHT systems. This figure shows that if we keep the total population constant and increase the number of clusters, the bandwidth utilization greatly decreases for Chord–Kademlia HDHT but slightly decreases for Chord–Chord HDHT, since fewer nodes transmit maintenance messages. The trend for Chord–Kademlia HDHT is mainly caused by the decreasing application traffic, since there are fewer nodes in each cluster at higher *C*.

Fig. 13(d) shows the *Bandwidth Consumption* with respect to *Node Lifetime* of the two churning HDHT systems. The figure shows that for Chord–Kademlia HDHT, the bandwidth utilization decreases as the node lifetime increases. This behavior is caused by the mobile peers' bandwidth consumptions for application traffic similar to Fig. 13(b). When the system contains peers that are highly mobile, the mobile peers exhibit high bandwidth consumption due to higher application traffic caused by determining the destination nodes and by exchanging of DHT records.

5.4. Summary of the evaluation of Chord–Kademlia HDHT

Our results show that Chord–Chord and Chord–Kademlia HDHTs perform similarly except for the bandwidth consumption scenarios at varying R_{Reliable} and $E[t_{\text{on}}]$, where Chord–Kademlia's traffic is decreasing. Both HDHTs perform satisfactorily due to their cluster-based designs, at the cost of higher traffic and processing for superpeers. We can see that increasing the total population, while keeping the number of clusters constant, still degrades the HDHT system performance. Increasing the reliable population ratio, number of clusters, or node lifetime improves system performance of the two HDHTs, as well.

In no-churn events, the two HDHTs have comparable success ratios, but Chord–Kademlia HDHT has lower latency and lower traffic due to its fast lookup resolution. In most cases where churn is present, Chord–Kademlia HDHT performs better than Chord DHT in success ratios and latencies but suffers greatly from high traffic, due to the larger and more updated Kademlia buckets and due to Kademlia's update policy. The difference in performance is credited to the choice of overlay parameters.

6. Conclusions and future work

In this work, we have evaluated three systems, namely, Chord DHT, Chord–Chord HDHT and Chord–Kademlia HDHT, under various operating conditions. We have shown that the flat DHT performs poorly at high churn, prompting the need for a system with better support for mobile nodes. We have shown that the two hierarchical systems exhibit satisfactory performance even at the presence of churn. We discover that both HDHTs are more stable, more efficient and more scalable compared to the flat system and are therefore good candidates in alleviating the effects of churn in mobile networks.

Chord–Chord HDHT is better than Chord DHT because of the cluster-based design resulting to better fault tolerance. It is more stable, more efficient and more scalable than Chord DHT. Similar to Chord–Chord HDHT, Chord–Kademlia HDHT also shows good performance.

This work's Chord–Kademlia HDHT implementation can be employed in mobile networks that need to be reliable and delay-intolerant but are not bandwidth-constrained. Such networks require few lookup failures and fast lookup resolution to ensure that time-sensitive materials are received correctly and immediately. However if the main consideration is lightweight mobile application, our work's Chord–Chord HDHT implementation is the better option. Mobile devices have limited capacities, which means short battery life and unreliable connectivity. Such devices require an application that can provide good lookup success and fast lookup latency, along with small bandwidth utilization to optimize the limited resources. In conclusion, we have shown that by using any of the two hierarchical DHTs, we can integrate mobile devices into the P2P paradigm.

For a more realistic scenario, we suggest various modifications to the HDHT designs. First, we propose to introduce multiple superpeers per sub overlay for better fault tolerance in the face of superpeer unreliability. Second, we recommend to introduce more nodes in the network and to have uneven cluster sizes for HDHTs. Practically, a network may span to more than 8000 nodes and each autonomous system has different population sizes. Third, we can incorporate actual file exchange after the successful lookup operation to determine the effects of transmitting large data on the over-all HDHT performance.

Acknowledgment

This research was financially supported by the Engineering Research and Development for Technology (ERDT) project of the Department of Science and Technology (DOST) in the Philippines.

References

[1] S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies, ACM Comput. Surv. 36 (4) (2004) 335–371, doi:10.1145/1041680.1041681.

[2] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, S. Lim, A survey and comparison of peer-to-peer overlay network schemes, IEEE Commun. Surv. Tutor. 7 (2) (2005) 72–93, doi:10.1109/COMST.2005.1610546.

[3] M. Hattar, The internet of everything has begun, 2013, http://blogs.cisco.com/news/the-internet-of-everything-has-begun/ (accessed 01.06.13).

[4] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for internet applications, SIGCOMM Comput. Commun. Rev. 31 (4) (2001) 149–160, doi:10.1145/964723.383071.

[5] A.I.T. Rowstron, P. Druschel, Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems, in: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg (Middleware'01), Springer-Verlag, London, UK, 2001, pp. 329–350.

[6] I. Gupta, K. Birman, P. Linga, A. Demers, R. van Renesse, Kelips: Building an efficient and stable P2P DHT through increased memory and background overhead, in: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), Springer Berlin Heidelberg, 2003, pp. 160–169, doi:10.1007/978-3-540-45172-3_15.

[7] B.Y. Zhao, J.D. Kubiatowicz, A.D. Joseph, Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing, Technical Report, University of California at Berkeley, Berkeley, CA, USA, 2001.

[8] P. Maymounkov, D. Mazières, Kademlia: A peer-to-peer information system based on the XOR metric, in: Revised Papers from the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Springer-Verlag, London, UK, 2002, pp. 53–65.

[9] Q. Hofstatter, S. Zoels, M. Michel, Z. Despotovic, W. Kellerer, Chordella: A hierarchical peer-to-peer overlay implementation for heterogeneous, mobile environments, in: Eighth International Conference on Peer-to-Peer Computing, 2008 (P2P'08), IEEE, 2008, pp. 75–76, doi:10.1109/P2P.2008.42.

[10] B.Y. Zhao, Y. Duan, L. Huang, A.D. Joseph, J. Kubiatowicz, Brocade: Landmark routing on overlay networks, in: Revised Papers from the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02), Springer-Verlag, London, UK, 2002, pp. 34–44.

[11] L. Garces-Erice, E. Biersack, K.V. Ross, P.A. Felber, G. Urvoy-Keller, Hierarchical Peer-to-Peer Systems, Euro-Par 2003 Parallel Processing, 9th International Euro-Par Conference Klagenfurt, Austria, August 26-29, 2003 Proceeding, 2790, Springer Berlin Heidelberg, 2003, pp. 1230–1239, doi:10.1007/978-3-540-45209-6_166.

[12] Z. Xu, R. Min, Y. Hu, HIERAS: A DHT based hierarchical P2P routing algorithm, in: Proceedings of the 2003 International Conference on Parallel Processing, 2003, IEEE, 2003, pp. 187–194, doi:10.1109/ICPP.2003.1240580.

[13] M. Zulhasnine, C. Huang, A. Srinivasan, Topology-aware integration of cellular users into the P2P system, in: 2011 IEEE Vehicular Technology Conference (VTC Fall), IEEE, 2011, pp. 1–5, doi:10.1109/VETECF.2011.6092842.

[14] P. Ganesan, K. Gummadi, H. Garcia-Molina, Canon in G major: Designing DHTs with hierarchical structure, in: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 263–272.

[15] M. Sànchez-Artigas, P. García-López, J.P. Ahullo, A.F.G. Skarmeta, Cyclone: A novel design schema for hierarchical DHTs, in: Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 49–56, doi:10.1109/P2P.2005.5.

[16] M. Freedman, D. Mazières, Sloppy hashing and self-organizing clusters, in: Proceedings of the 2nd International Conference on Peer-to-Peer Systems (IPTPS'03), Springer Berlin Heidelberg, 2003, pp. 45–55, doi:10.1007/978-3-540-45172-3_4.

[17] H.N. Chan, K. Van, G.N. Hoang, Characterizing Chord, Kelips and Tapestry algorithms in P2P streaming applications over wireless network, in: Second International Conference on Communications and Electronics, 2008 (ICCE'08), IEEE, 2008, pp. 126–131, doi:10.1109/CCE.2008.4578945.

[18] G.N. Hoang, H.N. Chan, K.N. Van, T.L. thi Xuan, T.N. Manh, V.V. Thanh, Performance improvement of Chord distributed hash table under high churn rate, in: International Conference on Advanced Technologies for Communications, 2009 (ATC'09), IEEE, 2009, pp. 191–196, doi:10.1109/ATC.2009.5349535.

[19] Z. Ou, E. Harjula, O. Kassinen, M. Ylianttila, Feasibility evaluation of a communication-oriented P2P system in mobile environments, in: Proceedings of the 6th International Conference on Mobile Technology, Application & Systems (Mobility'09), ACM, New York, NY, USA, 2009, pp. 43:1–43:8, doi:10.1145/1710035.1710078.

[20] S. Saroiu, P.K. Gummadi, S.D. Gribble, A measurement study of peer-to-peer file sharing systems, in: Multimedia Computing and Networking, International Society for Optics and Photonics, 2002, pp. 156–170, doi:10.1117/12.449977.

[21] W.J. Bolosky, J.R. Douceur, D. Ely, M. Theimer, Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs, SIGMETRICS Perform. Eval. Rev. 28 (1) (2000) 34–43, doi:10.1145/345063.339345.

[22] R. Bhagwan, S. Savage, G.M. Voelker, Understanding availability, in: Proceedings of the 2nd International Conference on Peer-to-Peer Systems (IPTPS'03), Springer Berlin Heidelberg, 2003, pp. 256–267, doi:10.1007/978-3-540-45172-3_24.

[23] S. Guha, N. Daswani, R. Jain, An experimental study of the Skype peer-to-peer VoIP system, in: Proceedings of the 5th International Conference on Peer-to-Peer Systems (IPTPS'06), 2006. http://iptps06.cs.ucsb.edu/papers/Guha-skype06.pdf.

[24] S. Kim, X. Wang, H. Kim, T. Kwon, Y. Choi, Measurement and analysis of BitTorrent traffic in mobile WiMAX networks, in: 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), IEEE, 2010, pp. 1–4, doi:10.1109/P2P.2010.5569997.

[25] M. McNett, G.M. Voelker, Access and mobility of wireless PDA users, SIGMOBILE Mob. Comput. Commun. Rev. 9 (2) (2005) 40–55, doi:10.1145/1072989.1072995.

[26] M. Papadopouli, H. Shen, M. Spanakis, Characterizing the duration and association patterns of wireless access in a campus, in: 11th European Wireless Conference 2005—Next Generation Wireless and Mobile Communications and Services (European Wireless), VDE, 2005, pp. 1–7.

[27] O. Kassinen, Z. Ou, M. Ylianttila, E. Harjula, Effects of peer-to-peer overlay parameters on mobile battery duration and resource lookup efficiency, in: Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia (MUM'08), ACM, New York, NY, USA, 2008, pp. 177–180, doi:10.1145/1543137.1543173.

[28] I. Kelenyi, J. Nurminen, Energy aspects of peer cooperation measurements with a mobile DHT system, in: IEEE International Conference on Communications Workshops, 2008 (ICC Workshops'08), IEEE, 2008, pp. 164–168, doi:10.1109/ICCW.2008.36.

[29] S. Gurun, P. Nagpurkar, B.Y. Zhao, Energy consumption and conservation in mobile peer-to-peer systems, in: Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare'06), ACM, New York, NY, USA, 2006, pp. 18–23, doi:10.1145/1161252.1161258.

[30] S. Rhea, D. Geels, T. Roscoe, J. Kubiatowicz, Handling churn in a DHT, in: Proceedings of the Annual Conference on USENIX Annual Technical Conference (ATEC'04), USENIX Association, Berkeley, CA, USA, 2004, p. 10.

[31] Z. Ou, E. Harjula, M. Ylianttila, Effects of different churn models on the performance of structured peer-to-peer networks, in: 2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications, IEEE, 2009, pp. 2856–2860, doi:10.1109/PIMRC.2009.5450134.

[32] H.-C. Hsiao, C.-T. King, Mobility churn in DHTs, in: 25th IEEE International Conference on Distributed Computing Systems Workshops, IEEE, 2005, pp. 799–805, doi:10.1109/ICDCSW.2005.98.

[33] J. Li, J. Zhang, Z. Cao, W. Pei, Genre: A general replication scheme over an abstraction of dhts, in: 2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC), IEEE, 2013, pp. 43–52, doi:10.1109/COMPSAC.2013.10.

[34] O. Herrera, T. Znati, Modeling churn in P2P networks, in: 40th Annual Simulation Symposium, 2007 (ANSS'07), IEEE, 2007, pp. 33–40, doi:10.1109/ANSS.2007.28.

[35] R. Price, T.T.A. Dinh, G. Theodoropoulos, Analysis of a self-organizing maintenance algorithm under constant churn, in: Proceedings of the 2008 International Symposium on Applications and the Internet (SAINT'08), IEEE Computer Society, Washington, DC, USA, 2008, pp. 209–212, doi:10.1109/SAINT.2008.65.

[36] Z. Ou, E. Harjula, O. Kassinen, M. Ylianttila, Performance evaluation of a Kademlia-based communication-oriented P2P system under churn, Comput. Netw. 54 (5) (2010) 689–705, doi:10.1016/j.comnet.2009.09.022.

[37] O. Herrera-Ruiz, T. Znati, Performance of redundancy methods in P2P networks under churn, in: 2012 International Conference on Computing, Networking and Communications (ICNC), IEEE, 2012, pp. 327–331, doi:10.1109/ICCNC.2012.6167437.

[38] S. Kim, G. Park, J. Hong, Y. Cho, Adaptive link recovery scheme for structured peer-to-peer networks under churn, in: Proceedings of the 2011 ACM Symposium on Applied Computing (SAC'11), ACM, New York, NY, USA, 2011, pp. 611–615, doi:10.1145/1982185.1982315.

[39] M. Sànchez-Artigas, P. García-López, A. Skarmeta, A comparative study of hierarchical DHT systems, in: 32nd IEEE Conference on Local Computer Networks, 2007 (LCN'07), IEEE, 2007, pp. 325–333, doi:10.1109/LCN.2007.79.

[40] E. Anceaume, R. Ludinard, B. Sericola, Analytic study of the impact of churn in cluster-based structured P2P overlays, in: 2010 IEEE International Conference on Communications (ICC), IEEE, 2010, pp. 1–5, doi:10.1109/ICC.2010.5501827.

[41] H. Shen, G. Liu, L. Ward, A proximity-aware interest-clustered p2p file sharing system, IEEE Trans. Parallel Distrib. Syst. 26 (6) (2015) 1509–1523, doi:10.1109/TPDS.2014.2327033.

[42] M. Khan, L. Yeh, K. Zeitouni, C. Borcea, Mobistore: Achieving availability and load balance in a mobile p2p data store, in: 2014 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), IEEE, 2014, pp. 171–172, doi:10.4108/icst.mobicase.2014.257793.

[43] J. Kintner, Z. Mildon, S. Kim, Tiny bubbles: A fault-tolerant distributed overlay for networks with block-churn, in: Military Communications Conference, 2012 (MILCOM'12), IEEE, 2012, pp. 1–6, doi:10.1109/MILCOM.2012.6415835.

[44] M. Pandey, B. Chaudhary, A content-based publish/subscribe over two-tier dht utilizing domain ontology, in: 2011 IEEE International Conference on Advanced Information Networking and Applications (AINA), IEEE, 2011, pp. 300–306, doi:10.1109/AINA.2011.58.

[45] A. Mizrak, Y. Cheng, V. Kumar, S. Savage, Structured superpeers: Leveraging heterogeneity to provide constant-time lookup, in: Proceedings of the Third IEEE Workshop on Internet Applications (WIAPP'03), IEEE Computer Society, 2003, pp. 104–111, doi:10.1109/WIAPP.2003.1210294.

[46] T. Koskela, E. Harjula, O. Kassinen, M. Ylianttila, Robustness of a p2p community management system based on two-level hierarchical dht overlays, in: 2011 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2011, pp. 881–886, doi:10.1109/ISCC.2011.5983953.

[47] S. Zoels, Q. Hofstatter, Z. Despotovic, W. Kellerer, Achieving and maintaining cost-optimal operation of a hierarchical DHT system, in: IEEE International Conference on Communications, 2009 (ICC'09), IEEE, 2009, pp. 1–6, doi:10.1109/ICC.2009.5198690.

[48] OMNeT++ Community, http://www.omnetpp.org/ (accessed 25.03.13).

[49] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, I. Stoica, Towards a common API for structured peer-to-peer overlays, in: Proceedings of the 2nd International Conference on Peer-to-Peer Systems (IPTPS'03), Springer Berlin Heidelberg, 2005, pp. 33–44, doi:10.1007/978-3-540-45172-3_3.

[50] C.-L. Liu, C.-Y. Wang, H.-Y. Wei, Mobile Chord: Enhancing P2P application performance over vehicular ad hoc network, in: 2008 IEEE GLOBECOM Workshops, IEEE, 2008, pp. 1–8, doi:10.1109/GLOCOMW.2008.ECP.62.

[51] Z. Yao, D. Loguinov, Understanding disconnection and stabilization of Chord, in: 27th IEEE Conference on Computer Communications (INFOCOM'08), IEEE, 2008, pp. 1049–1057, doi:10.1109/INFOCOM.2008.159.

[52] J. Furness, M. Kolberg, Considering complex search techniques in DHTs under churn, in: 2011 IEEE Consumer Communications and Networking Conference (CCNC), IEEE, 2011, pp. 559–564, doi:10.1109/CCNC.2011.5766542.

[53] D. Liu, F. Chen, G. Yin, H. Wang, P. Zou, Lsb-chord: Load balancing in dht based p2p systems under churn, in: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 3, IEEE, 2010, pp. 466–470, doi:10.1109/ICCSIT.2010.5564528.

[54] D. Stutzbach, R. Rejaie, Understanding churn in peer-to-peer networks, in: Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement (IMC'06), ACM, New York, NY, USA, 2006, pp. 189–202, doi:10.1145/1177080.1177105.

[55] T. Silverston, O. Fourmaux, Measuring P2P IPTV systems, in: Proceedings of International Workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV'07), 2007. http://www.nossdav.org/2007/files/file-24-session4-paper3-silverston.pdf.

[56] O. Landsiedel, S. Götz, K. Wehrle, A churn and mobility resistant approach for DHTs, in: Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking (MobiShare'06), ACM, New York, NY, USA, 2006, pp. 42–47, doi:10.1145/1161252.1161263.

[57] Z. Liu, R. Yuan, Z. Li, H. Li, G. Chen, Survive under high churn in structured P2P systems: evaluation and strategy, in: Proceedings of the 6th International Conference on Computational Science—Vol. Part IV (ICCS'06), Springer-Verlag, Berlin, Heidelberg, 2006, pp. 404–411, doi:10.1007/11758549_58.