PERFORMANCE ANALYSIS OF R/KADEMLIA, PASTRY AND BAMBOO USING RECURSIVE ROUTING IN MOBILE NETWORKS

Farida Chowdhury[1] and Md. Sadek Ferdous[2]

[1]Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh

[2]Electronics and Computer Science, University of Southampton, Southampton, UK

ABSTRACT

We are experiencing a revolution in the wireless technology and mobile phone networks as they are becoming increasingly ubiquitous. On the other hand, there is another trend, gaining extreme popularity over the traditional wired Internet: Peer-to-Peer (P2P) networking. Specifically, Distributed Hash Table (DHT) based Structured P2P overlay networks are capable to facilitate information storage and retrieval among a potentially very large number of users without the use of central server components. Providing such P2P services in mobile networks will allow a user to harness the potential of P2P services while on the move. However, due to technical issues in mobile phone networks and limited resources available on the handsets, designing such a system is a challenging issue. While the research of P2P overlays in wired networks is plenty, there is a limited study on the suitability of different existing P2P overlays for mobile networks. This article investigates the performance and efficiency of a number of structured P2P overlays that use recursive approach for routing strategy: R/Kademlia, Pastry and Bamboo and finds their suitability for use on mobile networks. The simulation result shows that among the three overlays, Bamboo exhibits a marginally better performance in comparison to the others with a 98% to 99% routing efficiency and a lower bandwidth consumption in high churn environment. Hence, Bamboo protocol is a good choice to be used over Kademlia and Pastry in mobile networks.

KEYWORDS

Peer-to-Peer, Distributed Hash Table, Churn, Mobile Networks

1. INTRODUCTION

The computing capabilities of mobile phones have improved considerably supporting a vast range of third party applications. Simultaneously, Peer-to-Peer (P2P) overlay networks have experienced a tremendous growth in terms of usage as well as popularity in recent years particularly in a fixed wired network. A P2P network is defined as a multi-node communication network that has neither central control nor coordination. Each participant (either a computer or a mobile device) in a P2P network is simply referred to as a peer and the data available on the network is distributed among all the peers on the network. Each peer simultaneously acts as a client and a server and has equivalent capabilities and responsibilities. Additionally, each peer can join and leave the system at any arbitrary time. A typical P2P architecture creates an overlay on top of the existing IP Network which is the base for data delivery. The overlay orientation is a logical view; therefore, it is not necessary to directly form a symmetrical view of the physical network. With the rapid growth of the Internet and the advancement of computing technologies, peers provide much more information and computing resources than from a limited number of centralized servers. Overlay networks can offer a better utilization of the Internet information and resources, as routing in overlay networks is very flexible and on the basis of different parameters,

DOI: 10.5121/ijcnc.2017.9504

they can detect and avoid network congestion quickly. Due to the flexibility in routing, the peers in overlay networks are highly connected to each other. Peers can communicate with each other using the overlay while the physical network connections exist.

P2P architectures can be broadly categorized as *Unstructured* and *Structured* based on the overlay topology formation. However, in this article, we have focused Structured P2P overlays only. This kind of P2P overlays connects peers using a particular data structure or protocol to ensure that queries are more efficient and data discovery can be guaranteed. Structured P2P overlays utilize a Distributed Hash Table (DHT) to distribute index information about the shared data items among the participating peers. Each node is assigned a unique ID (e.g. a hashed IP address) and is responsible for storing a range of keys which are closest to its node ID. Each data item is also assigned a file ID (e.g. the hash of a file name) and each node is responsible for a particular range of the file ID space. If a node queries a key, the network returns the node ID where the associated file with that key is stored. Some well-known examples of structured P2P overlays are Chord [1], Kademlia [2], Pastry [3], CAN [4] and Bamboo[5].

DHT based structured P2P overlay networks offer major advantages to users of mobile devices and networks such as scalable, fault tolerant and self-managing infrastructure which does not exhibit single points of failure. Providing such P2P services in mobile networks will allow a user to harness the potential of P2P service while on the move. Designing such a system poses a number of challenges, such as restrictions in terms of data transmissions on cellular networks, limited battery power of the handsets, high level of churn due to variations in signal strength and placing voice calls which suspend the data link.

This article focuses on structured P2P overlays on mobile cellular networks. We investigate three structured P2P overlays: R/Kademlia [6], Pastry [3] and Bamboo [5] which use recursive routing for data lookup and identify which is/are suitable for cellular networks and mobile devices. We evaluate and compare the performance of these three overlays using OverSim [7], a P2P overlay simulator.

The contributions of our article are:

- Identifying the most suitable configuration of three DHT based structured P2P overlays: R/Kademlia Pastry, and Bamboo to achieve the optimal performance in churn intensive mobile environments.
- Performance analysis of the aforementioned P2P overlays under high levels of churn and bandwidth consumption as required in mobile networks.

The rest of the article is organized as follows. Section 2 highlights the related work and section 3 presents a brief overview of the selected P2P protocols. Section 4 explains the experiment methodology where the simulation platform, simulation metrics, and the simulation setup are described. Section 5 discusses how the suitable and best parameter configurations for each DHT have been selected for use on mobile networks. Section 6 provides the performance evaluations. Finally, we summarize and conclude in section 7.

2. RELATED WORK

The performance of a number of DHT based P2P overlays in wired and wireless network has been reported in some studies [8-15].

In [8], the authors have presented a Performance Versus Cost (PVC) framework to evaluate the cost and performance of Chord, Kademlia Kelips, One Hop and Tapestry; and showed that if parameters are optimized, all of these measured DHTs can achieve a reasonable performance.
A comprehensive study of a number of DHT based P2P overlays, has been presented in [9]. The authors have compared Chord, Koorde, Pastry, Bamboo, Kademlia and Broose where the evaluation methodology was based on the PVC evaluation framework. This work has focused on the stable fixed networks. However, we have studied P2P overlays under conditions as found in cellular networks such as high levels of churn and limited bandwidth consumption; and therefore identify suitable overlays for mobile networks. Our evaluation also focuses on recursive routing based P2P overlays and includes recursive Kademlia which has not been included in [9].

In [10], the authors have analyzed Chord, Kelips and Tapestry under high churn rate and considered the applicability over wireless networks. In [11], the authors have presented an analytical evaluation and comparison of Chord, Pastry, CAN, Tapestry, Bamboo and Kademlia from the viewpoint of interpersonal communication.

In [12], the authors have provided directions to design a DHT under churn. They have conducted an analytical study of three features - lookup strategy, lookup parallelism and lookup key replication to improve the DHT lookup performance under churn. They have compared the performance of two lookup strategies: Recursive Routing (RR) and Iterative Routing (IR). Furthermore, they have proposed two enhanced alternatives. One is a two-phase routing strategy (RR+IR), where the RR is used in the first phase and if RR fails, IR is conducted as the second phase. Another one is Recursive Routing with ACK strategy (RR+ACK), where an ACK is sent by an intermediate node, containing the address of the next hop to the originator along with the RR. Through simulation, the authors have shown that the RR+ACK technique performs better than others in different churn conditions. For lookup parallelism, the authors have shown that a parallelism degree of two or three can increase the overall lookup performance. Lookup Key Replication is another important feature in which one data key is replicated in multiple peers. They studied two replication policies: replication without repair mechanism and replication with repair mechanism and found that the necessity of a repair mechanism depends on the peer lifetime distribution. In Exponential Distribution, a repair mechanism is required no matter what the churn level is; however, in Pareto Distribution, if there is a very heavy tail, simple replication is enough to guarantee a sufficient long lifetime of the replicated key.

In one of our previous works [13, 14], we have evaluated the performance and efficiency of five popular DHT based structured P2P overlays: Chord, Pastry, Kademlia, Broose and EpiChord, considering the conditions as seen in mobile networks. Our experimentation has shown that Kademlia and EpiChord are good candidates for use in high churn mobile environments. In those papers, we have considered the P2P overlays which only use iterative routing for lookups. However, in this article, all experiments have been carried out using recursive routing, instead of iterative routing. Furthermore, we have experimented with Bamboo which was not repeated previously.

3. Protocol Overviews

This section provides a brief summary of the evaluated three DHT overlays.

3.1. R/Kademlia

R/Kademlia [6] is a recursive version of the popular P2P overlay Kademlia [2]. The only difference with Kademlia is in the routing and lookup procedure. Instead of using iterative routing, R/Kademlia uses recursive routing. Similar to Kademlia, R/Kademlia assigns a NodeId to each node in a 160-bit key space and maintains a routing table consisting of *k-buckets*. Each *k-bucket* contains a maximum of *k* entries with <IP address, UDP port, NodeId> triples of other nodes. The routing table forms a binary tree where nodes are represented as leaves and the *k-bucket* is represented as a series of sub-trees that do not contain the node itself.

3.2. PASTRY

Pastry [3] consists of a circular identifier space where each node and data item are assigned a 128-bit identifier with base 2^b (*b* is a configuration parameter of the algorithm), using a consistent hash function. Each Pastry node uses a routing table consists of $log_{2b}N$ rows and 2^b - *1* columns, where the entries in row *n* contain routing information of all nodes whose identifier has the same *n* first digits with the present node. In addition to the routing table, Pastry nodes maintain a *Leaf set* and a *Neighborhood set*. The leaf set *L* contains the *L/2* nodes with numerically closest larger nodes and *L/2* nodes with numerically closest smaller nodes, NodeId relative to the current node's nodes, The leaf set is used to store object replica and ensures reliable message delivery. The neighborhood set *M* contains the closest nodes in terms of proximity to the current node.

3.3. BAMBOO

The structure of Bamboo [4] is based on Pastry. It also assigns a unique 160-bit node ID, where the node IDs are uniformly distributed and generated by using a secure hash algorithm consisting of either a public key or the IP address and port number combination. Each node in Bamboo uses two sets of neighbor information: *leaf set* and *routing table* information. The leaf set contains the immediate successors and predecessors that are in the numerically closest in circular key space. A routing table consists of nodes that share a common prefix and is used to improve the lookup performance.

4. EXPERIMENTAL METHODOLOGY

4.1. SIMULATION PLATFORM

For all experiments throughout the article, the OverSim [7] platform has been used, due to its strong support of DHT based P2P algorithms. It provides a generic lookup mechanism and an RPC interface to facilitate additional protocol implementations. It allows large-scale simulations of simplified networks as well as complex heterogeneous underlay networks. It also provides an interactive GUI and real-time messages which are extremely useful for debugging. Overeem has been used in a number of other P2P works, making the results easily comparable to other researchers' works [9].

4.2. PERFORMANCE METRICS

The following performance metrics are measured from the simulation results:

- **Lookup Success Ratio:** The percentage of successful lookups in the overlay.
- **Mean Maintenance Traffic:** The mean number of sent maintenance bytes per second by each node. This metric will be considered as a parameter for bandwidth consumption in this article.

4.3. Simulation Setup

Throughout the experimentation discussed in this article, the following setup has been used, unless stated otherwise.

- **Network Size:** Experiments have been carried out using a 1,000 node network.
- **Churn:** The level of churn has been simulated by various node lifetime values between 100 seconds, 600 seconds and 1,000 seconds. A shorter lifetime means a higher level of churn and vice versa. The node lifetime of 100 seconds has been selected to simulate high churn scenarios according to the statistics presented in [16].
- **Churn Model:** The simulations use Overeem's Lifetime Churn model with Weibull Distribution, as this distribution has been shown to model churn in P2P overlays much more accurately [17].
- **Transport Protocol:** The selected P2P overlays originally use only UDP as a transport protocol. Therefore, UDP has been used in the experiments.
- **Routing:** The overlays have been configured to use Recursive routing.
- **Repetition:** Each simulation configuration has been repeated 5 times, and results have been averaged. To increase the credibility of the simulation results, the 95% confidence intervals have been calculated across the repetitions in the figures [18].

5. Best Parameter Selection

To select the best parameters, the simulation process has been done using two steps where in the first step, a wide range of parameter values of R/Kademlia, Pastry and Bamboo have been used as input. This step determines the values of key parameters for each overlay to exhibit the best possible performance under high churn. In the second step, we have compared the overlays with each other considering the selected ‘best’ configured parameter set obtained from the first step to evaluate the performance under high churn.

In the next subsections, we presented the simulation results concerning the first step.

5.1. R/Kademlia

For R/Kademlia, Table 1 lists the key parameters, which are *bucket size (k), siblings (s), bits per digit (b), lookup redundant nodes (r)* and *bucket refresh interval*, along with the varied range of values.

Table 1. R/Kademlia: Simulation parameters

P2P Protocol	Parameters	First step	Second step
R/Kademlia	Bucket size, k	2, 4, 8, 16, 20, 32, 40	4
	Number of siblings, s	2, 4, 8, 16, 32	16
	Bits per digit, b	1, 2, 4, 6, 8	2
	Lookup redundant nodes	1, 2, 4, 8, 16	1
	Bucket refresh interval (sec)	100, 600, 1000, 2000, 5000	1000

In the first step of the simulation, we have experimented R/Kademlia using different parameters as listed in the Table 1.

Each node in R/Kademlia maintains a routing table which consists of *k* number of buckets. Each bucket also holds *k* number of references to reach other nodes. Figure 1 shows the results for success ratio and bandwidth consumption (that is, sent maintenance bytes/s) with varying valuer *k* over different churn rates (i.e. lifetime 100s, 600s and 1000s). To achieve the best performance, the trade-off between a higher success ratio and the decreased maintenance traffic has been evaluated, and therefore the value of 4 was chosen for the bucket size.

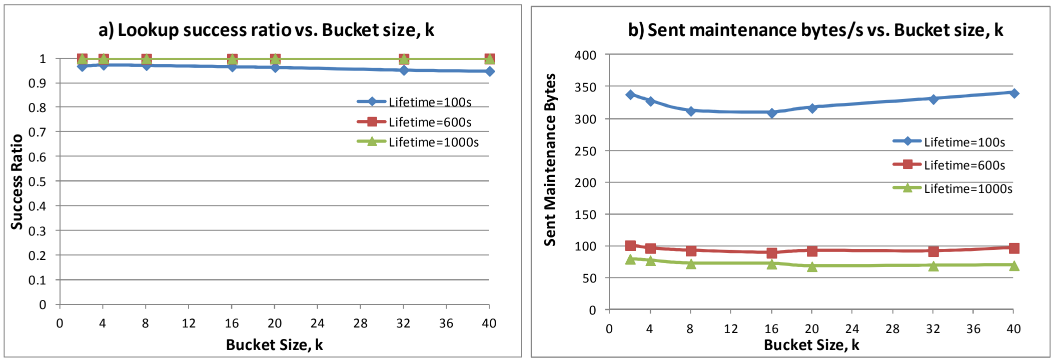

Figure 1: R/Kademlia: Testing on *bucket size, k* over different churn rates; a) Lookup success ratio vs. bucket size and b) Sent maintenance bytes/s vs. bucket size

In R/Kademlia, siblings (denoted with parameter *s*) are responsible for a certain (key, value) pair that is needed to be stored in the hash table. Comparing different values of *s*, it has been found that 16 sibling nodes are optimal to achieve the best performance in high churn scenarios according to lookup success ratio and bandwidth consumption (Figure 2).

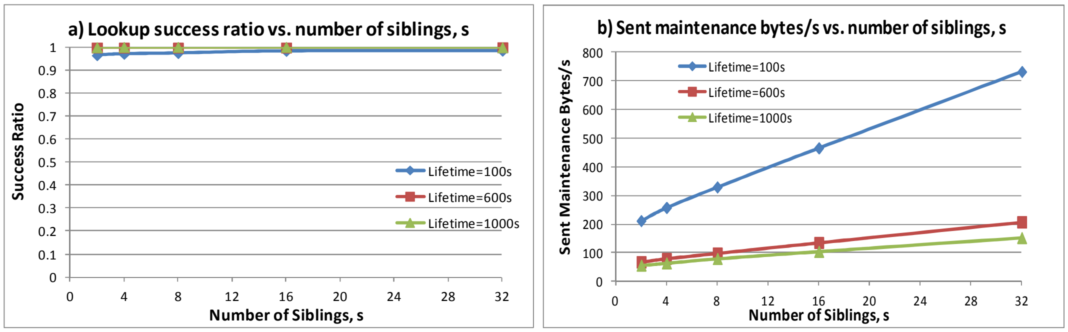

Figure 2: R/Kademlia: Testing on *siblings, s* over different churn rates; a) Lookup success ratio vs. number of siblings and b) Sent maintenance bytes/s vs. number of siblings

The number of bits per digit (*b)* has been varied in Figure 3. The results in the figure exhibit that the success ratio remains quite the same with higher values; however, the maintenance traffic increases drastically. Therefore, a value of 2 is appropriate.

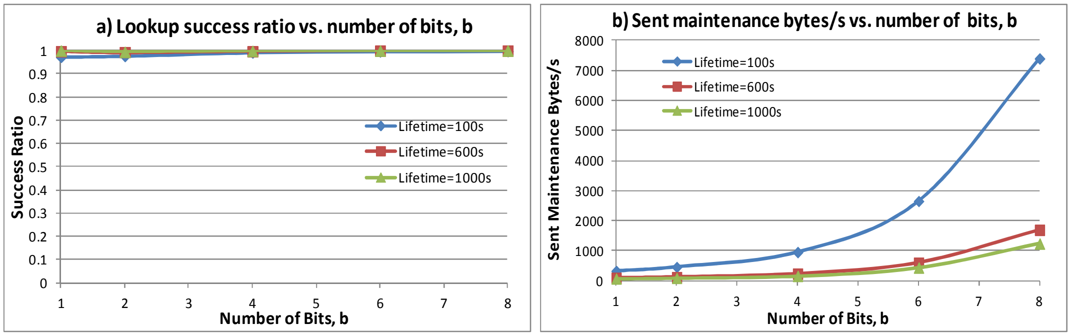

Figure 3: R/Kademlia: Testing on *bits per digits, b* over different churn rates; a) Lookup success ratio vs. bits per digit and b) Sent maintenance bytes/s vs. bits per digit

In R/Kademlia, the parameter *bucket refresh interval* is used to keep the bucket up-to-date and current. Figure 4 shows the experiment results on bucket refresh interval over different churn rates. An interval of 1000 sec has been selected to keep the buckets consistent.

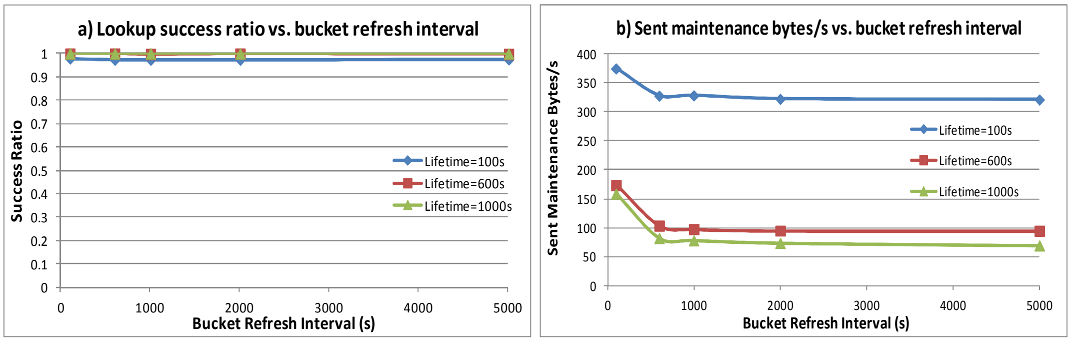

Figure 4: R/Kademlia: Testing on *bucket refresh interval* over different churn rates; a) Lookup success s ratio vs. bucket refresh interval and b) Sent maintenance bytes/s vs. bucket refresh interval

Finally, in Figure 5, the number of redundant nodes has been altered. It is evident from the result (shown in the figure) that increasing the number of redundant nodes does not impact the lookup success ratio and maintenance bytes that much, therefore, only 1 redundant node is sufficient to achieve the best result.

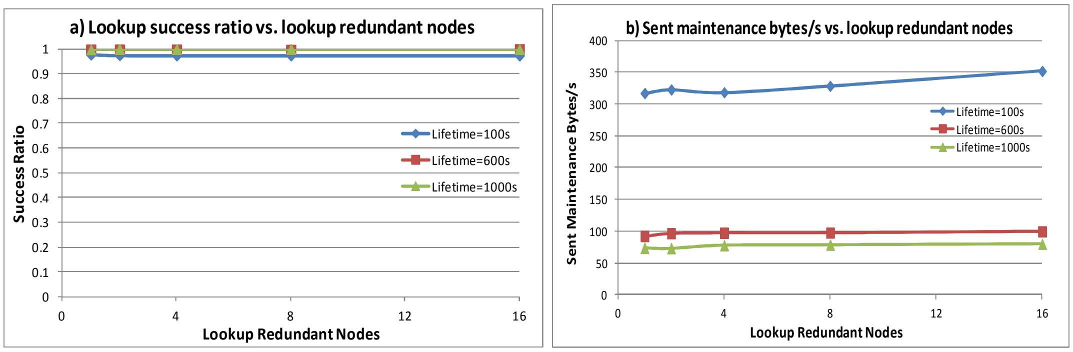

Figure 5: R/Kademlia: Testing on *lookup redundant nodes* over different churn rates; a) Lookup success ratio vs. lookup redundant nodes and b) Sent maintenance bytes/s vs. lookup redundant nodes

5.2. Pastry

Table 2 shows the key parameters of Pastry which are *bits per digit, number of leaves* and *number of neighbors*.

Table 2. Pastry: Simulation parameters

P2P Protocol	Parameters	First step	Second step
Pastry	Bits per digit	1, 2, 4, 6, 8	2
	Number of leaves	4, 8, 16, 32	8
	Number of neighbors	0, 2, 4, 8, 16	0

The results of varying levels of *bits per digit* are shown in Figure 6. Increasing the number of bits does not affect the lookup success ratio but it significantly increases the maintenance cost. Therefore, a value of 2 for bits per digit has been selected.

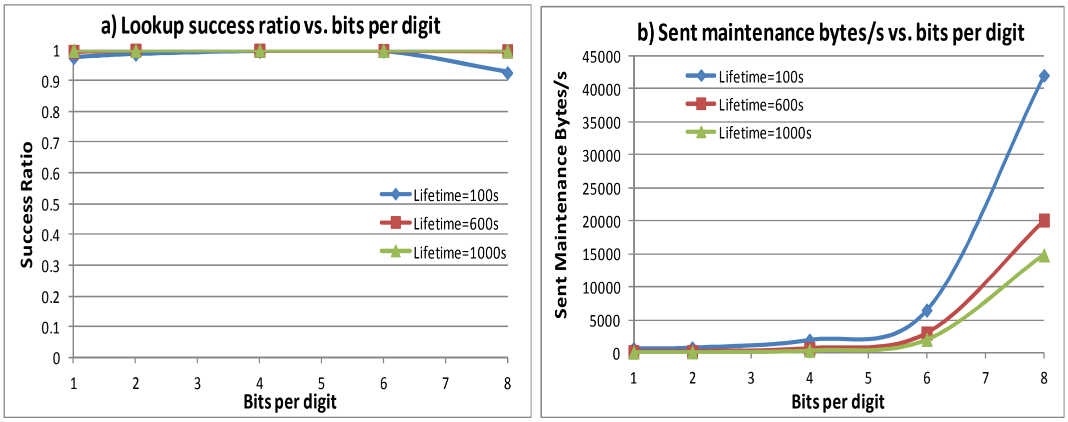

Figure 6: Pastry: Testing on *bits per digit* over different churn rates; a) Lookup success ratio vs. bits per digit and b) Sent maintenance bytes/s vs. bits per digit

Figure 7 shows the effect of different *numbers of leaf* nodes. In medium low churn scenarios, there is no significant performance difference, whereas, we observed the highest success ratio at the value of 8 in high churn scenarios. Hence, the number of leaf nodes of 8 was selected.

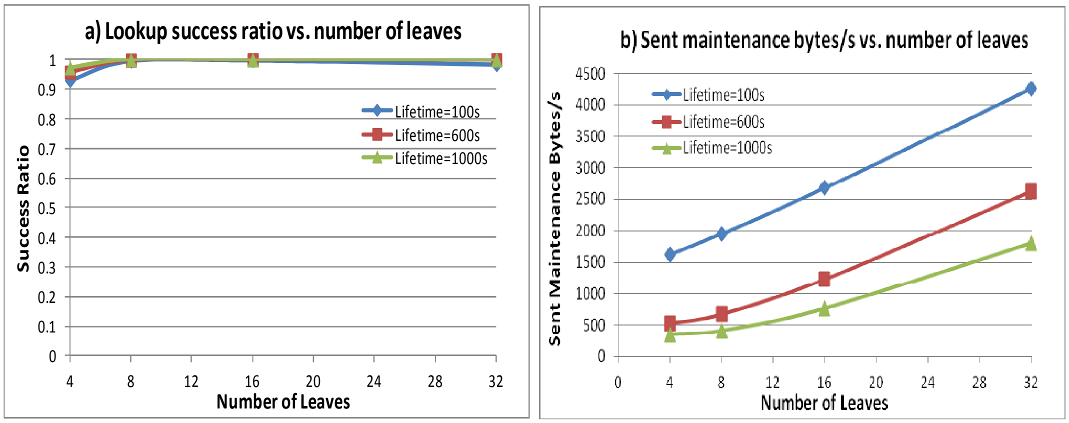

Figure 7: Pastry: Testing on *number of leaves* over different churn rates; a) Lookup success ratio vs. number of leaves and b) Sent maintenance bytes/s vs. number of leaves

In Pastry, neighbor nodes contain nearby nodes based on proximity metrics. Figure 8 shows the simulation results with different number of neighbor nodes. Using a neighbor node does not result in considerable improvements in the success ratio and maintenance cost. Therefore, we have decided not to use any neighbors.

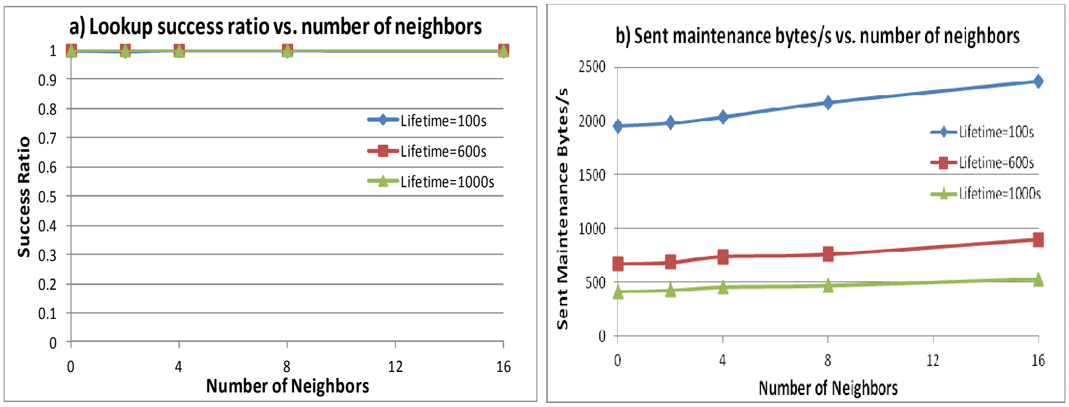

Figure 8: Pastry: Testing on *number of neighbors* over different churn rates; a) Lookup success ratio vs. number of neighbors and b) Sent maintenance bytes/s vs. number of neighbors

5.3. BAMBOO

Bamboo has been experimented with its key parameters: leaf set size, bits per digit, leaf set maintenance interval, local tuning interval and global tuning interval, as shown in Table 3. Figure 9 to 13 illustrate the success ratio and bandwidth consumption for different parameters of Bamboo.

Table 3. Bamboo: Simulation parameters

P2P Protocol	Parameters	First step	Second step
Bamboo	Number of leaves	4, 8, 16, 32	16
	Bits per digit	1, 2, 4, 6, 8	6
	Leaf set maintenance interval (sec)	5, 10, 30, 60, 120, 180, 300	5
	Local tuning interval (sec)	5, 10, 20, 30, 60, 120, 180, 300	5
	Global tuning interval (sec)	5, 10, 20, 30, 60, 120, 180	5

The *number of leaves* and number of *bits per digit* were varied in Figure 9 and 10 respectively. To achieve the best performance, the traded-off between a higher success ratio and the decreased maintenance traffic has been evaluated. Thus, the value of 16 for leaf set and 6 for the bits per digit have been selected.

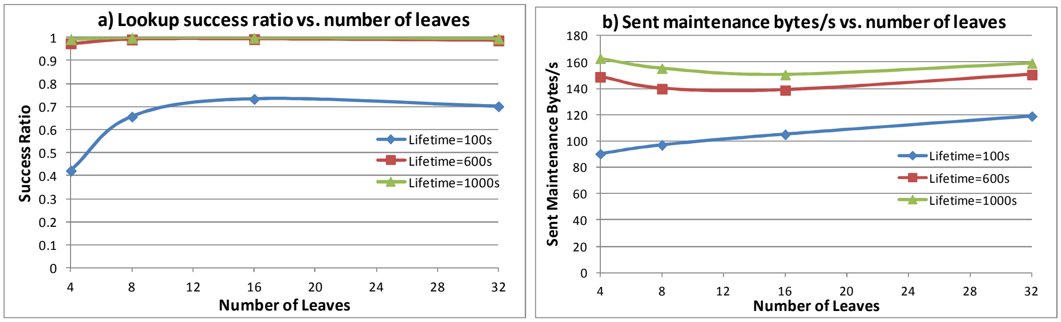

Figure 9: Bamboo: Testing on *number of leaves* over different churn rates; a) Lookup success ratio vs. number of leaves and b) Sent maintenance bytes/s vs. number of leaves

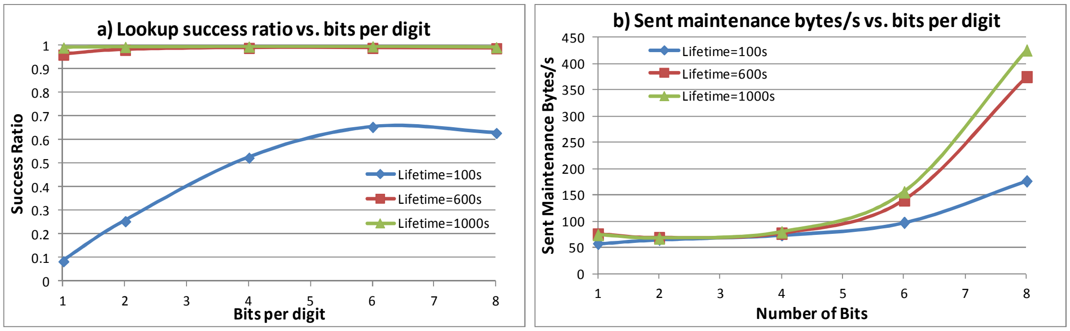

Figure 10: Bamboo: Testing on *bits per digit* over different churn rates; a) Lookup success ratio vs. bits per digit and b) Sent maintenance bytes/s vs. bits per digit

In Figure 11, the *leaf set maintenance interval* has been altered. As shown in the Figure 11(a), increasing the leaf set maintenance interval time decreases the success ratio in high churn scenario (i.e. lifetime = 100 sec), an interval of 5 sec has been chosen.

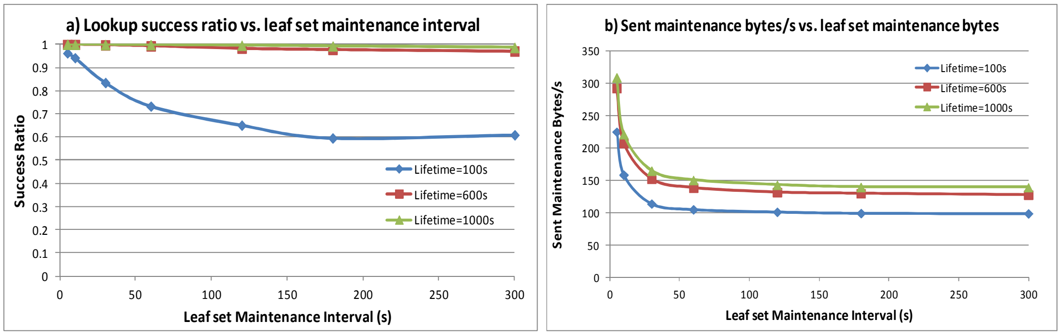

Figure 11: Bamboo: Testing on *leaf set maintenance interval* over different churn rates; a) Lookup success ratio vs. leaf set maintenance interval and b) Sent maintenance bytes/s vs. leaf set maintenance interval

The results of varying levels of *local tuning interval* and *global tuning interval* have been shown in Figure 12 and 13 respectively. Increasing the interval time cannot improve the success ratio as well as maintenance bytes. Hence, we have selected 5 sec for local tuning interval and 5 sec for global tuning interval.

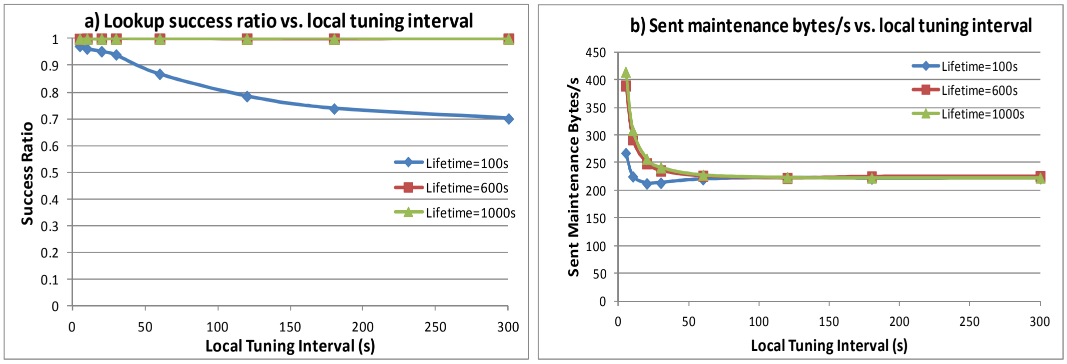

Figure 12: Bamboo: Testing on *local tuning interval* over different churn rates; a) Lookup success ratio vs. local tuning interval and b) Sent maintenance bytes/s vs. local tuning interval

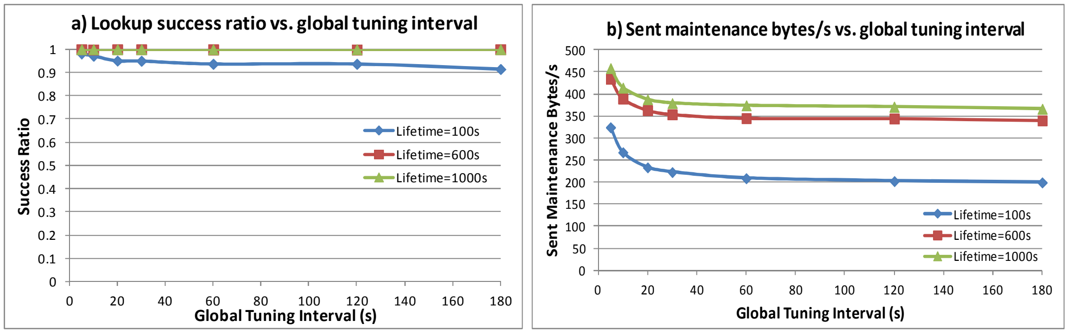

Figure 13: Bamboo: Testing on global tuning interval over different churn rates; a) Lookup success ratio vs. global tuning interval and b) Sent maintenance bytes/s vs. global tuning interval

6. Performance Analysis

This section provides the performance analysis of the selected P2P overlays with the most suited parameters obtained from the previous step under high level of churn according to lookup success rate and bandwidth consumption.

6.1. Effects Of High Churn

In the high churn scenario which a closely matched lookup success ratio in medium churn environment (i.e. for a node lifetime of 600 sec). However, for a node lifetime of 1000 sec, Bamboo exhibits nearly 99.99% success ratio, whereas, Pastry and R/Kademlia general, 98.62%, i.e is for a node lifetime of 100 sec; R/Kademlia, Pastry and Bamboo exhibit 98.67and 98.15% success ratio respectively (Figure 14). Similarly, all the three overlays exhibit environment and suitable for use in churn intensive mobile networks. In this section, the performance of the three overlays: R/Kademlia, Pastry and Bamboo under varying levels of churn have been evaluated. Figure 14 shows the lookup success ratio under different levels of churn from 100 sec to 1,000 sec. .

.

In the high% 99.83% and99.82%. and 98.15% success ratio respectively (Figure 14). However, for a node lifetime of 1000 sec, Bamboo exhibits nearly 99.99% success ratio, whereas, Pastry and R/Kademlia generate 99.83% and 99.82% success ratio respectively. This clearly shows that all these three overlays are quite robust even in high churn

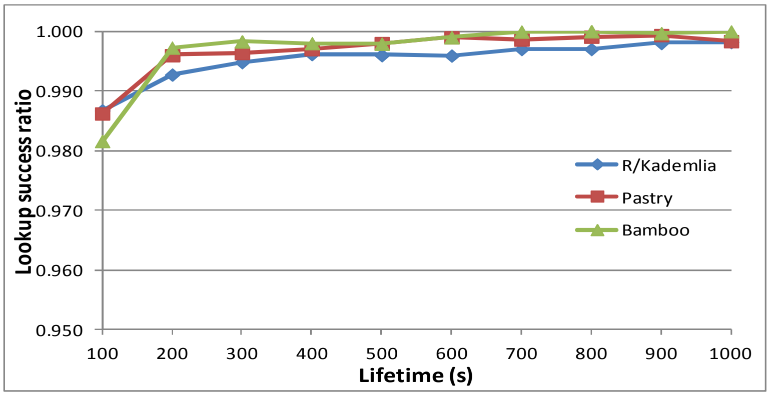

Figure 14: Lookup success ratio of R/Kademlia, Pastry and Bamboo operated on a 1,000 node networks under various levels of churn.

6.2. Effects Of Bandwidth Consumption

Band width We also have analyzed the bandwidth consumption of three aforementioned overlays according to high churn.

The simulation results for bandwidth consumption of R/Kademlia, Pastry and Bamboo are shown in Figure 15. The sent maintenance bytes have been measured under varying levels of churn to investigate how much is consumed by each overlay.

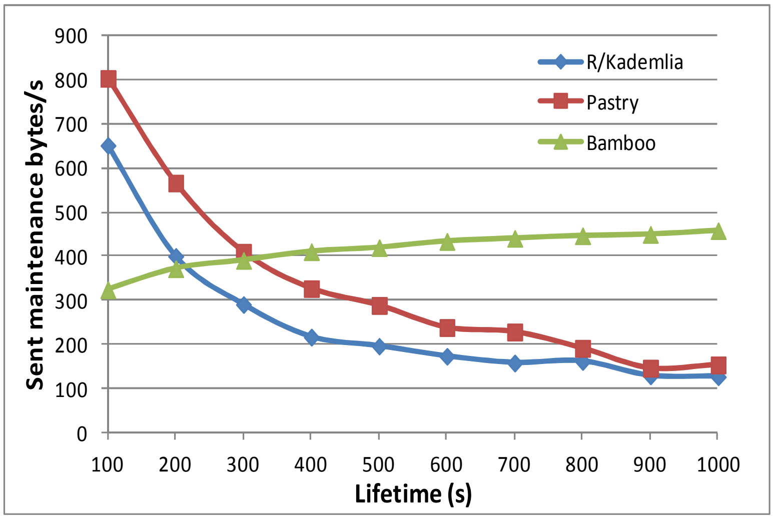

Figure 15: Bandwidth consumption of R/Kademlia, Pastry and Bamboo operated on a 1,000 node networks under various levels of churn.

In a high churn environment when the node lifetime is 100 seconds, Bamboo has the lowest bandwidth cost using 324 bytes/s. Whereas, R/Kademlia and Pastry consume 651 bytes/s and 80s/s respectively under high churn. In medium and low churn scenario (i.e. when the node life times are 600 sec and 1000 sec), R/Kademlia and Pastry consume less bandwidth as the overlays become more stable during medium churn environment and they use less amount of bandwidth. R/Kademlia uses 173 bytes/s and 127 bytes/s under medium and low churn scenarios respectively, whers, Pastry consumes 238 bytes/s and 154 bytes/s when the node life times are 600 sec and 1000 sec respectively. It is noticeable that Bamboo behaves differently, unlike R/Kademlia and Pastry overlays compared here, in scenarios when the node lifetime increases the maintenance traffic slowly increases instead of decreasing. It is due to the fact that Bamboo does the aknowledgements and periodic maintenance and therefore it sends some other time related messages to update the routing table which increases the bandwidth as nodes stay in the network for longer.

6.3. Discussion

these all three overlays are well suited for mobile environments. However, Bamboo demonstrates a marginally better performance in comparison to the others with a 98% to 99% routing efficiency and a lower bandwidth consumption in high churn From the simulation results, we can establish the following observations:

- Considering the lookup success ratio, all the three overlays: R/Kademlia, Pastry and Bamboo exhibit quite the similar performance and almost 98% lookup success ratio at very high churn and therefore environment.
- Regarding bandwidth consumption, pastry consume highest bandwidth (802 bytes/s) in high churn environments. R/Kademlia shows moderate bandwidth consumption (651 bytes/s) and Bamboo consumes the lowest bandwidth (324 bytes/s) in high churn. However, in low churn environment, R/Kademlia and Pastry become stable and generate 127 bytes/s and 154 bytes/s respectively. Conversely, Bamboo uses a bit high bandwidth (457 bytes/s) during low churn environment.

7. CONCLUSIONS

Peer-to-Peer (P2P) overlay networks are widely used for their self-managing infrastructure and highly scalable features. Extending P2P in mobile networks can provide users the advantages of both technologies: P2P and wireless networks. However, enabling P2P applications in mobile networks poses a number of challenges, such as high level of churn and bandwidth consumption. This article presents a performance evaluation of R/Kademlia, Pastry and Bamboo using recursive routing in the presence of high churn and investigated their suitability for mobile networks. The simulation results suggest Bamboo as the most appropriate P2P overlay to implement on the mobile network according to look up success ratio under high churn environment. Bamboo also consumes a moderate level of bandwidth compared to Pastry and R/Kademlia which is also acceptable in mobile networks.

REFERENCES

[1] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. “Chord: A scalable peer-to-peer lookup service for internet applications.” In *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’01)*, San Diego, California, United States; ACM, 2001.

[2] Maymounkov, P. and Mazières, D. “Kademlia: A Peer-to-Peer Information System Based on the XOR Metric.” In *Proceedings of the International Workshop on Peer-to-Peer Systems*. Springer, Berlin, Heidelberg, March 2002.

[3] Rowstron, A. and Druschel, P. “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 and Open Distributed Processing*, Springer, Berlin, Heidelberg, November 2001.

[4] Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S. “A Scalable Content-addressable Network.” In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM ’01)*, San Diego, California, United States; ACM, 2001.

[5] Rhea, S., Geels, D., Roscoe, T. and Kubiatowicz, J. “Handling churn in a DHT.” In *Proceedings of the USENIX Annual Technical Conference* (Vol. 6, pp. 127-140), June 2004.

[6] Heep, B. “R/Kademlia: Recursive and topology-aware overlay routing.” In *Proceedings of the Telecommunication Networks and Applications Conference (ATNAC), Australasian*, IEEE, 2010.

[7] Baumgart, I., Bernhard, H. and Stephan, K. “OverSim: A flexible overlay network simulation framework.” In *Proceedings of the IEEE Global Internet Symposium,* IEEE, May 2007.

[8] Li, J., Stribling, J., Morris, R., Kaashoek, M.F. and Gil, T.M.. “A performance vs. cost framework for evaluating DHT design tradeoffs under churn”. In *Proceedings of the 24th IEEE Infocom* (Vol. 1, pp. 225–236), March 2005.

[9] Baumgart, I. and Heep, B. “Fast but economical: A simulative comparison of structured peer-to-peer systems.” In *Proceedings of the Next Generation Internet (NGI), 8th EURO-NGI Conference on* (pp.87-94). IEEE, June, 2012.

[10] Chan, H. N., Van K. N., Hoang, G. N. “Characterizing Chord, Kelips and Tapestry algorithms in P2P streaming applications over wireless network”, In *Proceedings of the International Conference on Communications and Electronics ICCE 2008*, Hoi An, Vietnam, June 2008.

[11] Hautakorpi, J. and Gonzalo, C. “Evaluation of DHTs from the viewpoint of interpersonal communications.” In *Proceedings of the 6th international conference on Mobile and ubiquitous multimedia*. ACM, 2007.

[12] Di, W., Tian, Y. and Ng, K.W. “Analytical study on improving DHT lookup performance under churn.” In *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing, 2006.* IEEE, 2006.

[13] Chowdhury, F. and Kolberg, M. “Performance evaluation of structured peer-to-peer overlays for use on mobile networks.” In *Proceedings of the 6th International Conference on Developments in eSystems Engineering (DeSE),* pp. 57-62, IEEE, December 2013.

[14] Chowdhury, F., Furness, J. and Kolberg, M. “Performance analysis of structured Peer-to-Peer overlays for mobile networks.” In *International Journal of Parallel, Emergent and Distributed Systems*, pp.1-27, July 2016.

[15] Chowdhury, F. and Kolberg, M. “Performance Evaluation of EpiChord under High Churn.” In *Proceeding of the 8th ACM Performance Monitoring, Measurement and Evaluation of Heterogeneous Wireless and Wired Networks (PM2HW2N) Workshop*, Barcelona, Spain, 2013.

[16] Dave, H. “Time in app increases by 21 across all apps.” Available at http://info.localytics.com/blog/time-in-app-increases-by-21-across-all-apps. Accessed on 21 May 2017.

[17] Stutzbach, D. and Rejaie, R. “Understanding churn in peer-to-peer networks.” In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (pp. 189-202), ACM, October 2006.

[18] Pawlikowski, K., Jeong, H.D. and Lee, J.S. “On credibility of simulation studies of telecommunication networks.” In *IEEE Communications magazine*, *40*(1), pp.132-139, 2002.

AUTHORS

Farida Chowdhury is an Assistant Professor at the department of Computer Science and Engineering of the Shahjalal University of Science & Technology, Bangladesh. She received her PhD degree at the Institute of Computing Science and Mathematics at the University of Stirling, Scotland. Her research interests focus on Network security, P2P networks, Pervasive computing, Next-Generation Wireless Networks, Wireless Sensor Networks, Internet of Things and Social Networks.

Md Sadek Ferdous is a Research Fellow at the Electronics and Computer Science of the University of Southampton. He holds a PhD in the area of Mobile Identity Management at the School of Computing Science of the University of Glasgow in 2015. His research interest includes Network Security, Distributed Ledger, Identity Management, Trust Management and Privacy Enhancing Technologies.