Routing performance of structured overlay in Distributed Hash Tables (DHT) for P2P

Article *in* Bulletin of Electrical Engineering and Informatics · March 2019

DOI: 10.11591/eei.v8i2.1449

Routing performance of structured overlay in Distributed Hash Tables (DHT) for P2P

Rafiza Ruslan[1], Ayu Shaqirra Mohd Zailani[2], Nurul Hidayah Mohd Zukri[3], Nur Khairani Kamarudin[4], Shamsul Jamel Elias[5], R. Badlishah Ahmad[6]
[1,2,3,4]Department of Computer Science, Universiti Teknologi MARA Perlis, Perlis, Malaysia
[5]Universiti Teknologi MARA (UiTM) Kedah, Malaysia
[6]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Malaysia

Article Info

Article history:

Received Oct 16, 2018
Revised Dec 15, 2018
Accepted Mar 12, 2019

Keywords:

Chord
DHT
Kademlia
P2P
Pastry

ABSTRACT

This paper presents a routing performance analysis of structured P2P overlay network. Due to the rapid development and hectic life, sharing data wirelessly is essential. P2P allows participating peers move freely by joining and leaving the network at any convenience time. Therefore, it exists constraint when one measuring the network performance. Moreover, the design of structured overlay networks is fragmented and with various design. P2P networks need to have a reliable routing protocol. In order to analyse the routing performance, this work simulates three structured overlay protocols-Chord, Pastry and Kademlia using OMNeT++ with INET and OverSim module. The result shows that Pastry is the best among others with 100% routing efficiency. However, Kademlia leads with 12.76% and 18.78% better than Chord and Pastry in lookup hop count and lookup success latency respectively. Hence, Pastry and Kamelia architectures will have a better choice for implementing structured overlay P2P network.

Corresponding Author:

Rafiza Ruslan,
Department of Computer Science,
Universiti Teknologi MARA Perlis, Perlis, Malaysia.
Email: rafiza.ruslan@perlis.uitm.edu.my

1. INTRODUCTION

Peer-to-peer (P2P) systems are widely used for data sharing that exploits a self-organization for optimizing resource allocation. P2P is an example of overlay networks that implements a network service that is not available in the physical network. In a simplest form, P2P network is connected between two or more PCs in a decentralized environment similar to Mobile Ad-Hoc Network (MANET) that consist of nodes communicate each other without any fixed infrastructure through the wireless devices [1]. P2P overlay networks are divided into two classes which is *unstructured* and *structured* [2]. Unstructured P2P networks use flooding search protocols where the demand is communicated to all companions inside a specific range with Time to Live (TTL) instrument and also have no precise control over resource emplacements such like Gnutella. Other examples of application file sharing that apply P2P system are Napster, Morpheus, Freenet, BitTorrent and Skype. On the other hand, structured P2P network refers to the tightly controlled topology that is maintain over a network graph where resources are not placed randomly on nodes but in a deterministic manner, i.e. using Distributed Hash Table (DHT). P2P need to have better structured overlay algorithm for P2P due to the increasing number of users in P2P network [3]. DHT is one of the distributed memory method where it distributes the contructions of hash table [4]. In DHT based system, identifier (Id) for each peer and item is generated using hash function. The mapping of Ids into set of peers of structure overlay is done by DHT in a distributed fashion; called routing or lookup [5]. Structured overlay network protocol that use DHT are Chord, Pastry, Kademlia etc [6].

Chord is the first DHT-based structured P2P overlay protocol [7]. It uses consistent hashing and hashes each peer and key into a same m-bit circular identifier (Id) space, where m is a system parameter [7]. Chord peer uses routing tables to route the lookup message to the destination peer in O(log N) complexity, where N is total number of peers. Chord uses periodic stabilization mechanism to correct the routing table periodically. Pastry organizes each peers in 128-bit circular Id space that uses prefix matching to route the messages in O(logb N) complexity, where N is total number peers and b is system parameter [8]. Neighbouring peers periodically exchange keep-alive messages in order to maintain the routing table. Finally, Kademlia organizes each peer by assigning 160-bit Id and the each peer contains list of entries known as routing buckets [6]. Kademlia uses XOR based closeness algorithm to calculate the closest peers, then sends parallel lookup messages to multiple buckets at complexity of O(logb N), where N is total number of peers and b is system parameter. Kademlia peer employs maintenance algorithm which periodically checks the routing buckets through which there has not been a lookup since last stabilization. Kademlia only ensures that at least one entry in each bucket is alive since last stabilization, where as Chord and Pastry ensures all routing entries are alive and up to date.

Structured overlays use its corresponding routing algorithm and the routing table to route the messages to the destination peer. The random join and leave of peers in P2P system causes the routing table to be incorrect and out of date. This routing timeout may results in lookup failure. The rate of failure is dependent on the rate of peer join and leave or known as mean lifetime of the peers. Each structured overlay uses the maintenance algorithm to keep the routing table up_to_date. The correctness of the routing table depends on how frequently the routing table is maintained. Hence, the routing performance increases with the increase in frequency of maintenance. The asymptotic analysis of routing performance of the structured overlay protocols - Chord, Pastry and Kademlia are logarithmic. This paper present the analysis of routing performance on structured overlay in DHT which are Chord, Pastry and Kademlia to find better structured overlay for P2P network. In this study, the analysis is according to the performance metric in terms of routing efficiency, lookup hop count and lookup success latency. However, due to the simulation resource limitation, we only simulate 100-500 peers of P2P networks and range of 100-1000 sec of mean lifetime.

There are also previous related works that compare the structured DHT overlay network. In [5], the routing performance comparison are also among Chord, Pastry and Kademlia, where only routing efficiency is being considered. Routing efficiency is calculated as the ratio of successfully routed messages to total number of messages sent. The simulation increases with mean lifetime of peers. The result shows that Kademlia has the best routing efficiency followed by Pastry and Chord. A simulation carried out by [9] on Chord, Pastry, Kademlia, Broose and EpiChord in mobile networks having high churn rate. From the result, Kademlia are also chosen to be the best P2P overlay to implement on the mobile nodes based on routing efficiency. Due to the importance of lookup latency and routing efficiency, [10] developed a mathematical model that combine these two metrics. They found that value of effective latency using the mathematical model matches their simulations. This reason motivates to simulate the other performance metric which is the workload per node or the lookup hop count. Thus, this work analyzed three performance metrics which are routing efficiency, lookup success latency and lookup hop count.

The remainder of the paper is organized as follows. Section 2 illustrate the research method on simulating Chord, Pastry and Kademlia. Section 3 demonstrates the results and analysis on the routing performance according to the experiments. Finally, the whole paper is concluded in Section 4.

2. RESEARCH METHOD

This work simulates structured overlay protocols using OMNeT++ with INET [11] and OverSim module. OverSim is an open-source P2P simulation framework that has 3-tier architecture designed using the NED language [3]. OverSim has been described as a powerful tool for evaluating and developing P2P applications [12, 13]. Figure 1 shows the 3-tier architecture consists of three layers such as application layer, overlay layer and underlay layer.

In application layer, key-based routing (KBR) is implemented where it periodically sends test messages to random nodes and records different parameters such as number of lookup, message sent, the number of lookup failed, messages delay and hop-counts [8]. In the same layer, P2P Apps for example DHT TestApp is used to test the DHT application, by performing DHT PUT and GET requests with randomly generated data items [14]. Important to realize, these applications are implemented as modules and interface with overlays through KBR API, which represent basic capabilities common to all structure overlays. In the overlay layer, OverSim provides a number of different network models, for both structured (Chord, Kademlia, Pastry, Broose and Koorde etc.) and unstructured overlays (GIA etc) [14]. In the underlay network, the INET framework can be used for routing protocol implementation and also support wireless and mobile simulation as well [5]. In addition, OverSim provides multiple underlay models to allow detail simulation; and provides single host underlay for simulation of a single node.

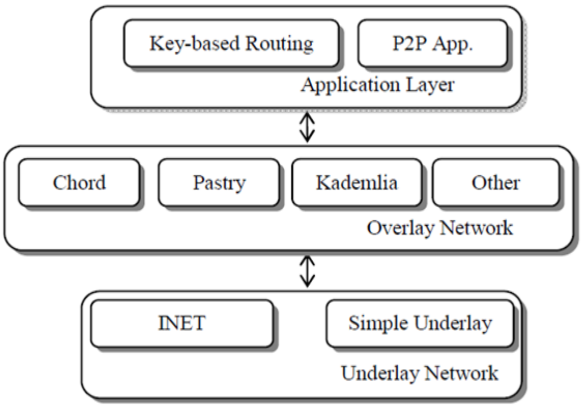

Figure 1. 3-tier simulation framework implemented in OverSim simulator [3]

In this simulation, the routing efficiency, lookup hop count and lookup success latency of the peers are calculated and recorded. The data presented in section 3 is the mean of 3 independent experiments. The following performance metrics have been analyzed in order to measure the performance [15].

Routing Efficiency: Routing efficiency also known as lookup success rate is calculated by taking the ratio of total number of lookup messages managed to succees before timeout and the total number of lookup messages broadcastes by the system. This metric is calculated using (1).

$$Routing\ efficiency = \frac{Successfull\ Lookups}{Total\ Lookups}\ X\ 100 \quad (1)$$

Lookup Hop count: Hop count or workload per peer is calculated by taking the ratio of total broadcasted complex query messages and total number of nodes required to successfully forward the messages to the destination node. This is defined in (2).

$$Lookup\ hop\ count\ (ratio) = \frac{Total\ broadcast\ messages}{Total\ nodes\ required\ to\ forward\ the\ messages} \quad (2)$$

Lookup Success Latency: Lookup success latency is defined as the time taken for a packet of data to get from one designated point to another and sometimes measured as the time required for a packet to be returned to its sender.

This work simulates the routing performance for different network size and mean lifetime of the peers as mentioned in Table 1. In this simulation, the number of peers were limit to 500. The mean lifetime of the peers are varied between 100 sec to 1000 sec. This limitation is due to the limited resources for executing the simulation.

Table 1. Parameters used for simulation

Name of the Parameters	Architecture		
	Chord	Pastry	Kademlia
Number of peers	100, 200, 300, 400, 500	100, 200, 300, 400, 500	100, 200, 300, 400, 500
Mean life-time of the peer (sec)	100, 300, 500, 700, 1000	100, 300, 500, 700, 1000	100, 300, 500, 700, 1000

3. RESULTS AND ANALYSIS

3.1. Routing efficiency

Two scenarios has been conducted to collect the routing efficiency results. First experiment which is increasing network size from 100 to 500 peers with the fixed mean lifetime of 1000 sec. Figure 2 illustrates the routing efficiency vs network size.

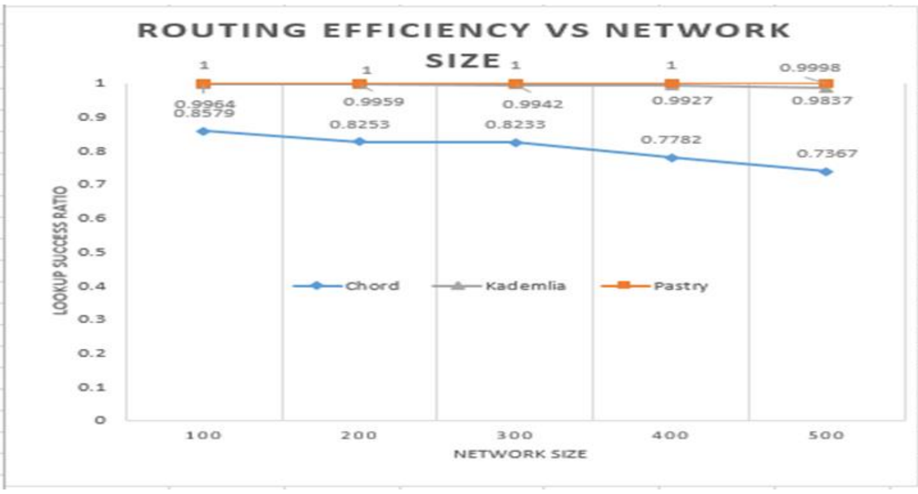

Figure 2. The results on Chord, Pastry and Kademlia based on routing efficiency vs network size

Chord has the lowest routing efficiency over network size among Pastry and Kademlia. The lowest routing efficiency ratio is Chord where the ratio is 0.8579 at 100 peers and gradually decrease when the network size increase to 500 peers. Pastry maintained the routing efficiency to 1.0 until network size 500 peers then drop to 0.9998. Kademlia has a slightly lower ratio compared to Pastry. If the routing efficiency is 1.0, meaning that it is a good routing with high network stability. Chord organizes its routing table in single-dimensional space, whereas Pastry and Kademlia have bigger routing table and organized its routing table in two-dimentional space. Second scenarios for routing efficiency are increasing mean lifetime with the fixed network size (100 peers). Figure 3 illustrates the routing efficiency vs mean lifetime.

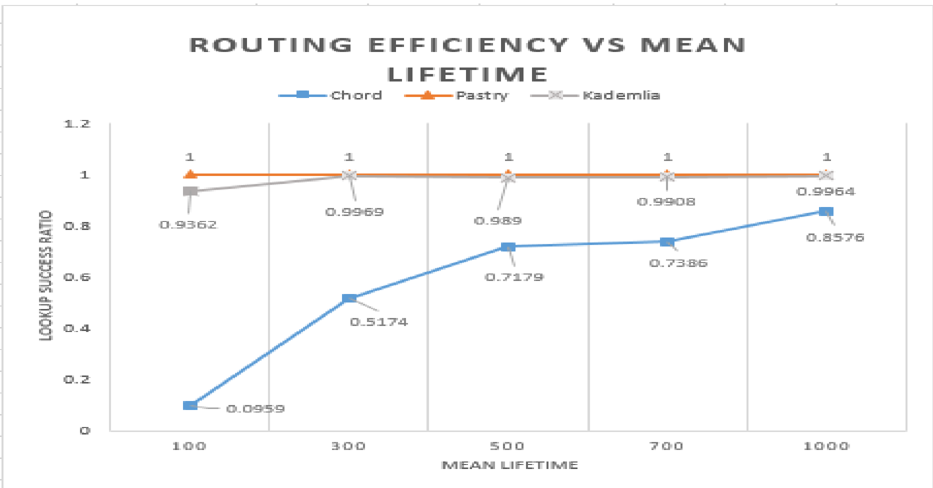

Figure 3. The results on Chord, Pastry and Kademlia based on routing efficiency vs mean lifetime

Based on Figure 3, the results are similar where, Chord has the lowest compared to Pastry and Kademlia algorithm. The ratio is 0.0959 (Chord algorithm) on 100 seconds means lifetime. The ratio in Chord is linearly increased as the mean lifetime increases. Kademlia has a slightly drop compared to Pastry. The highest routing efficiency is the ratio of 1.0 which Pastry on 100 seconds to 1000 seconds mean lifetime. This proves that Pastry are the best algorithm for routing efficiency.

3.2. Lookup hop count

In lookup hop count, the similar scenarios were conducted. First, increasing network size from 100 to 500 peers with fixed mean lifetime 1000 sec. Figure 4 illustrates the lookup hop count vs network size.

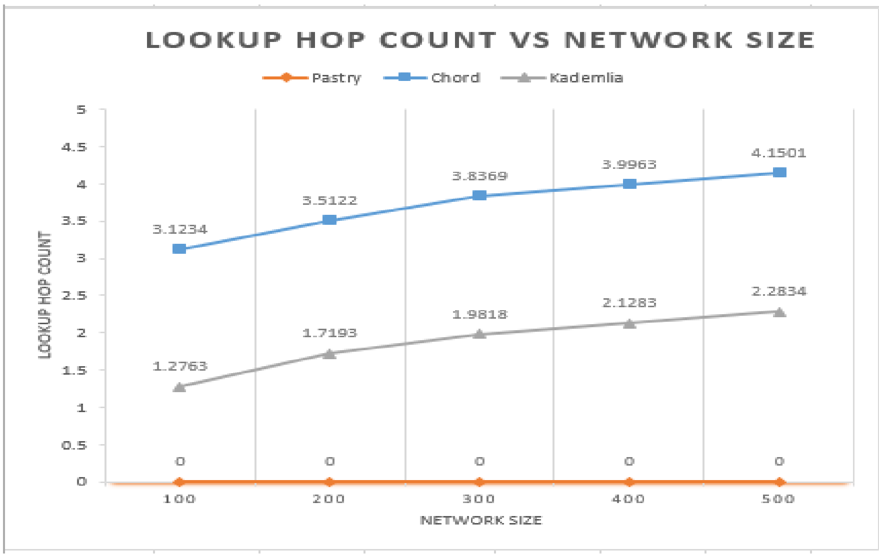

Figure 4. The results on Chord, Pastry and Kademlia based on lookup hop count vs network size

In Figure 4, lookup hop count for Pastry shows a constant ratio of 0 where the values were too small. This is due to the roundup value process. Between Kademlia and Chord, Kademlia has the better lookup hop count at 100 peers, where lookup hop count is 1.2763. Chord has the ratio of 3.1234. When the network increases to 500 peers, both mean lookup hop count for Kademlia and Chord increased respectively. The result shows Pastry has the lowest lookup hop count compared to Chord and Pastry. The lower the lookup hop count, the better the routing protocol. This result replicates the previous routing efficiency result, where Pastry has ratio of 1.0 routing effiency. Second simulation is increasing mean lifetime from 100 to 1000 sec with fixed network size of 100 peers. Figure 5 illustrates the lookup hop count vs mean lifetime.

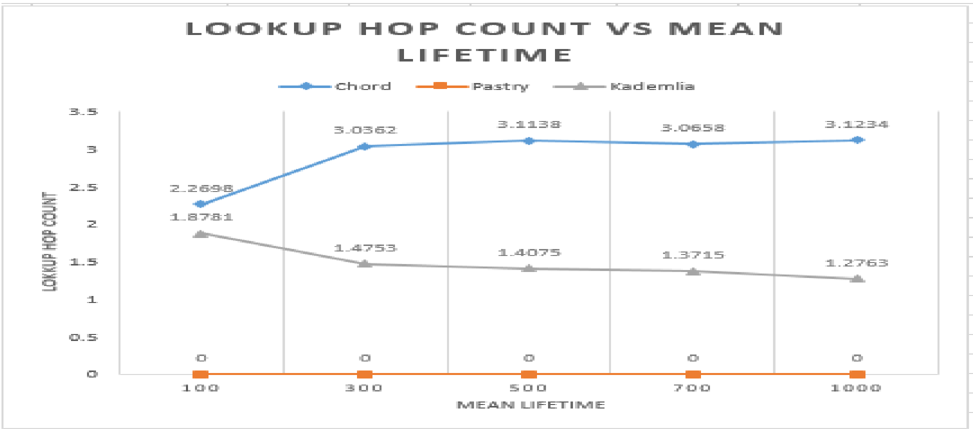

Figure 5. The results on Chord, pastry and kademlia based on lookup hop count vs mean lifetime

Based on Figure 5, Pastry has similar results with lookup hop count vs network size, where the lookup hop count is near to zero. Chord has the highest lookup hop count. Therefore, Pastry shows better routing in lookup hop count among other algorithms.

3.3. Lookup success latency

Similar lookup success latency, two scenarios have been conducted with increasing network size from 100 nodes to 500 nodes with the fixed mean lifetime 1000s. Figure 6 shows the plotted graph for the lookup success latency vs network size.

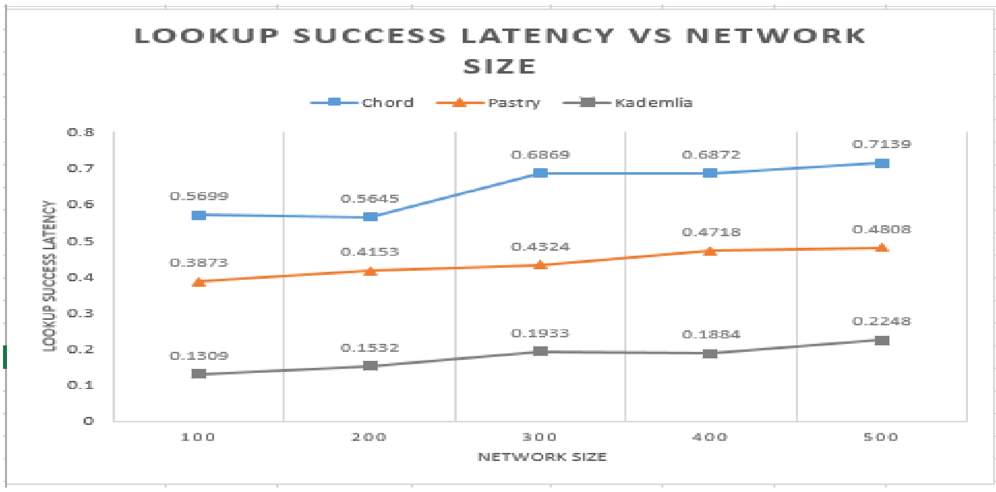

Figure 6. The results on Chord, Pastry and Kademlia based on lookup success latency vs network size

Lookup success latency is where latency time takes for data packets to be stored or retrieved. The lower lookup success latency are the better in the routing performance. As illustrated in Figure 6, Chord has the highest lookup success latency and Kademlia has the lowest lookup success latency. Second simulation were the increasing mean lifetime with the fixed network size of 100 peers. Figure 7 illustrates the lookup success latency vs mean lifetime. Chord has the highest lookup success latency compared to Pastry and Kademlia. Kademlia has the lowest lookup success latency.

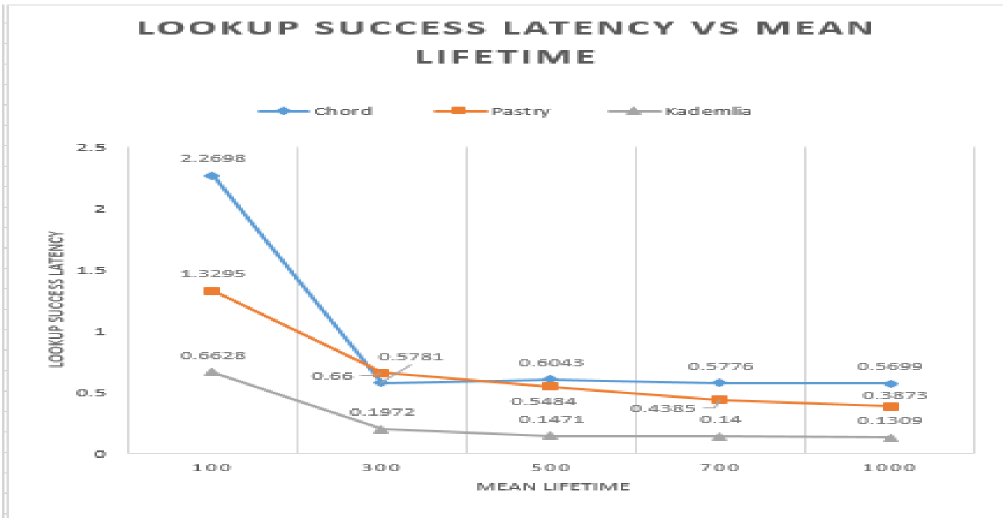

Figure 7. The results on Chord, Pastry and Kademlia based on lookup success latency vs mean lifetime

This work simulates three performance metrics which are routing efficiency, lookup hop count and lookup success latency. The parameter setting for network size is from 100 to 500 peers and the mean lifetime is from 100-1000 sec. This set of simulation is considered a small-scale of network environment due to the simulation resource limitation. The result shows that Pastry has the highest ratio in routing efficiency. The results are not similar to the previous research by [5]. However, the difference of routing efficiency between Pastry and Kademlia is not significant. It is assume that if the simulation executes in a large scale of network and in a higher mean lifetime, Kademlia has the possibility to obtain highest ratio in routing efficiency similar to the result obtained by [5]. Chord has the lowest routing efficiency due to Chord organizes its routing table in single-dimensional space, whereas Pastry and Kademlia have bigger routing table and organized its routing table in two-dimentional space. Lookup hop count is defined as the workload per peer in the network [15]. A low ratio of lookup hop count is good for routing where the workload per peer is low as well. The results for lookup hop count are that Pastry has the lowest ratio, followed by Kademlia and Chord. The value obtained for Pastry from the simulation is rounded to 0. Here, the results for lookup hop count are parallel to the results in routing efficiency for Pastry in this work. Finally, the last simulation shows that Kademlia obtained the lowest lookup success latency ratio, followed by Pastry and Chord. The lookup success latency between Kademlia and Pastry also has not much difference.

4. CONCLUSION

In conclusion, this project was completed and achieved its objective. The main objective of this research was to simulate the structured overlay protocol Chord, Pastry, and Kademlia that use DHT using OverSim Simulator using NED language. The second objective is to evaluate three DHT-based structure overlays which are Chord, Pastry and Kademlia on the network performance in terms of routing efficiency, lookup hop count and lookup success latency for P2P. The simulations have been divided into two scenarios, the first scenario was created according to the increasing number of nodes and the second scenario were created according increasing mean lifetime. For the first scenario in term of routing efficiency Pastry has the best routing efficiency is 1.0. Next, in term of lookup hop count and lookup success latency Kademlia is the best algorithm when has the lowest lookup hop count and lookup success latency. For the second scenario, performance measure in term of routing efficiency the best algorithm is Pastry but not have much different with Kademlia which slightly drop from Pastry. For lookup hop count and success latency Kademlia has the lowest for these two performance measurements. This means Kademlia is the best algorithm among Pastry and Chord.

ACKNOWLEDGEMENTS

We thank Dr. Saiful Khan from University of Oxford's e-Research Centre and Dr. Mr. Mohammad Elsheikh from Universiti Malaysia Perlis for the useful discussions, guidance, and assistance in this research.

REFERENCES

[1] M. Caleffi and L. Paura, "P2P over MANET: Indirect tree-based routing," in *7th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2009*, 2009, pp. 18–22.
[2] S. A. Abid, "An Application Of 3D Logical Structure In DHT Paradigm For Efficient Communication In MANET-PhD Thesis," Unversity of Malaya, Kuala Lumpur, 2014.
[3] J. Leng and T. Li, "Research on P2P Network Resource Search Model," in *International Conference Advanced Engineering and Technology Research (AETR 2017) Research*, 2018, vol. 153, no. Aetr 2017, pp. 55–60.
[4] C. Yan, "Distributed hash table based routing algorithm for wireless sensor networks," in *Proceedings-2014 5th International Conference on Intelligent Systems Design and Engineering Applications, ISDEA 2014*, 2014, pp. 430–433.
[5] S. Khan, A. Gani, and M. Sreekandath, "The routing performance of logarithmic-hop structured P2P overlay," *2011 IEEE Conf. Open Syst. ICOS 2011*, pp. 208–213, 2011.
[6] P. Maymounkov and D. Mazi, "Kademlia : A Peer-to-peer Information System Based on the XOR Metric," *PeertoPeer Syst.*, pp. 53–65, 2002.
[7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable P2P Lookup Service for Internet Applications," *Proc. of SIGCOMM*, pp. 1–23, 2001.
[8] S. Khan, A. Gani, S. Raviraja, and Shashikantha, "A study of broadcasting over structured overlay," *2015 7th Int. Conf. Commun. Syst. Networks, COMSNETS 2015-Proc.*, 2015.
[9] F. Chowdhury, "Performance Evaluation of Structured Peer-to-Peer Overlays for Use on Mobile Networks," 2014.
[10] N. Varyani, S. Nikhil, and V. S. Shekhawat, "Latency and routing efficiency based metric for performance comparison of DHT overlay networks," in *Proceedings-IEEE 30th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2016*, 2016, pp. 337–342.
[11] OMNeT++, *INET Framework for OMNeT++*. 2012.
[12] J. Moorhouse, L. Liu, Z. Li, and Z. Ding, "Modelling and Simulation of Peer-to-Peer Overlay Network Protocols using OverSim," *2013 UKSim 15th Int. Conf. Comput. Model. Simul.*, pp. 144–149, 2013.
[13] T. Kunz, B. Esfandiari, S. Ngozi, and F. Ockenfeld, "P2P Overlay Performance in Large-Scale MANETs," *Int. J. Commun. Netw. Syst. Sci.*, pp. 147–173, 2018.
[14] J. Furness, M. Kolberg, and M. Fayed, "An evaluation of chord and pastry models in OverSim," *Proc.-UKSim-AMSS 7th Eur. Model. Symp. Comput. Model. Simulation, EMS 2013*, pp. 509–513, 2013.
[15] S. Khan, "Complex Searching Over Structured Overlay Under Dynamic Environment-Master Thesis," University Of Malaya, Kuala Lumpur, 2011.