Effects of different churn models on the performance of structured peer-to-peer networks

Zhonghong Ou, Erkki Harjula, Mika Ylianttila
MediaTeam, University of Oulu
Inf. Proc. Laboratory, P.O.BOX 4500
FIN-90014 University of Oulu, Finland
firstname.lastname@ee.oulu.fi

Abstract—**We present the effects of different churn models on the performance of structured peer-to-peer (P2P) networks in this paper. Specifically, Exponential distribution (ED), Pareto distribution (PD), and Weibull distribution (WD) are evaluated to provide a comparative analysis. Kademlia-based Peer-to-Peer Protocol (P2PP) is utilized as the underlying signaling protocol. Through simulations, we conclude that the simulated different churn models do not have a significant effect on the performance of the simulated structured P2P network. Quantitatively, ED and PD result in better performance compared to WD from the viewpoints of lookup success rate, mean network traffic load, and mean number of messages.**

Keywords- structured peer-to-peer network; churn model; exponential distribution; pareto distribution; weibull distribution;

I. Introduction

Peer-to-Peer (P2P) system has attracted a great deal of attention from the industry, academia and media since the date of its birth in the late 90's, represented by Napster, because of the scalable, robust, extensible, autonomous etc., characteristics. According to the varied application environments, P2P systems can be divided into different categories: communication and collaboration, distributed computation, internet service support, database system, and content distribution [1]. The commonalities behind these categories are that nodes firstly join the P2P system, contribute some resources, e.g. Central Processing Unit (CPU) processing capability, storage, and lookup service, and then leave the system. The join-participate-leave cycle is called a *session*, and the collective effect created by the independent arrival and departure of thousands-or millions-of peers is named as *churn* [2].

Churn has a significant effect on the performance of P2P systems, e.g. frequent nodes joining and leaving result in stale routing information in the routing table and inconsistency of the stored resource items, the distribution of session length affects the overlay topology and key design parameters, just to name a few. Consequently, the effects of churn should be taken into account when design or evaluate a P2P system. Furthermore, every simulation and analysis study of churn effect is built on the premise of a churn model. Several research activities [3]-[15] have focused on this topic in order to accurately characterize and model the peer dynamics and characteristics, i.e. churn, in P2P systems.

Some research activities built the model of churn following Exponential distribution (ED) [3] [4] [5] [6] [7] [8], while some others constructed the churn model as Pareto distribution (PD) [9] [10] [11] [12], still some others argued that the Weibull distribution (WD) could best characterize the dynamics of peers participating in P2P systems through real life measurements [13] [14] [15]. However, surprisingly, there are few studies focusing on the effects of different models of churn on the performance of P2P systems.

In this paper, we analyze the effects of different models of churn on the performance of structured P2P systems by utilizing three typical distributions, i.e. ED, PD, and WD. We build a simulation model based on the Peer-to-Peer Protocol (P2PP) [16] signaling protocol, and utilize Kademlia [17] as the underlying Distributed Hash Table (DHT) algorithm. Through simulations, we conclude that the effects of different models of churn on the performance of structured P2P systems are quantitative instead of qualitative.

The remainder of the paper is structured as follows: Section II gives the related work; Section III presents an overview of different distributions; Section IV introduces the used signaling protocol in general; Section V provides the simulation results. In Section VI, we conclude the paper.

II. Related Work

ED is widely used to model the time of a component to fail in reliability theory. It is also extensively used in modeling the session time of nodes taking part in P2P systems. Through extensive trace analysis and modeling of BitTorrent, Guo *et al.* [3] conclude that both the lingering and downtime distributions are exponentially distributed. The unified cost versus performance framework put forth in [4] and [5] is also based on the churn model that follows ED. Meanwhile, ED is used as the basis of availability assumptions in some other studies, namely Chord [6], Tapestry [7], and [8] respectively.

PD is another largely used model in P2P systems. Bustamante and Qiao [9] monitor peers in Gnutella to motivate preferential neighbor selection based on nodes uptime. Their measurements show that nodes session length fits well with PD. By utilizing two churn models of ED and PD, Wu *et al.* [10] present an analytical study of three strategies on improving DHT lookup performance under churn, i.e. lookup strategy, lookup parallelism and replication strategy. In other contexts, e.g. process lifetime estimation [11] and network performance [12], "long-tailed" distributions, especially PD, are used as the fundamental models for analysis.

Though ED and PD are used extensively as the fundamental models to analyze the effect of churn on the performance of P2P systems, some studies doubt the use of ED or PD to characterize the session lengths. Nurmi *et al.* [13] analyze the suitability of different statistical distribution for describing machine availability in three different data sets. Their results indicate that either a hyper-exponential or Weibull model effectively represent machine availability in enterprise and Internet computing environments. Stutzbach *et al.* [14] present a thorough analysis of Churn in three real life P2P systems, i.e. Gnutella, Kad and BitTorrent, and conclude that session-lengths are not heavy-tailed or Pareto, instead they are more accurately modeled by a Weibull distribution. Steiner *et al.* [15] explore the peer behavior, e.g. the total number of peers online and their geographical distribution, by crawling a real system Kad continuously for six months. They find that the distribution of the session lengths is best characterized by a Weibull distribution, with shape parameter $k < 1$.

In this paper, we choose three typical distributions, i.e. ED, PD, and WD, to provide a comparative analysis of different churn models on the performance of structured P2P system.

III. Overview of Typical Distributions

In this section, we introduce the basic characteristics of the analyzed distributions, i.e. density function, distribution function, expected value, and generating function.

A. Exponential Distribution

The probability density function (PDF) and cumulative distribution function (CDF) for an ED are given by the following equations:

$$f_e(x;\lambda) = \begin{Bmatrix} \frac{1}{\lambda} e^{-x/\lambda}, x \geq 0, \\ 0, \qquad x < 0. \end{Bmatrix}. \tag{1}$$

$$F_e(x;\lambda) = \begin{Bmatrix} 1 - e^{-x/\lambda}, x \geq 0, \\ 0, \qquad x < 0. \end{Bmatrix}. \tag{2}$$

where $\lambda > 0$ is a *scale* parameter of the distribution. The mean or expected value of an exponentially distributed variable X with scale parameter λ is given by the following equation:

$$E_e(X) = \lambda. \tag{3}$$

The function to generate an exponential variable is as follows:

$$T_e = \lambda(-\ln U). \tag{4}$$

where U is a random variable that follows the uniform distribution on the unit interval (0,1).

B. Pareto Distribution

The PDF and CDF for a PD are given by the following equations:

$$f_p(x;\alpha,\beta) = \begin{Bmatrix} \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}, x \geq \beta, \\ 0, \qquad x < \beta. \end{Bmatrix}. \tag{5}$$

$$F_p(x;\alpha,\beta) = \begin{Bmatrix} 1 - (\frac{\beta}{x})^{\alpha}, x \geq \beta, \\ 0, \qquad x < \beta. \end{Bmatrix}. \tag{6}$$

where β is a *scale* parameter of the distribution, and is the minimum possible value of the random variable X that is necessarily positive, while α stands for the *heavy-tailed* degree. The mean or expected value of a random variable X that follows PD with *scale* parameter β is given by the following equation:

$$E_p(X) = \frac{\alpha\beta}{\alpha - 1}. \tag{7}$$

One point to be noted here is that if $\alpha \leq 1$, the expected value is infinite. The function to generate a Pareto variable is as follows:

$$T_p = \frac{\beta}{U^{1/\alpha}}. \tag{8}$$

where U is a random variable that follows the uniform distribution on the unit interval (0,1).

C. Weibull Distribution

The PDF and CDF for a PD are given by the following equations:

$$f_w(x;\lambda,k) = \begin{Bmatrix} \frac{k}{\lambda}(\frac{x}{\lambda})^{k-1} e^{-(x/\lambda)^k}, x \geq 0, \\ 0, \qquad x < 0. \end{Bmatrix}. \tag{9}$$

$$F_w(x;\lambda,k) = \begin{Bmatrix} 1 - e^{-(x/\lambda)^k}, x \geq 0, \\ 0, \qquad x < 0. \end{Bmatrix}. \tag{10}$$

where $k > 0$ is called the *shape* parameter and $\lambda > 0$ is called the *scale* parameter. When $k = 1$, it is equivalent to an ED; when $k = 2$, it equals to the Rayleigh distribution. The mean or expected value of a random variable X that follows WD with *shape* parameter k and *scale* parameter λ is given by the following equation:

$$E_w(X) = \lambda\Gamma(1 + \frac{1}{k}). \tag{11}$$

where Γ is the Gamma function. The function to generate a random variable that follows WD is as follows:

$$T_w = \lambda(-\ln U)^{\frac{1}{k}}. \tag{12}$$

where U is a random variable that follows the uniform distribution on the unit interval (0,1). According to the page limitation, the proof of these generating functions is omitted in this paper.

IV. SIGNALING PROTOCOL

A. Peer-to-Peer Protocol

P2PP [16] is a binary protocol for creating and maintaining an overlay for participating nodes. It can support both structured and unstructured P2P protocols, e.g. Kademlia [17] Chord [18], Gnutella. In our simulation model, we choose Kademlia as the fundamental DHT algorithm and make some modifications to it. It is noteworthy that the simulation model can support other DHT algorithms, e.g. Chord, without much effort. The following messages listed in Table I have been implemented.

TABLE I. MESSAGE TYPE AND ITS FUNCTIONALITY

Message Type	Functionality
Bootstrap	Return the IP address and port of a node already in the overlay.
Join	Node joins the overlay.
Publish	Publish a resource item in the overlay.
Lookup	Look up a resource item in the overlay.
Exchange	Update the routing table, sent periodically.
KeepAlive	Detect the aliveness of the nodes and remove the stale routing items from the routing table, sent periodically.
Leave	Notify a peer's routing neighbors about its leaving from the overlay.
Transfer	Transfer resource items to another node in the overlay.

The transport protocol utilized to carry the P2PP protocol is User Datagram Protocol (UDP). Accordingly, the ACK message is used in our simulation model to guarantee the reliability of message delivery. One transaction is made up of a Request, a Response and an ACK messages. One point to be noted here is that, if an ACK message follows directly a Response message, then the ACK message piggybacks onto the Response message to save network traffic. Furthermore, as we choose Kademlia as the underlying DHT algorithm, correspondingly, the iterative routing mechanism (the preference of Kademlia) is used in our simulation model.

B. Kademlia-based Algorithm

There are three categories of operations with regard to Kademlia, i.e. nodes joining and leaving, overlay maintenance and resource-related operations.

1) Nodes Joining and Leaving: There is one centralized Bootstrap server existing that provides the associated Bootstrap functionality. The node joining process is serial, which means each time the joining node just sends one request to, and accordingly, receives one response from the node already in the overlay. This is slightly different from the parallel mechanism used in the original Kademlia algorithm. We simulated this mechanism in this paper as it is also common in some other DHT algorithms, e.g. Chord. In the leaving process, the *graceful* leaving mechanism is implemented in our simulation. It means the leaving node transfers all the resource items it stores to its closest neighbor and notifies its leaving to all its routing neighbors by sending Leave messages. In this process, the Bootstrap, Join, Leave, Transfer and ACK messages are utilized.

2) Overlay Maintenance: To decrease the effect of churn on the overlay stability, Exchange and KeepAlive messages are implemented. According to the limitation of UDP packet size, each Exchange response message includes maximum 15 unduplicated routing items. To avoid including stale routing information into routing table, additional KeepAlive messages are sent to check the aliveness of the routing items received by Exchange response message. Moreover, KeepAlive messages are also sent periodically to remove stale routing items from routing tables that are resulted from nodes leaving. In this process, Exchange, KeepAlive and ACK messages are utilized.

3) Resource-related Operations: There are two operations regarding resource items, i.e. Publish and Lookup. Once more, the serial mechanism is utilized here. The Publish and Lookup operations are highly similar to the Join operation aforementioned. One noteworthy point here is that, if the Publish or Lookup message is sent to an offline node, the entire Publish or Lookup process results in a failure without any re-trying to other nodes. The reason is that it is hard to decide which node is the second closest node to the resource item as we just return one closest routing item in each response message (this is how iterative routing mechanism works). Meanwhile, to keep the resource items stored in the overlay up to date, the owner of the resource items also re-publishes them periodically. The Republish process is exactly the same as the Publish process. In this process, the Publish, Lookup and ACK messages are utilized.

V. SIMULATION RESULTS

In this section, we evaluate the effects of different models of churn on the performance of P2PP through simulations. The level of churn is determined by the mean session time, or mean online time. Shorter mean online time means a higher level of churn, and vice versa.

A. Performance Metrics

The following performance metrics are used in this paper:

- *Success rate*: the ratio of successfully finding the resource items already published in the overlay.
- *Mean traffic load*: the mean number of bytes sent and received in one second by each node, i.e. bytes/node/s.
- *Mean number of messages*: the mean number of messages sent and received in one second by each node, i.e. messages/node/s.

One point that helps to understand the metrics is that each single message, e.g. a Join request message, is calculated twice in our evaluation as it is one outgoing message for the initiating node and one incoming message for the targeting node. At both ends, it causes network traffic and consumes battery in the case of mobile terminals.

B. Simulation Setup

We utilize the NetHawk EAST software to simulate the abovementioned P2PP protocol. NetHawk EAST is a test automation and traffic generation tool to simulate telecommunication networks. It has the functionality of supporting binary encoding and decoding that makes it suitable for simulating the binary formatted P2PP and makes the traffic load as close to real life as possible. We make use of a dedicated server to run the simulations. The hardware of the server is Xeon (TM) CPU 3.06 GHz, 3.05 GHz, 2.00 GB of RAM. The software environment is Microsoft Windows XP Professional Version 2002 Service Pack 2 and NetHawk_EAST_IMS_v2.0.1_U1.1.

According to the hardware limitation and UDP buffer size, we can only simulate overlays with 400 nodes. For each simulation, nodes stay online and offline (crash and rejoin the overlay) for time intervals that follow ED, PD, and WD, respectively. The corresponding random variables are generated by utilizing Eq. (4), Eq. (8), and Eq. (12) respectively. As we set the same mean online and offline time intervals for the overlay nodes, i.e. $m_{online} = m_{offline}$, around half of the 400 nodes, i.e. 200 nodes, stay online on average at any given moment. By changing the mean online and offline time intervals, we can change the associated churn levels. Each simulation runs for 2 hours. The nodes use the same IP addresses and Node IDs when they rejoin the overlay after the offline time interval.

We use a dedicated Bootstrap server that stays online throughout each experiment to bootstrap the overlay. In the *initiation* stage, the nodes join the overlay at a rate of 2 nodes/s. After 200 nodes have joined the overlay, a *stabilization* interval of 200 seconds is used to fill their routing tables with proper routing items from the overlay. Then the *churn* stage is started, the online nodes leave the overlay (after their online time expires) and the offline nodes join the overlay (after their offline time expires), and so on until the end of the experiment. The data are collected within the *churn* stage.

For the resource related operations, i.e. Publish and Lookup, a database is utilized to record the resource items already published in the overlay. After the owner of one resource item leaves the overlay, the resource item will be removed from the database accordingly after the mean online time to keep the resource items up to date. For the Lookup operation, nodes only look up resource items from the database to make the success rate meaningful. We make use of the MD5 algorithm to generate node Identifiers (IDs) and resource IDs that both have a length of 128 bits in our simulation.

C. Simulation Results

To make the analysis clearer, we firstly give the parameters and the selected values used in the simulated P2PP overlays. The *k*-value for Kademlia is 3 and serial lookup is utilized. The intervals for sending the associated messages are as follows: $t_{publish}$=100s, $t_{republish}$=60s, t_{lookup}=125s, $t_{exchange}$=60s, $t_{keepalive}$=100s, $m_{online} \in \{200, 400, 600, 800, 1000, 2000, 3000, 4000\}\, s$.

The success rate, mean network traffic load, and mean number of messages with respect to different levels of churn under the three churn models, i.e. ED, PD, and WD, are shown in Figure 1, Figure 2, and Figure 3, respectively.

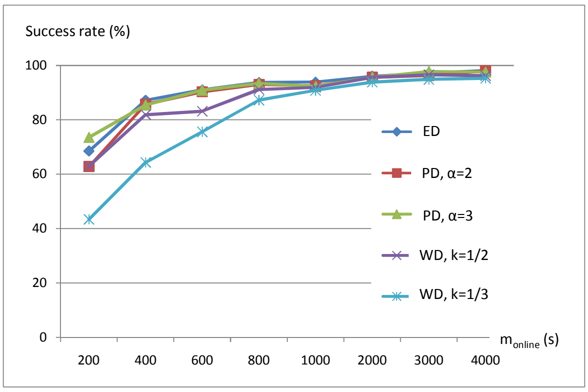

Figure 1. Success rate with respect to different churn models.

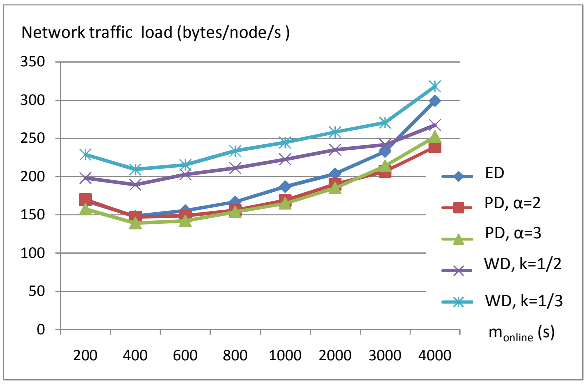

Figure 2. Mean network traffic load with respect to different churn models.

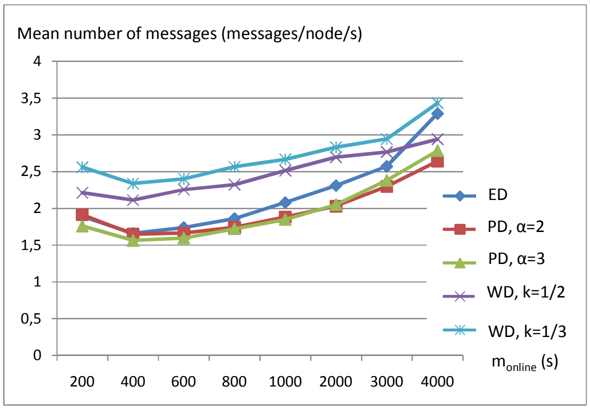

Figure 3. Mean number of messages with respect to different churn models.

For the analysis of PD, we firstly fix the value of the *heavy-tailed* parameter α, and change the value of the *scale* parameter β according to the associated value of mean online time m_{online}. The varied value of β is reflected by the varied value of m_{online} according to Eq. (7), where $E_p(X)$ is synonymous with m_{online}. Then we change the value of α to see its own effect on the performance of P2PP.

For the analysis of WD, similar to PD, we firstly fix the value of the *shape* parameter k and change the value of the *scale* parameter λ, and then we change the value of k to see its effect on the performance of P2PP.

From Figure 1, we can see that there is no significant difference among the three churn models from the viewpoint of success rate. The overall trend is that the success rate increases as the level of churn decreases. This makes sense as in a higher level of churn environment, the resource items are stored in the right nodes of the overlay with lower probability that results in lower success rate. Quantitatively, ED and PD result in better performance of success rate compared to WD. This phenomenon can be explained by the characteristics of the expected value and the generating function of the three churn models. Under the same expected value, i.e. mean online time m_{online}, WD has more widely dispersed samples that result in more small sample values and fewer large sample values than ED and PD, which in turn incurs more frequent nodes joining and leaving. Accordingly, the success rate in WD is lower than in ED and PD. This phenomenon becomes more distinct when the value of the *shape* parameter k decreases.

From Figure 2 and Figure 3, no significant differences are observed regarding the mean network traffic load and mean number of messages. The overall trends with these two performance metrics are that they firstly decrease as the value of m_{online} increases, after some critical point, i.e. $m_{online} = 400s$, they start to increase. This is because in an comparatively high churn environment, i.e. $m_{online} = 200s$ in this paper, the frequent joining and leaving of nodes causes a great deal of network traffic in the overlay, while in the simulated extremely low churn environment, i.e. $m_{online} = 4000s$, most of the nodes stay online for a large fraction of the simulated time that causes a lot of maintenance traffic in the overlay. Again, quantitatively, the same phenomenon with regard to the three different churn models can be observed here, i.e. ED and PD have slightly better performance than WD. The reason is the same as for the success rate.

As a whole, we can conclude that there is no significant difference among the effects of the three simulated churn models on the performance of the simulated structured P2P network. Quantitatively, ED and PD result in better performance compared to WD from the viewpoints of success rate, mean network traffic load, and mean number of messages.

VI. CONCLUSION

We analyzed the effects of different models of churn on the performance of structured P2P networks in this paper. Specifically, Exponential Distribution (ED), Pareto Distribution (PD), and Weibull Distribution (WD) were evaluated to provide a comparative analysis. Kademlia-based P2PP was utilized as the underlying signaling protocol. Through simulations, we concluded that the simulated different churn models did not have a significant effect on the performance of structured P2P networks. Quantitatively, the success rate of WD was slightly lower than the other two churn models, i.e. ED and PD, while the mean network traffic load, and mean number of messages of WD were slightly higher than the other two models. In the future, we will extend the analysis to include other DHT algorithms, e.g. Chord, to get an overall analysis of different churn models to different structured P2P networks. Possible real life prototype will be implemented to further confirm the conclusions we made in this paper.

ACKNOWLEDGMENT

The authors would like to thank Tekes, Ericsson, Nokia, and Nethawk for financially supporting this work in Decicom.

REFERENCES

[1] S. Androutsellis-Theotokis, D. Spinellis, A survey of peer-to-peer content distribution technologies, ACM Computing Surveys, Vol. 36, No. 4, December 2004, pp: 335–371.

[2] D. Stutzbach and R. Rejaie. Understanding churn in peer-to-peer networks. In Proc. Internet Measurement Conference (IMC), Oct. 2006.

[3] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, Analysis, and Modeling of BitTorrent-like Systems. In Internet Measurement Conference, 2005.

[4] J. Li, J. Stribling, T.M. Gil, R. Morris, F. Kaashoek, Comparing the performance of distributed hash tables under churn, in: The 3rd International Workshop on Peer-to-Peer Systems (IPTPS), San Diego, CA, February 2004.

[5] J. Li, J. Stribling, F. Kaashoek, R. Morris, and T. Gil, A Performance vs. Cost Framework for Evaluating DHT Design Tradeoffs under Churn, in INFOCOM, Miami, FL, Mar. 2005.

[6] I.Stoica, R.Morris, D. Liben-Nowell, D.R.Karger, M.F.Kaashoek , F. Dabek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for Internet applications. IEEE/ACM Transactions on Networking, Volume 11, Issue 1, Feb. 2003 Page(s):17 - 32.

[7] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, Tapestry: An infrastructure for fault-tolerant wide-area location and routing, Tech. Rep. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, Apr 2001.

[8] A. Binzenhöfer and H. Schnabel, Improving the Performance and Robustness of Kademlia-based Overlay Networks, University of W¨urzburg, Institute of Computer Science, Report No. 405, April 2007.

[9] F. E. Bustamante and Y. Qiao. Friendships that last: Peer lifespan and its role in P2P protocols. In International Workshop on Web Content Caching and Distribution, 2003.

[10] D. Wu, Y. Tian, and K.W. Ng, Analytical Study on Improving DHT Lookup Performance under Churn, in Proc. of IEEE P2P 2006, Cambridge, UK, Sep. 2006.

[11] M. Harcol-Balter and A. Downey. Exploiting process lifetime distributions for dynamic load balancing. ACM Transactions on Computer Systems, 15(3):253–285, 1997.

[12] V. Paxon and S. Floyd. Wide area traffic: the failure of poisson modeling. IEEE/ACMTransactions on Networking, 3(3), 1995.

[13] D. Nurmi, J. Brevik, and R. Wolski, Modeling machine availability in enterprise and wide-area distributed computing environments, in Proc. Euro-Par, Aug. 2005, pp. 432–441.

[14] D. Stutzbach and R. Rejaie, Understanding Churn in peer-to-peer networks, in Proc. ACM Internet Measurement Conf. (IMC), Oct. 2006, pp. 189–202.

[15] M. Steiner, T. En-Najjary, E. W. Biersack, A Global View of KAD, in Proc. of the 7th ACM Internet measurement, IMC'07, October 24-26, 2007, San Diego, California, USA, pp: 117-122.

[16] S. Baset, H. Schulzrinne, M. Matuszewski, Peer-to-Peer Protocol (P2PP), draft-baset-p2psip-p2pp-01, November 19, 2007 (work-in-progress).

[17] P. Maymounkov and D. Mazires. Kademlia: A peer-to-peer information system based on the xor metric. In IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002, pp: 53 - 65.

[18] I.Stoica, R.Morris, D. Liben-Nowell, D.R.Karger, M.F.Kaashoek , F. Dabek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup protocol for Internet applications. IEEE/ACM Transactions on Networking, Volume 11, Issue 1, Feb. 2003, pp: 17 - 32.