Analytic Study of the Impact of Churn in Cluster-Based Structured P2P Overlays

E. Anceaume
IRISA/CNRS UMR 6074
France
emmanuelle.anceaume@irisa.fr

R. Ludinard*, B. Sericola
IRISA/INRIA Rennes Bretagne-Atlantique
France
{romaric.ludinard,bruno.sericola}@irisa.fr

***Abstract*—In this paper we present an analytic study of the impact of churn in cluster-based overlay networks. Cluster-based overlays keep the best of unstructured and structured overlays in terms of scalability, fault-tolerance and stability. Most of join and leave events have no impact on the overall overlay topology making these overlays highly robust to high churn. The only situations that effectively give rise to topology modifications are when clusters need to split because they exceed some maximal size or need to merge because they fall under some minimal size. Although these operations are scalable, they are intricate in the sense that they need synchronization among nodes involved in these operations. In this paper we accurately predict the frequency at which the topology of the overlay changes according to the number of join/leave operations. Our analysis improves upon existing studies by showing that these relevant topological changes are very infrequent, namely $\Theta(N)$ join/leave events are required before any of these topological operations occur, where N is the number of peers currently in the system. Such a result clearly demonstrates the appropriateness of these overlays to high churn.**

***Index Terms*—Peer-to-peer, churn, analytical study**

I. Introduction

A comprehensive set of measurement studies have shown a very high rate of churn in peer-to-peer systems [1]. Churn is the propensity of peers to continuously join and leave the system (voluntarily or not). This dynamics, if not efficiently managed, quickly gives rise to dropped messages, data inconsistency, increased latency, and increased bandwidth. To guarantee successful lookup operations, a recurrent prerequisite that should hold in all P2P systems is their ability to deal with churn. Most popular P2P systems (e.g., Gnutella [2], kaZaA [3]) rely on unstructured overlays. These overlays are based on random graphs, i.e., logical relationships among peers are mostly set according to a random process which makes joining and leaving operations constraint free. Data placement enjoy the same absence of constraints. Any data can be placed on any peer thereby imposing flooding or random walk techniques to retrieve them. It is well known that flooding does not scale due to network overflow (e.g. Gnutella V0.4), while random walk method performs badly when the graph is not regular (i.e., when peers degree show a high discrepancy) [4]. This has led to structured overlays (also called Distributed Hash Tables (DHTs)) in which peers self-organise in structured graphs. Typically, peers self-organise according to a distance function based on their identifier, allowing to partition the identifier space among all the peers of the system. Most proposed structured overlays (e.g., [5], [6], [7]) are highly satisfactory in terms of efficiency and scalability when evolving in weakly dynamic environments (i.e., their key-based routing interface guarantees operations whose complexity in messages and latency usually scale logarithmically with system size). However, in the presence of very frequent peers connections and/or disconnections, a very large number of join and leave operations are locally triggered engendering accordingly multiple and concurrent maintenance traffic. Ensuring routing tables consistency quickly becomes unbearable, leading to misrouting, and to possible partitioning of the system [8].

Castro et al. [9] have shown that both structured and unstructured graphs can be improved to be more robust to churn at the expense of increased maintenance overhead (e.g., augmentation of the frequency of failures detection probes, using of the graph structure to aggregate routing tables entries [7], usage of stabilisation algorithms, messages acknowledgements), while Godfrey et al [10] advocate for ”intelligent” failed peers replacement strategies (based on peers characteristics for predictive fixed strategies, or based on randomisation for agnostic strategies) according to the focused applications (e.g. storage, anycast server selection or tree-based multicast). In all these approaches, reducing churn turns out to augment the number of communication messages among peers either to decrease failure detection latency or to select the most appropriate replacement peer. These approaches are therefore conflicting with reducing communication overhead.

A different approach to build a churn-resilient overlay network has been proposed with cluster-based structured overlays (e.g., [11], [8], [12], [13]). In these overlays, peers which are close to each other according to a given proximity metric group together into clusters that form the vertices of the structured topology. Most of the communication traffic is captured by clusters which severely limits both topology changes and communication overhead imposed by the above solutions. However these clusters rely on two cluster management operations, namely the `split` and `merge` operations which are triggered each time a cluster becomes over populated or under populated. Handling these operations requires high synchronization among involved peers. Clearly this may tend to overwhelm

*Supported by the Direction Générale des Entreprises - P2Pim@ges project.

the benefit of cluster-based DHTs if both operations are too frequent with respect to join and leave events.

A. Contributions of the paper

In this paper we demonstrate cluster-based overlay networks robustness. Here, robustness is the ability of a system to continue to operate correctly despite high churn rate. This is achieved by accurately predicting the minimal number of join and leave events that need to be globally triggered in the system to give rise to the first `split` or `merge` operation in the system. We show that $\Theta(N)$ join/leave events are required before any of these topological operations occur, where N is the number of peers currently in the system. This is an important result as it shows that due to the rare occurrence of those relatively costly topological operations, it is simple for the network to correctly update routing tables in due time and thus maintain the graph structure. This solves the Achilles heel of DHTs, i.e., the cost induced to maintain nodes routing tables consistency in presence of high churn. From a practical point of view this is interesting as it shows the appropriateness of these overlay networks as substrate for large scale applications demanding in terms of routing latency and topology stability such as multimedia streaming platforms [14], and persistent data storage [15]. Prior to this analysis presented in Section III, we present the main features of cluster-based structured overlays networks in Section II. We conclude our paper in Section IV.

II. Cluster-based DHT Overlays in a Nutshell

Cluster-based DHT overlays mainly consist in the clusterized version of DHT overlays. Groups of peers substitute peers at the vertices of the graph. These groups of peers, typically called swarms [11], [13], clusters [12], cliques [8], and buckets [16] are populated by peers that are close to each other according to a given proximity metric D. This metric can be logical (as in [11], [13], [12], [16]), or geographical (as in [8]). These clusters form the vertices of the structured topology. Clusters in the system are uniquely labelled. Clusters size is lower (resp. upper) bounded. The lower bound, named S_{min} in the following, usually satisfies some constraint based on the assumed failure model. For instance $S_{min} \geq 4$ allows Byzantine tolerant agreement protocols to be run among these S_{min} nodes. The upper bound, that we call S_{max} in the sequel, is typically in $\mathcal{O}(logN)$ to meet scalability requirements, where N is the current number of nodes in the system.

Let us briefly describe how cluster-based overlays typically evolve according to these properties. When peer n enters the system, it joins the cluster whose label matches the proximity metric D. Once a cluster size exceeds S_{max}, this cluster splits into two smallest clusters, each one populating with the peers that are closer to each other. When n leaves, it simply leaves its cluster. When a cluster size reaches S_{min} this cluster merges with its closest neighbour. According to the proposed overlays, all cluster members or only a subset of them (but at least S_{min} of them) are in charge of routing lookup requests, replicating all data-items that match the cluster label, and handling cluster operations (split/merge and create). In the following we assume that only S_{min} peers are in charge of these operations. We call these peers *core members* of a cluster. The other peers of the cluster (if any) are inactive until they replace left core members. We call these nodes *spare members* of the cluster. In PeerCube [12], spare members allow to make join and leave events transparent to the overlay topology by simply having any peer that joins a cluster to join it as a spare member.

As a consequence, by clusterizing DHTs, both node degree and network diameter are reduced, making maintenance and routing protocols more efficient. For instance, routing in PeerCube [12] is achieved in $\log_2(N/\log_2 N)$ hops w.h.p. and require $\mathcal{O}(\log N)$ messages. Data structures (routing tables or finger tables) are locally maintained by moving the information they store among cluster members, instead of reconstructing them each time a new node joins the system which is typically the case for ”classical” DHTs network overlays. As an illustration, Figure 1 shows the maintenance cost of a classic hypercubic topology (where each node represents a vertex of the graph) compared to a cluster-based hypercubic one, i.e., PeerCube [12]. The number of routing tables updates per node in a network of around $N = 5,000$ nodes is depicted. Bursts of joins and leave are cyclically generated. For PeerCube, the upper bound on clusters size is set to $Smax = 12$, and the lower bound is set to $Smin = 4$. Curves depict the number of routing tables updates per node generated in PeerCube (denoted by `PeerCube`) and in a pure hypercube (denoted by `Hypercube`). As expected, one can notice the benefit of both clustering and using newcomers as hot spares as it drastically reduces the number of routing tables updates for both joins and departures events.

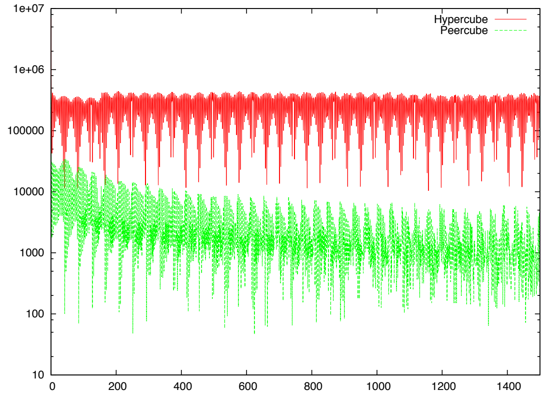

Fig. 1: Number of routing table updates per node as a function of join and leave events.

III. Analysing the churn impact on cluster-based topologies

In this section, we accurately analyse the frequency at which `split`, and `merge` operations occur with respect to `join` and `leave` operations. Specifically, we derive the minimum number of join and leave events needed to trigger a `split` or `merge` operation, in expectation and with very high probability. This formally shows the impact of churn on cluster-based topologies. We model this problem as the "balls-and-urns" problem, where each cluster is modelled as an urn and each join or leave event as a ball.

A. The model

Consider a system composed of n urns initially empty in which balls are thrown one by one. The throwings are independent from each other and each ball is thrown into urn i with probability p_i. For every $i = 1, \dots, n$, we denote by $U_i(m)$ the number of balls in urn i after throwing the m-th ball. It is well-known that the joint distribution of the $U_i(m)$ is given, for $k \geq 1$ and integers $m_1, \dots, m_n$ such that $m_1 + \cdots + m_n = m$, by

$$\mathbb{P}\{U_1(m) = m_1, .., U_n(m) = m_n\} = \binom{m}{m_1 \cdots m_n} \prod_{i=1}^{n} p_i^{m_i}$$

Moreover, for every $i = 1, \dots, n$, we have

$$\mathbb{P}\{U_i(m) = k\} = \binom{m}{k} p_i^k (1 - p_i)^{m-k}.$$

For every $h = 1, \dots, n$, we denote by $p(h)$ the probability distribution defined by $p(h) = \frac{1}{s_h}(p_1, \dots, p_h)$, where $s_h = p_1 + \cdots + p_h$. Clearly, we have $s_n = 1$. We denote by $M(m, h)$ the maximum content in the urns after throwing the m-th ball, when the number of urns is equal to h and the throwing distribution is $p(h)$, i.e.

$$M(m, h) = \max\{U_1(m), \dots, U_h(m)\}.$$

It is easy to check that, with probability 1, we have

$$M(m, n) \in \{\lceil m/n \rceil, \dots, m\},$$

where, for every $x \in \mathbb{R}$, $\lceil x \rceil$ is the smallest integer $\geq x$.

Theorem *1:* For every $m \geq 1$, $n \geq 1$ and $k \geq 1$, we have

$$\mathbb{P}\{M(m, n) \leq k\} = \begin{cases} 1 & \text{if} \quad k \geq m \\ 0 & \text{if} \quad k \leq \lceil m/n \rceil - 1 \end{cases}$$

and, for $n \geq 2$ and $\lceil m/n \rceil \leq k \leq m - 1$,

$$\mathbb{P}\{M(m, n) \leq k\} = \sum_{j=0}^{k} \binom{m}{j} p_n^j (1 - p_n)^{m-j} \mathbb{P}\{M(m - j, n - 1) \leq k\}.$$

Proof: Let $k \geq 1$. The result is trivial in the 2 first cases. We denote by $\underline{U}_n(m)$ the row vector $(U_1(m), \dots, U_n(m))$ and by $\mathbb{1}_n$ the row vector of dimension n with all its entries equal to 1. For $n \geq 2$ and $\lceil m/n \rceil \leq k \leq m - 1$, we have

$$\begin{aligned}
&\mathbb{P}\{M(m, n) \leq k\} = \mathbb{P}\{\underline{U}_n(m) \leq k\mathbb{1}_n\} \\
&= \sum_{j=0}^{k} \mathbb{P}\{\underline{U}_{n-1}(m) \leq k\mathbb{1}_{n-1}, U_n(m) = j\} \\
&= \sum_{j=0}^{k} \mathbb{P}\{U_n(m) = j\} \mathbb{P}\{\underline{U}_{n-1}(m) \leq k\mathbb{1}_{n-1} | U_n(m) = j\} \\
&= \sum_{j=0}^{k} \binom{m}{j} p_n^j (1 - p_n)^{m-j} \mathbb{P}\{\underline{U}_{n-1}(m - j) \leq k\mathbb{1}_{n-1}\} \\
&= \sum_{j=0}^{k} \binom{m}{j} p_n^j (1 - p_n)^{m-j} \mathbb{P}\{M(m - j, n - 1) \leq k\}.
\end{aligned}$$

■

Note that in this theorem, the random variable $M(m - j, n - 1)$ is the maximum content in the urns after throwing $m - j$ balls into $n - 1$ urns, when the throwing distribution is $p(n - 1)$. From Theorem 1 we can derive the expected value of the maximum $M(m, n)$ as:

$$\begin{aligned}
E(M(m, n)) &= \sum_{k=1}^{\infty} \mathbb{P}\{M(m, n) \geq k\} \\
&= m - \sum_{k=\lceil m/n \rceil}^{m-1} \mathbb{P}\{M(m, n) \leq k\} \qquad (1)
\end{aligned}$$

where, as usual convention, an empty sum equals 0.

Lemma *2:* For every $m \geq 1$, and $n \geq 1$, we have

$$\frac{m}{n} \leq E(M(m, n)) \leq \log n + m \log(1 + q_n),$$

where $q_n = \max\{p_i, i = 1, \dots, n\}$.

Proof: The lower bound is straightforward since $M(m, n) \geq \lceil m/n \rceil$. Following the idea of [8], in which only the uniform distribution is considered, we introduce the random variable $2^{M(m,n)}$. It is easy to check that

$$2^{M(m,n)} \leq \sum_{i=1}^{n} 2^{U_i(m)}.$$

This leads to

$$\begin{aligned}
E(2^{M(m,n)}) &\leq \sum_{i=1}^{n} E(2^{U_i(m)}) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{m} 2^j \binom{m}{j} p_i^j (1 - p_i)^{m-j} \\
&= \sum_{i=1}^{n} (1 + p_i)^m \leq n(1 + q_n)^m.
\end{aligned}$$

The Jensen's inequality for concave functions applied to logarithm function in base 2 leads to

$$\begin{aligned} E(M(m,n)) &= E(\log 2^{M(m,n)}) \\ &\leq \log E(2^{M(m,n)}) \\ &\leq \log n + m \log(1+q_n). \end{aligned}$$

■

When the throwing distribution is uniform, i.e. when $p_i = 1/n$ for every $i = 1,\ldots,n$, we obtain a result similar to the one presented in [8]:

$$E(M(m,n)) \quad \leq \quad \log n + m\log(1+\frac{1}{n}). \qquad (2)$$

B. Evaluating the impact of churn in cluster-based overlays

We are now ready to derive the minimum number of join and leave events needed to trigger a topological change in cluster-based overlays. More precisely, we first derive the minimum number of events m_0 needed to trigger the first expected split or merge operation. Then, we refine this result by calculating the minimum number of events m_2 needed to trigger the first split or merge operation with probability $1-\varepsilon$ with $\varepsilon \leq 10^{-3}$. We assume that it is equiprobable that the next event occurring in the system is a join or a leave event. In the following, the lifetime of a cluster is defined as the number of join/leave operations that occur in that cluster until that cluster gives rise to a split or a merge operation.

Lemma *3:* Let $L = S_{max} - S_{min}$, where S_{max} and S_{min} respectively represent the upper bound and lower bound of clusters size, and H be the maximal expected lifetime of any cluster before it splits or merges. Then

$$H = \lfloor L^2/4 \rfloor.$$

Proof: From cluster-based overlays operations, the minimal condition for a cluster to split is that its size exceeds S_{max}, while the condition for it to merge with another cluster is that its size undershoots S_{min}. This process can be modelled as a gambler's ruin game with states $0, 1, \ldots, L$, where state k means that the number of peers in the spare set of a cluster is equal to k. States 0 and L are absorbing states. In state 0 the cluster has to merge, while in state L the cluster has to split into two new clusters. Assuming that join and leave events have the same probability, equal to $1/2$, to occur in a given cluster, then the average lifetime E_k of the game starting from state k, with $0 \leq k \leq L$, satisfies the following equations

$$E_k = \begin{cases} 0 & \text{for } k = 0 \text{ or } k = L \\ 1 + \frac{E_{k-1}}{2} + \frac{E_{k+1}}{2} & \text{for } k = 1,\ldots,L-1 \end{cases}$$

It is easily checked that the solution to these equations is given, for every $k = 0,\ldots,L$, by $E_k = k(L-k)$. By definition, H is the maximal value of E_k. It is obtained for $k = \lfloor L/2 \rfloor$ and $k = \lceil L/2 \rceil$. At both points, the value of the maximum is the same and is given by $H = \lfloor L^2/4 \rfloor$, which represents the maximal expected number of events occurring in a cluster before this cluster splits or merges. ■

In our context, Equation 1 can be interpreted as the expected maximal number of events (both join and leave events) that may occur in any given cluster of the system. Thus by setting $H = E(M(m,n))$, we deduce the minimum number m_0 of join and leave events occurring in the system, in expectation, before any cluster splits or merges. More precisely, m_0 being an integer, it is given by

$$m_0 = \inf\{m \geq 0 \mid E(M(m,n)) \geq H\} \qquad (3)$$

where n is the number of clusters of the overlay. Indeed, since $E(M(m,n))$ is increasing with m, the value m_0 exists. Note that the result obtained in [8] is obtained using an the upper bound of $E(M(m,n))$ given in Relation (2). The authors thus compute a lower bound m_1 of m_0 with

$$m_1 = \inf\{m \geq 0 \mid \log n + m\log(1+1/n) \geq \lfloor L^2/4 \rfloor\}$$

which leads to

$$m_1 = \left\lceil \frac{\lfloor L^2/4 \rfloor - \log n}{\log(1+1/n)} \right\rceil^+, \qquad (4)$$

where x^+ is defined by $x^+ = \max\{x, 0\}$. Note that when $H - \log n \leq 0$ we have $m_1 = 0$, meaning that the system is frozen since there are no joins nor leaves. Such a situation is reported in Table I, where for $N = 200$ peers in the system (actually for all values of N less than or equal to 2^8) the lower bound m_1 of m_0 is null. In the experiments below, $S_{max} = \lceil \log N \rceil$, and $n = \lceil N/S_{max} \rceil$. We set S_{min} to 4.

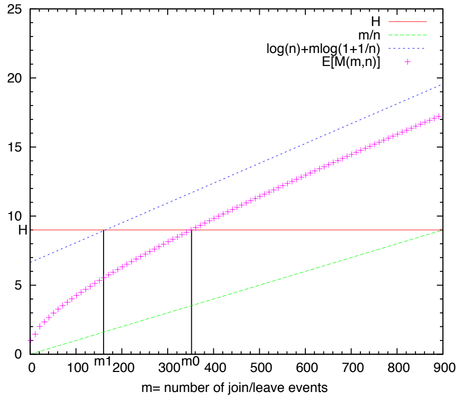

Fig. 2: Lower bounds, upper bounds and exact values of the expectation $E(M(m,n))$ as functions of m for $N = 1,000$ peers (i.e., $n = 100$ and $H = 9$).

Figure 2 plots three curves representing the respectively the expectation $E(M(m,n))$, and both its lower and upper bounds. These curves have been calculated for a population of $N = 1,000$ peers, which gives $n = 100$ clusters, and $H = 9$. From these curves we deduce both values of m_0 (as derived in equation (3)) and m_1 (equation (4)). We get $m_0 = 352$ while $m_1 = 160$. This shows the gap between the exact value of the expectation $E(M(m,n))$ and its upper bound. As we could guess, the lower bound is trivially far from $E(M(m,n))$. The main lesson learnt from these experiments is that, in expectation, a large number of join and leave events is required before the first relevant topological change occurs in cluster-based structured overlays. So far the value of m_0 has been derived by using the expectation $E(M(m,n))$ as given in Equation 1. A more accurate evaluation of the number of join and leave events occurring in the system before the first cluster splits or merges can be done at the same cost using the distribution of $M(m,n)$ instead of its expectation (see Theorem 1). Thus we define m_2 as the minimum number of join and leave events such that $M(m_2,n) > \lfloor L^2/4 \rfloor$ with a high probability, say $1-\varepsilon$, where ε is close to 0 i.,e.,

$$m_2 = \inf\{m \geq 0 \mid \mathbb{P}\{M(m,n) > H\} \geq 1-\varepsilon\}.$$

Note that, since $M(m,n)$ is increasing with m, so is $\mathbb{P}\{M(m,n) > H\}$ from 0 to 1 so that m_2 exists.

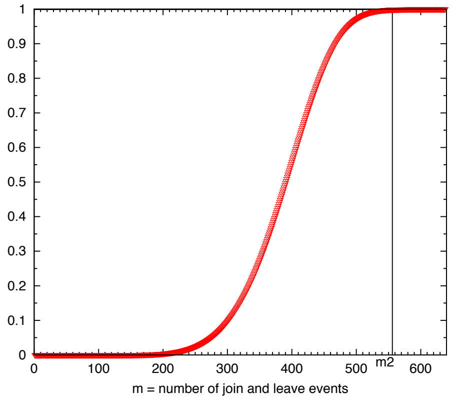

Fig. 3: $\mathbb{P}\{M(m,n) > H\}$ as a function of m for $N = 1,000$ (i.e., $n = 100$ and $H = 9$). Value $m_2 = 556$ is obtained with $\varepsilon = 10^{-3}$.

Figure 3 plots the curve of the complementary cumulative distribution function of $M(m,n)$ at point $H = \lfloor L^2/4 \rfloor$ as a function of m for a population of $N = 1,000$ peers, which gives $n = 100$ clusters, $L = 6$ and $H = 9$. We get that $m_2 = 556$ events (both join and leave) are necessary for the first cluster to split or to merge with probability 10^{-3}. This curve clearly confirms the trend that a large number of join and leave events is required before the first topological change occurs in cluster-based overlays.

Observations of both Figures 2 and 3 for specific values of N are confirmed in Table I. This table summarises for different values of the system size the corresponding values of m_0, m_1, and m_2 with $\varepsilon = 10^{-3}$, 10^{-5}, and 10^{-7}.

N	H	m_2			m_0	m_1
		$\varepsilon = 10^{-3}$	10^{-5}	10^{-7}		
200	4	72	81	87	34	0
500	6	210	233	249	116	8
600	9	363	395	417	229	130
700	9	413	449	474	261	141
800	9	462	502	530	291	150
900	9	509	553	584	322	158
1,000	9	556	604	638	352	165
2,000	12	1309	1407	1471	910	568
4,000	16	3193	3384	3518	2380	1763

TABLE I: Computation of m_0, and m_1 as function of N, and m_2 as function of both N and ε.

IV. Conclusion and Future Works

This work has precisely analysed robustness of cluster-based overlays in terms of their capability to continue to operate correctly despite high churn. It has shown that a very large number of join and leave events are globally necessary before the first topological change occurs in those overlay networks. As future work we intend to accurately predict long term behavior of cluster-based overlays by studying whether right after the first split or merge operation is triggered, a cascade of other clusters may or not reach their maximal expected lifetime, and thus may undergo a split or merge operation. In that respect the current result is a first step to a future analysis that should closely model the long term behavior of cluster-based overlays.

References

[1] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, and J. Zahorjan, "Measurement, modeling and analysis of a peer-to-peer file-sharing workload," in *Proc. of the 19th ACM Symposium of Operating Systems Principles (SOSP)*, 2003.

[2] http://en.wikipedia.org/wiki/Gnutella.

[3] http://www.kazaa.com/.

[4] C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid search schemes for unstructured peer-to-peer networks," in *Proc. of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. of the ACM SIGCOMM*, 2001.

[6] I. Stoica, D. Liben-Nowell, R. Morris, D. Karger, F. Dabek, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. of the ACM SIGCOMM*, 2001.

[7] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proc. of the Int'l Conference on Distributed Systems Platforms (Middleware)*, 2001.

[8] T. Locher, S. Schmid, and R. Wattenhofer, "eQuus: A provably robust and locality-aware peer-to-peer system," in *Proc. of the Int'l Conference on Peer-to-Peer Computing (P2P)*, 2006.

[9] M. Castro, M. Costa, and A. Rowstron, "Debunking some myths about structured and unstructured overlays," in *Proc. of the 2nd Conference on Symposium on Networked Systems Design & Implementation (NSDI)*, 2005.

[10] P. B. Godfrey, S. Shenker, and I. Stoica, "Minimizing churn in distributed systems," in *Proc. of the ACM SIGCOMM Conference*, 2006.

[11] A. Fiat, J. Saia, and M. Young, "Making chord robust to byzantine attacks," in *Proc. of the Annual European Symposium on Algorithms (ESA)*, 2005.

[12] E. Anceaume, F. Brasiliero, R. Ludinard, and A. Ravoaja, "Peercube: an hypercube-based p2p overlay robust against collusion and churn," in *Proc. of the IEEE Int'l Conference on Self Autonomous and Self Organising Systems (SASO)*, 2008.

[13] B. Awerbuch and C. Scheideler, "Group spreading: A protocol for provably secure distributed name service," in *Procs of the Int'l Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[14] S. Asaduzzaman, Y. Qiao, and G. Bochmann, "Cliquestream: An efficient and fault-resilient live streaming network on a clustered peer-to-peer overlay," in *Proc. of the Int'l Conference on Peer-to-Peer Computing (P2P)*, 2008.

[15] H. Ribeiro and E. Anceaume, "Exploiting rateless coding in structured overlays to achieve data persistence," in *Proc. of the 24th IEEE Int'l Conference on Advanced Information Networking and Applications (AINA)*, 2010.

[16] D. Malkhi, M. Naor, and D. Ratajzcak, "Viceroy: Scalable emulation of butterfly networks for distributed hash tables," in *Proc. of the ACM Annual Symposium on Principles of distributed computing (PODC)*, 2003.