Peer Rewiring in Semantic Overlay Networks under Churn
(Short Paper)

Paraskevi Raftopoulou[1,2] and Euripides G.M. Petrakis[2]

[1] University of Peloponnese (UOP), Tripoli, 22100, Greece
[2] Technical University of Crete (TUC), Chania, 73100, Greece
praftop@uop.gr, petrakis@intelligence.tuc.gr

Abstract. Semantic overlay networks have been proposed as a way to organise peer-to-peer networks; peers that are semantically, thematically or socially similar are discovered and logically organised into groups. Efficient content retrieval is then performed by routing the query towards peer groups based on their likelihood to match the query. In this paper, we study the behaviour of semantic overlay networks that support full-fledged information retrieval in the presence of peer churn. We adopt a model for peer churn, and study the effect of network dynamics on peer organisation and retrieval performance. The overlay network is evaluated on a realistic peer-to-peer environment using real-world data and queries, and taking into account the dynamics of user-driven peer participation. Using this evaluation, we draw conclusions on the performance of the system in terms of clustering efficiency, communication load and retrieval accuracy in such a realistic setting.

1 Introduction

The main idea behind peer-to-peer (P2P) is that instead of relying on central components, functionality is provided through decentralised overlay architectures, where peers typically connect to a small set of other peers. Specifically in Semantic Overlay Networks (SONs), peers that are semantically, thematically or socially similar are *organised* into groups. Queries are then selectively forwarded to those groups that have the potential to provide content matching the queries. SONs, while being highly flexible, improve query performance and guarantee high degree of peer autonomy [5]. Unlike what their name imply, SONs do not necessarily use semantics in the traditional sense (e.g., ontologies), however this is the term first proposed in the literature [3].

SONs technology has proven useful not only for information sharing in distributed environments, but also as a natural distributed alternative to Web 2.0 application domains, such as decentralised social networking in the spirit of Flickr or del.icio.us. Contrary to structured overlays that focus on providing accurate location mechanisms (e.g., [20]), SONs are better suited for loose P2P architectures, which assume neither a specific network structure nor total control

R. Meersman et al. (Eds.): OTM 2010, Part I, LNCS 6426, pp. 573–581, 2010.

over the location of the data. Additionally, SONs offer better support of semantics due to their ability to provide mechanisms for approximate, range, or text queries, and emphasise peer autonomy. Naturally, the one technique does not exclude the other, since semantic overlays can appear over structured overlays in an attempt to combine their good properties [4].

The management of large volumes of data in P2P networks has generated additional interest in methods supporting information retrieval (IR) (e.g., [6,2]). Most of these research proposals, while exploiting certain architectural [6] or modelling [2] aspects of peer organisation, assume for their experimental evaluation an ideal scenario where peers never leave or join the network. However, studies of P2P content-sharing systems have concluded that peers are typically dynamic (e.g., [21]). A peer joins the network when a user starts the application. While being connected, the peer can contribute resources to the network and search for resources provided by other peers. The peer leaves the system when the user exits the application. Stutzbach and Rejaie [21] define such a join-participate-leave cycle as a *session*. The independent arrival and departure of peers creates the collective effect called *churn*. These user-driven dynamics of peer participation is a critical issue, since churn affects the overlay structure [22], the resiliency of the overlay [8,25], the selection of key design parameters [9], and the content availability which in turn, affects retrieval effectiveness.

To the best of our knowledge, the work presented in this paper is the first to address the issues involved in the design and the evaluation of the SONs when introducing peer churn. We adopt a model for peer churn proposed by Yao et al. [25], and study the effect of network dynamics on peer organisation and retrieval performance. The overlay network is evaluated on a realistic P2P environment using real-world data and queries. Based on the results of this evaluation, we draw conclusions on the performance of the system in terms of clustering efficiency, communication load and retrieval accuracy.

The remainder of the paper is organised as follows. SON-like structures supporting IR functionality are reviewed in Section 2. Section 3 presents the model used to describe peer churn, while Section 4 presents a SON architecture and the related rewiring protocol. Finally, the experimental evaluation of the dynamic network is presented in Section 5.

2 Related Work and Background

Initial IR approaches implementing SON-like structures and supporting content search in a distributed collection of peers include the work of Li et al. [10], where semantic small world (SSW) is proposed. SSW aims to construct a self-organising network based on the semantics of data objects stored locally to peers. Along the same lines, Schmitz [18] assumes that peers share concepts from a common ontology and proposes strategies for organising peers into communities with similar concepts. i*Cluster* [15] extends these protocols by allowing peers with multiple and dynamic interests to form clusters.

Loser et al. [12] introduce the concept of semantic overlay clusters for super-peer networks. This work aims at clustering peers that store complex heterogeneous schemes by using a super-peer architecture. Along the same lines, Lu et al. [13] propose a two-tier architecture. In this architecture, a peer provides content-based information about neighbouring peers and determines how to route queries in the network. In [6], Klampanos et al. propose an architecture for IR-based clustering of peers. In this architecture, a representative peer (hub) maintains information about all other hubs and is responsible for query routing.

The notion of peer clustering based on similar interests rather than similar documents is introduced by Sripanidkulchai et al. [19]. According to this work, peers are organised on the top of the existing Gnutella network to improve retrieval performance. Emphasising on small world networks, Voulgaris et al. [23] propose an epidemic protocol that implicitly clusters peers with similar content. In [11], Loser et al. propose a three-layer organisation of peers (based both on peer content and usefulness estimators) and suggest combining information from all layers for routing queries.

Aberer at al. [1] introduce a decentralised process that, relying on pair-wise local interactions, incrementally develops global agreement and obtains semantic interoperability among data sources. Koloniari et al. [7] model peer clustering as a game, where peers try to maximise the recall for their local query workload by joining the appropriate clusters.

Most of the above presented research proposals, while exploiting certain architectural or modelling aspects of peer organisation, assume for their experimental evaluation an ideal scenario where peers never leave or join the network. In this paper, we address the issues involved in the design and the evaluation of the SONs when introducing the dynamics of user-driven peer participation.

3 Churn Model

Building upon previous work [25,8,21], we present a model of user behaviour characterising peer arrivals and departures in a P2P system. The model takes into account heterogeneous browsing habits, formalises recurring user participation in P2P systems and explains the relationship between the various lifetime distributions observable in P2P networks.

Churn Model. We consider a P2P network with N peers. Each peer p_i is either *alive* (i.e., present in the system) at a specific moment or *dead* (i.e., logged-off). This behaviour is modelled by a renewal process $\{Z_i(t)\}$ for each peer p_i, $0 \leq i \leq N$, as in [25]. This process is 1 if p_i is alive at time t, and 0 otherwise. The following assumptions are made:

1. To capture the independent nature of peers, we assume that peers behave independently of each other and processes $\{Z_i(t)\}$ and $\{Z_j(t)\}$, for any $i \neq j$, are independent. This means that peers do not synchronise their arrivals or departures and generally exhibit uncorrelated lifetime characteristics.
2. Although the model is generic enough to allow dependencies between cycle lengths, without loss of generality we treat all on-line processes (or lifetime) and off-time processes as independent and use identical distributed sets of variables. Thus, for each process $\{Z_i(t)\}$ its on-line durations $\{L_{i,t}\}$

are described by some distribution $F_i(x)$ and its off-line durations $\{D_{i,t}\}$ are described by another distribution $G_i(x)$. By this, for each peer its on-line and off-line durations are independent.

Lifetime Distribution. The on-line distribution commonly considered in the literature [25,8] is the heavy-tail distribution. The same consideration holds for the off-line distribution. To allow for arbitrarily small lifetimes, a Pareto distribution [25] is used to represent the on-line durations $F(x) = 1 - (x/x_m)^{-k}$, $x_m > 0,\ k > 1$, where x stands for the on-line durations. Off-line durations are respectively represented by an alike Pareto distribution $G(x)$. Each Pareto distribution is parameterised by the quantities x_m and k, which stand for the *scale* and the *shape* of the distribution respectively. The scale parameter sets the position of the left edge of the probability density function. The shape parameter determines the skewness of the distribution.

Peer Availability. The *availability* a_i of peer p_i is the probability that p_i is in the system at a random moment. Intuitively, we expect this probability to represent the lifetime of a peer over the entire lifetime of the system. Yao et al. [25] define the average on-line duration (or lifetime) of a peer p_i as $l_i = E[L_i]$ and its average off-line duration as $d_i = E[D_i]$. The availability a_i of peer p_i is then calculated, in the spirit of [17], as $a_i = \lim_{t\to\infty} P(Z_i(t) = 1) = \frac{l_i}{l_i+d_i}$.

According to this model, the only parameters that control a peer's availability are the on-line l_i and the off-line d_i durations. Notice that these parameters are *independent* and *unique* for each peer. Parameters l_i and d_i are drawn independently from two Pareto distributions. Once pair (l_i, d_i) is generated for each peer p_i, it remains constant for the entire evolution of the system.

4 A Semantic Overlay Network

The present work uses i*Cluster* P2P network [15], which extends the idea of peer organisation in small-world networks by allowing peers to have multiple and dynamic interests. i*Cluster* peers are responsible for serving both users searching for and users contributing information to the network. Each i*Cluster* peer is characterised by its information content (i.e., its document collection), which may be either automatically (by text analysis) or manually (e.g., tags or index terms) assigned to each document. To identify its *interests*, a peer categorises its documents by using an external reference system (i.e., an ontology as in [18] or a taxonomy such as the ACM categorisation system) or by clustering. Thereupon, a peer may be assigned *more than one* interests. Interests are created and deleted dynamically to reflect the documents a peer contributes to the network.

Each peer maintains (for each interest) a *routing index* (RI) holding information for short- and long-range links to other peers. Short-range links connect peers inside a SON (i.e., peers with similar interest), while long-range links are used to interconnect SONs. Entries in the routing index contain the IP addresses of the peers the links point to and the corresponding interests of these peers.

The basic protocols that determine the way peers join the overlay network, connect to and disconnect from the network, and the way queries are processed are thoroughly presented in [16]. Below, we present the protocol that specifies the way peers dynamically self-organise into SONs.

Rewiring Protocol. Peer organisation proceeds by establishing new connections to similar peers and by discarding old ones. Each peer p_i periodically (e.g., when joining the network or when its interests have changed) initiates a rewiring procedure (independently for each interest) by computing the intra-cluster (or *neighborhood*) similarity $NS_i = \frac{1}{s} \cdot \sum_{\forall p_j \in RI_i} sim(I_i, I_j)$, where s is the number of short-range links of p_i according to interest I_i, p_j is a peer contained in RI_i that is on-line, I_j is the interest of p_j, and $sim()$ can be any appropriate similarity function (e.g., the cosine similarity between the term vector representations). The neighborhood similarity NS_i is used here as a measure of *cluster cohesion*, indicating whether peers with similar interests are gathered together.

If NS_i is greater than a threshold θ, then p_i does not need to take any further action, since it considers that it is surrounded by peers with similar interests. Otherwise, p_i issues a $\textsc{FindPeers}(ip(p_i), I_i, L, \tau_R)$ message, where L is a list and τ_R is the time-to-live (TTL) of the message. List L is initially empty and will be used to store tuples of the form $\langle ip(p_j), I_j \rangle$, containing the IP address and interest of peers discovered while the message traverses the network. System parameters θ and τ_R need to be known upon bootstrapping.

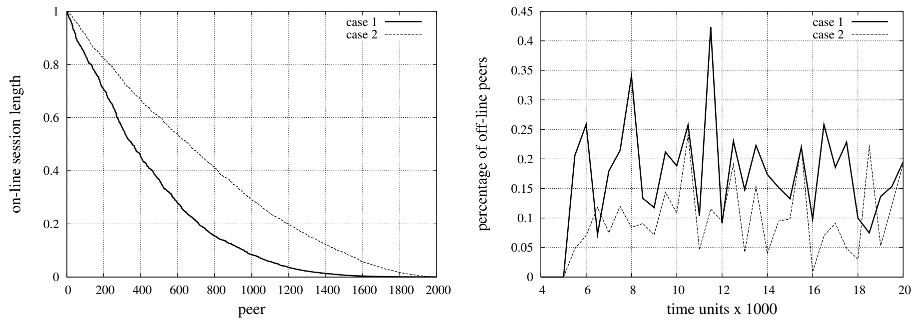
A peer p_j receiving the $\textsc{FindPeers}()$ message appends its IP address $ip(p_j)$ and its interest I_j (or the interest most similar to I_i if p_j has multiple interests) to L, reduces τ_R by one, and forwards the message to m random neighbouring peers ($m \leq s$). This message forwarding technique is referred to in the literature as *random walk* (RW) [18,16]. When $\tau_R = 0$, the $\textsc{FindPeers}()$ message is sent back to the message initiator p_i. When the message initiator p_i receives the $\textsc{FindPeers}()$ message back, it uses the information in L to update its routing index RI_i by replacing old short-range links with new links to peers with more similar interests. For the update of the long-range links, peer p_i uses a random walk in the network [16].

5 Evaluation

In this section, we evaluate the effect of peer churn on SONs, using the i*Cluster* rewiring protocol and a realistic dynamic setting. Typically, in the evaluation of P2P IR systems performance is measured in terms of *network traffic* (i.e., the number of rewiring/search messages sent over the network) and retrieval effectiveness (i.e., *recall*). In our setting precision is always 100% since only relevant documents are retrieved. Peer clustering is measured by *clustering efficiency* $\bar{\kappa}$ [14], which gives information about the underlying network structure.

5.1 Experimental Testbed

Dataset. The dataset, also used in the evaluation of [24,16], contains over 556,000 web documents from the TREC-6 collection belonging in 100 categories.



(a) Distributions of on-line session lengths (b) Off-line peers over time

Fig. 1. Lifetime distributions

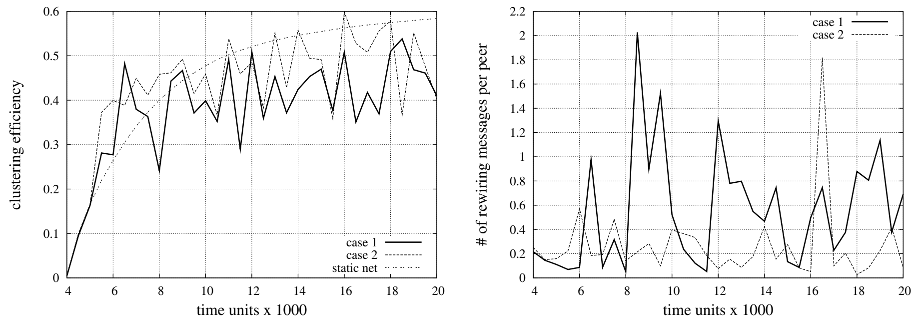
The queries employed in the evaluation of the corpus are strong representatives of document categories (i.e., the topics of the categories).

Setup. We consider 2000 loosely-connected peers, each of which contributes documents in the network from a single category. At the bootstrapping phase, peers join the network and are connected as described in [16]. We experimented with different values of the parameters. We consider that a given parameter value is better than another if it results in better clustering and retrieval for less communication load. These results are out of the scope of this paper and the interested user may refer to [16]. The simulator used to evaluate the rewiring protocol was implemented in C/C++ and all experiments were run on a Linux machine. Our results were averaged over 25 runs (5 random initial network topologies and 5 runs for each topology). Query processing is carried out as described in [16].

Lifetime Distributions. We experimented with different lifetime distributions. Figure 1(a) illustrates two different on-line session length distributions. By definition, the more skewed the distribution is, the smaller the lifetimes of the most peers are. The first case in Figure 1(a) corresponds to a difficult scenario compared to the second case, since peers are on-line for shorter time periods and leave the network more often. Figure 1(b) presents the percentage of off-line peers as a function of time. In the first scenario the percentage of the peers that are logged-off reaches up to 42% ($t = 11.5K$), while in the second scenario the percentage of the logged-off peers every moment is kept under 25%.

5.2 Experimental Evaluation

Rewiring Effectiveness. Figure 2(a) presents clustering efficiency as a measure of network organisation over time. The plots presented in the figure correspond to the two dynamic scenarios discussed in the previous section. A static network (with peers always connected to the network) is used as the baseline for our evaluation. Initially ($t \leq 4K$), the clustering efficiency is very low indicating that peers are still randomly connected. We observe that during the initial convergence period ($t \leq 5K$) peers self-organise into clusters improving the clustering



(a) Clustering efficiency (b) Rewiring messages

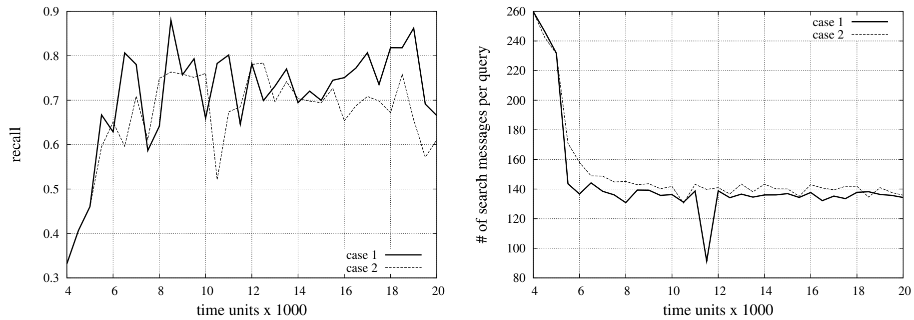
Fig. 2. Rewiring effectiveness over time

efficiency. In the case of the static network, rewiring finally converges towards a stable network organisation and a constant value of clustering efficiency. In the case of the dynamic network, peers arbitrarily leave and join the system and by this, links pointing to off-line peers as well as newly-connected peers emerge. As a result, peer connections continuously change and peers try (by rewiring) to recover network organisation, a situation continuously repeated.

As shown in Figure 2(a), the rewiring protocol manages to organise the network and clustering efficiency is eventually kept high compared to the unorganised network. Naturally, the network loses its clustering cohesion at moments of high churn, e.g. in the first case when $t = 8K$ and 35% of the peers are off-line, but the network manages to quickly recover in all cases. We are thus, driven to the conclusion that SONs are resilient to churn; the rewiring protocol manages to quickly reorganise the network and keep the values of clustering efficiency high. Notice, by collating Figures 2(a) and 1(b), that clustering efficiency is narrowly related to the percentage of peers that are off-line; when the percentage of off-line peers increases clustering efficiency decreases, and vice versa.

In terms of rewiring messages, peer churn imposes a continuous need for peer organisation, which in turn leads to message traffic. Figure 2(b) presents the number of rewiring messages over time. As churn increases, more peers need to rewire their links and more messages traverse the network: in the first scenario rewiring messages are on average 3 times more than the second, less dynamic, one. Compared to a static network, the rewiring procedure in a dynamic setting needs on average 2 times more and on the worst case 7 times more messages to keep the SON organised.

Retrieval Effectiveness. Figure 3(a) presents the performance of retrievals under churn as a function of time. We observe that recall attains high values throughout the entire evolution of the system. In particular, during the initial convergence period ($t \leq 5K$), the peers self-organise into clusters and the retrieval performance is almost doubled compared to the initial unorganised network. After this point, peers continuously leave and re-join the network. The arbitrarily connected peers naturally cause troubles to the retrieval performance



(a) Retrieval efficiency (b) Search messages

Fig. 3. Retrieval effectiveness over time

of the network. However, Figure 3(a) shows that recall is always kept high (in the the worst case of the second scenario ($t = 10.5K$) recall is 55% better compared to the recall on an unorganised network).

Figure 3(b) shows the number of messages per query over time for the two different scenarios. When the network is not organised ($t = 4K$), a high number of search messages is needed to retrieve the available data relevant to a query. However, this message overhead is decreased (more than 85%) as peers continuously get organised into clusters with similar interests. Notice though, that the number of search messages is kept low throughout the evolution of the system, regardless the continuous changes in the network structure. This happens because the number of search messages is related to the number of neighbouring peers, that under a dynamic scenario is interpreted to the number of *available* neighbouring peers. In cases of high churn, a noticeable number of peers have gone off-line, the network size has been reduced and the overlay has lost its structure. The rewiring mechanism reinstates the network organisation, but the number of search messages is kept low since the neighborhoods are sparsely populated (notice that the lowest number of messages is achieved in the first scenario at $t = 11.5K$, when the percentage of off-line peers is the higher achieved).

References

1. Aberer, K., Cudre-Mauroux, P., Hauswirth, M.: The Chatty Web: Emergent Semanics Through Gossiping. In: WWW 2003 (2003)
2. Backes, M., Hamerlik, M., Linari, A., Maffei, M., Tryfonopoulos, C., Weikum, G.: Anonymous and Censorship-resistant Content-sharing in Unstructured Overlays. In: PODC 2008 (2008)
3. Crespo, A., Garcia-Molina, H.: Routing Indices for P2P Systems. In: ICDCS 2002 (2002)
4. Doulkeridis, C., Vlachou, A., Norvag, K., Vazirgiannis, M.: Handbook of Peer-to-Peer Networking. Springer, Heidelberg (2010)
5. Garcia-Molina, H., Yang, B.: Efficient Search in P2P Networks. In: ICDCS 2002 (2002)

6. Klampanos, I., Jose, J.: An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks. In: ACM SAC 2004 (2004)
7. Koloniari, G., Pitoura, E.: Recall-based Cluster Reformulation by Selfish Peers. In: NetDB 2008 (2008)
8. Leonard, D., Yao, Z., Rai, V., Loguinov, D.: On Lifetime-Based Node Failure and Stohastic Resilience of Decentralised Peer-to-Peer Networks. IEEE/ACM Transactions on Networking 15(3) (2007)
9. Li, J., Stribling, J., Kaashoek, F., Morris, R., Gil, T.: A Performance vs. Cost Framework for Evaluating DHT Design Tradeoffs under Churn. In: INFOCOM 2005 (2005)
10. Li, M., Lee, W.-C., Sivasubramaniam, A.: Semantic Small World: An Overlay Network for Peer-to-Peer Search. In: ICNP 2004 (2004)
11. Loser, A., Tempich, C.: On Ranking Peers in Semantic Overlay Networks. In: Althoff, K.-D., Dengel, A.R., Bergmann, R., Nick, M., Roth-Berghofer, T.R. (eds.) WM 2005. LNCS (LNAI), vol. 3782, Springer, Heidelberg (2005)
12. Loser, A., Wolpers, M., Siberski, W., Nejdl, W.: Semantic Overlay Clusters within Super-Peer Networks. In: DBISP2P 2003 (2003)
13. Lu, J., Callan, J.: Content-based Retrieval in Hybrid Peer-to-peer Networks. In: CIKM 2003 (2003)
14. Raftopoulou, P., Petrakis, E.: A Measure for Cluster Cohesion in Semantic Overlay Networks. In: LSDS-IR 2008 (2008)
15. Raftopoulou, P., Petrakis, E.G.M.: iCluster: a Self-Organising Overlay Network for P2P Information Retrieval. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, Springer, Heidelberg (2008)
16. Raftopoulou, P., Petrakis, E.G.M., Tryfonopoulos, C.: Rewiring Strategies for Semantic Overlay Networks. Distributed and Parallel DataBases 2–3(26) (2009)
17. Saroiu, S., Gummadi, K.P., Gribble, S.D.: A Measurement Study of Peer-to-Peer File Sharing Systems. In: MMCN 2002 (2002)
18. Schmitz, C.: Self-Organization of a Small World by Topic. In: P2PKM 2004 (2004)
19. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient Content Location using Interest-Based Locality in Peer-to-Peer Systems. In: INFOCOM 2003 (2003)
20. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Frans Kaashoek, M., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking 11(1) (2003)
21. Stutzbach, D., Rejaie, R.: Understanding Churn in Peer-to-Peer Networks. In: INFOCOM 2006 (2006)
22. Stutzbach, D., Rejaie, R., Sen, S.: Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. In: IMC 2005 (2005)
23. Voulgaris, S., van Steen, M., Iwanicki, K.: Proactive Gossip-based Management of Semantic Overlay Networks. Concurrency and Computation: Practice and Experience 19(17) (2007)
24. Xu, J., Croft, W.B.: Cluster-Based Language Models for Distributed Retrieval. In: ACM SIGIR 1999 (1999)
25. Yao, Z., Leonard, D., Wang, X., Loguinov, D.: Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks. In: ICNP 2006 (2006)