Peer Rewiring in Semantic Overlay Networks under Churn

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

[1]Dept. of of Computer Science and Technology
University of Peloponnese (UOP), Tripoli, 22100, Greece
praftop@uop.gr
[2]Dept. of Electronic and Computer Engineering
Technical University of Crete (TUC), Chania, 73100, Greece
petrakis@intelligence.tuc.gr

Abstract. In this paper, we study the behaviour of a semantic overlay network that supports 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. Our work is the first to demonstrate the issues involved in the design of semantic overlay networks when introducing peer churn. 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

Peer-to-peer (P2P) systems offer the potential for low-cost sharing of information, while ensuring autonomy and privacy of the participating entities. The main idea behind P2P is that instead of relying on central components, functionality is provided through decentralised overlay architectures. In overlay networks, peers typically connect to a small set of other peers. Queries are then, propagated to the network searching for information qualifying query criteria by utilising existing connections and following a predetermined query forwarding strategy. The popularity of earlier systems, like Gnutella[1] and Freenet[2], built upon the idea of unstructured overlay networks, has propelled research in this direction, while lately, the proliferation of social networking has added yet another interesting dimension to the problem of searching for content.

In Semantic Overlay Networks (SONs), peers that are semantically, thematically or socially similar are *organised* into groups. SONs, while being highly flexible, improve query performance and guarantee high degree of peer autonomy [2, 15, 10]. This technology has proven useful not only for information sharing in

[1] http://www.gnu.org/
[2] http://freenetproject.org/

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[3] or del.icio.us[4]. Contrary to structured overlays that focus on providing accurate location mechanisms (e.g., [20, 11]), SONs are better suited for loose P2P architectures, which assume neither a specific network structure nor total control 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.

The management of large volumes of data in P2P networks has generated additional interest in methods for effective network organisation based on peer contents and consequently, in methods supporting information retrieval (IR) (e.g., [4]). Most of these research proposals, while exploiting certain architectural [4] or modelling [1] 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 [5, 25], the selection of key design parameters [6], 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 and also research on modeling peer dynamics are reviewed in Sec. 2. Section 3 presents the model used to describe peer churn, while Sec. 4 presents a SON architecture and the related rewiring protocol. The experimental evaluation of the system is presented in Sec. 5, followed by conclusions and issues for further research in Sec. 6.

[3] http://www.flickr.com/
[4] http://www.del.icio.us.com/

2 Related Work and Background

This section provides a brief survey of the work related to data organisation and retrieval in SONs, along with research on modelling the dynamics of P2P networks.

2.1 Semantic Overlay Networks

Initial IR approaches implementing SON-like structures supporting content search in a distributed collection of peers include the work of Lu et al. [8], where a two-tier architecture is proposed. In this architecture, a peer provides content-based information about neighbouring peers and determines how to route queries in the network. Along the same lines, Klampanos et al. [4] 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 in the work of Sripanidkulchai et al. [17].

Additional work on peer organisation using SONs is based on the idea of "small-world networks". Li et al. [7] propose creating a self-organising semantic small world (SSW) network based on the semantics of data objects stored locally to peers. Voulgaris et al. [23] propose an epidemic protocol that implicitly clusters peers with similar content. Along the same lines, Schmitz [15] assumes that peers share concepts from a common ontology and this information is used for organising peers into communities with similar contents. Most of these works assume a static peer network for their evaluation.

2.2 Churn Models

Characterising churn requires fine-grained and unbiased information about the arrival and departure of peers in a network. This task is rather challenging due to the large size and highly dynamic nature of P2P systems. Several studies (e.g., [3]) present a high level view of churn by analysing its characteristics (e.g., median session length) in large scale P2P systems. Gummadi et al. [3] measure session lengths by monitoring a router at the University of Washington. Sen et al. [16] analyse P2P traffic by monitoring flows in FastTrack[5], Gnutella and DirectConnect[6]. Sripanidkulchai et al. [18] study the live streaming workload of a large content delivery network, and present an analysis characterising popularity, arrival process and session length.

All studies infer that session lengths follow some known probability distribution ranging from Poison to heavy-tailed (or Pareto) distributions. Stutzbach and Rejaie [21] identify the key challenges in characterising churn, determine common pitfalls in measuring churn such as biased peer selection, which they believe are the main factors for the conflicting results, and develop techniques

[5] http://www.kazaa.com/
[6] http://www.neo-modus.com/

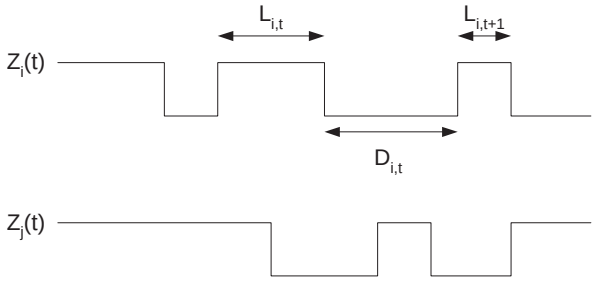


Fig. 1. On-line and off-line behaviour for peers p_i and p_j

to address these difficulties. Along the same lines, Leonard et al. [5] present a realistic model for peer lifetimes in P2P networks and investigate the resilience of random graphs to lifetime-based peer failure. Yao et al. [25] introduce a generic model that captures the heterogenous behaviour of peers. They view each peer as an alternating renewal process and consider that on-line/off-line durations are independent and unique for each peer.

3 Churn Model

Building upon previous work [25, 5, 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 participating 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 as in [25]:

$$Z_i(t) = \begin{cases} 1, p_i \text{ is alive at time } t; \\ 0, \text{ otherwise.} \end{cases}, \quad 0 \leq i \leq N. \tag{1}$$

This framework is illustrated in Fig. 1 that presents the renewal processes $\{Z_i(t)\}$ and $\{Z_j(t)\}$ for peers p_i and p_j respectively during time period t. The random variables $L_{i,t} > 0$ and $D_{i,t} > 0$ represent the on-line and off-line durations for peer p_i respectively.

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 lifetime and off-time processes as independent and we use identical distributed sets of variables. Thus, for each process $\{Z_i(t)\}$ its on-line durations $\{L_{i,t}\}$ are described by some joint distribution $F_i(x)$ and its off-line durations $\{D_{i,t}\}$ are described by another joint distribution $G_i(x)$. This means that for each peer its on-line and off-line durations are independent.

Lifetime Distribution. A Pareto distribution [25] is used to represent the on-line durations:

$$F(x) = 1 - (x/x_m)^{-k},\ x_m > 0,\ k > 1 \tag{2}$$

Off-line durations are respectively represented by an alike Pareto distribution $G(x)$. 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. The shape parameter determines the skewness of the distribution.

Peer Availability. 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 (i.e., the probability that p_i is in the system at a random moment) is calculated, in the spirit of [14], as:

$$a_i = \lim_{t\to\infty} P(Z_i(t) = 1) = \frac{l_i}{l_i + d_i}. \tag{3}$$

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. Note 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 for Information Retrieval

The present work builds on i*Cluster* P2P network [10], 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 information 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 assigned to each document (e.g., tags or index terms). To identify its *interests*, a peer categorises its documents by using an external reference system (i.e., an ontology as in [15] or a taxonomy such as the ACM categorisation system) or by clustering [19]. Thereupon, a peer may be assigned more than one interests. Interests are created and deleted dynamically to reflect a peer's variety in the documents it contributes to the network.

Each peer maintains a *routing index* (RI) holding information for short- and long-range links to other peers:

short-range links correspond to *intra-cluster* information (i.e., links to peers with similar interests)

long-range links correspond to *inter-cluster* information (i.e., links to peers having different interests and thus belonging to different clusters)

Entries in the routing index contain the IP addresses of the peers the links point to and the corresponding interests of these peers.

The idea is to let peers *self-organise* into *clusters* with similar content. Peer organisation is achieved through a *rewiring protocol* that is (periodically) executed independently by each peer. The purpose of this protocol is to establish connections among peers with *similar* interests. Eventually, by creating new connections to peers and by discarding connections that are outdated, dynamic clusters of peers *emerge*. Overall, rewiring is a highly dynamic procedure that involves simultaneous operations by many peers inducting to peer clusters. Queries can then be resolved by routing the query towards clusters based on their likelihood to match the query. Once reaching a cluster, the peer receiving the query is responsible for forwarding it to other peers within the same cluster.

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 [10]. Below, we present the protocol that specifies the way peers dynamically self-organise into clusters with similar content.

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 similarity (or *neighborhood similarity*)

$$NS_i = \frac{1}{s} \cdot \sum_{\forall p_j \in RI_i} sim(I_i, I_j), \tag{4}$$

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 [13]). The neighborhood similarity NS_i is used here as a measure of *cluster cohesion*. If NS_i is greater than a threshold θ, then p_i does not need to take any further action, since it is surrounded by peers with similar interests. Otherwise, p_i issues a 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 FindPeers() message appends its IP address $ip(p_j)$ and its interest I_j to L (or the interest most similar to I_i if p_j has multiple interests), reduces τ_R by one, and forwards the message to the m neighbouring peers ($m \leq s$) with interests most similar to I_i. This message forwarding technique is referred to in the literature as *gradient walk* (GW) [12, 15]. When $\tau_R = 0$,

Procedure Rewiring($p_i, I_i, \tau_R, \theta, m$)
Initiated by p_i when neighborhood similarity NS_i drops below θ.

input: peer p_i with interest I_i and routing index RI_i
output: updated routing index RI_i

1: compute $NS_i = \frac{1}{s} \cdot \sum_{\forall p_j \in RI_i} sim(I_i, I_j)$
2: **if** $NS_i < \theta$ **then**
3: $L \leftarrow \{\ \}$
4: create FindPeers()
5: $p_k \leftarrow p_i$
6: **repeat**
7: $L \leftarrow L :: \langle ip(p_k), I_k \rangle$
8: send FindPeers() to
m neighbours of p_k with interests most similar to I_i
9: $p_k \leftarrow$ every peer receiving FindPeers()
10: $\tau_R \leftarrow \tau_R - 1$
11: **until** $\tau_R = 0$
12: return list L to p_i
13: update RI_i with information from L

Fig. 2. The rewiring protocol

the FindPeers() message is sent back to the message initiator p_i. Figure 2 illustrates the above rewiring procedure in algorithmic steps.

When the message initiator p_i receives the FindPeers() message back, it utilises the information contained in L to update its routing index RI_i by replacing old short-range links corresponding to peers with less similar interests with new links corresponding to peers with more similar interests.

Peers also store long-range links in their routing indexes which stand as short paths to dissimilar clusters. For the update of the long-range links, peer p_i uses a random walk[7] in the network with TTL τ_R.

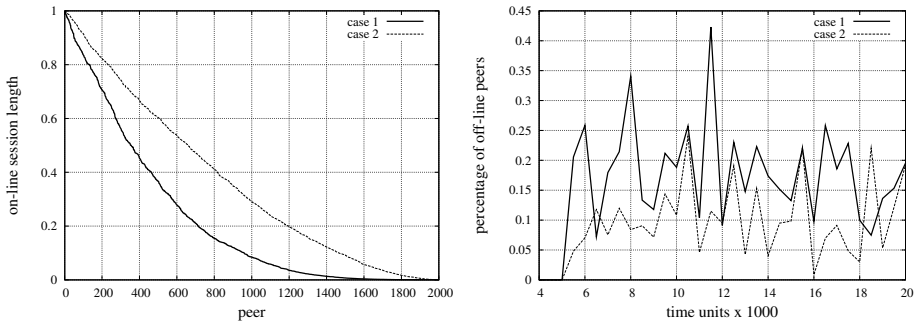
5 Evaluation

In this section, we evaluate the rewiring protocol of i*Cluster* under churn in a realistic setting using realistic data and queries.

5.1 Performance Measures

As it is typical in the evaluation of P2P information retrieval systems, performance is measured in terms of network traffic and retrieval effectiveness. The *network traffic* is measured by the number of rewiring (respectively search) messages sent over the network during rewiring (respectively querying). The retrieval effectiveness is evaluated using *recall* (i.e., the number of relevant documents retrieved over the total number of relevant documents in the network). Notice that in our setting precision is always 100% since only relevant documents are

[7] A random walk in a P2P network means visiting successive random peers.



(a) Different distributions of on-line session lengths

(b) Percentage of off-line peers as a function of time

Fig. 3. Lifetime distributions

retrieved. To describe the effect of peer clustering, we use *clustering efficiency* $\bar{\kappa}$ [9], a measure that gives information about the underlying network structure. Clustering efficiency takes values in the interval $[0, 1]$. The higher its value is the better the organisation of the network is considered.

5.2 Experimental Testbed

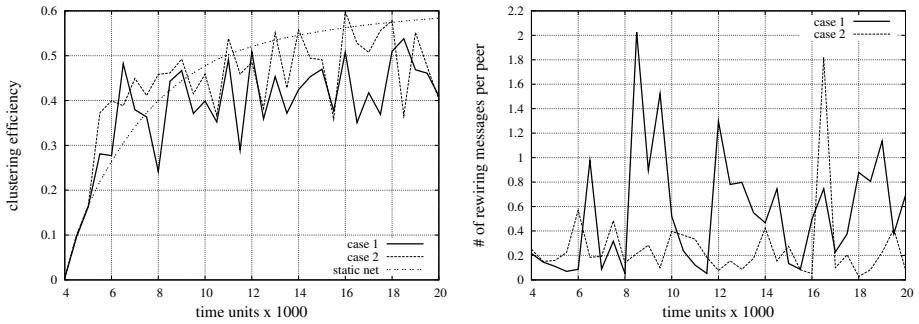
Dataset. The dataset contains over 556,000 web documents from the TREC-6[8] collection belonging in 100 categories, and has been previously used to evaluate information retrieval algorithms over distributed document collections (e.g., [24]). The queries employed in the evaluation of the corpus are strong representatives of document categories (i.e., the topics of the categories).

Setup. We consider N loosely-connected peers, each of which contributes documents in the network from a single category. The base unit for time used in the experiments is the period t. The start of the rewiring procedure for each peer is randomly chosen from the interval $[0, 4000]$ and its periodicity is randomly selected from a normal distribution of 2000, in the spirit of [15]. Therefore, each peer starts (and goes over again) independently the rewiring process. We start recording the network activity at time $4000 \cdot t$, when all peers have initiated the rewiring procedure at least once.

We experimented with different values of similarity threshold θ, message forwarding TTL τ_R and query forwarding TTLs τ_b, τ_f. We consider that a given parameter value is better than another if it results in better clustering and retrieval for less communication load.

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).

[8] http://boston.lti.cs.cmu.edu/callan/Data/



(a) Clustering efficiency as a function of time

(b) Rewiring messages as a function of time

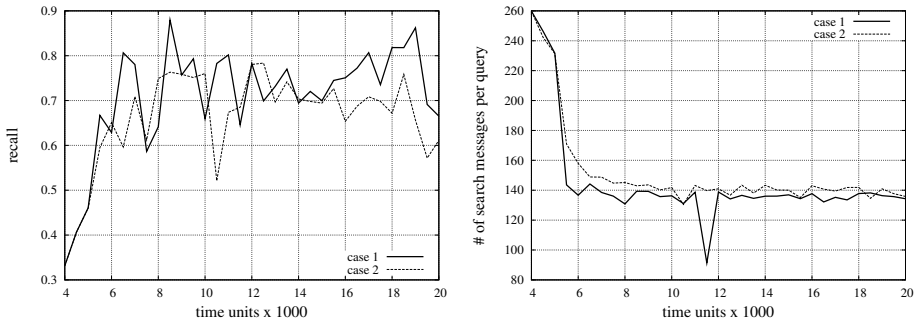
Fig. 4. Rewiring effectiveness

Lifetime Distributions. We experimented with different lifetime distributions. Figure 3(a) illustrates two different on-line session lengths distributions. By definition, the most skewed the distribution is, the smaller the lifetimes of the most peers are. The first case in Fig. 3(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 3(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% (for $t = 11.5K$), while in the second scenario the percentage of the peers that are logged-off every moment is kept under 25%.

5.3 Experimental Evaluation

Rewiring Effectiveness. Figure 4(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. We observe that during the initial convergence period ($t \leq 5K$) peers self-organise into clusters improving the clustering 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 Fig. 4(a), the peer organisation protocol manages to organise the network and clustering efficiency is eventually kept high compared to the unorganised network ($t = 4K$). Naturally, the network looses its clustering cohesion at moments of high churn, as for example 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 the rewiring protocol manages



(a) Retrieval efficiency as a function of time

(b) Search messages as a function of time

Fig. 5. Retrieval effectiveness

to quickly reorganise the network and keep the values of clustering efficiency high. Notice, by collating Fig. 4(a) and 3(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 continual message traffic. Figure 4(b) presents the number of rewiring messages as a function of time. Specifically, as churn increases more peers need to rewire their links and more messages traverse the network: 3 times more rewiring messages are sent on average in the first scenario (where off-line session lengths are larger) compared to the second scenario that is considered easier. Compared to a static network, the rewiring task needs on average 2 times more and on the worst case 7 times more messages when operating under a dynamic network. As a conclusion, i*Cluster* manages to keep the network organised at the exchange of high network traffic.

Retrieval Effectiveness. Figure 5(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 leave and re-join the network. Because of this, links pointing to off-line peers or to non-organised peers, as well as newly joined peers emerge. The arbitrarily connected peers naturally cause troubles to the retrieval performance of the network. However, Fig. 5(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). This remark coincides with the results presented in Fig. 4(a) inducting thus, that the network recovers quickly and that rewiring proves to be resilient to peer churn.

Figure 5(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 are 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 many peers are off-line and the neighborhoods are sparsely populated. To confirm, notice that the lowest number of messages is achieved in the first scenario when the percentage of off-line peers is the higher achieved ($t = 11.5K$).

Summarising, the overlay structure is directly related to peer churn: the more dynamic the network is, the more often peers connect to and disconnect from the network affecting the organisation of the network. i*Cluster* tries and eventually manages to retain a clustered organisation of peers by rewiring. However, this organisation does not come for free. Network organisation affects in turn, the retrieval performance. By rewiring the system manages to retain high values of recall and low values of search messages per query by keeping the network organised even under high peer churn.

6 Conclusions

In this work, we studied the performance of P2P networks under churn. Building upon established peer churn models describing user behaviour in a realistic setting (where peers connect and disconnect from the network), we evaluated the effects of churn on network organisation and the sufficiency of rewiring to retain a clustered organisation of peers. For this we relied on i*Cluster* [10], an approach for organising peers into highly dynamic networks and supporting efficient information retrieval. Our experimental results demonstrate that rewiring achieves high clustering degree and high recall. This is the first work to address the issues involved in the design and the evaluation of a rewiring protocol when introducing peer churn. Incorporating replication and caching (to ensure consistency between redundant resources, to improve reliability, fault-tolerance, or accessibility) in self-organising dynamic P2P networks, and studying more elaborate methods for load balancing (in terms of processing and communication activity) are important issues for future research.

References

1. K. Aberer, P. Cudre-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semanics Through Gossiping. In *WWW*, 2003.
2. H. Garcia-Molina and B. Yang. Efficient Search in Peer-to-Peer Networks. In *ICDCS*, 2002.

3. 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 *SOSP*, 2003.
4. I. Klampanos and J. Jose. An Architecture for Information Retrieval over Semi-Collaborating Peer-to-Peer Networks. In *ACM SAC*, 2004.
5. D. Leonard, Z. Yao, V. Rai, and D. Loguinov. On Lifetime-Based Node Failure and Stohastic Resilience of Decentralised Peer-to-Peer Networks. *IEEE/ACM Transactions on Networking*, 15(3), 2007.
6. 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*, 2005.
7. M. Li, W.-C. Lee, and A. Sivasubramaniam. Semantic Small World: An Overlay Network for Peer-to-Peer Search. In *ICNP*, 2004.
8. J. Lu and J. Callan. Content-based Retrieval in Hybrid Peer-to-peer Networks. In *CIKM*, 2003.
9. P. Raftopoulou and E. Petrakis. A Measure for Cluster Cohesion in Semantic Overlay Networks. In *LSDS-IR*, 2008.
10. P. Raftopoulou and E.G.M. Petrakis. iCluster: a Self-Organising Overlay Network for P2P Information Retrieval. In *ECIR*, 2008.
11. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *ACM SIGCOMM*, 2001.
12. J. Sacha, J. Dowling, R. Cunningham, and R. Meier. Discovery of Stable Peers in a Self-Organising Peer-to-Peer Gradient Topology. In *DAIS*, 2006.
13. G. Salton. *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*. Addison-Wesley, 1989.
14. S. Saroiu, K. P. Gummadi, and S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In *MMCN*, 2002.
15. C. Schmitz. Self-Organization of a Small World by Topic. In *P2PKM*, 2004.
16. S. Sen and J. Wang. Analyzing Peer-To-Peer Traffic Across Large Networks. *IEEE/ACM Transactions on Networking*, 12(2), 2004.
17. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient Content Location using Interest-Based Locality in Peer-to-Peer Systems. In *INFOCOM*, 2003.
18. K. Sripanidkulchai, B. Maggs, and H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In *IMC*, 2004.
19. M. Steinbach, G. Karypis, and V. Kumar. A Comparison of Document Clustering Techniques. In *KDD Workshop on Text Mining*, 2000.
20. I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. Frans Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Transactions on Networking*, 11(1), 2003.
21. D. Stutzbach and R. Rejaie. Understanding Churn in Peer-to-Peer Networks. In *INFOCOM*, 2006.
22. D. Stutzbach, R. Rejaie, and S. Sen. Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. In *IMC*, 2005.
23. S. Voulgaris, M. van Steen, and K. Iwanicki. Proactive Gossip-based Management of Semantic Overlay Networks. *Concurrency and Computation: Practice and Experience*, 19(17), 2007.
24. J. Xu and W.B. Croft. Cluster-Based Language Models for Distributed Retrieval. In *ACM SIGIR*, 1999.
25. Z. Yao, D. Leonard, X. Wang, and D. Loguinov. Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks. In *ICNP*, 2006.