Self-organizing Peer-to-Peer Network for Collaborative Document Tracking

Hathai Tanta-ngai Evangelos E. Milios Vlado Kešelj
Faculty of Computer Science, Dalhousie University
Halifax, Nova Scotia, Canada
http://www.cs.dal.ca/~{hathai,eem,vlado}

ABSTRACT

Given a set of peers with overlapping interests where each peer wishes to keep track of new documents that are relevant to their interests, we propose a self-organizing peer-to-peer document-tracking network based on common interest profiles. The goal of a document-tracking network is to disseminate new documents as they are published. Peers collaboratively share new documents of interest with other peers. There is no explicit profile exchange between peers and no global information available. We describe a strategy for peers to discover the existence of other peers and learn about their interests locally, based on information carried in the document metadata that propagates through the network. Peers are connected based on their observed common interests. We compare our proposed common interest strategy with a randomly connected network. The experimental results, based on simulated environment using the ACM digital library metadata, demonstrate that the proposed strategy gives the best dissemination performance. We also demonstrate that our self-organizing networks follow the characteristics of social networks.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: On-line Information Services—*Data sharing*; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries—*Dissemination*

General Terms

Algorithm, Experimentation, Measurement

Keywords

Peer-to-peer, Social networks, Document tracking, Dissemination

1. INTRODUCTION

Many tools have been developed to help researchers find documents of interest over the Internet. These tools include search engines, mailing lists, online archives, and online document sharing services such as BibSonomy [1] and Citeulike [2]. However, existing tools pose a number of challenges for researchers in getting informed about newly published documents, or new documents for short. For instance, to keep track of new documents of interest, researchers need to revisit search engines or online archives multiple times and issue the same query. Mailing lists can also be used for keeping track of new documents; however, researchers need to find where to post and subscribe for information. Moreover, these tools usually lack autonomous mechanisms to select documents based on individual interests. Even though Citeulike provides a "watchlist" option for users to keep track of new documents that are relevant to each page, the users need to manage the watchlists manually. Researchers usually have long-term interests and researchers who work in the same area are typically interested in the same type of documents. Their efforts in sharing and keeping track of new documents of interest can benefit from an autonomous collaborative environment. Shrack is a P2P framework introduced in 2007 to support such collaborative environment [13].

Although there are many proposed peer-to-peer networks based on common interest such as in [8, 9] and [14], Shrack is unique in that (1) its goal is to form a collaborative group of peers to keep track of future documents, not to search for a document on instant queries; and (2) pull communication is the only communication that is used in Shrack. As a result, (1) there is no explicit query in Shrack, hence each peer needs to learn about interests of its associated user; and (2) Shrack peers discover other peers and learn about their interests locally using information available in dissemination messages without explicitly exchanging their profiles.

In the previous work, Shrack architecture is presented along with a dissemination protocol in which peers disseminate every message they receive to other peers [13]. This paper improves upon the previous work with the following main contributions: (1) the improvement of the Shrack dissemination protocol in which peers disseminate messages based on their interests, section 4; (2) a peer profile learning algorithm whereby a peer learns the interests of its users and other peers in the network, section 5; (3) a method for a peer to discover other peers in the network, section 5.2; and (4) a neighbour selection algorithm based on common interests between peers, section 6.

2. RELATED WORK

[1] http://www.bibsonomy.org
[2] http://www.citeulike.org

CNIKM '09 Hongkong, China

Several P2P networks have been developed to improve data and document sharing over the Internet. Data-sharing networks based on small-world networks are proposed in [5] and [10]. The networks are comprised of several clusters, each is a community with overlapping interests. The systems support searching and locating documents within each cluster. In the first approach [5], peers periodically update global information of all peers and document locations within a cluster through a push-based "gossip" protocol [7]. Hence, each peer can immediately locate documents within its cluster. In the second approach [10], populated files are replicated among peers in the same cluster. Peers use limited flooding of requests to search for files within their clusters. Our system follows a similar concept as in [5]; whereby the clusters are formed dynamically and adaptively. Peers in Shrack form soft clusters as opposed to the well-defined rigid clusters of [5].

The filtering and dissemination system pFilter [12] uses a publish/subscribe model where peers register their profiles for persistent queries and documents to filter and disseminate new documents on a semantic overlay of a structured P2P network. A drawback of this system is that users need to explicitly submit the persistent queries.

PlanetP [3], improves distributed search in P2P communities by maintaining global addresses of peers in the communities and global inverted term-to-peer index. To search for documents, a peer forwards a request to peers containing documents with the requested term based on the inverted index and a ranking algorithm.

To improve search in P2P document-sharing networks, selective forwarding of requests to peers based on their semantic topology is proposed in [4]. Peers advertise their expertise to the network. Peers forward a request based on matching the subject of a query and the expertise according to their semantic similarity. Simulation experiments show that this approach outperforms random peer selection.

While there are many initiatives to build P2P systems for research collaborations, the existing systems focus on document searching such as [2, 4] and [16]. We focus on document tracking based on the common interest of the users.

3. OVERVIEW OF THE SHRACK FRAMEWORK

Shrack is an unstructured P2P network, in which peers who have common interests and who are willing to collaboratively establish pull connections to one another to share and keep track of new documents. To join the Shrack network, the associated user first configures the peer (the software) with an initial set of neighbours, with whom the user has real world collaboration or connections. After joining Shrack, peers learn about new neighbours from the Shrack messages. Each Shrack message contains document metadata for receiver peers to learn about the document such as the location (e.g. URI), the title, the keywords, and the abstract. Shrack messages are disseminated among peers through pull connections. Each Shrack message contains a message propagation history consisting of a list of peers that the message has travelled through from the *publisher peer* to the local peer (visited peers), and TTL (Time-To-Live indicating the number of hops after which the message will be discarded). We call the other peers of which the local peer is aware of as *known peers*. For any given peer p_i, its known peers include its neighbourhood peers as well as other peers that p_i has observed in the received message history. The neighbourhood peers are peers to which p_i get connected, i.e., p_i establishes connections to pull Shrack messages.

To publish a document, the publisher peer creates a Shrack message containing the document metadata, adds itself in to the list of visited peers of the message, and places the message into its *shared directory*. Periodically, each peer pulls new Shrack messages from its neighbours to keep track of new documents that become available in the network. Upon receiving Shrack messages, a peer makes Shrack messages containing metadata of documents that are relevant to its interests available in its shared directory for other peers to pull. With successive pulls, information about the new document is disseminated to all connected peers with common interest within the message's predefined TTL. After a peer learns about new relevant documents, the peer can automatically download the documents for the user, or the user can later retrieve the documents using information about their locations embedded in the Shrack messages. Further details of the Shrack framework can be found in [13].

4. PULL-ONLY INFORMATION DISSEMINATION PROTOCOL

Shrack peers use the pull-only information dissemination protocol to distribute information about new documents and learn about other peers in the network. The new Shrack dissemination protocol is described in Algorithm 1. We redefine the dissemination protocol, i.e., peers only share messages that are of interests in their shared directories for other peers to pull. As a result, each peer will filter out irrelevant messages for other peers that have common interest, creating a community filtering system. We also present how peer profile learning and peer neighbour selection processes are incorporated in the dissemination protocol.

We define a *pull interval* as a predefined interval, specifying how frequently a peer wishes to pull its neighbours to learn about new documents. In this work, we assume that peers pull Shrack messages from all of their neighbours using the same pull interval. In practice, pull intervals might be different. When a peer requests to pull document metadata from each neighbour, the peer will attach an *update time* indicating the oldest messages the peer wants to pull from the neighbour's shared directory. In general, the update time is the latest time the peer visits the neighbour. To control the oldest messages the peer wants to update, each peer has a predefined *maximum update time*. If the update time exceeds the maximum update time, the peer will set its update time to the maximum update time.

When receiving a Shrack message, the receiver peer checks if the message contains new document metadata by comparing the document metadata's identifier with the peer's history list. If the identifier of the document metadata is not in the history list, it will be detected as a new document metadata. If the new document metadata d is relevant to the receiver peer, d will be added to the peer's local archive and the peer will update its local peer profile according to d. Then, the receiver peer decreases the TTL of the message containing d by one. If the TTL of the message is greater than zero, the receiver peer will append its peer contact ID to the message and place the message in its shared directory for further dissemination. After that, the receiver peer updates known peer profiles of peers in the visited list (in

Algorithm 1 p_i.pull()

```
1: for each neighbour p_j of peer p_i do
2:     peer p_i pulls a set of Shrack messages, m, from p_j's
       shared directory that arrived at p_j after the update
       time.
3:     for each Shrack message m do
4:         d is a document metadata embedded in m
5:         if d is new to p_i then
6:             if d is relevant to the peer's interest then
7:                 keep d in the local archive
8:                 update local profile of p_i according to d
9:                 decrease the m.ttl by one
10:            if m.ttl is greater than 0 then
11:                append its identifier to m
12:                add m to the share directory of p_i.
13:    p_i.updateKnownPeerProfile(m)
14: p_i.updateNeighbours()
```

every messages). At the end of each pull cycle, the receiver peer evaluates its known peers and updates its neighbourhood. We describe how a peer updates its local profile and known peer profiles in section 5, and how a peer updates its neighbourhood in section 6. Note that for simplicity of our simulation, we assume that a peer gets an immediate feedback from its user to identify relevant documents based on their metadata. In practise, the peer waits for a feedback from the user.

5. PEER PROFILES

To enable a self-organizing P2P network, each peer learns about the interests of its associated user and also interests of other peers in the network. We call the interest of the peer's associated user a *local peer profile*, and the interest of other peers in the network as a *known peer profile*. The local peer profile contains a set of relevant document metadata that a peer receives and a set of document metadata that the peer publishes. The known peer profile contains a set of relevant document metadata that are relevant to each known peer according to the information that the peer receives.

Since the set of document metadata that are available in the network changes with time according to which documents are published at which time, we define sets of document metadata with time as follows.

DEFINITION 1. *$S^t(p_i)$ is the set of document metadata that peer p_i publishes up to time t.*

DEFINITION 2. *P^t is the set of document metadata that are published up to time t; that is, $P^t = \cup_{i=1}^{z} S^t(p_i)$, where z is the number of peers*

DEFINITION 3. *$R^t(p_i)$ is the set of document metadata that are published up to time t and are relevant to peer p_i; that is,*

$$R^t(p_i) = \{d \mid d \in P^t \text{ and } d \text{ is relevant to } p_i\} \quad (1)$$

DEFINITION 4. *$D^t(p_i)$ is the set of document metadata that peer p_i receives up to time t, excluding the self-published documents $S^t(p_i)$.*

$D^t(p_i)$ includes all document metadata that p_i receives up to time t, regardless whether the documents are relevant to p_i. Note that $R^t(p_i) \not\subset D^t(p_i)$ due to propagation delay or network connections. We assume that peers are always interested in the documents that they publish.

5.1 Local Peer Profile

A local peer profile of peer p_i is a set of document metadata that are relevant to p_i as identified by the user feedback. However, since each peer is only aware of document metadata that they publish or receive up to a given time, we define the local peer profile $L^t(p_i)$ of peer p_i at time t as

$$L^t(p_i) = S^t(p_i) \cup (R^t(p_i) \cap D^t(p_i)) \quad (2)$$

When a peer publishes or receives a new document metadata, d, that is relevant to its interest, it updates the local peer profile with the new document metadata, line 8 in Algorithm 1, as following.

$$L^{t+1}(p_i) = L^t(p_i) \cup \{d\} \quad (3)$$

5.2 Known Peer Profile

To discover the existence of other peers and learn about their interests, when a local peer receives a Shrack message, the local peer will obtain information about the visited peers to update its known peer profile. Since peers only share document metadata that is relevant to their interests, the list of visited peers represents a list of peers that are interested in the associated document metadata. A peer can receive Shrack messages that contain the metadata of a given document from different paths, which will be used to update the known peer profile.

DEFINITION 5. *$K^t(p_i, p_j)$ is a set of document metadata that a local peer p_i receives via peer p_j up to time t; that is, for each document metadata $d \in K^t(p_i, p_j)$, the peer p_j is in the list of visited peers associated with d.*

We use $K^t(p_i, p_j)$ as the profile of p_j in the known peer profile of p_i at time t. The known peer profile of p_i consists of a set of $K^t(p_i, p_j)$, where $p_j \in K^t$ and K^t is the set of all peers known to p_i at time t.

When a local peer pulls Shrack messages from its neighbour, all the messages that the local peer receives will be used to update its known peer profile. For each message, the local peer will use the embedded document metadata to update the known peer profile of peers in the visited list as shown in Algorithm 2. In addition to peers in the visited list

Algorithm 2 p_i.updateKnownPeerProfile(m)

```
1: d is a document metadata embedded in m
2: for each visited peer p_j in m do
3:     K^{t+1}(p_i, p_j) = K^t(p_i, p_j) ∪ {d}
```

of dissemination messages, each peer also learns about the existence of other peers when they request to pull from its shared directory. Each peer keeps a contact of a new peer who pulls messages from its shared directory as its known peer with an empty profile; because the peer does not yet have knowledge of relevant documents that are of the interest to the new peer, until it pulls from the shared directory of the new peer.

6. PEER NEIGHBOURHOOD SELECTION

With the assumption that peers usually have overlap of interests and each peer would like to keep track of new documents that are relevant to their interests, we hypothesize that a self-organizing P2P collaborative network in which

peers get connected, pull shared information, based on common interests will improve quality of the document metadata received by each peer over the randomly connected network. This section describes how peers get connected based on different neighbourhood selection strategies.

6.1 Common Interest Strategy

In the common interest strategy, a peer gets connected based on common interest with its known peers. We define a common interest score between a local peer p_i and a known peer p_j at time t, denoted $C^t(p_i, p_j)$. We quantify the common-interest score based on the modification of a *Jaccard index*[3] by measuring the similarity of a local peer profile, $L^t(p_i)$, and each known peer profile, $K^t(p_i, p_j)$.

Furthermore, we assume that a new peer p_j who requests to pull shared messages from a local peer p_i would tentatively have a common interest with the local peer, hence we set its common-interest score to 1, the maximum value. As a result, the local peer will select the new peer p_j as one of its neighbours the next time the local peer updates its neighbourhood to learn about the interests of p_j. After the local peer pulls information from p_j, the common-interest score between the local peer and the peer p_j will be computed regularly. In other word, this is how a peer p_j introduces itself to p_i, by initiating a pull request to p_i.

$$C^t(p_i, p_j) = \begin{cases} 1 \text{ if } p_j \text{ is new to } p_i \\ \dfrac{|L^t(p_i) \cap K^t(p_i, p_j)|}{|L^t(p_i) \cup K^t(p_i, p_j)|} \text{ otherwise} \end{cases} \quad (4)$$

$C^t(p_i, p_j)$ does not satisfy the commutative property. Since each peer creates its known peer profile using information that it receives locally, for a given document metadata d that is relevant to both p_i and p_j, there may exist a dissemination path of d from p_j to p_i but not from p_i to p_j.

At each time t, when a peer wants to update its neighbours, the peer will compute common-interest scores of its known peers. The top-N ranked known peers according to their common interest scores will be selected as a new set of the peer's neighbours. Ties in the scores are resolved by a random selection.

6.2 Random Strategy

Since the gossip protocol [7], a well-known scalable and reliable dissemination protocol for a large-scale network, disseminates information among peers based on random connections, a random strategy is selected as a baseline. In this strategy, a peer simply updates its neighbours by randomly selecting N known peers as its new neighbours without considering the common-interest score between the local peer and each known peer.

6.3 Hybrid Strategy

The hybrid strategy is chosen to reduce the effect of the greedy behaviour of the common interest strategy by allowing peers to randomly explore peers in the network. In this strategy, each peer selects its neighbours from its top-ranked known peers with probability $1 - \beta$, or randomly from its known peers with probability β, where β is an exploration parameter and $0 \leq \beta \leq 1$. The β parameter indicates how much peers want to explore the network.

[3]The Jaccard index can be expressed as an extension of the cosine similarity measure to binary attributes.

7. EXPERIMENTS

7.1 Authorship User Interest Model

To evaluate performance of the peer neighbourhood selection strategies, we create an artificial user associated with each peer in the simulated environment. The interests of peers are defined by the interests of the associated user. Each artificial user defines a set of documents that the associated peer publishes and a set of relevant documents of which the associated peer should keep track during the simulation. Users may have overlap of interests. We introduce an artificial user model called an *authorship user interest model*, whereby the users' interests are defined by the topics of their published documents.

We create the authorship user interest model from a collection of documents containing information of their authors. In our experiments, we use the ACM metadata collection. Each author in the collection is viewed as an artificial user associated with a peer in the simulation. The list of documents that each user has published is the set of documents that the associated peer publishes. The interest of each user is modelled based on the ACM Computing Classification System (CCS)[4] according to the real ACM metadata collection. Each document in the ACM collection has been assigned to CCS classes by the document authors. We assume that the interest of each user is identified by the work that he/she publishes. In our case, the interest of each user is described by the CCS classes of the documents that he/she has published. Subsequently, the set of documents of which each peer should keep track during the simulation is the set of documents that is relevant to the interest of the peer's associated user. For example, the peer whose associated user has published documents in class H.3.3 and H.2.1 would be interested in keeping track of all documents in class H.3.3 and H.2.1 that are available in the system. As a result, we create the overlap of interests of the users naturally based on the classes of documents they published.

7.2 Dataset Preparation

We create our artificial users from authors who published documents in class H.3.3, information search and retrieval, in the year 2008, in the ACM metadata collection. We use a set of documents in class H.3.3 that these authors published since the year 2000 as our document dataset. There are 7 subclasses in class H.3.3. These subclasses are used to define interests of the artificial users and document classes in our simulation. Each document can be labelled by multiple subclasses, which define the interests of the document's authors. We select the top 1,000 authors according to the number of documents they published as the set of the artificial users. From these authors, we have 1,639 documents in our simulation. Table 1 shows the number of users and documents in each subclass. Out of 1,000 users, 38% are interested in one subclass, 31% are interested in two subclasses, the rest are interested in three or more subclasses. The majority of documents, 72%, are labelled with one subclass.

7.3 Performance Evaluation Metrics

We compare the performance of the peer neighbourhood selection strategies based on the *quality of received document metadata* and *dissemination speed/distance of relevant docu-*

[4]http://www.acm.org/about/class/1998

Table 1: The number of users and documents in each subclass

Subclass Name	#Users	#Docs
Clustering	235	222
Information filtering	268	238
Query formulation	327	354
Relevance feedback	148	151
Retrieval models	579	629
Search process	437	493
Selection process	144	110

ment metadata to the local peer. Moreover, we analyze the properties of the result networks to observe whether they form a self-organizing social network.

In definition 1 to 3, we define sets of document metadata published from the start of the simulation until time t. We extend these definitions below to define sets of document metadata published during an arbitrary time slot $\tau_k = t_{k_r} - t_{k_s}$, where t_{k_s} and t_{k_r} are the start and the end of the time slot.

DEFINITION 6. *$S^{\tau_k}(p_i)$ is the set of document metadata published by p_i during time slot τ_k; i.e., $S^{\tau_k}(p_i) = S^{t_{k_r}}(p_i) \setminus S^{t_{k_s}}(p_i)$.*

DEFINITION 7. *P^{τ_k} is the set of document metadata published during time slot τ_k; i.e., $P^{\tau_k} = P^{t_{k_r}} \setminus P^{t_{k_s}}$.*

DEFINITION 8. *$R^{\tau_k}(p_i)$ is the set of document metadata relevant to peer p_i that are published during time slot τ_k; i.e., $R^{\tau_k}(p_i) = R^{t_{k_r}}(p_i) \setminus R^{t_{k_s}}(p_i)$.*

Each document metadata takes time to be disseminated. We divide the simulation times into time slots and observe the dissemination of documents that are published in each time slot. The dissemination performance of a set of document metadata P^{τ_k} published during time slot τ_k is measured when the dissemination of documents in P^{τ_k} ends, denoted $t_{k_{end}}$. We use a heuristic criterion to determine $t_{k_{end}}$, which is "a time when no more document metadata received that were published in P^{τ_k} for a sufficiently long time t_p".

7.3.1 Quality of received document metadata

The quality of document metadata published during time slot τ_k that peer p_i receives, $P^{\tau_k} \cap D^{t_{k_{end}}}(p_i)$, is measured in terms of precision, recall and F-score, defined as follows.

DEFINITION 9. *$Precision^{\tau_k}(p_i)$ is the fraction of documents published during time slot τ_k, excluding self-published documents, received by peer p_i that are relevant to p_i.*

$$Precision^{\tau_k}(p_i) = \frac{|R^{\tau_k}(p_i) \cap D^{t_{k_{end}}}(p_i)|}{|P^{\tau_k} \cap D^{t_{k_{end}}}(p_i)|} \quad (5)$$

DEFINITION 10. *$Recall^{\tau_k}(p_i)$ is the fraction of documents published during time slot τ_k relevant to peer p_i, excluding self-published documents, that are received by p_i.*

$$Recall^{\tau_k}(p_i) = \frac{|R^{\tau_k}(p_i) \cap D^{t_{k_{end}}}(p_i)|}{|R^{\tau_k}(p_i) \setminus S^{\tau_k}(p_i)|} \quad (6)$$

DEFINITION 11. *$Fscore^{\tau_k}(p_i)$ is the harmonic means of $Precision^{\tau_k}(p_i)$ and $Recall^{\tau_k}(p_i)$. That is,*

$$Fscore^{\tau_k}(p_i) = \frac{2 \cdot Precision^{\tau_k}(p_i) \cdot Recall^{\tau_k}(p_i)}{Precision^{\tau_k}(p_i) + Recall^{\tau_k}(p_i)} \quad (7)$$

We use terms precision, recall and F-score to refer to an average $Precision^{\tau_k}(p_i)$, an average $Recall^{\tau_k}(p_i)$ and an average $Fscore^{\tau_k}(p_i)$ over all peers in the network in each time slot, respectively.

7.3.2 Dissemination speed/distance of relevant document metadata

The dissemination speed/distance of relevant document metadata are measured in terms of pull delay and path length of relevant document metadata that a peer receives, which are defined as follows:

DEFINITION 12. *Given a peer p_i and a time slot τ_k, the **relevant pull delay**, $RelPullDelay^{\tau_k}(p_i)$, of documents that p_i receives, which are relevant to p_i and published during time slot τ_k, is defined as the average, over all such document metadata $d \in R^{\tau_k}(p_i) \cap D^{t_{k_{end}}}(p_i)$, of the time delay from when d is published until p_i first observes d.*

DEFINITION 13. *Given a peer p_i and a time slot τ_k, the **relevant path length**, $RelPathLength^{\tau_k}(p_i)$, of documents that p_i receives, which are relevant to p_i and published during time slot τ_k, is defined as the average hop count over all such document metadata $d \in R^{\tau_k}(p_i) \cap D^{t_{k_{end}}}(p_i)$, when p_i first observes d.*

We use the terms $RelPullDelay$ and $RelPathLength$ to refer to the average $RelPullDelay^{\tau_k}(p_i)$ and the average $RelPathLength^{\tau_k}(p_i)$ over all peers in each time slot, respectively. A peer is considered to experience better performance if it has lower relevant pull delay and lower relevant path length.

7.3.3 Self-organizing network property

We analyze the resulting network topologies to determine whether they form a social network by examining their clustering coefficients, characteristic path lengths, and degree distributions, which are defined as follow.

DEFINITION 14. Clustering Coefficient (CCO) *is the average of the clustering coefficient of p_i over all p_i. We use the clustering coefficient formula for directed graph,*

$$CCO(p_i) = \frac{|E(p_i)|}{|N(p_i)|(|N(p_i)| - 1)} \quad (8)$$

where $E(p_i)$ is the set of connections between neighbours of p_i, and $N(p_i)$ is the set of neighbours of p_i.

DEFINITION 15. Characteristic path length (CPL) *is the average shortest path between any two peers in the network. In the case of a disconnected network, CPL is the average shortest path between any two peers in the largest strongly connected component.*

DEFINITION 16. Degree distribution *is defined in terms of the in-degree distribution, where the in-degree of a peer p_i is a number of incoming pull connections that p_i receives. Since every peer has the same fixed number of neighbours, the out-degree distribution is not considered.*

Many studies [1, 11, 15] report that social networks usually have small world properties such as large CCO, small CPL, and power-law scaling in degree distributions. Watts and Strogatz [15] shows that small-world networks are highly clustered like regular lattices with a much higher CCO than random graphs of the same parameter, but have small CPL similar to random graphs.

7.4 Experimental Setup

We built a simulation of Shrack on PeerSim [6]. Each cycle has 1,000 simulation clock units. We assume that point-to-point network communication delay is negligible. Thus, the pull delay depends only on the pull intervals. We run the simulation on a network with 1,000 peers with the initial connections are formed randomly. We compare three peer neighbourhood selection strategies; (1) the common interest strategy, denoted *Jac*, (2) the random strategy, denoted *Ran*, and (3) the hybrid strategy with varying β values between 10^{-7} and 10^{-1}. For clarity of presentation, we report results for the hybrid strategy with $\beta \in \{10^{-1}, 10^{-2}, 10^{-3}\}$, denoted *Hybrid1, Hybrid2, Hybrid3*, respectively. Collectively, the hybrid strategies are denoted *Hybrid*. For $\beta < 10^{-3}$, the hybrid strategy behaves silimarly to Jac on all metrics. Each performance metric, except for the degree distribution, is averaged over 10 simulations with different random seeds. Documents are published in the system by the peer associated with the first author. The publishing time follows a Poisson distribution with an average publishing rate of 1 document per 4 cycles [5]. The pull interval is fixed at 20 cycles with a different (random) starting time for each peer. The experiments are conducted with a TTL value of 8 (selected experimentally) and different sizes of peer neighbourhood varying from 3 to 15. Each peer sets the maximum update time to 160 cycles—the product of the TTL (8) and the pull interval (20 cycles)

We evaluate the performance of the selection strategies as a function of time using a sliding window. The parameter t_p of the criterion determining dissemination end is set at 200 cycles. We use a window size of 400 cycles in duration, with 200 cycles overlap between successive windows, i.e., $t_{k_s} = \{0, 200, 400, ..., 5400\}$, where $k = \{1, 2, 3, ..., 29\}$[6]. The network property is measured every 200 cycles.

7.5 Experimental Results

We average each evaluation metric, except the degree distribution, over the last 10 time slots, τ_{20} through τ_{29}, to measure the performance and network property of each peer neighbourhood selection strategy in each configuration.

7.5.1 *Quality of received document metadata*

Figures 1, 2, and 3 show the performance in terms of precision, recall, and F-score. As the size of peer neighbourhood increases, the recall increases but the precision decreases, except for RAN where they remain unchanged. In all configurations, Jac gives higher precision and lower recall than Ran. As the size of peer neighbourhood increases, initially, Jac and Hybrid gain more recall than the loss of precision, as a result, their F-score increase. When their recall approaches 1, the increase in Recall is not sufficiently larger than the loss of precision, resulting in the decline of F-score. Hybrid shows the mixture effects of Jac and Ran; as β increases, the precision decrease and the recall increase. Jac and Hybrid with $\beta \leq 10^{-2}$ show significant improvement over Ran in terms of F-score. Hybrid outperforms Jac in terms of F-score, when the size of peer neighbourhood is less than or equal to five. When the size of peer neighbourhood is greater than five, Hybrid does not provide significantly improvement over Jac. At their best, when the size of peer neighbourhood equals eight, Hybrid3 and Jac give a 27% improvement in F-score over Ran.

[5]the document publication time in the simulation is independent of the time of the actual ACM publication.

[6]We ignore the last 4 time slots, τ_{30} through τ_{33}, because when the simulation ends, the dissemination of documents published in these time slots are not end.

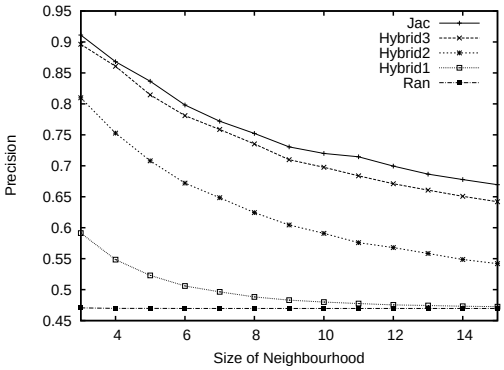

Figure 1: Precision vs. the size of peer neighbourhood

7.5.2 *Dissemination speed/distance of relevant document metadata*

Figures 4 and 5 show that Ran has the highest RelPullDelay and the lowest RelPathLength. In all strategies, RelPullDelay exhibit a negative correlation with the size of peer neighbourhood and Jac always gives the lowest RelPullDelay. Similarly, RelPathLength negatively correlates with the size of peer neighbourhood, except for Ran where it has no effect.

7.5.3 *Self-organizing network property*

Figure 6 shows the clustering coefficient CCO of the networks in each selection strategy. Overall, Jac and Hybrid3 give the best CCO. The CCO of Hybrid decreases as β increases. Ran has the lowest CCO. The Characteristic Path Length CPL of the networks in each selection strategy are shown in Figure 7. Jac and Hybrid3 have the highest CPL followed by Hybrid2, Hybrid1, and Ran. The CPL of all the networks decrease as the size of peer neighbourhood increases.

Figure 8 shows the in-degree complementary cumulative distribution function (CCDF) of the networks under different selection strategies on a log-log scale. We observe that the in-degree distribution of the Jac and Hybrid3 networks follows the power law distribution fairly closely for in-degree

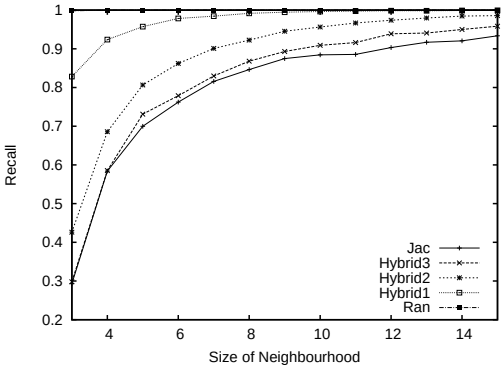

Figure 2: Recall vs. the size of peer neighbourhood

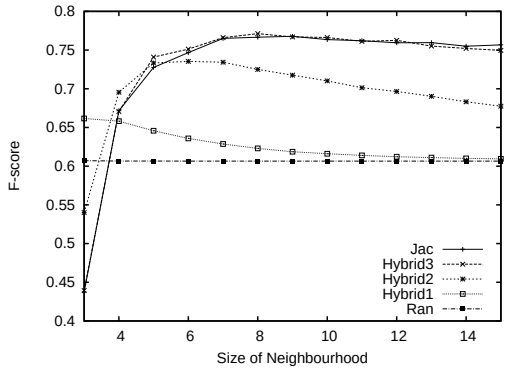

Figure 3: F-score vs. the size of peer neighbourhood

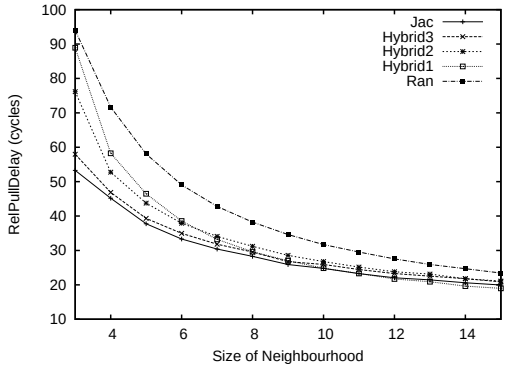

Figure 4: RelPullDelay vs. the size of peer neighbourhood

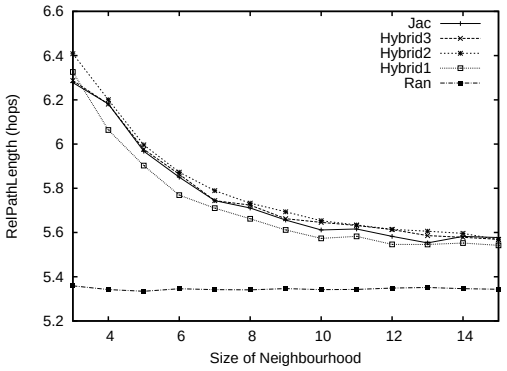

Figure 5: RelPathLength vs. the size of peer neighbourhood

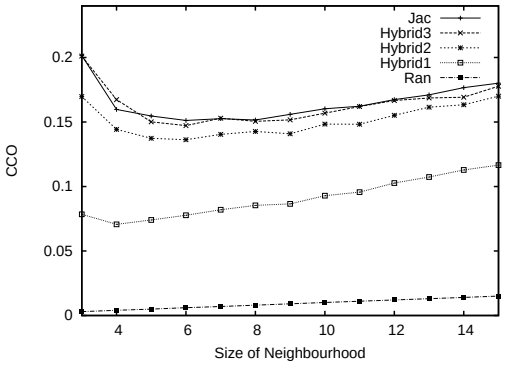

Figure 6: CCO of the network vs. the size of peer neighbourhood

greater than 3, with $\alpha = 2.1$. Conversely, the Ran network does not follow the power law distribution as closely; even for in-degree values greater than 9, its in-degree distribution fits a power law function with $\alpha = 9.6$ which is outside the typical range of $2 > \alpha > 3$ for social networks.

8. DISCUSSION

The results show that peers in Jac and Hybrid can automatically form a self-organizing network of peers with common interest. By sharing and disseminating only messages that are of interest, peers automatically filter out messages that are not relevant to their group, creating a community filtering system.

The relatively low recall of Jac can be attributed to peers not receiving relevant messages from some peers that have low common interests. Another possible reason is the group of peers having common interests becoming disconnected from the network, either by a small number of TTL or the size of peer neighbourhood.

Jac peers with multiple interests may have difficulty in keeping track of documents in an interest group with fewer documents (imbalanced document sets), because the common interest score favours peers with more documents. A possible solution is for the user to use multiple peers, one for each interest. Thus, peers responsible for sparse interests will perform a random walk until they find a relevant peer.

Ran guarantees that peers receive all documents published (recall of 1.0), similar to the gossip protocol. However, since not all the documents are relevant to all peers, the average precision of Ran is low.

The speed/distance performance shows that Jac disseminates relevant messages faster than Ran, because Jac peers usually pull messages from peers in the same group. Hybrid show the mixture effects between Jac and Ran with less sensitivity to β than the quality performance metrics.

The analysis the network properties gives evidence that the Jac and Hybrid3 networks have different characteristics than the Ran network. Jac and Hybrid3 networks are self-organizing into a topology that follows the characteristics of social networks, namely the small-world property [1, 11, 15]. Particularly, the Jac and Hybrid3 networks have significantly larger CCO than Ran, small CPL similar to Ran, and power-law scaling in degree distribution.

We anticipated the hybrid strategy would reduce the effect of the greedy behaviour in Jac, increasing the recall and F-score. Experimental results show, however, that for size of peer neighbourhood greater than 5, Hybrid does not provide a significant improvement in the F-score over Jac. This shows that increasing the neighbourhood diversity by increasing the size of peer neighbourhood results in better overall performance than by creating some random connections.

9. CONCLUSION

We propose a new pull-based dissemination strategy and protocol for the Shrack document sharing and tracking system modelled after social networks. Through simulated experiments using real world document metadata form the ACM digital library, we explore the information dissemination performance of our proposed peer neighbourhood selection strategies and compare them to random networks. We

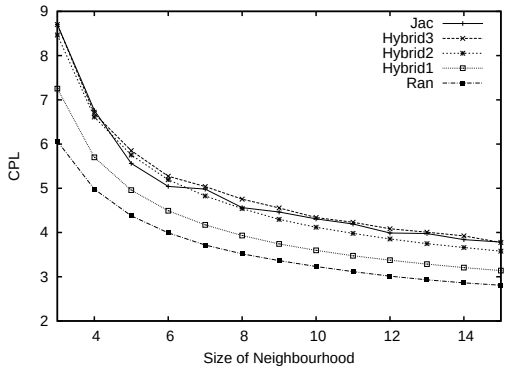

Figure 7: CPL of the network vs. the size of peer neighbourhood

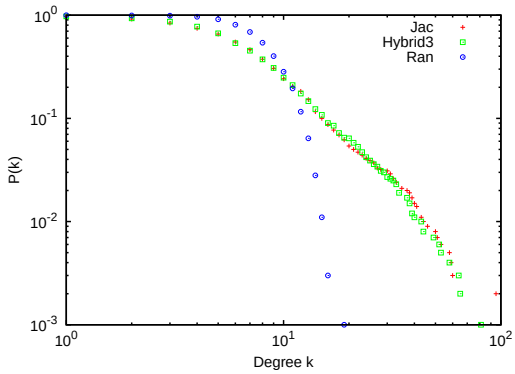

Figure 8: In-degree complementary cumulative distribution function of the network

show that our network self-organizes into a small-world network, giving noticeable improvement in the dissemination performance in terms of F-score up to 27% over a random network. Our main contributions are (1) a pull-based information dissemination strategy in which peers disseminate messages based on their interests; (2) a method for a peer to discover other peers in the network; (3) a peer profile learning algorithm whereby a peer learns the interests of other peers in the network; and (4) an autonomous neighbour selection algorithm based on common interests between peers. The experimental results show that the best F-score is achieved with Jac by increaseing the size of peer neighbourhood. In future work, we plan to analyze the system with unlimited TTL, evaluate the performance of the network in the presence of sparse interest groups, devise an algorithm for adaptively determining the optimal neighbourhood size for each peer, and incorporate term-based similarity.

Acknowledgement

We are grateful for the financial support of the Natural Sciences and Engineering Research Council of Canada, MITACS, and GenieKnows.com. We also would like to thank ACM for supplying its bibliographic metadata.

10. REFERENCES

[1] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW'07: Proc. of the 16th Intl. Conf. on World Wide Web*, pages 835–844, New York, NY, 2007.

[2] P. A. Chirita, A. Damian, W. Nejdl, and W. Siberski. Search strategies for scientific collaboration networks. In *P2PIR '05: Proc. of the ACM Workshop on IR in P2P networks*, pages 33–40, New York, NY, 2005.

[3] F. M. Cuenca-Acuna, C. Peery, R. P. Martin, and T. D. Nguyen. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. In *12th IEEE Intl. Symp. on High Performance Distributed Computing*, June 2003.

[4] P. Haase, J. Broekstra, M. Ehrig, M. Menken, P. Mika, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich. Bibster — A Semantics-Based Bibliographic Peer-to-Peer System. In *Proc. of the 3rd Intl. Semantic Web Conference (ISWC 2004)*, pages 122–136, Hiroshima, Japan, 2004.

[5] A. Iamnitchi, M. Ripeanu, and I. Foster. Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations. In *1st Intl. Workshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.

[6] M. Jelasity, A.Montresor, G. P. Jesi, and S. Voulgaris. The Peersim simulator. `http://peersim.sf.net`, Accessed May 2008.

[7] A.-M. Kermarrec, L. Massoulie, and A. J. Ganesh. Probabilistic Reliable Dissemination in Large-Scale Systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(3):248–258, Mar. 2003.

[8] M. Li, W.-C. Lee, and A. Sivasubramaniam. Semantic small world: An overlay network for peer-to-peer search. In *ICNP*, pages 228–238, 2004.

[9] A. Löser, C. Tempich, B. Quilitz, S. Staab, W. T. Balke, and W. Nejdl. Searching dynamic communities with personal indexes. In *Proc. 4th Intl. Semantic Web Conf., ISWC 2005*, volume 3729 of *LNCS*, pages 491 – 505, Galway, Ireland, NOV 2005.

[10] J. Mitre and L. Navarro-Moldes. P2P Architecture for Scientific Collaboration. In *the 13th IEEE Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*, 2004.

[11] M. E. J. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Physical Review E*, Volume 64, June 2001.

[12] C. Tang and Z. Xu. pfilter: Global information filtering and dissemination using structured overlay. In *FTDCS*, 2003.

[13] H. Tanta-ngai, V. Keselj, and E. E. Milios. Shrack: Description and performance evaluation of a P2P system for document sharing and tracking using pull-only information dissemination. (in HotP2P) *Proc. of IPDPS*, page 471, 2007.

[14] C. Tempich, A. Löser, and J. Heizmann. Community Based Ranking in Peer-to-Peer Networks. In *OTM Confederated Intl. Conf.'s CoopIS, DOA, and ODBASE*, pages 1261–1278, Agia Napa, Cyprus, 2005.

[15] D. J. Watts. Networks, Dynamics, and the Small-World Phenomenon. *American Journal of Sociology*, 105(2):493–527, Sept. 1999.

[16] L.-S. Wu and F. Menczer. Diverse peer selection in collaborative web search. In *SAC '09: Proc. of the 2009 ACM Symp. on Applied Computing*, pages 1709–1713, New York, NY, 2009.