On Using Peer Profiles to Create Self-Organizing P2P Networks

Elizabeth Daly, Alan Gray and Mads Haahr
Department of Computer Science, Trinity College Dublin
{eldaly, agray, haahrm}@cs.tcd.ie

Abstract

Searching and organization of peers are fundamental challenges in P2P networks. Unstructured networks, such as Gnutella, inefficiently use broadcast searches and random neighbors. Structured networks are similarly inefficient, as they generally rely on globally unique identifiers (GUIDs) which are assigned irrespective of content, which prevents fuzzy semantic searches. In both types of network search, neighbors establish trust between themselves, regardless of whether or not their content is likely to satisfy searches. We present the idea of using context-based profiles to describe peers. This enables self-organizing clusters of similar peers. A profile represents a peer's expertise based on content and responsiveness. By refining the search process using these profiles, more efficient directed searches are possible. Moreover, expertise provides a basis for trust establishment.

1. Introduction

In ubiquitous computing environments, there is an obvious need to be aware of what entities are deemed to be reputable and worthy of interaction. With the possible absence of a centralized certification authority, entities must use proof of ownership or other techniques to perform entity recognition [1]. Once the identity has been established, a trust engine must determine if the level of trustworthiness in the entity, when weighed against the risk inherent in the interaction, makes the cost/benefit ratio acceptable to the user that the application supports.

There has been much interest and research into these issues of trust/reputation management and security in ubiquitous and peer-to-peer (P2P) environments. However, the majority of these techniques subsume that the interaction is requested and should be permitted or denied purely based on the requesting entity's trustworthiness. Of a more fundamental nature is the question of why an entity should request a given interaction in the first place.

We contend that the likelihood of an entity's ability to satisfy a request for an action does not purely depend on its trustworthiness, but also on its expertise, or ability to perform the requested function. For example, it is preferable to ask a reasonably reputed biochemist a question on biochemistry than the most world-renowned computer scientist. Expertise in the context/area of a request should also be factored into the decision of whom to forward the request.

We propose to augment unstructured P2P overlay networks to include a profile of each peer. A peer can use such profiles to cluster with similar and trustworthy peers, or those offering services it seeks. Because these profiles are known before any interaction takes place, we can achieve a degree of self-organization in the network.

Clustering similar peers provides an organized, logical base, upon which webs of trust can be built. Additionally, these profiles can be used to improve the performance of searches in the network, as a peer can forward a query to those neighbors whose profiles show that they have expertise in the area. In this paper, we limit the discussion of P2P network to document-sharing networks, where a peer's profile describes the collection of documents it is sharing and its performance. However, the clustering concept presented can be used in any environment where a profile of a peer can be generated, for example, describing services it offers/seeks or keywords of hobbies/interests in friend-of-a-friend networks.

This document is organized as follows: Section 2 introduces structured and unstructured P2P networks. Unstructured P2P networks in related work are discussed in Section 3. Our peer descriptors are explained in Section 4, while Section 5 discusses inter-peer trust in our network. A mechanism for self-organization using peer descriptors and inter-peer trust is presented in Section 6. Finally, Section 7 discusses future work and concludes.

Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)
0-7695-2342-0/05 $20.00 © 2005 IEEE

2. Structured and Unstructured P2P Networks

This section discusses why unstructured topologies are better suited to self-organization than structured ones. In the P2P domain, there are two general categories of topology – structured and unstructured [2]. Structured networks, such as Chord and CAN [3, 4], are often based on distributed hash tables (DHTs). These have fixed topologies, where a peer's position, and hence neighbors, in the network are determined by a randomly generated GUID. Each peer manages a subset of the key space that corresponds to its place in the network, with each resource mapped to a specific position on the key space. Queries for known resources are routed efficiently and accurately, within a bounded number of hops, to the peer responsible. The key space is unrelated to content, meaning similar documents are randomly distributed throughout the network. As a result, fuzzy semantic searches are inefficient and often impossible in these structured P2P topologies.

By comparison, unstructured P2P networks do not have fixed topology. Due to the extra flexibility afforded to them, they can be designed to be more adaptive and resilient than the statistical models [5] and are more suited to highly dynamic P2P networks. Routing in unstructured P2P networks does not follow a statistically determinate protocol, because of the inherent unpredictability of the graph of connections between peers. When a query comes into a peer, there is no fixed neighbor to route it to, so the peer must decide if it can satisfy the query, or which neighbor can best satisfy the query and forward it to them. Therefore, we say that unstructured P2P networks use Decision Based Routing (DBR). DBR allows for great heterogeneity between peers [6]. This is because peers can use whatever decision-making criteria they require/choose for whom to forward a query to, if at all. This permits heterogeneity of trust and security models, computational resources and also contextualization of the decision and reasoning domains, which is vital in ubiquitous environments. As a result, we restrict our related work discussion to unstructured networks.

3. Related Work

Unstructured networks use differing organizing principles. Gnutella (*http://www.gnutella.com*) organizes peers purely by connections to existing peers in the network. Peers join by querying well known "bootstrap" peers to discover peers currently on the network. Once the peer has connections to other peers, they are used to flood searches and queries with a given time to live (TTL) in hops. If the sought content is beyond a peer's view horizon, it is unreachable to that peer. The random connectivity, view horizon and inefficient search in Gnutella-like P2P networks make them inherently unscalable [2].

In Freenet [7], peers are randomly connected to other peers in the network. Freenet was not designed for efficiency, but for anonymity and to prevent censorship. A hash is computed on each file on Freenet, and peers build up expertise in a particular key space over time. To retrieve a file, a user must first know the hash of the file contents, meaning that searches are not possible. FASD [8] provides a search mechanism for Freenet and runs above the Freenet protocol. Before a document is inserted into Freenet, a metadata key consisting of a vector of term weights and a pointer to the document it represents is generated and inserted into Freenet; and users can then search these keys to obtain the document pointers satisfying their query. However, it has been noted that session lifetimes in Freenet need to be of the order of weeks and months before this expertise emerges, whereas session lifetimes in typical ubiquitous scenarios are of the order of minutes or hours. In a highly dynamic P2P or ubiquitous environment, this amount of longevity is very improbable.

Kazaa (*http://www.kazaa.com*) and Grokster (*http://www.grokster.com*) use super-peer based topologies, in which peers with a lot of resources can optionally be promoted to super-peers, depending on the users' inclinations towards acting as super-peers. Each super-peer acts as a "directory-service" for other peers, whereby peers with fewer resources can register with a super-peer. Search is done in the normal Gnutella blind forwarding between super-peers. Each hop in a search therefore encompasses a greater number of peers. However, searching is still blind flooding within a TTL horizon, with the scaling problems this brings still present.

Semantic Overlay Networks (SONs) [9] use static profiling to organize P2P networks to improve search in P2P data-sharing networks by clustering peers with semantically similar content. Each peer classifies its content according to some globally defined classification hierarchy. SONs corresponding to each class are generated. Queries are restricted to the SON they are best suited to. SONs have been shown to greatly reduce the query-processing overhead. Each piece of data must be assigned manually to a globally predefined classification hierarchy maintained by some authority; which breaks the model of a truly decentralized network. These hierarchies do not allow any heterogeneity across contexts or peers.

The multi-agent system in [10] organizes a network to optimize distributed search in a P2P Information

Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)
0-7695-2342-0/05 $20.00 © 2005 IEEE

Retrieval (IR) system. In this approach, peers are described by a collection descriptor, which is a language model built to describe the documents a peer is sharing. The network is organized by peers maintaining a constant number of connections to their most similar neighbors. During search, queries are compared to a peer's collection descriptor. If the similarity is above a certain threshold, the peer returns results and forwards the query to its neighbors who are most like the query and also its most highly connected neighbors. However, this approach does not allow for the addition of a trust metric, and blindly relies on peers providing accurate descriptions of their own, and others', content.

The Adaptive P2P Topologies (APT) [11] protocol uses direct outcomes count to organize the network. Each peer maintains a history of the peers it has interacted with and the result (i.e. positive or negative) outcome of each interaction. "Peers connect to those peers that have high scores and, disconnect from peers with low scores" [11]. This enables the network to self-organize into clusters of peers that are mutually benevolent. This is, in essence, organization by trust values based on direct outcomes count, which inherently takes time. When a peer enters the network, there are no past interactions upon which to determine node location. This means that the node is potentially in a sub-optimal location in the network until several queries have been satisfied. The APT protocol organizes peers into clusters of mutually-trusting peers. It cannot accommodate peers acting in different roles or contexts. Search is done through blind query forwarding limited by a TTL value, as in Gnutella.

4. Peer Descriptors

Here we present peer descriptors as a mechanism for semantically representing the expertise of a peer. A peer descriptor is a weighted vector of keywords describing a peer in an application specific context. For example, the descriptor could describe the category of documents a given peer shares, services a peer offers/seeks or keywords of hobbies/interests. The peer descriptor vector $\vec{D}$ representing peer p is given in (1), where $W_p(t_i)$ represents the weight W held by peer p in term t_i, and n is the number of terms.

$$\vec{D} = \{W_p(t_1), W_p(t_2), W_p(t_3) \ldots W_p(t_n)\} \qquad (1)$$

As this paper concentrates on document-sharing networks, a semantic representation of the documents shared by a peer is required. Each document can be analyzed individually and then the collection of documents can be used to represent the peer by using an aggregate of the collection.

The documents are first parsed using Porter stemming [12] and the removal of stop words [13], thus reducing the number of terms and removing common terms that do not provide information about the semantic content of the document.

Once the documents have been simplified, the importance of terms in the documents can be determined. The Term Frequency (TF) metric counts the number of occurrences of a word in the document. In order to account for different document lengths, the value is multiplied by a constant of $1/|d|$, where $|d|$ is the length of the document.

An improvement on the TF method combines term frequencies with an "absolute" measure of a term's importance called the Inverse Document Frequency (IDF) value [14]. The IDF value of a term decreases as the number of documents that contain the term increases. Thus rare words receive a higher IDF weight and frequently used common terms receive a lower IDF weight. Therefore a high TF-IDF value is reached either by a high frequency term in a specific document, or a low frequency term in the overall document collection.

A text document can be represented by high-dimensional vector space where terms are associated with vector components. For example, a document d can be represented as a sequence of terms, d = {t1, t2, t3, …, t|d|} where |d| is the length of the document.

Once the term vector for each of the documents has been generated, the peer descriptor is created by combining the document vectors and selecting the highest weighted terms. This method is aimed at parsing documents with purely textual content. Therefore, the parsing methodology can be varied to exploit differing types of document. For example, HTML tags can be used to discover key information about a document. Any application specific parsing scheme may be used, such that it results in a weighted term vector representing the peer.

5. Inter-Peer Trust

In this section, we introduce trust between peers. Each peer's view of the network is subjective; therefore the network perspective of the trust model is local [15]. In other words, peers have their own opinions on the trustworthiness of others. In this paper, we consider trust to be defined as interpersonal trust based on past interactions between peers [16]; meaning the probability of an entity being a good source for a given domain depends on the performance for past interactions in that domain.

Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)
0-7695-2342-0/05 $20.00 © 2005 IEEE

In this paper we use the SECURE trust value [17]. However, any trust model that is a local centralized scalar trust metric (as defined in [15]) can be substituted at any of the nodes, as DBR allows for heterogeneity of trust models. The SECURE trust value represents an event as a (*s*,*i*,*c*)-triple, where *s* is the number of events that support, *i* is the number of events that have no information or are inconclusive about the outcome and *c* is the number of events that contradict the expected outcome. A trust value is maintained for each term in the peer descriptor based on the results of searches. The trust vector $\vec{T}$ is comprised of a trust value for each term of the descriptor vector. The trust value $T_p(t_k)$, given by (2), represents the trust *T* of peer *p* in term t_k, where $s_p(t_k)$, $i_p(t_k)$, $c_p(t_k)$ represent the number of requests peer *p* successfully, inconclusively and unsuccessfully satisfied containing term t_k respectively.

$$T_p(t_k) = \frac{s_p(t_k)}{s_p(t_k) + i_p(t_k) + c_p(t_k)} \tag{2}$$

6. Self-Organizing by Profile

Here we introduce peer profiles which are a combination of a peer descriptor and the trust in each term of the peer descriptor. The trust value and the peer descriptor value for each term are mapped onto 2-dimensional space as shown in Figure 1.

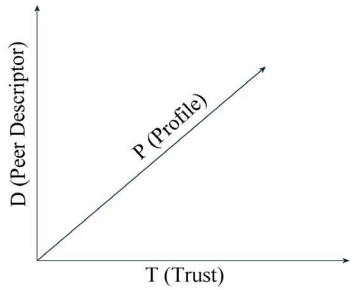

Figure 1. A profile (P) mapped in 2-dimentional space of peer description (D) and trust (T).

The profile value for each term t_k is given by the length of the line:

$$P_p(t_k) = \sqrt{\left(W_p(t_k)\right)^2 + \left(T_p(t_k)\right)^2}\,, \tag{3}$$

where $P_p(t_k)$ represents the profile value of peer *p* in term t_k. $W_p(t_k)$ describes the weight of term t_k for peer *p* and $T_p(t_k)$ represents the trust in peer *p* for term t_k. The result is that as a peer *p* starts to successfully satisfy searches for a given term t_k, the profile value $P_p(t_k)$ for that term increases, however if the peer performs badly for a given term t_k, *p*'s profile value $P_p(t_k)$ will decrease.

The network is organized by clustering similar nodes together. The peer profile can be represented in vector-space where each component is the profile value for a given term:

$$\vec{P} = \left\{P_p(t_1), P_p(t_2), P_p(t_3) \ldots P_p(t_n)\right\}, \tag{4}$$

where $P_p t_k$ represents the profile value of peer *p* in term t_k.

The cosine coefficient is used to determine peer profile similarity, this metric was chosen because queries can also be represented as a vector, and can then be compared to profiles using the same means. The cosine coefficient (5), is the measure of the angle formed by the vector-space representation of the two peers, where P_i and P_j are profiles of peers *i* and *j*, and $P_i(t_k)$ and $P_j(t_k)$ describe the profile value of the term t_k in the respective vectors of peers *i* and *j*, and *n* is the number of terms.

$$\cos(\vec{P_i}, \vec{P_j}) = \frac{\sum_{k=1}^{n} (P_i(t_k) \cdot P_j(t_k))}{\sqrt{\sum_{k=1}^{n} (P_i(t_k))^2} \cdot \sqrt{\sum_{k=1}^{n} (P_j(t_k))^2}} \tag{5}$$

The self-organizing clustering is initiated during the peer join request. In order to join the network, the peer must be able to contact at least one live peer on the network. The peer's descriptor is sent along with a join request to the live peer, and the descriptor is then broadcast for *n* hops. Each peer replies with its own descriptor. The joining peer then compares its descriptor with each descriptor received using (5) to compare the vector representation and determine the most similar descriptor. The responding peers are ranked according to similarity with the joining peer descriptor. The *k*-nearest peers and a number of random peers now form the routing table. The random peers prevent the network from becoming disjointed. Over time as searches are seen and new nodes discovered, the routing table is updated with peers that are the most similar in profile, thus clustering similar nodes enabling fuzzy semantic searches. The results of searches are used to maintain the trust vector for each peer. As a peer starts to successfully satisfy queries, the peer's rank for that term increases, however if the peer performs badly for a given term the peer's rank for a given term will decrease. Since the list of neighbors depends on the peer ranking, peers that consistently fail to satisfy queries will be moved to the outskirts of the network.

Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)
0-7695-2342-0/05 $20.00 © 2005 **IEEE**

7. Discussion and Future Work

We have presented a mechanism for self-organizing P2P networks based on content and trust. The aim is to build up trust only in peers that will be useful in satisfying peer searches by using peer content as a starting point for trust when no history is available.

In [10], peers are organized based purely on a collection description with no regard for peer performance or trust. As a result, a peer could manipulate the network through falsifying a collection description, allowing free-riding. In our network, a peer may initially lie about a descriptor, but as the peer continues to unsuccessfully satisfy queries, the trust metric will be reduced. Thus, reducing the value of the peer profile and causing peers to drop connections to the malicious peer. In [11] peers randomly enter the network and establish trusted connections over time. There is no notion of building up expertise in a semantically representable manner and so searches must resort to Gnutella-like flooding. In our network, a peer enters the network and clusters with similar nodes and attempts to build up trust with these peers, moving towards more trusted, more similar peers over time. Peers can be compared not just on trustworthiness but also on the expertise in a given area. This allows for the possibility of directed searches, which are known to be more efficient than flooding.

Future work will apply peer profiles to a P2P network, to enable directed semantic search. We intend to simulate and evaluate a document-sharing P2P network. We also plan to show that the peer profile framework can be used in a variety of P2P applications such as distributed work, distributed storage, publish/subscribe networks and messaging.

8. References

[1] J.-M. Seigneur, S. Farrell, C. D. Jensen, E. Gray, and Y. Chen, "End-to-end Trust Starts with Recognition", *In Proc. of the First International Conference on Security in Pervasive Computing*, Germany, 2003.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-like P2P Systems Scalable", *In Proc. of the ACM SIGCOMM '03 Conference*, August 2003.

[3] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications", *In Proc. of ACM SIGCOMM'01*, CA, 2001, pp.149-160.

[4] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S., "A Scalable Content-Addressable Network", *In Proc. of ACM SIGCOMM'01*, CA, 2001, pp.161-172.

[5] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of Scale-Free Networks: Error and Attack Tolerance", *In Physica A n. 320*, Elsevier, 2003,

[6] Gray, A., and Haahr, M., "Personalised, Collaborative Spam Filtering", *In Proc. of the First Conference on Email and Anti-Spam (CEAS 2004)*, CA, 2004.

[7] I. Clarke "A distributed decentralized information storage and retrieval system", *Unpublished report, Division of Informatics, University of Edinburgh*, UK, 1999.

[8] A. Kronfol, "FASD: A Fault-tolerant, Adaptive, Scalable, Distributed Search Engine", *Princeton University Technical Report*, USA, 2002.

[9] A. Crespo, and H. Garcia-Molina, "Semantic Overlay Networks for P2P Systems", *Technical Report, Computer Science Department, Stanford University*, USA, 2002.

[10] H. Zhang, W.B. Croft, B. Levine, and V. Lesser, "A Multi-agent Approach for Peer-to-Peer-based Information Retrieval Systems", *In Proc. of the Third International Joint Conference on Autonomous Agents and MultiAgent Systems*, NYC, 2004, pp. 456 – 464.

[11] T. Condie, S.D. Kamvar and H. Garcia-Molina, "Adaptive Peer-To-Peer Topologies", *In Proc. of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, Switzerland, 2004, pp. 53-62.

[12] M.E. Porter, "An algorithm for suffix stripping", *Program*, 14(3), 1980, pp. 130-137.

[13] C. Fox, "Lexical analysis and stoplists", *In Information Retrieval: Data Structures and Algorithms (ed. W. B. Frakes and R. Baeza-Yates)*, NJ: Prentice Hall, NJ, 1992, ch. 7.

[14] G. Salton and M.J. McGill, "*Introduction to Modern Information Retrieval*", McGraw-Hill Inc., NY, 1986.

[15] C.-N. Ziegler and G. Lausen, "Spreading Activation Models for Trust Propagation", *In Proc. of the International Conference on e-Technology, e- Commerce, and e-Service*, IEEE, 2004.

[16] D.H. McKnight and N.L. Chervany, "What is trust? A Conceptual Analysis and an Interdisciplinary Model", *In Proc. of the AMCIS 2000*, CA, 2000.

[17] N. Mogens, M. Carbone, and K. Krukow, "An Operational Model of Trust", *SECURE Deliverable 1.2,2004*, http://secure.dsg.cs.tcd.ie, 2004.

Proceedings of the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)
0-7695-2342-0/05 $20.00 © 2005 **IEEE**