Hybrid P2P Architecture for Transaction Management

Steven Marrocco[1] and Rachid Anane[2]
[1]School of Computer Science, University of Birmingham, UK
s.marrocco@cs.bham.ac.uk
[2]Faculty of Engineering and Computing, Coventry University, UK
r.anane@coventry.ac.uk

Abstract

In contrast with many centralised schemes P2P systems are flexible, scalable and highly dynamic. They offer an attractive distributed platform despite concerns over security. This work is motivated by the need to address explicitly the main issues that arise in the deployment of P2P systems in a business environment. A systematic approach is proposed in the development of a secure and trusted system to support authentication, non-repudiation and trust in business transactions. This involves two stages. Firstly, the identification of the functional components designed to facilitate security through authentication and non-repudiation, and those aimed at insuring trust; secondly, their implementation and integration into a hybrid P2P architecture, where the entry point server plays a central role. This integration is facilitated by the layering of functional components. Secure and trusted file transactions are further enhanced by a community management layer.

1. Introduction

The popularity of P2P systems is due largely to their scalability, their adaptation and the absence of a single point of failure [1]. They are decentralized, and owe their resilience to the symmetric and autonomous role that each peer is expected to play. The lack of a central server in P2P systems enhances their flexibility but can be a source of vulnerability.

Despite their use in a range of applications P2P systems are mainly associated with file-sharing applications such as in Napster [2] and BitTorrent [3]. Although the suitability of P2P systems for business has been highlighted [4, 5], concerns over security and reliability have hindered their wider adoption in e-business. Authentication, non-repudiation and trust have presented a significant challenge in e-business; they are particularly difficult to implement in a pure decentralised P2P system. Authentication is used to determine the identity of an agent, whereas contractual obligations between agents are enforced by non-repudiation; both are usually part of a security mechanism and are designed to create a relatively safe environment. Trust, on the other hand, is viewed as a concept with many facets, and consists of three factors: ability, benevolence and integrity [6, 7]. Ability refers to the competence of an agent in meeting requests and providing services. It is usually assessed by the quality of the service or the information provided, often in terms of accuracy and reliability. Benevolence is the expectation that an agent is well-disposed and has the best intentions towards other agents. Finally, integrity is the expectation that an agent would behave according to established ethical norms. Studies have confirmed that that trust in business tends to encourage greater participation by users and to foster long-term relationships [8].

This work is motivated by the need to address explicitly some of issues that arise in the deployment of P2P systems in a business environment. A systematic approach is proposed in the development of a secure and trusted system. Public encryption and social control mechanisms are combined in order to support authentication, non-repudiation and trust. Relevant functional components are identified, implemented and integrated into a layered architecture. Secure and trusted transactions are further enhanced by a community management component.

The remainder of the paper is organised as follows. Section 2 presents the technological context. Section 3 describes the layered architecture of the proposed system and outlines the functionality of the layers. Section 4 offers a brief discussion of relevant issues with pointers for further work, and Section 5 concludes the paper.

2. Technological context

A consideration of the technological characteristics of P2P systems is helpful in selecting relevant architectures.

2.1 P2P systems

In a P2P system an intervention by a peer involves two stages. The first stage facilitates access by providing a mechanism for peers to locate and join a network. New nodes need the address of an access point, which is usually well publicised. In Napster this task was achieved by means of centralised indices

[1]. This was however, considered a potential bottleneck and a single point of failure, which was overcome in Gnutella by the distribution of independent lists on the Web [9]. Irrespective of the degree of its centralisation, the entry point mechanism has traditionally been confined to a passive role.

The second stage is concerned with looking up services. In some systems this is mediated by a centralised directory [1], which holds information on all peers. In unstructured systems [9] the flooding mechanism fulfills this role through the propagation of queries from peer to peer, whereas in structured systems [10] it is implemented by distributed hash tables (DHT). Although variants of these schemes have been proposed [11], the general consensus is that centralised directories are incompatible with a pure P2P approach. All these architectures are vulnerable to malicious attacks [12]. Centralised systems and unstructured systems, such as Gnutella, are prone to denial of service attacks while structured systems, such as Chord [10], can be subject to malicious routing.

The core issue that underlines the vulnerability of P2P systems is the difficulty of ensuring the identity of a peer. The openness of P2P systems and the relative anonymity of the transactions that can be conducted in a P2P envirnement can be exploited by malicious peers and abused by free riders [13]. This concern can be addressed by the setting up of a Public Key Infrastructure (PKI) by the deployment of a trust and reputation system [14], or by a hybrid solution.

2.2 Public key infrastructure (PKI)

A PKI promotes a centralized approach to security. Its appeal stems from the association of public keys with their respective identities and their guarantee by a certificate authority. With its implied centralisation a fully-fledged deployment of a public key infrastructure may go against the fundamental principles of a pure P2P system. It has however, the advantage of facilitating authentication and of enforcing non-repudiation through signatures. Furthermore, it can also ensure confidentiality through encryption.

2.3 Trust management

Researchers have resorted to social control mechanisms as a way of assessing the trustworthiness of peers. Trust and reputation systems [15, 16] were introduced in P2P systems in order to create a safe environment for collaboration, to deter uncooperative behaviour and free-riding, and to counter malicious attacks such as content pollution and denial of service (DOS). Trust defined as the belief that a peer is

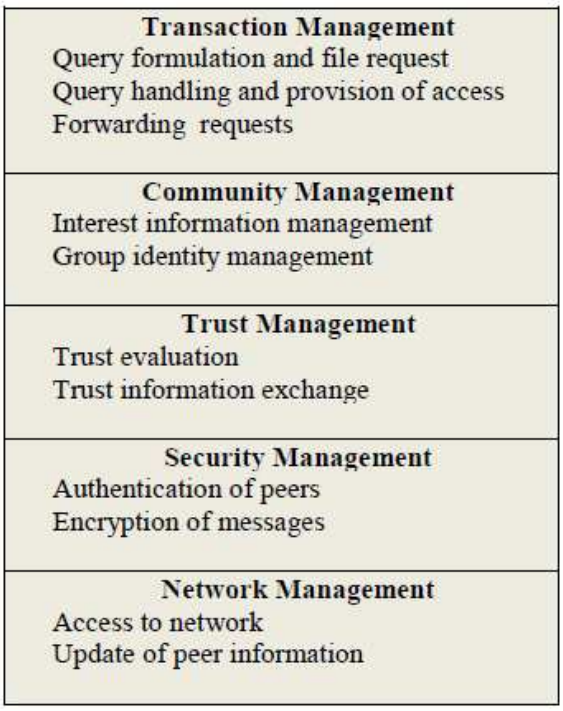

Figure 1. Layered architecture

reliable, well-intentioned and honest, is built gradually from the experience of direct interactions between any two peers [17, 18]. Reputation is defined as the public perception of the trustworthiness of a peer; it is generated from the recommendations or references from other peers. While trust is personalized and subjective, reputation is a collective measure.

In trust-based systems a peer assigns a trust level to another peer based on an assessment of its past behaviour. Threshold values are used to discriminate between trusted and un-trusted peers, and to influence patterns of interaction. Trust assessment can be refined by associating a trust level with a peer and with a group of peers.

3. A hybrid layered architecture

As noted earlier, authentication and non-repudiation can be difficult to enforce in a decentralized P2P system. A hybrid architecture, which combines partial centralisation for security enforcement, with autonomous peer behaviour seems more appropriate. A trust management system can also be grafted onto the P2P network. As trust has many facets the mapping of ability, benevolence and integrity onto a P2P system may be problematic.

In meeting security and trust requirements a number of functional components were identified. Network

management, security management, trust management, community management and transaction management were integrated into a layered architecture. This type of architecture offers a number of benefits. Functional components can be developed independently with the added advantage of reuse. Layering facilitates enhancement, configurability and adaptivity [19]. The proposed architecture is presented in Figure 1 with a brief outline of the most important functions of each layer. In the hierarchy of layers, the lowest is the network and the highest is the transaction management layer. The significant feature of this P2P architecture is its hybrid mode of operation. The P2P entry point mechanism acts as a certificate authority and therefore introduces some partial centralisation.

3.1 Network management

Flexibility in mode of operation was one of the key factors in the design of the system. This was facilitated by a combination of hybrid solutions employed in the network layer. An unstructured network, similar to Gnutella, was implemented with flooding as a vehicle for query propagation, controlled by a decrement hop count. This type of network is suitable for highly dynamic and heterogeneous environments. When a peer receives a query it first returns its own result and then, if the number of hops remaining for that query is greater than one, it decrements the number of hops and forwards the query to its neighbours. Peers keep up-to-date information on other peers by sending 'periodic keep alive' messages. If a peer fails to reply to three successive ping attempts it is assumed dead, or at least uncooperative, and is removed from the neighbour set.

Access to the P2P network is via an entry point server (EPS). The first contact of a peer with the entry point server is initiated explicitly by a potential newcomer. A successful admission to the network is rewarded by the provision of a list of neighbour peers, which will be subsequently contacted directly without any further mediation by the server.

3.2 Security management

The security layer is concerned mainly with the authentication of the peers and the secure transmission of messages. In the creation of a secure environment the entry point server acts as the trusted third party. Its role as a certificate authority (CA) is to validate the association of identity with public key, without any bearing on the trustworthiness of the peer holding that certificate [18]. Peers who wish to verify certificates need to have access to the public keys of at least some of the high level CAs, who are assumed to be trustworthy [16].

The asymmetric encryption scheme is also exploited in securing communication paths. For efficiency reasons both public and private keys are combined in the hybrid encryption of messages. Data is encrypted using a secret key, which is then encrypted and encapsulated in a message using the public key of the recipient. The recipient can then decrypt the data by using the encapsulated secret key, which is decrypted with its private key. Although the main drawback of the use of a PKI is the implicit centralisation that it promotes its deployment mitigates the effect of impersonation by malicious peers. It has also the advantage of supporting non-repudiation through signatures.

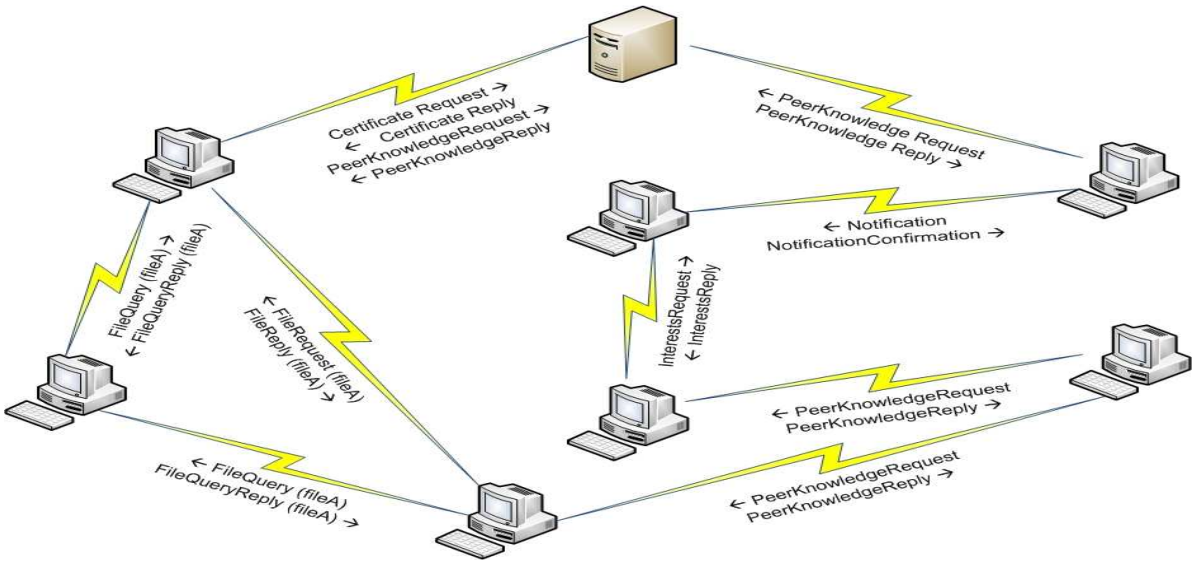

Figure 2. Peer authentication and interaction

Peer initiation

The dynamics of the P2P system initiation is illustrated in Figure 2. It demonstrates the functional interactions that straddle the network and the security layers. A P2P system consists of an entry point server, which also acts as a certificate authority for the network and of a set of peers as client applications. Communication with the server requires the server's digital certificate, which is publicly available. A peer contacts the server to request a digital certificate which represents the identity supplied by another peer. If the identity has not been previously presented the server generates a certificate, signs it and returns it to the client peer; otherwise the server notifies it that the identity is not available. Once the peer obtains a valid digital certificate it requests a list of neighbours from the server, which then returns a random selection of online peers. The requesting peer informs them that it is aware of their existence and forwards its digital certificates. If the peers agree to communicate with it they return their own digital certificate. Once a peer has populated its set of neighbours it no longer needs to contact the server. All other communication in the system takes place directly between peers.

3.3 Trust management

Trust management is relatively complex because it is context sensitive, multi-faceted and dynamic [17]. The level of trust of the P2P network involves two components: the reliability of a peer and the quality of service provided by the peer. The reliability is assessed in terms of the ability of a peer to respond to requests. The reliability of peers may be affected by machine status or network latency, a state of affairs that is often outside the control of peers. The trust rating assigned to a peer should reflect past interactions and take into account the experience of other peers in the system.

Besides the intrusive and inefficient nature of recommendations, a reliance on reputation only may be detrimental to new peers; this is often referred to as the 'cold start problem'. The exclusion of newcomers to the system is prevented by assigning an initial threshold trust value that enables them to interact with other peers. The quality of service is determined by the rating assigned by the user to the files received. This covers a range of possibilities from a harmful file to a file of excellent quality. The quality of service of a peer is the cumulative total of the service quality for each transaction. Although the quality of service is the most critical factor in determining a level of trust, it may be affected by the reliability of a peer. The calculation of trust relates more to the way trust is established between people, and identifies a continuous range of values rather than a mere binary evaluation [18, 20].

Local trust

Trust *(T)* is a function of the reliability *(R)* of a peer and of the quality of service *(S)* for the transactions of that peer, $T = f(R, S)$. The reliability of a peer is calculated as the average of the reliability ratings of all the transactions with that peer. The reliability factor is used to scale the quality of the files as determined by the user. This factor is designed to give a better indication of the quality of the context of interaction and provide a more realistic assessment of the trust. Trust *T* is computed as

$$T = (if(T > 0) then\ R\ else\ 0) \times S\ ,\ \text{where}$$

$$R = \left(\frac{\sum_{i=0}^{Number\ of\ Transactions-1} Reliability_{\ i}}{NumberOfTransactions}\right),$$

$$\boldsymbol{S} = \left(\frac{\sum_{k=0}^{Number\ of\ File\ Ratings-1} QualityOfService_k}{NumberOfFileRatings}\right)$$

Distributed trust

For a peer with any recorded historical behaviour peers issue a request for distributed trust, which is an aggregate of the local trust values returned by known peers. It represents the reputation of a peer. A request for reference is sent to peers who are known and who have a local trust value greater than a user configurable threshold value. The recipients of the request determine their own local trust value for the peer in question, and return it to the requesting peer. Each returned value is scaled by the trust value of the sending peer, as a credibility factor [10, 16, 17]. The average of the received local trust values becomes the distributed trust value for the unknown peer [17, 20, 21].

Trust bias

The calculation of the trust level for a peer involves mainly an assessment of the ability component of trust. This encompasses file quality and response time. As this work does not deal explicitly with free-riding or malicious behaviour, the benevolence and integrity components are subsumed in the evaluation of the ability factor. It is assumed therefore that a peer, who provides files of high quality, within an acceptable interval of time, is benevolent and has some integrity.

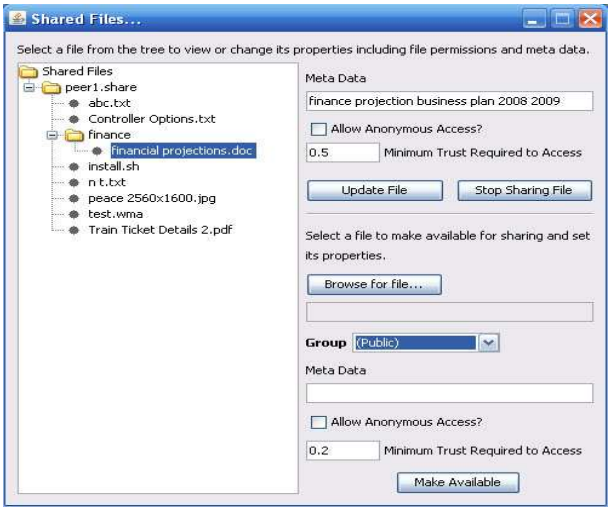

Figure 3. File management

3.4 Community management

In community management peers have the opportunity to overcome the limitations of a flat network structure, by introducing semantically-based structures [22]. An unmanaged scheme without a coordinator presides over community formation. Peers assume responsibility for discovering groups of interest and for joining and leaving them. An explicit declaration of interest in a particular topic by a peer identifies implicitly a group, which eventually includes all the peers that share this interest. This explicit method offers a peer some discretion over its level of commitment to a group while at the same time it minimises storage and processing requirements.

The search for peers with the same interest involves potentially two stages. In the first instance peers in a neighbourhood are queried about an interest. When a match is found the peer requests a list of peers that belong to the corresponding group. If the neighbours do not share the interests of the peer they forward the request to their own neighbours. Each peer in a group maintains an up-to-date list of the peers in that group. Each peer has the ability to query any member of the group and to view the resources that members of the group are making available. A group can be selectively targeted by a peer through directed flooding to request information. Members of a community are more likely to share files based on common interests, and to hold each other in high esteem.

The existence of communities often leads to better organisation of information and its dissemination [22, 23]**.** While the transactions conducted within a community are still constrained by security and trust requirements, they are also subjected to an implicit 'community trust'.

3.5 Transaction management

File management is taken as an example of business transactions and as an illustration of the access to and exchange of resources. The mediation of file sharing is assumed by the transaction management layer and is supported by the lower layers.

File management

A user can decide which files are shared by placing them in a special folder, which may contain subfolders; they hold files specific to a particular interest group. All shared files are held in a tree structure of folders, where the user can view and edit file properties. The customisation process is further refined by granting users some control over file sharing; ten levels of access to files can be set for the other peers (Figure 3).

File Search

The search for files can be performed either by file name or by metadata, to allow for more useful results to be returned. One incentive for peers to provide metadata for the files they are sharing is that they are more likely to receive a higher trust rating if they provide useful files to peers. This in turn will grant them access to a greater range of resources held and shared by the other peers.

Anonymous file transfer is also supported in the P2P system. When files are requested anonymously, the message is forwarded from peer to peer through the network until it reaches the host. On receipt of the message, the host responds to the requesting peer with a download ticket. This peer then sends another download ticket to the peer it received the request from, and so on through the chain until the peer who originally requested the file receives a download ticket and starts to download the file. The file is relayed through all peers involved in the query propagation chain. Each peer knows only the peer it is receiving the file from and the peer it is relaying the file to. Anonymity requires a trade off in security as any file transferred in this way, although encrypted between peers, will be visible to those peers. A user must explicitly allow a file to be accessed anonymously, as trust can no longer be relied upon to protect that file from untrustworthy peers.

The retrieval of file duplicates by clients is pre-empted by the return of an MD5 hash of a file with its content. This hash acts as a file handle that uniquely

identifies each file. When query results are received and collated only the first response for each unique MD5 hash is added to the list of results. It is assumed that the result received first is most likely to originate from a peer who may be physically closer or has more system resources available.

Client interface

A graphical user interface enables client peers to initiate and control their interactions with other peers (Figure 4). The design of the interface reflects to a large extent the hierarchy of layers of the P2P architecture. At the top of the window the network layer is indicated by the identity and the connection status. The behaviour of the security layer is outside the control of the user and is therefore transparent. The trust layer manifests itself in the setting of a threshold minimum value and in the display of distributed trust ratings for a particular peer. In the group management layer provision is made for peers to join and leave groups, and to view the resources shared by peers in a group. The most important part of the window is devoted to the transaction management layer where a high level of customisation can be set in the search for files.

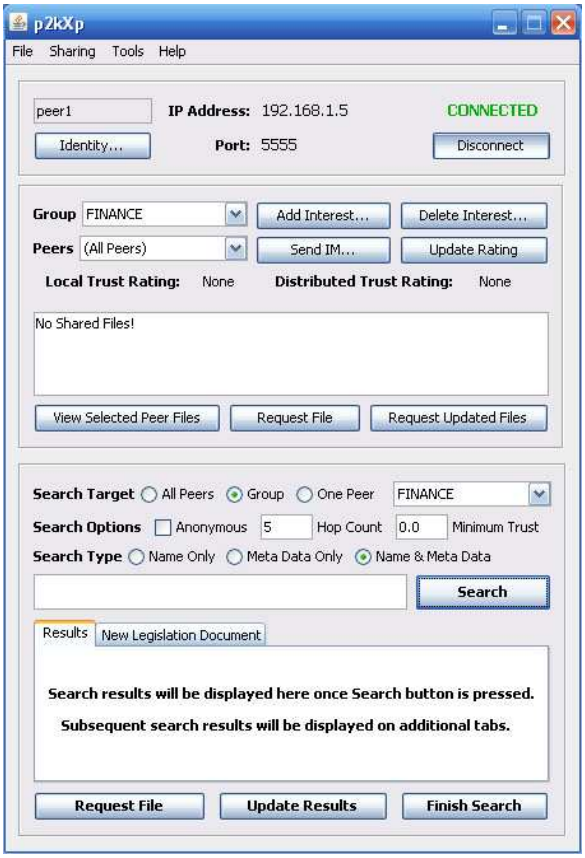

Figure 4. Transaction management

4. Discussion

The proposed hybrid architecture offers adequate mechanisms for ensuring security and trust. There are however, a number of issues for consideration:

- In meeting some of the fundamental requirements in e-business, the proposed P2P architecture has deviated from the pure P2P model. It has conferred to the entry point server (EPS) a central role in the PKI. This partial centralisation underlines the potential vulnerability of the EPS as a bottleneck and as a single point of failure. This can be alleviated by the provision of a number of entry point servers.
- The design and implementation of the system has benefited from the layering approach. The different functional components were integrated seamlessly. Trust management was enhanced by the incremental functionality of the different layers.
- The focus of trust management has been on the determination of the ability of a peer. A more comprehensive mapping of trust should also give a greater weight to integrity by identifying and sanctioning explicitly, for example, malicious behaviour. Trust evaluation is however enhanced by community management. A tighter community is bound to lead to a higher level of trust.
- In its management of trust the system conforms to the 'pull model' where a peer sends its local trust value to another peer on request. A peer does not propagate its adjustments to trust levels following a 'bad' or' good' experience with other peers. Propagation on a wider scale might obscure trust assessment and increase communication and processing overheads. The propagation of a re-evaluation of trust may be more relevant within a community where a high level of integrity can be assumed and maintained.
- File sharing was used as a demonstration of the functionality of the transaction management layer. Despite its limitations, this application has managed to illustrate the interaction and behaviour of the different layers. A more business-based application would have shed more light on features such as non-repudiation.
- The P2P approach offers a viable alternative to the increasing centralisation and control of many initiatives such as Cloud computing.

5. Conclusion

A layered P2P architecture was presented as a platform for the conduct of some e-business transactions. In meeting security and trust requirements partial centralisation was introduced into a hybrid system, by deploying a public key infrastructure. Authentication and non-repudiation were supported by the deployment of a PKI, whereas trust was established mainly by the determination of the level of the ability of peers. The community management layer provides further refinement in trusted transactions.

Although, the architecture forms an adequate basis for a safe environment, its functionality can be extended by taking into account the integrity of peers, and by introducing spatial parallelism to enhance its resilience.

6. References

[1] Chopra D., Schulzrinne H., Marocco E. and Ivov E. (2009). Peer-Peer Overlays for Real-Time Communication: Security Issues and Solutions, *IEEE Communications Surveys & Tutorials, Vol. 11, No 1*, First Quarter 2009.

[2] Napster, *http://www.napster.com/*.

[3] BitTorrent–what is a tracker? *http://support.bittorrent.com/*

[4] White A., Peterson K., and Lheureux B. New P2P solutions will redefine the B2B supply chain. *Technical report, Gartner Research Note*, February 2003.

[5] Pankaj P., Hyde M.and Rodger J.A. P2P Business Applications: Future and Directions. *Communications and Network, Vol. 4 No. 3*, 2012, pp248-260.

[6] Butler J.K. toward understanding and measuring conditions of trust: evolution of a conditions of trust inventory. *Journal of management 17(3)*, 1991, pp643-663.

[7] Jarvenpaa S.L., Knoll K. and Leidner D.E. Is anybody out there?: antecedents of trust in global virtual teams. *Journal of Management Information Systems - Special section: Managing virtual workplaces and teleworking with information technology, Volume 14 Issue 4*, March 1998, pp29-64

[8] Caceres R.C. and Paparoidamis N.G., Service quality, relationship satisfaction, trust, commitment, and business-to-business loyalty. *European Journal of Marketing, Vol. 41, Issue 7/8*, pp836-867

[9] Ripeanu M. (2001), Peer-to-Peer Architecture Case Study: Gnutella network. *http://www.cs.uchicago.edu/%7Ematei/PAPERS/gnutella-rc.pdf*

[10] Stoica I., Morris R., Karger D., Kaashoek M.F. and Balakrishnan H. (2001), Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", *SIGCOMM'01*, August 2001, San Diego, California, USA. MIT Laboratory for Computer Science, *http://pdos.lcs.mit.edu/chord/*

[11] Lu J. and Callan J. (2003), Content-based retrieval in hybrid peer-to-peer networks. *Proc. of the twelfth international conference on Information and knowledge management, CIKM '03* pp199 – 206.

[12] Stutzbac D. and Rejaie R.(2008) Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems. *IEEE/ACM Transactions on Networking, Vol. 16, No. 2*, April 2008.

[13] Mekouar L., Iraqi Y. and Boutaba R.: Peer-to-peer's most wanted: Malicious peers, *Computer Networks 50(4)*, 2006, pp545-562.

[14] Cornelli F., Damiani E., De Capitani di Vimercati S., Paraboschi S. and Samarati P., Implementing a Reputation-Aware Gnutella Servent, *Proc. of the International Workshop on Peer-to-Peer Computing*, Pisa, Italy, May 2002, pp321-334.

[15] Liu L. and Shi W. Trust and Reputation management, *IEEE Internet Computing*, September/October 2010.

[16] Marti S. and Garcia-Molina H. (2006) Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks, Volume 50, Issue 4, Elsevier*, 15 March 2006, pp472–48

[17] Wang Y. and Vassileva J. (2003) Trust and Reputation in Peer-to-Peer Networks, *Proc. of the 3rd IEEE International Conference on Peer-to-Peer Computing*, Linkoping, Sweden, 2003, pp150-157.

[18] Li H. and Singhal M. (2007). Trust Management in Distributed Systems. *Computer , 40 (2)*, February, 2007, pp45-53.

[19] Aberer K., Alima L.O., Ghodsi A., Girdzijauskas S., Haridi S. and Hauswirth M. (2005). The essence of P2P: a reference architecture for overlay networks, *Fifth IEEE International Conference on Peer-to-Peer Computing*, (P2P 2005), August 2005, pp11-20.

[20] Yu B., Singh M.P and Sycara K. (2004). Developing trust in large-scale peer-to-peer systems, Proc. of the *First IEEE Symposium on Multi-Agent Security and Survivability*, Philadelphia, USA, *2004*, pp1-10.

[21] Wang, Y. and Varadharajan, V. (2006). DynamicTrust: The Trust Development in Peer-to-Peer Environments. *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing*, Taichung, Taiwan, 2006, pp302- 305.

[22] Khambatti M. , Ryu K., Dasgupta P. (2002) Efficient Discovery of Implicitly Formed Peer-to-Peer Communities, *International Journal of Parallel and Distributed Systems and Networks, Vol. 5, No. 4*, 2002.

[23] Tian H., Zou S., Wang W. and Cheng S. (2006). A Group Based Reputation System for P2P Networks, Autonomic and Trusted Computing, *Lecture Notes in Computer Science, 2006, Volume 4158/2006*, pp342-351.