Enhancing Security and Response Time for Secure Search in Unstructured Peer to Peer Network

A. Suganya[1], P. Laura Juliet[2]

[1]M. Phil Research Scholar, Department of Computer Science
Vellalar College for Women, Erode-12, Tamilnadu, India
suganyaemerlin@yahoo.co.in

[2]Assistant Professor, Department of Computer Applications (PG)
Vellalar College for Women, Erode-12, Tamilnadu, India
p.laurajerome@yahoo.co.in

Abstract: ***In a peer-to-peer (P2P) network, every machine plays the role of client and server at the same time. Peer to peer networks can be categorized into structured and unstructured peer to peer networks. In structured networks, the peer or system which start to search a file into other peers by establishing paths (i.e. the source system knows where the searching happen are). The unstructured P2P networks do not have a well-known architecture. In unstructured networks, there is no relationship between the source with other peers except neighbor peer and its location. Our proposed work is to search a file in structured and unstructured peer to peer network with authenticity, integrity, and non-reputation.***

Keywords: peer to peer network, distributed hash table, transport layer security, secure socket layer.

1. Introduction

1.1 Peer to Peer Network

A peer-to-peer (abbreviated to P2P) computer network is one in which each computer in the network can act as a client or server for the other in the network, allowing shared access to various resources such as files, peripherals, and sensors without the need for a central server. P2P networks can be set up within the home, a business, or over the Internet.

1.2 Architecture of P2P Systems

Peer-to-peer systems often implement an abstract overlay network, built at Application Layer, on top of the native or physical network topology. Such overlays are used for indexing and peer discovery and make the P2P system independent from the physical network topology. Figure 1 shows the architecture of p2p systems.

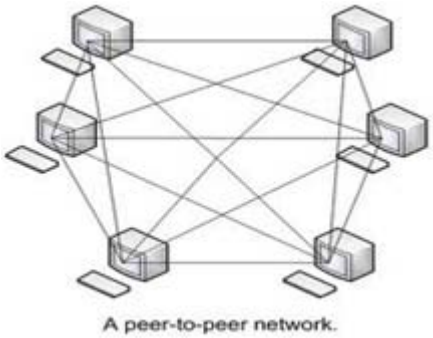

Figure 1: Architecture of p2p Systems

A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client–server model where communication is usually to and from a central server.

1.3 Classification of Peer To Peer Network

We can classify the P2P networks as,

- Structured
- Unstructured

1.3.1 Structured Network

Structured P2P networks employ a globally consistent protocol to ensure that any node can efficiently route a search to some peer that has the desired file/resource, even if the resource is extremely rare. Such a guarantee necessitates a more structured pattern of overlay links.

1.3.2 Unstructured Network

Unstructured P2P networks do not impose any structure on the overlay networks. Peers in these networks connect in an ad-hoc fashion based on some loose set of rules. An unstructured P2P network is formed when the overlay links are established arbitrarily.

2. Existing System

2.1 The MAXDISJOINT Replica Placement

The properties of tree-based routing can be used to construct a replica placement that creates disjoint routes to provide the reader with some intuition and then move toward a more formal definition. After defining the placement, which we call MAXDISJOINT evaluate the necessary replication degree to create a desired number of disjoint routes. Then introduce the notion of a run and provide an expression for the maximum tolerable run length for a given replication degree. MAXDISJOINT is a more adaptive and flexible solution than equally spaced replication. Finally, outline the basic elements of an implementation of the MAXDISJOINT placement.

2.2 Replica placement in pastry

To replicate an object with id 101 in this Pastry ring. Node 121 routes to this object through the routing table entry marked “10x” in Fig. 2. Suppose replicate the object with the id 111 to target the routing table entry “11x” in the example. This approach creates an additional disjoint route for any lookups for object 101 originating at node 121. One route is forwarded via the entry “10x” and the other is forwarded via “11x”.

However, consider another source node 221. This node routes to the object 101 and 111 through the same entry marked “1xx” and, therefore, does not gain an additional disjoint route. To move toward a more effective approach, consider all the replicas of object 101 that create an additional disjoint route for node 121. These are: 001, 111, 120, 122, 123, 131, 201, and 301. Note that there are total of nine possible disjoint routes3 (including the route to the object 101), which is the number of routing table entries for node 121. Of these replicas, there are only three that can create an additional disjoint route for every possible source node: 001, 201, and 301.

These replicas create disjoint routes by targeting entries in the first row of the routing table. Note that targeting an entry in the first row of the routing table requires a single replica whose id differs from that of the master object in the first digit. To target entries deeper in the routing table, a larger number of replicas are required. Suppose we wish to create five disjoint routes for all possible source nodes.

Four routes can be created for every possible source node using the three replicas already discussed (001, 201, and 301) in addition to the object 101. To create the fifth route, we must target an entry deeper in the routing table. In the case of node 121, we may choose the replica 111. As alluded to before, this replica only creates a disjoint route for those source nodes whose ids start with the prefix “1” because these are the only nodes with an entry for “11x.” Since there are four possible values for the prefix (B ¼ 4), four replicas are required to target this routing table entry: 011, 111, 211, and 311. One of these four replicas will create an additional route for every possible source node depending on its prefix.

The remaining three will be routed through previously used routing table entries overlapping a previous route. This is shown graphically in Figure 2. Five disjoint routes are created for node 121, one each for the replicas R001 (or R011), R101, R111, R201 (or R211), and R301 (or R311). In a similar fashion, we can create a sixth disjoint route using the replicas 021, 121, 221, and 321; and a seventh using 031, 131, 231, and 331. This pattern continues until the entire id space is exhausted. Note that in Pastry each node partitions the id space using prefixes and, therefore, we place replicas by varying their prefixes.

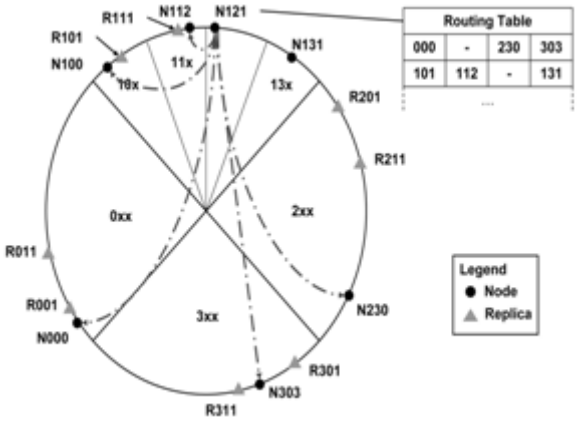

Figure 2: Replica Placement in Ring

2.3 Replica Placement and Neighbor Set Routing

Replica placement is an efficient way of creating disjoint routes because it does not require significant modification to the underlying DHT routing scheme. Although these approaches do not create provably disjoint routes, there is value in introducing some additional form of route diversity.

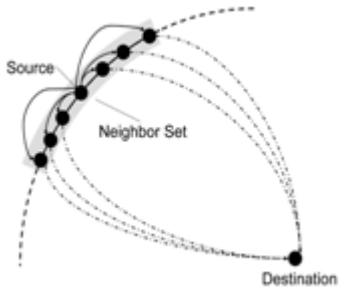

Figure 3: Representation of Neighbor Set Routing

Use neighbor set routing to find diverse routes toward the neighborhood of a key. To create diverse routes, messages are routed via the neighbors of the source node. This is depicted graphically in Figure 3. Castro et al. claim that this technique is sufficient in the case when replicas are distributed uniformly over the identifier space, as in CAN and Tapestry. We consider the ability of neighbor set routing to create diverse routes to a replica to enhance the routing robustness of MAXDISJOINT.

3. Proposed Methodology

3.1 Steps for Digital Signature

A key generation algorithm is that selects a private key uniformly at random from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.

A key generation algorithm involves the use of two keys:

A public key, which may be known by anybody, and can be used to encrypt Messages
A private key, known only by the recipient, and used to decrypt messages

A signing algorithm that, given a message and a private key, produces a signature:

A signature verifying algorithm is that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.

3.2 The Protocols of Secure Sockets Layer and Transport Layer Security

Transport Layer Security (TLS) is a protocol that ensures privacy between communicating applications and their users on the Internet. When a server and client communicate, TLS ensures that no third party may eavesdrop or tamper with any message. TLS is the successor to the Secure Sockets Layer (SSL). TLS/SSL can be used to create a secure environment for web browsing, emailing, or other client-server applications. For example, TLS can be used to create a secure connection between your organization's donation web page and a donor's web browser. The donor's financial or other personal information is encrypted in such a way that only you and the donor can access and use it.

TLS/SSL encryption requires the use of a digital certificate, which contains identity information about the certificate owner as well as a public key, used for encrypting communications. These certificates are installed on a server – typically a web server if the intention is to create a secure web environment, although they can also be installed on mail or other servers for encrypting other client-server communications. The Transport Layer Security is layered on top of the Transport Layer such as TCP.

TLS is composed of two layers:

TLS Record Layer
TLS Handshake Layer

3.2.1 TLS Record Layer

The TLS Record Layer is used for encapsulation of various higher level protocols such as the handshake protocol, the alert protocol, the change cipher spec protocol, and the application data protocol.

The TLS Record Layer is used for encapsulation of various higher level protocols such as the handshake protocol, the alert protocol, the change cipher spec protocol, and the application data protocol.

The TLS Record Protocol provides connection security with some encryption method such as the Data Encryption Standard (DES). The TLS Record Protocol can also be used without encryption.

3.2.2 TLS Handshake Layer

The TLS Handshake Protocol allows the server and client to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before data is exchanged.

The TLS Handshake Layer consists of the handshake protocol, the alert protocol and the change cipher spec protocol. The orange box with the HTTP1 layer and TLS Record layer combined constitutes Hypertext Transfer Protocol Secure (HTTP2) as shown in Figure 4.

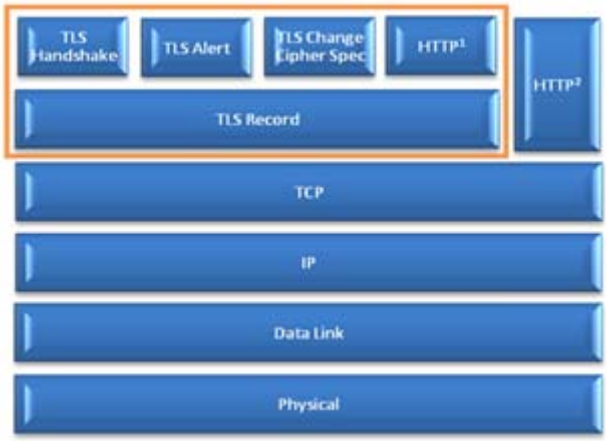

Figure 4: SSL/TLS Protocol Layers

4. Result and Discussion

The proposed protocol was evaluated with a simulating 11 peers participating in data transactions per instance simulation, at unstructured p2p network. The peers were assigned unique IP addresses. Additionally, each peer was assigned a goodness factor to account for the fact that a good peer is likely to participate in higher number of transactions (as a provider) than a bad peer. Hence, the reputation of a good peer is likely to escalate faster than the reputation of a bad peer. This was done to ensure that a peer lost more reputation on performing a malicious transaction as compared to the reputation gained by doing a good transaction. The percentage of malicious peers was varied from 10 percent to 90 percent. The probability that a peer would cheat was set to 1/2 in order to account for the fact that in the real world honesty is not constant and varies with time and stakes.

For each iteration of a simulation, a randomly selected peer became the provider and another randomly selected peer assumed the role of a requester. After the transaction, the requester gave a recommendation to the provider. For each recommendation received by the provider, its reputation was incremented by its goodness factor. After 20,000 transactions, the ranks of the peers were calculated without using the proposed identity management mechanism. The differences of the ranks were averaged for all peers in the network and the results were statistically analyzed.

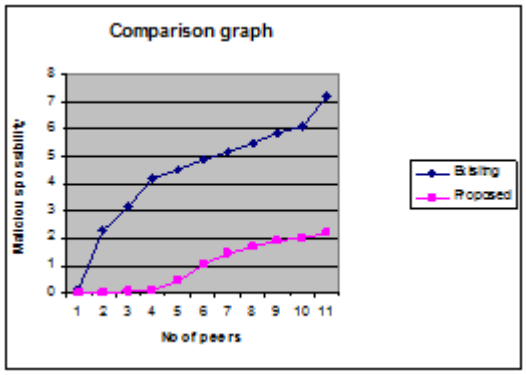

Figure 5: No of peers Vs Malicious possibility (Malicious transaction)

In order to evaluate the combined benefit of self-certification, the cryptographic protocol, we modified the experiments done for the evaluation of the cryptographic protocol, and added an availability factor, AF, to each node. The availability factor accounts for the erratic availability of the past recommenders of a given peer. AF values from 50 percent to 90 percent were randomly allocated to peers. The number of malicious transactions were counted and compared with the results obtained. As is visible in above fig the number of malicious transactions in the system is reduced when the proposed protocol is used along with self-certification instead of without self certification. The total number of malicious transactions increased considerably with an increase in the number of transactions when the proposed model was not used but are more or less constant when the proposed model was used the rate of increase in the number of malicious transactions was much less when reputations were used

5. Conclusion and Future Enhancement

This paper presents self-certification, an identity management mechanism, reputation model, and a cryptographic protocol that facilitates generation of global reputation data in a P2P network, in order to expedite detection of rogues. The self-certification-based identity generation mechanism reduces the threat of liar farms by binding the network identity of a peer to his real-life identity while still providing him anonymity.

The Identity mechanism is based on the fundamental that the ranks of the peers are more relevant than the absolute value of their reputation. The cost of this security is the difference in the ranks of the providers because of the use of the proposed mechanism. The global reputation data are protected against any malicious modification by the third party peer and are immune to any malicious modifications by their owner. The proposed protocol reduces the number of malicious transactions and consumes less bandwidth per transaction than the other reputation systems proposed in its category.

Thus, our future work focus on novel overlay formation algorithm for unstructured P2P networks. Based on the file sharing pattern exhibiting the power-law property, our future work will unique in that it poses rigorous performance guarantees.

References

[1] Cyrus Harvesf and Douglas M. Blough, “Replica Placement for Route Diversity in Tree-Based Routing Distributed Hash Tables” IEEE Transactions On Dependable And Secure Computing, Vol.8, No.3, May/June 2011.

[2] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. Wallach, “Secure Routing for Structured Peer-to-Peer Overlay Networks,” Proc. Symp. Operating Systems Design and Implementation (OSDI ’02), pp. 299-314, 2002.

[3] Y. Chen, R.H. Katz, and J. Kubiatowicz, “Dynamic Replica Placement for Scalable Content Delivery,” Proc. Int’l Workshop Peer-to-Peer Systems (IPTPS ’02), pp. 306-318, 2002.

[4] M. Srivatsa and L. Liu, “Vulnerabilities and Security Threats in Structured Peer-to-Peer Systems: A Quantitative Analysis,” Proc. IEEE Ann. Computer Security Applications Conf. (ACSAC ’04), pp. 252-261, 2004.

[5] A. Ghodsi, L.O. Alima, and S. Haridi, “Symmetric Replication for Structured Peer-to-Peer Systems,” Proc. Int’l Workshops Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P ’05), pp. 74-85, 2005.

Author Profile

A. Suganya received the B. Sc (physics) degree in 2008 from Bharathiar University, Coimbatore and MCA degree in 2011 from Anna University, Coimbatore. She is currently doing M.Phil degree in Computer Science from Bharathiar University, Coimbatore.

P. Laura Juliet was born on 29-06-1976; she received the B. Sc (Physics), MCA degree from the Bharathidasn University, Trichy in 1996 and 1999 respectively. She received the M.Phil (computer Science) degree from Mother Teresa women’s university, Kodaikanal in 2004. She is an Assistant Professor in the PG Department of computer Application Vellalar College for women (Autonomous) Erode. Totally she has twelve years teaching experiences. She published one paper in national journal. Also she has presented more than twenty papers in the international, national and state level Conference and seminars.