Trust Based Content Distribution for Peer-To-Peer Overlay Networks

S.Ayyasamy[1] and S.N. Sivanandam[2]

[1]Asst. Professor, Department of Information Technology,
Tamilnadu College of Engineering, Coimbatore-641 659, Tamil Nadu, INDIA.
Email: ayyasamyphd@gmail.com

[2]Professor and Head, Department of Computer Science and Engineering,
PSG College of Technology, Peelamedu, Coimbatore-641 004, Tamil Nadu, INDIA.

Abstract

In peer-to-peer content distribution the lack of a central authority makes authentication difficult. Without authentication, adversary nodes can spoof identity and falsify messages in the overlay. This enables malicious nodes to launch man-in-the-middle or denial-of-service attacks. In this paper, we present a trust based content distribution for peer-to-peer overlay networks, which is built on the trust management scheme. The main concept is, before sending or accepting the traffic, the trust of the peer must be validated. Based on the success of data delivery and searching time, we calculate the trust index of a node. Then the aggregated trust index of the peers whose value is below the threshold value is considered as distrusted and the corresponding traffic is blocked. By simulation results we show that our proposed scheme achieves increased success ratio with reduced delay and drop.

Keywords

Replica, Overlay, Clusters, QoS, Content, Routing

1. Introduction

Peer -to-Peer Overlay Network

Peer-to-peer networks follow a different paradigm than client-server based systems. A key underlying attribute is that each node participates in the network by offering and using services at the same time. There is no central control and the network organizes itself in a dynamic way. A peer-to-peer (P2P) system is defined as follows: “Peer-to-peer (P2P) systems are distributed systems without any centralized control or hierarchical organization, where the software running at each node is equivalent in functionality”

Because P2P networks are built at the application layer and use the underlying network for the exchange of messages, P2P systems are also called overlay networks. P2P networks have evolved in recent years. Early systems (e.g., Gnutella) used flooding for message routing in the network. Any node receiving a search request would broadcast this message to all its neighbors. The message has a time-to-live value which is reduced at every hop to prevent messages from being routed in the network forever. These systems cannot give any formal guarantees that a message in the network will reach its destination. Furthermore, broadcast messages impose an unnecessary traffic burden on the network [1].

10.5121/ijnsa.2010.2211

Emerging peer-to-peer (P2P) applications benefit from the large amount of resources provided by many individual peers. Using sophisticated techniques for aggregation and Replication of these resources, P2P-based systems are able to provide a much higher robustness and performance than traditional client/server-based applications. For example, file-sharing applications like eMule or Bit Torrent [2] are able to provide access to huge amount of content in a reliable way. At the same time, an increasing number of applications make use of basic P2P network infrastructures like Chord [3] or Pastry and benefit from the good scalability properties of these systems [4]**.**

Peer-to-Peer Content Distribution

Peer-to-Peer (P2P) content sharing service has grown in significance on the Internet, both in terms of the number of participating users and the traffic volume. However, due to the self-organization and self-maintenance nature of P2P overlay networks, each participating user has to manage the potential risks involved in the application transactions without adequate experience and knowledge about other users [5].

Content distribution via peer-to-peer networks goes a step beyond towards a completely distributed structure involving the resources of the peers interested in the content. P2P content distribution allows for more flexibility in the overlay network, which may be structured according to different content e.g. by trackers for each item in the Bit Torrent network or according to other criteria. The size of the overlay can automatically adjust to the population of peers and thus user demand with a replication strategy for the data being set up by the P2P protocol [6].

Security Challenges

Because P2P systems are inherently different from client-server systems, new challenges for security arise. For instance, the lack of a central authority makes authentication in a pure P2P network difficult. Without authentication, adversary nodes can spoof identity and falsify messages in the overlay. This enables malicious nodes to launch man-in-the-middle or denial-of-service attacks. Douceur showed in that without a trusted agency which certifies identities, adversary nodes can control a large fraction of an overlay network. Castro *et al*. identify three requirements for secure structured overlay networks: secure node-ID assignment, secure routing table maintenance, and secure message forwarding [1].

Threats specific to P2P-SIP include subversion of the identity-mapping scheme attacks on the overlay network routine scheme, bootstrapping communications in the presence of malicious first-contact nodes, identity enforcement (Sybil attacks), traffic analysis and privacy violation by intermediate nodes, and free riding by nodes that refuse to route calls or otherwise participate in the protocol other than to obtain service for themselves (selfish behavior).

The modern P2P systems need to deal with selfish (a.k.a “leechers” or “free-riders”) or malicious users1, P2P worms , Byzantine faults and Sybil attacks, Eclipse attacks, flash crowds, etc. Some of these problems are particularly challenging for large-scale, peer-to-peer systems [7].

Current research efforts have mainly focused on trust management techniques to recognize trustworthy peers on P2P network. For collecting peers trust values in the P2P network, majority of approaches presented in this area uses special algorithms. In this paper, we present a trust based content distribution for peer-to-peer overlay networks, which is built on the trust management scheme. To evaluate traffic from other peers and dynamically update their trust values by a peer, this trust scheme is used. This paper is an extension of our previous work [12].

2. Related Works

Ruichuan Chen et al [5] have proposed a unique poisoning-resistant security framework based on the idea that the only trusted sources to verify the integrity of the requested content would be the content providers. A content provider publishes the information of his shared contents to a group of content maintainers self-organized in a security overlay, to present the mechanisms of availability and scalability. Hence a content requestor can confirm the integrity of the requested content from the associated content maintainers. They have devised a scalable probabilistic verification scheme, to further enhance the system performance.

B. Mortazavi et al [7] have first provided a survey of related systems. They then have focused on content distribution networks that have honest but sensibly selfish users. Their focus is to expand a novel reputation framework, in which in the absence of misrepresentation of the reputation values can reveal the tendency to cooperate of the peers. Based on their framework, a game is designed in which users play to maximize the files received from the system by adjusting their cooperation level and gaining a better reputation as a result.

Thomas Repantis et al [8] have proposed a decentralized trust management middleware, based on reputation for unstructured, ad-hoc, peer-to-peer networks. In their middleware to requests for data or services, the reputation information of each peer is stored in its neighbors and piggy-backed on its replies. In self-organizing networks the lack of structure and the dynamic nature of the network are usually regarded as barriers in managing trust information. Their approach utilizes these characteristics to build a self-organizing, non-intrusive trust management infrastructure resistant to tampering and collusions.

V. Valli Kumari et al [9] have offered a weighted feedback based reputation computation. With dynamic correction, this mechanism attempts to solve the reputation computation problem. The simulation results with highly dynamic peer behaviors, changing malicious feedbacks and the dynamic corrections are presented. The feasibility of their mechanism with respect to minimum overheads of storage and retrieval are also discussed.

Mujtaba Khambatti et al [10] have proposed an approach for trust management in P2P systems. They have established an optimistic role-based model for trust amongst peers and prove that it is scalable, dynamic, revocable, secure and transitive. Their proposed solution allows asymmetric trust relationships that can be verified by any peer in the system through a simple, low-cost algorithm. The authors have introduced a metric known as iComplex which combines a peer's trust value for each of its roles into a single, relative, probabilistic guarantee of trust. Finally, they have also discussed the no repudiation of peer relations and how their trust model allows peers to revoke relationships with malicious peers.

G.H. Nguyen et al [11] have proposed a probabilistic model to handle trust in a P2P setting. It supports a local computation and a simple form of propagation of the trust of peers into classes of other peers. They have claimed that it is well suitable to the dynamics of P2P networks and to the choice of each peer within the network to have different perspectives towards the peers with which it interacts.

3. QoS-Aware Content Distribution and Data Retrieval

System Model

Let us consider a collection of N server nodes which form a peer to peer (P2P) overlay network. In addition to being part of the overlay, each node functions as a server responding to requests (queries) which come from clients outside of the overlay network. An example could be that each node is a web server with the overlay linking the servers and clients being web browsers on remote machines requesting content from the servers.

We assume each node always stores one copy of its own content item which it serves to clients and that it has additional storage space to store k replicated content items from other nodes which it can also serve [3]. The object is associated with an authoritative *origin server* (OS) in the network where the content provider makes the updates to the object. The object copy located at the origin server is called the *origin copy* and an object copy at any remaining server is called a *replica*.

Replica Placement

In our QoS aware topology, nodes are grouped into strong and weak clusters based on their weight vector which comprises the following parameters:

- Available capacity
- CPU speed
- Memory size
- Access Latency

In the replica placement algorithm, we classify the content as Class I and Class II, based on their access patterns. (i.e.) The most frequently accessed contents are ranked as Class I and the less frequently accessed contents as Class II. Then more copies of Class I content are replicated in strong clusters (having high weight values). Routing is performed hierarchically by broadcasting the query only to the strong clusters.

Query Search and Data Retrieval

A route discovery algorithm is needed to determine if and where the requested item is replicated when the requester does not have knowledge of the destination.

By reducing the communication cost, the speed and efficiency of the information retrieval mechanism can be improved. So, the number of messages exchanged between the nodes and the number of cluster nodes that are queried for each query request, are to be minimized. For this, a robust searching algorithm is proposed.

In this algorithm, each node maintains a profile which contains the details of queries processed by its neighbors, within the last t seconds.

Node Id	(Ni)
Query Id	(Qid)
Query Hits,	(Qhit)
No.of Results	(NoR)

This profile information is used to forward the queries to the neighbors, which are having more chances of replying to those queries.

In order to forward a query Q, to its neighbors, a node N1 assigns a score to each of its neighbors based on their profile. To calculate the score of each node Nj, (j=2, 3…) N1 compares .Q with all queries stored in Nj's profile. If there is a query hit for Q, then the score of Nj can be calculated as

$$\text{Score (Nj, Q)} = \sum_{k=0}^{m} \text{NoR (Nj, Qk)}^{\alpha}$$

Where NoR (Nj, Qk) is the number of results returned by Nj for query Qk, which are similar to Q. So the node which returns more results, get the higher score.

The value α allows us to add more weight to the most similar queries. For example, when α is large, then the query with the largest similarity NoR (Nj, Qk) dominates the formula. If we set α = 1, all queries are equally counted, whereas setting α=0 allows us to count only the number of results returned by each peer.

(i) When a data request is initiated at a client, it first looks for the data item in its own cache (local hit). If there is a local cache miss, the client sends the *request to* the set of strong cluster nodes.

(ii) On receiving the request, each strong cluster node which has the requested content, will send an ack packet to the query client to acknowledge that it has the data item. The ack packet will contain the following fields: time stamp Ts and weight W. The time stamp field helps to choose the latest copy of the searched item and the weight value field helps to choose the best client node.

(iii) When the query client receive *ack* packets from the strong cluster, it selects the best node Sbest with $\max(T_s, W)$

(iv) The client then check the reputation of Sbest using the trust evaluation algorithm described in the next section.

(v) If the node Sbest is a trusted node, then the client sends a confirm packet to it. The *ack* packets for the same item received from other nodes are discarded.

(vi) When the node Sbest receives a *confirm* packet, it responds back with the actual data value to the requested query node.

(vii) Suppose if the requested data is not available in any strong cluster nodes, the request is directed to the server from the query client. Then the necessary data is sent to the client from the server. If the client has the available memory size (MZ) and bandwidth (BW), then it caches the data in its buffer. Then it is also considered as a strong cluster node and it is propagated to other nodes as

{Nid, Clid (“S”), d1}

Where Nid is the node id, Clid is the cluster id and d1… is the content database id.

4. Trust Management Scheme

Current research efforts have mainly focused on trust management techniques to recognize trustworthy peers on P2P network. For collecting peers trust values in the P2P network, majority of approaches presented in this area use special algorithms. In this paper, we present a trust based content distribution for peer-to-peer overlay networks, which is built on the trust management scheme. The main concept is, before sending or accepting the traffic, the trust of the peer must be validated. Based on the success of data delivery and searching time, we calculate the trust index of a node. Then the aggregated trust index of the peers whose value is below the threshold value is considered as distrusted and the corresponding traffic is blocked. To evaluate traffic from other peers and dynamically update their trust values by a peer, this trust scheme is used.

Calculation of Trust Index (TI)

The trust value is calculated based on the factors Success of Data Delivery Ratio and Search Time. Depending on the outcome, the Trust Index of the target t assigned by the node m $\{TI_{mt}\}$ can be calculated as follows:

$$TI_{mt} = w1 * T_{req} \qquad (1)$$

Where T_{req} is the total number of requests and

$$w1 = (SDR/ST)*100 \qquad (2)$$

Here w1 represent the trust percentage value of single data, where SDR and ST represent Success of Data Delivery Ratio and Search Time respectively.

The Trust Index of the node increases when SDR increases and it decreases when ST increases.

Trust Evaluation

When the data is retrieved from a target node, each node involved in content distribution, calculates the trust index (TI) value of the target based on the retrieved data using (1) and (2). These calculated TI values are transmitted to a set of nodes called recommenders, which maintains a table called Trust Index Table (TIT) to store the TI values of all the targets.

When the source s wants to retrieve data from a target t, it selects the recommender nodes from the Recommender list. Then it sends a trust request packet for the target to all recommenders R1, R2…Rn. The recommenders send their corresponding Trust Index of the target from the TIT to the requester. (Refer fig.1)

The aggregated trust index of t (ATI_t) is calculated from all the trust index values of t assigned by the nodes m_1, m2…m_n

$$ATI_t = \sum_{j=1}^{n} TI_{mjt} \qquad (3)$$

Traffic from a peer is accepted or rejected depending on its trust index and trust threshold scale of the peer calculating its reputation.

Two thresholds Th1 and Th2 represent host thresholds for aggregated trust index (ATI). where, peers with ATI below Th1 are distrusted and consequently no traffic is accepted from them. Th2 denotes full trust, peers with trust index above this value are considered trustworthy and therefore their traffic can be fully accepted. If ATI falls between Th1 and Th2, average trust is assigned to a peer and only part of its traffic is accepted. This trust scheme allows a peer on the network to evaluate traffic from other peers and dynamically update their trust values.

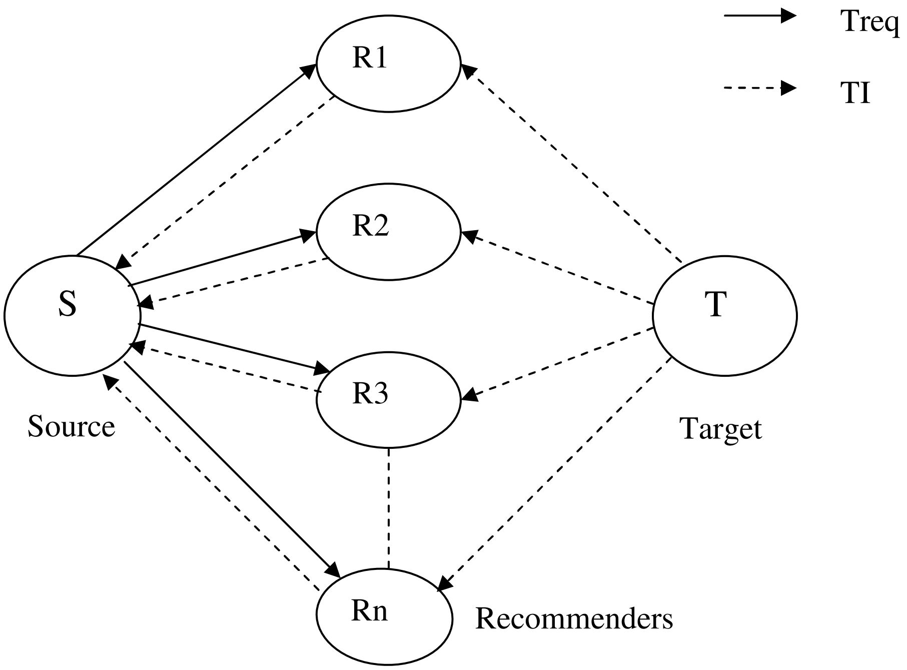

Fig. 1. Trusting in Peer to Peer Network

Trust Evaluation Algorithm

1. Source finds the target node for data retrieval using our query search algorithm.
2. Source selects the recommender nodes from the Recommender list.
3. The source sends a trust request packet for the target to all recommenders.
4. All recommenders send their corresponding Trust Index of the target to the requester.
5. Requestor now calculates aggregate TI (ATI) from the entire TI received from the recommenders.
6. It then compares the ATI with the thresholds Th1 and Th2.
7. If ATI < Th1, then

   The target is distrusted.
8. Else if ATI > Th2, then

```
        The target is fully trusted.
Else if ATI >= Th1 and ATI <=Th2, then
        The target is partially trusted.
End if
```

5. Experimental Results

Simulation Setup

This section deals with the experimental performance evaluation of our algorithms through simulations. In order to test our protocol, the NS2 simulator [13] is used. We have used the Bit Torrent packet-level simulator for P2P networks [14]. Based on the assumption that the bottleneck of the network is at the access links of the users and not at the routers, we use a simplified topology in our simulations. We model the network with the help of access and overlay links. Each peer is connected with an asymmetric link to its access router. All access routers are connected directly to each other modeling only an overlay link. This enables us to simulate different upload and download capacities as well as different end-to-end (e2e) delays between different peers.

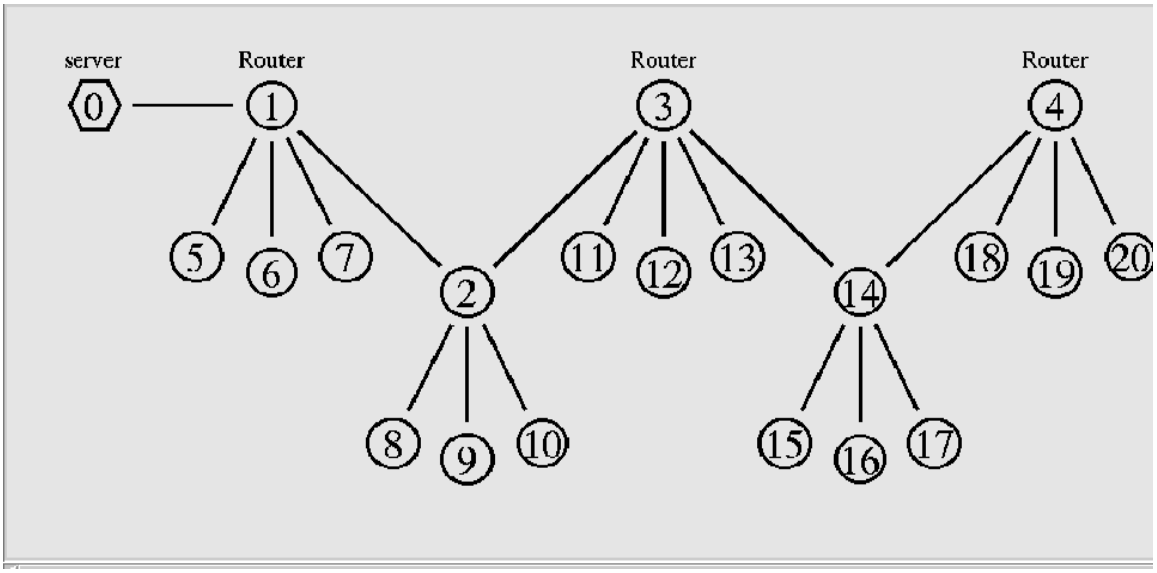

Fig. 2: Topology of P2P Overlay Network

Simulation Results

We have conducted simulations for both trust-based and no trust based content distribution scenarios. We evaluate the following metrics:

Success Ratio: It is measured as the successful downloads made per number of queries.

Delay: It is measured as the total delay occurred in sending a query and obtaining the results.

Drop: It is the number of packets dropped during the downloading.

A. Varying Rate

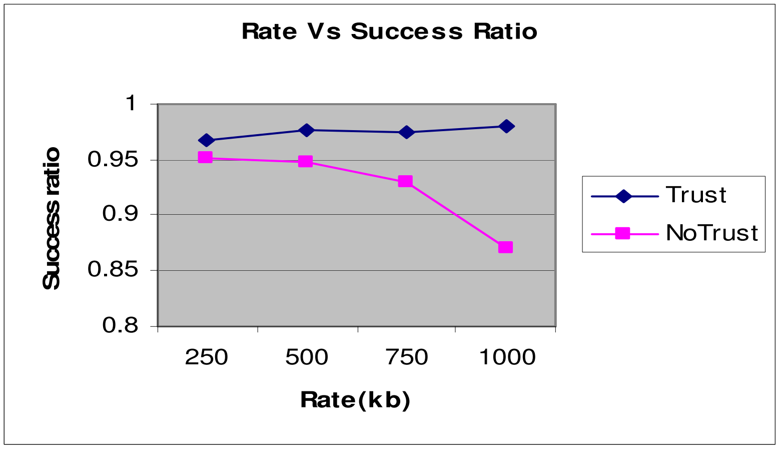

Figure 3: Rate Vs Success Ratio

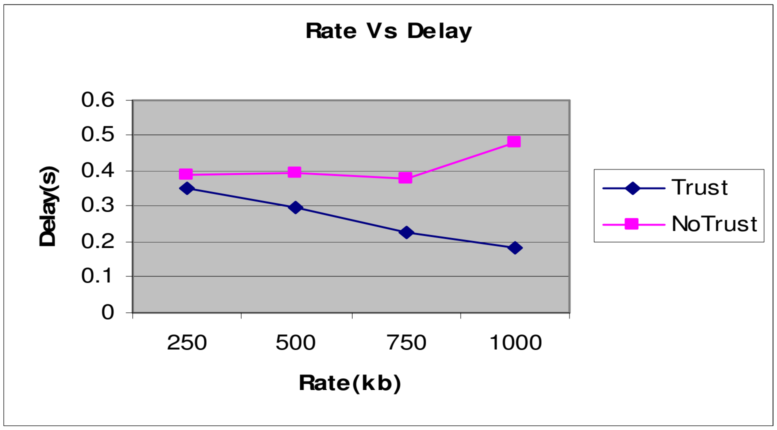

Figure 4: Rate Vs Delay

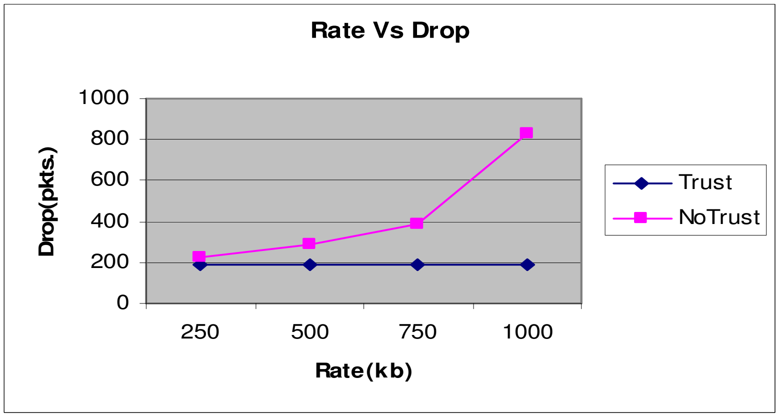

Figure 5: Rate Vs Packet Drop

In our first experiment, the query sending rate is varied from 250Kb to 1Mb. Figure 3 shows that the success ratio of the client nodes increases when trust evaluation is applied but it decreases when trust is not applied. From Figure 4, we can see that the delay increases in case of no trust scenario whereas it decreases in our trust based scenario. Figure 5 presents the packets dropped when the rate is increased. From the figure we can see that the drop is more and increased in case of no trust case but it is almost constant in the trust based case.

B. Varying Dishonest Nodes

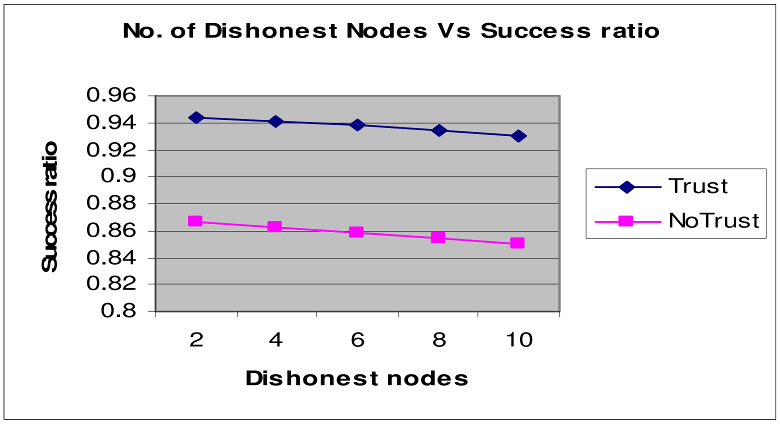

Figure 6: Dishonest Nodes Vs Success Ratio

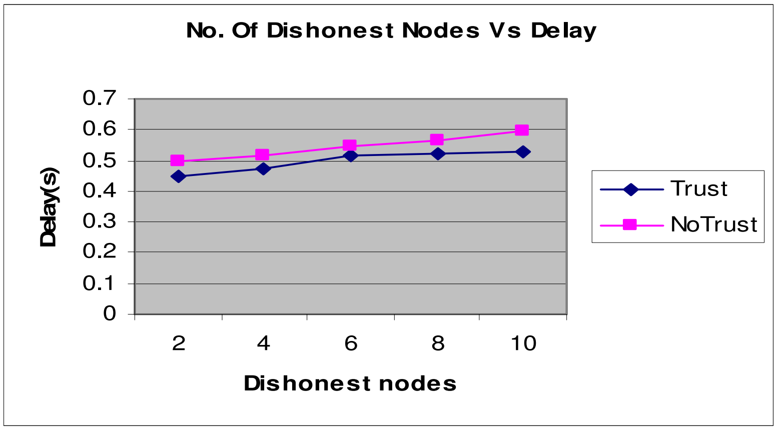

Figure 7: Dishonest Nodes Vs Delay

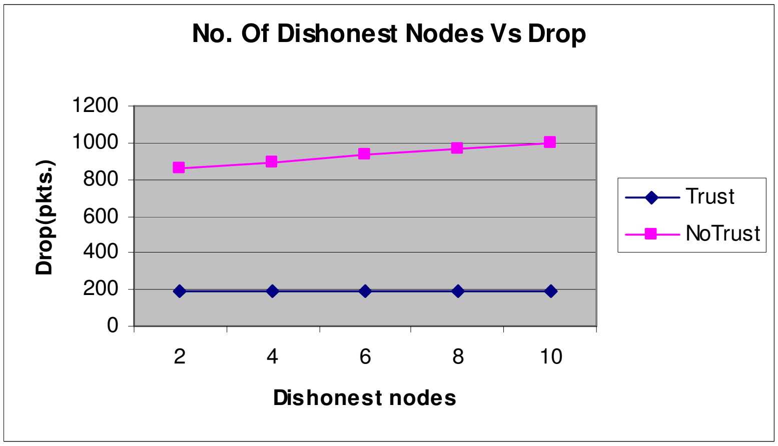

Figure 8: Dishonest Nodes Vs Packet Drop

In our second experiment, the number of dishonest nodes is varied as 2,4,6,8 and 10. Figure 6 shows that the success ratio of the client nodes is more when trust evaluation is applied but it is less when trust is not applied. From Figure 7, we can see that the delay is more in case of no trust scenario whereas it is less in our trust based scenario. Figure 8 presents the packets dropped when the dishonest nodes are increased. From the figure we can see that the drop is more and increased in case of no trust case but it is almost constant in the trust based case.

6. Conclusion

In this paper, we have presented a trust based content distribution for peer-to-peer overlay networks, which is built on the trust management scheme. The main concept is, before sending or accepting the traffic, the trust of the peer must be validated. When the data is retrieved from a target node, each node involved in content distribution, calculates the trust index (TI) value of the target based on the factors Success of Data Delivery Ratio and Search Time. These calculated TI values are transmitted to a set of nodes called recommenders, which maintains a table called Trust Index Table (TIT) to store the TI values of all the targets. When the source wants to retrieve data from a target, it selects the recommender nodes from the Recommender list. Then it sends a trust request packet for the target to all recommenders. The recommenders send their corresponding Trust Index of the target from the TIT to the requester. Then the aggregated trust index of the peers whose value is below the threshold value is considered as distrusted and the corresponding traffic is blocked. By simulation results we have shown that our proposed scheme achieves increased success ratio with reduced delay and drop.

References

[1] Jan Seedorf, “Security Challenges for Peer-to-Peer SIP”, IEEE Network, September/October 2006.

[2] B. Cohen, “Incentives Build Robustness in BitTorrent”, Workshop on Economics of Peer-to- Peer Systems, Berkeley, CA, USA, June 2003.

[3] Stoica, R. Morris, D. Karger, M. Kaashoek, H. Balakrishnan: “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, ACM SIGCOMM 2001, pp. 149-160, San Diego, CA, USA, August 2001.

[4] David Hausheer, Burkhard Stiller, “PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications”, Networking 2005, pp. 40-52, IFIP 2005.

[5] Ruichuan Chen, Eng Keong Lua, Jon Crowcroft, Wenjia Guo, Liyong Tang, Zhong Chen, “Securing Peer-to-Peer Content Sharing Service from Poisoning Attacks”, pp. 22-29, 8th International Conference in p2p computing, IEEE, 2008.

[6] Halldor Matthias Sigurdsson, Ulfur Ron Halldorsson, Gerhard Hasslinger, “Potentials and challenges of peer-to-peer based content distribution”, Telematics and Informatics 24, pp. 348-365, 2007 Elsevier Ltd.

[7] B. Mortazavi_ and G. Kesidis, “Cumulative Reputation Systems for Peer-to-Peer Content Distribution”, pp. 1546-1552, 2006 IEEE.

[8] Thomas Repantis Vana Kalogeraki, “Decentralized Trust Management for Adhoc Peer to Peer Networks”, ACM, MPAC: Vol.182, Proceedings of the 4th international workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006), USA.

[9] V. Valli Kumari, B. Dinesh Reddy, T. Sri Devi, Ramaprasad R. Kalidindi and KVSVN Raju, “Credibility Based Corrective Mechanism for Reputation Computation in Peer-to-Peer Communities”, International Journal of Computer Science and Network Security, Vol. 8, No. 5, May 2008.

[10] Mujtaba Khambatti, Partha Dasgupta, Kyung Dong Ryu, "A Role-Based Trust Model for Peer-to-Peer Communities and Dynamic Coalitions", Second IEEE International Information Assurance Workshop (IWIA'04), pp. 141-154, April 2004.

[11] G.H. Nguyen, P. Chatalic, M.C. Rousset, "A Probabilistic Trust Model for Semantic Peer to Peer Systems", pp. 59-65, Proceedings of the 2008 international workshop on Data management in peer-to-peer systems(DaMaP;Vol.261), 2008.

[12] S.Ayyasamy and Dr.S.N.Sivanandam, "A QoS-Aware Intelligent Replica Management Architecture For Content Distribution In Peer-to-Peer Overlay Networks", International Journal on Computer Science and Engineering, Vol. 1(2), pp. 71-77, 2009.

[13] Network Simulator, http://www.isi.edu/nsnam

[14] Kolja Eger,Tobias Hoßfeld, Andreas Binzenhofer, "Efficient Simulation of Large-Scale P2P Networks:Packet-level vs. Flow-level Simulations", in proceedings of 2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks, pp: 9-16, June 2007.

About the Authors

Mr.S.Ayyasamy completed his B.E. (Electronics and Communication Engineering) in 1999 from Maharaja Engineering College and M.E. (Computer Science and Engineering) in 2002 from PSG College of Technology, both under Bharathiar University, Coimbatore. Currently he is pursuing PhD degree from Anna University, Coimbatore. He is working as an Assistant Professor in the Department of Information Technology at Tamilnadu College of Engineering, Coimbatore. He is a member of various professional bodies like ISTE, CSI and IAENG. His research areas include P2P networks, Overlay Networks, Load Balancing and Quality of Services and having 9 years of teaching experience in Engineering Colleges.

Dr.S.N.Sivanandam completed his B.E. (Electrical Engineering) in 1964 from Government College of Technology, Coimbatore, and MSc (Engineering) in Power Systems in the year 1966 from PSG College of Technology, Coimbatore. He acquired PhD in control systems in 1982 from Madras University. He received best teacher award in the year 2001 and **Dhakshina Murthy Award** for teaching excellence from PSG College of technology. He received the citation for best teaching and technical contribution in the year 2002, Government College of Technology, Coimbatore. His research areas include Modeling and Simulation, Neural Networks, Fuzzy Systems and Genetic Algorithm, Pattern Recognition, Multidimensional system analysis, Linear and Non linear control system, Signal and Image processing, Control System, Power System, Numerical methods, Parallel Computing, Data Mining and Database Security. He is a member of various professional bodies like IE (India), ISTE, CSI, ACS and SSI. He is a technical advisor for various reputed industries and engineering institutions.