International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.1, January 2013

A MALICIOUS USERS DETECTING MODEL BASED ON FEEDBACK CORRELATIONS

Yong WANG[1], Yang BAI[2], Jie Hou[3] and Yuan-wei TAN[4]

[1]Department of Computer Science and Engineering, University of Electronic and Science Technology of China, Chengdu, China
cla@uestc.edu.cn
[2]Department of Computer Science and Engineering, University of Electronic and Science Technology of China, Chengdu, China
baiyang.cncq@gmail.com

ABSTRACT

The trust and reputation models were introduced to restrain the impacts caused by rational but selfish peers in P2P streaming systems. However, these models face with two major challenges from dishonest feedback and strategic altering behaviors. To answer these challenges, we present a global trust model based on network community, evaluation correlations, and punishment mechanism. We also propose a two-layered overlay to provide the function of peers' behaviors collection and malicious detection. Furthermore, we analysis several security threats in P2P streaming systems, and discuss how to defend with them by our trust mechanism. The simulation results show that our trust framework can successfully filter out dishonest feedbacks by using correlation coefficients. It can effectively defend against the security threats with good load balance as well.

KEYWORDS

Peer-to-peer Streaming Network, Trust Framework, Malicious Detection, Correlation Coefficient

1. INTRODUCTION

In the recent years, P2P streaming systems have achieved tremendous commercial success. In P2P streaming systems, a peer makes partnerships with a subset of other peers in the network, each participating peers periodically sends to its neighbours "buffer maps", which indicate the chunks it has ready for sharing, Meanwhile, it also forwards requested chunks to its neighbours. Compared to the traditional centralized applications, P2P streaming systems have several advantages, such as higher scalability, lower deployment cost, and more efficient. The measurement results show that P2P streaming applications consume about 40% per cent of Internet traffic [1].

However, with the increasing popularity of P2P steaming systems, many security issues have to be faced by both end users and service providers. P2P streaming brings new intellectual property protection (IPR) problems, because piracy can be much easier distributed in the network. Digital right management (DRM) [22] can protect users and service providers legal right effectively. Worms, Trojans, and viruses are the most typical problems threatening P2P streaming systems, traditional defence mechanisms such as intrusion detection, virus detection, and firewall may be used to secure the systems. Finally, Attacks utilizing P2P streaming system features are the potentially devastating security issues, such as free-riding, data pollution, routing attacks, and index poisoning. Trust and reputation mechanisms, incentive methods, and message signature are involved to defend against these new threats. Malicious attacks and selfish behaviours will cause the drop of the service quality and the system efficiency by affecting the selection of peers in the network. Trust and reputation mechanisms, by assigning

DOI : 10.5121/ijcnc.2013.5104

peers in the system with a uniform trust value, provide an effective way for selecting dependable neighbours. Consequently, various trust models have been proposed [2][3][4][5], some of which concentrating on the calculation of the trust value, others focus on the detection of the malicious peers in the network utilizing peers' trust values. Eigentrust [2] evaluates a peer's global trust value based on its history upload-behaviours, with which a peer can reject low value peers to avoid getting bad quality services. But this model did not consider strategic altering behaviours. Consequently, Chang et al. proposed the DyTrust [6] model with punishment schemes to reduce the affection of strategic altering behaviours.

In this paper, we propose a two-layered overlay to provide the function of peers' behaviours collection and malicious user detection. The lower level of the overlay is organized as a mesh by streaming peers, which is responsible for medium downloading and sharing. The upper level of the overlay is based on KAD [7], which stores the history evaluation of the peers and calculates their state-of-art trust values.

We also propose a trust framework integrating service objects, network community, and trust value correlations. A series experiments have been carried out to analyse the performance of our model. The evaluating results show that our model can effectively filter out dishonest feedbacks, detect malicious behaviours and resist altering behaviours.

The rest of the paper is organized as follows: in section 2, related work on trust model and P2P streaming is discussed; in section 3, the trust framework and the system architecture and algorithms are presented; in section 4, we analyse the complexity and security of our model, simulation experiments and results are shown in this section; in section 5, a brief conclusions and future works are presented.

2. RELATED WORK

In this section, we present P2P streaming systems and trust mechanism proposed by others researchers.

2.1. P2P Streaming System

Traditional streaming system is based on Server-Client model [8], which has good performance on service providing. But when the user's number becomes large, the root server becomes the bottleneck, facing the threat of one point failure. Content Delivery Network (CDN) and IP multicast can basically provide streaming services on the Internet. CDN pushes the services from centre to edge by large number of servers. The CDN can provide good service quality, with reducing stresses of centre server and the backbone network. However, CDN has its drawbacks: high cost and implementation complexity. IP multicast delivers same data once in the same physical link. That is, root server need not send a streaming data through the same physical link repeatedly. Thus, it reduces package sending redundancy and relieves root server's loads. Because of the problems as scalability, reliability of transmission, congestion of controlling and complexity of deployment, IP multicast system has not been widely deployed.

Then peer-to-peer (P2P) based streaming system was proposed to provide scalable, low cost solutions on the Internet. According to the network structure, the P2P streaming system can be divided into two classes: tree-based and mesh-based systems.

In tree-based system, nodes are formed as a multicast-tree. The tree-based system includes three kinds of nodes: root, middle nodes, and leaf nodes. Furthermore, tree-based system can be split into two subclasses: single tree based system, such as PeerCast [9], ZigZag [10], etc. In PeerCast system, each node gets streaming service from its parent, and forward it to several

other nodes, as the result, this system will relieve promulgator's stress. However, the system has bad robustness, and delivery delay will be added with the increasing level of the nodes. Zig-Zag [10] system limits the depth of tree structure, to reduce the delivery delay, a Zig-Zag system whose size is N, its nodes' level will be restricted to $O(\log N)$. Splitstream [11] and P2PCast [12] belong to multi-tree based streaming system. The Splitstream system uses multiple descriptions coding (MDC) to divide streaming data into many independent code streams, and then establishes a multicast tree for each code stream. Users can receive several code streams at one time. Each node in the system should be a middle node in one tree, and be a leaf in other multicast trees. This method will decrease the influence of node's one point failure, and free-riding attacks.

Mesh-based system, for example, CoolStreaming [13] and Prime [14] have no well-organized structure. An initial mesh and multiple trees constitute the topological structure of CoolStreaming system. In the system, each node share and download streaming service with several other nodes. In Prime system, nodes play equal role with other, and each node can connect the rest nodes randomly. In mesh-based network, a node keep neighbourhood with other nodes. Compared with tree-based system, it's not necessary to build a structured topology, as each node may change data with others; these save large amount of buffer memory for the system, and the system would have better scalability and redundancy. Therefore, we introduce mesh-based structure in our system structure. The structure detail describes in section III-B.

2.2. Security Threats in P2P Streaming System

Although streaming systems bring us convenience for medium sharing, many security threats have to be faced by these systems. Generally, these security threats can be grouped into four categories: threats on streaming sources, threats on streaming index, threats on streaming contents, and threats on user-nodes.

-Threats on streaming sources. In P2P streaming systems, the distribution of the medium starts from the source nodes. If the source nodes do not behave fairly or honestly, the efficiency of the distribution may drop severely. On the other hand, if source nodes are not authorized for medium sharing, pirate and sensitive contents may spread and cause the loss of content providers' benefits. DRM is adopted to protect the digital copyright of the medium [22], at the same time, it is very necessary to establish mechanisms to ensure the trustworthiness of the source nodes.

-Threats on streaming index. Users in P2P streaming systems can utilize index service to find medium sources and their neighbors. As the results, because the index service is responsible for the network connectivity, attacks on streaming index may cause the network partitioning. Furthermore, the incorrect index can cause the failure of peers and sources finding process, which leads to the failure of the medium distributions. Unfortunately, there are few ways to ensure the correctness of the index data, the index verifying mechanism was proposed to test the index data which has to cost of additional network bandwidth consumption.

-Threats on streaming contents. Forgery [23], data pollution [24], and contents modifying are most common threats to the streaming contents. These security threats may damage the streaming contents' integrity and availability. Moreover, the spreading of polluted data can cause the lost of the network bandwidth and affect the performance of the streaming systems. Hash signature, PKI framework, and blacklist can be used to defend these threats [25]. However, it is hard to filter out malicious peers distributing forgery contents because of the absence of reputation mechanisms.

-Threats on user-nodes. Users' may connect to malicious users, which leads to receiving purposly generated fake messages, lost of connection to others, or failure of request response. These security threats can be strengthened by collusion of malicious users in the network. Sybil

attack and white-washing are still serious security issues to be tackled. Other previous works show that free-ridding and spammer can limit the network scalability and drop the quality of service. Reputation and auditing mechanisms play an important role to defense these malicious behaviors. We will introduce the trust and reputation mechanism in the next subsection.

2.3. Trust Models

Based on the way of building trust relationship, the trust models can be divided into two categories: trusted third party based models (single CA trust model [17], eBay [18]) and feedback-evaluation based models (PeerTrust[4], DyTrust[6]). Trusted third party based model adopts PKI technology, by utilizing the trusted third party to monitor the performance of entire network, the malicious nodes will be regularly filtered out. When a user wants to get a service, he would check service providers' trust values from the trusted third party, then chose the most trustworthy provider to send request-message.

In feedback-evaluation based models, users evaluate each service and calculate the trust value based on the evaluations. Some of Feedback-evaluation based models provide global trust value, others provide local trust value. P2Prep proposed a reputation sharing protocol for Gnutella [19], in which node keeps track and shares reputation with others. P2Prep uses polling algorithm to ensure nodes' anonymity in streaming sharing process. Bayesian network-based trust model provides a flexible method to represent differentiated trust in aspects of each other's capability and combine different aspects of trust [20].With the limitation of Bayesian network, the computing cost rises with the size of the network. As the consequence, this model is only applicable in relative small-size networks. DyTrust [6] model was proposed with a punishment schemes, and can restrain the affection of strategic altering behaviours. These models can decrease both network expense and trust-building delay. Eigentrust [2], peerTrust [4] and RMS_PDN [21] are global trust provided models. Eigentrust [2], based on service provider's history upload-behaviours, calculates out a global trust value for each node, with which low quality service downloading can be decreased. In reference [4], PeerTrust presented a coherent adaptive trust model for quantifying and comparing the trustworthiness of peers based on a transaction-based feedback mechanism. These models have effective defines for free-riding and cooperative attacks [4][2][6].

2.4. Kademlia

Kademlia was first proposed in [28], as a peer-to-peer distributed hash table (DHT). The Kademlia has many advantage not simultaneously offered by any previous DHT (Chord, CAN, Pastry). Firstly, it minimizes the configuration messages number; secondly, it uses parallel, asynchronous queries to avoid timeout delays from failed node; Thirdly, the algorithm to record each other's existence resists DoS attacks. Specially, the system routes queries and locates nodes using XOR-based algorithm, this algorithm improved the speed of routing.

As its merits, Kadmelia has been introduced in to many system BiTtorrent realized its 4.1.0 version with Kademlia based DHT technology. Besides, Emule implemented the Kademlia based technology in its system. We would introduce the Kademlia into our system structure as well. Our system structure will be introduced in the next section.

3. TRUST FRAMEWORK

3.1. Basic Idea

The purpose of our work is to select trustworthy users and detect malicious behaviours by building up trust and reputation framework in P2P streaming systems. Generally, the goals of the trust framework are as follows:

-It should have good convergence for building and milking users' trust values;

-It should directly response to the users' services quality in P2P streaming systems;

-It can defend classic attacks as well as users' selfish behaviours, such as dishonest feedback and strategically altering behaviours.

To fulfil the above goals, our trust framework integrates service, network community, and trust correlation coefficient to evaluate users' behaviours. A two-layered overlay is adopted to provide the function of users' trust value evaluation and malicious user detection. The lower level of the overlay is organized as a mesh by streaming peers, which is responsible for medium downloading and sharing. The upper level of the overlay is based on KAD [7], which is consisted of a set of stable nodes, storing the history evaluation of the streaming peers and calculating their state-of-art trust values.

We make some assumptions for the trust framework of this paper as:

-The peers of the upper level are more stable with higher storage and computing capacity compared with those of the lower level;

-Each user can provide a concrete evaluation of the services he received.

3.2. Overlay Structure and Communication Messages

The upper level of the overlay is based on KAD, while the lower level of the overlay is structured as a mesh. Peers of the upper level (namely upper nodes) gather and store the evaluating information. The upper nodes are also responsible to the trust value calculating and retrieving. When a peer of the lower level (namely lower node) joins into the system, it chooses several lower nodes as its neighbour randomly.

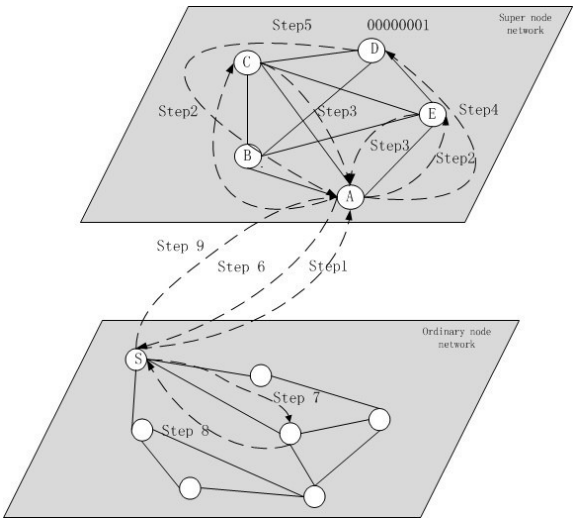

Figure 1. Communication process of the trust framework

There are three classes of communication messages in the trust framework overlay structure:

-**Messages between lower node and upper node**

REPORT message: it is to feedback the evaluation and nodes' information to the upper nodes.

SERVICE_FINDING message: if a lower node wants to get most trustworthy nodes information sharing the specific service R, it sends SERVICE_FINDING message to the upper nodes.

SERVICE_FINDING_RESPONSE message: when upper nodes retrieved the nodes information sharing the service R, the SERVICE_FINDING_RESPONSE message is used to carry the results back to the lower nodes.

-**Messages between upper nodes**

These messages are inherited from KAD communication messages, which are:

FIND_NODE message: request to find out upper nodes who stored service R's nodes information.

FIND_VALUE message: request to find service R's nodes information, such as IPs, ports, and their trust values.

FIND_VALUE_RESPONSE message: it carries nodes information back to the requester.

STORE message: request to save report information of lower nodes into the corresponding upper nodes.

-**Messages between lower nodes**

When a lower node needs streaming data, it sends data request to the most trustworthy neighbour who has the data, after which, the neighbour response with the corresponding streaming data. The communication process is shown in Fig.1.

In step 1, lower nodes (S) send the SERVICE_FINDING message to a corresponding upper node (A).

In step 2-5, upper nodes use FIND_NODE, FIND_VALUE, and FIND_VALUE_RESPONSE messages in KAD to find out the information of the most trustworthy node (D).

In step 6, the upper node A sends SERVICE_FINDING_RESPONSE message carrying the information of D to the lower node S.

In step 7-9, after S received the requested data from D, it sends a REPORT message to the corresponding upper node A. The node A uses the STORE message to save the report information into the corresponding upper node.

3.3. Trust Model and Computation

We assume the lower nodes get streaming data and report their evaluations in certain duration $[t_{start}, t_{end}]$, we divide it into several sub segments which are called timeframes.

(1)The direct evaluation

The direct evaluation is the average of several evaluation values between two fixed nodes with a same resource. It's the original data based on feedback. In our model, *i* represents the lower node requesting streaming data, *j* represents the lower node responding with the data, *R* denotes the requested service. *m* stands for the frequency that *i* getting *R* from *j*, in the n^{th} timeframe; Let

$e_{ij}^{n}(R)$ denotes the estimated value *i* to *R* in node *j*; the direct evaluation from *i* to R of *j*, $D_{ij}^{n}(R)$ is defined as formula (1).

$$D_{ij}^{n}(R)=\frac{1}{m}\sum e_{ij}^{n}(R)\cdots\cdots 1$$

(2)The expected direct evaluation value

The direct evaluation is the average of several evaluation values between several nodes with the same resource. It reflects the resource's general situation. In n^{th} timeframe, node *b* denotes one of user set *clt(j)*, which gets *R* from node *j,* the element number of set *clt(j)* is denoted by *m*. The expected value of *R's* directly evaluation is defined as formula (2).

$$D_{j}^{n}(R)=\frac{1}{m}\sum_{b\in Clt(j)}D_{bj}^{n}(R)\cdots\cdots 2$$

(3)The correlation coefficient based filter function

The correlation coefficient based filter function utilize Pearson correlation coefficient to metric the correlation ship between current node's direct evaluation and the expected direct evaluation value. Let $[D_{ij}^{1}(R)\cdots D_{ij}^{t}(R)\cdots D_{ij}^{n}(R)]$ denote the direct evaluation vector from *i* to R of *j*, from the 1^{th} timeframe to the nth timeframe. Let $[D_{j}^{1}(R)\cdots D_{j}^{t}(R)\cdots D_{j}^{n}(R)]$ denote the expected value of all lower nodes' direct evaluation vector, each element of the direct evaluation vector can be calculated with formula (2). We use the Pearson correlation coefficient to measure the correlation between the vector $[D_{ij}^{1}(R)\cdots D_{ij}^{t}(R)\cdots D_{ij}^{n}(R)]$ and $[D_{j}^{1}(R)\cdots D_{j}^{t}(R)\cdots D_{j}^{n}(R)]$, which is defined as formula (3).

$$Corr_{ij}^{n}(R)=\frac{1}{n-1}\sum_{t=1}^{n}\left(\frac{D_{ij}^{t}(R)-\frac{1}{n}\sum_{k=1}^{n}D_{ij}^{k}(R)}{\sigma_{D_{ij}^{t}(R)}}\right)\left(\frac{D_{j}^{t}(R)-\frac{1}{n}\sum_{k=1}^{n}D_{j}^{k}(R)}{\sigma_{D_{j}^{t}(R)}}\right)\cdots\cdots 3$$

In (3), *n* denotes the current timeframe number, let $\sigma_{D_{ij}^{t}(R)}$ and $\sigma_{D_{j}^{t}(R)}$ denote respectively the standard deviations of $[D_{ij}^{1}(R)\cdots D_{ij}^{t}(R)\cdots D_{ij}^{n}(R)]$ and $[D_{j}^{1}(R)\cdots D_{j}^{t}(R)\cdots D_{j}^{n}(R)]$. According to Pearson correlation coefficient, if $corr_{ij}^{n}\geq corrThreshold$, it indicates that the relation between $[D_{ij}^{1}(R)\cdots D_{ij}^{t}(R)\cdots D_{ij}^{n}(R)]$ and $[D_{j}^{1}(R)\cdots D_{j}^{t}(R)\cdots D_{j}^{n}(R)]$ is acceptable; $corr_{ij}^{n}<corrThreshold$ manifests that the relation between $[D_{ij}^{1}(R)\cdots D_{ij}^{t}(R)\cdots D_{ij}^{n}(R)]$ and $[D_{j}^{1}(R)\cdots D_{j}^{t}(R)\cdots D_{j}^{n}(R)]$ have large discrepancy. Namely, the user's direct evaluation is not in accordance with the expected one, which may mean that the user is not honest. As the result, the model keeps suspicious attitude to the evaluations, and will filter out these dishonesty evaluations by using the $flag_{ij}^{n}(R)$. The filtering mechanism is defined as formula (4), (5) and (6).

$$flag_{ij}^{n}(R)=\begin{cases}0, corr_{ij}^{n}(R)<corrThreshold_{j}^{n}(R)\\ 1, else\end{cases}\cdots\cdots 4$$

$$corrThreshold_{j}^{n}(R)=\frac{1}{k}\sum_{t\in clt(j)}corr_{ij}^{n}\cdots\cdots 5$$

(4)The accumulation feedback quality

$$Q_{ij}^{n}(R)=\frac{\sum_{b\in Clt(j)} flag_{bj}^{n}(R)*Cu_{ib}^{n}*D_{bj}^{n}(R)}{\sum_{b\in Clt(j)} flag_{bj}^{n}(R)*Cu_{ib}^{n}}\cdots\cdots 6$$

In formula (6), *b* is *a* lower node receiving streaming data from *j*, Cu_{ib}^{n} denotes the credibility from node *b* to *i*, and Q_{bj}^{n} denotes the accumulation feedback quality of service *R* of *j* during the n^{th} timeframe.

(5)The current trust value

The current trust value defines the integration of history trust value and the accumulation feedback quality of a service. It reflects the peer's behaviour and the quality of service that provided. Let $S_j^n(R)$ denotes the current trust value of service *R* of *j*, which can be defined as formula (7) and (8).

$$S_j^n(R)=(1-\rho)S_j^{n-1}(R)+\rho\times Q_j^n(R)\cdots\cdots 7$$

$$\rho=\begin{cases}\rho_1, Q_j^n(R)-S_j^{n-1}(R)\geq-\varepsilon\\ \rho_2, otherwise\end{cases}\cdots\cdots 8$$

Where ρ_2 is the destructive factor, ρ_1 is the constructive factor. Where $\rho_1 < \rho_2$. It indicates that building the node's trust value is harder than milking it. Let ε denote the tolerance of errors due to the possible noise during the evaluation process. If $S_j^{n-1}(R)-Q_j^n(R)<-\varepsilon$, the abusing trust A_{ij}^n will be involved to punish the strategic altering nodes to slow down the increasing of its trust value by combining the constructive factor ρ_1(see formula (9) and (10)). The constant *c* is used to control decreasing speed of ρ_1.

$$\rho_1=\rho_1\times\frac{c}{c+A_j^n}\cdots\cdots 9$$

$$A_j^n=\begin{cases}A_j^{n-1}+(S_j^{n-1}(R)-Q_j^n(R)), S_j^{n-1}(R)-Q_j^n(R)<-\varepsilon\\ A_j^{n-1}, other\end{cases}\cdots\cdots 10$$

(6)The group trust

The group trust reflects the influence degree form peer's location group to other groups. Considering clustering property of P2P streaming networks, we define the group trust value T_g^n in the n^{th} timeframe as (11), which depends on the trust values between nodes inside and outside the group.

$$T_g^n = \frac{1}{k}\sum_{j\in e}\frac{1}{r}\sum_{l\notin g}[D_{lj}^n \times \frac{2E_j}{k_j(k_j-1)}] \cdots\cdots 11$$

Where k denotes the number of edge nodes in g, and e denotes the set of edge nodes in a group g. Let j denote one of the edge nodes. l stands for a set of nodes outside of the group g connecting with j. The parameter r is the node number in the set e. Where k_j represents the number of connected nodes of node j in g; and E_j is the real connected edge number between these nodes. $\frac{2E_j}{k_j(k_j-1)}$ denotes the clustering coefficient of node j in group g. In a group, if the clustering coefficient of a node is higher, the influence of its trust value between group members would be larger.

(7)The global trust

We define the global trust is constituted by the current trust value and the group trust value. So we consider the resource's trust not only with its QoS, the provider's behaviour, but also the influence degree. Let θ be the weighted factor of trust value. The global trust of R of j can be described as:

$$GT_j^n(R) = \theta \times S_j^n(R) + (1-\theta)T_g^n \cdots\cdots 12$$

4. EXPERIMENTAL EVALUATION

Experiments were carried out to evaluate the validity of our trust framework. The convergence of the trust model is verified. Furthermore, we analysed its resistance to the typical security scenarios, including strategically altering behaviours of malicious peers and dishonest feedback. Finally, the load balance of the trust framework is analysed as well. We extended the Peersim [26] to simulate the streaming network with our trust framework.

4.1. Convergence

To verify the convergence of our trust model, two experiments are carried out. The first one is to observe the building up and milking down of user's trust values, while the second is to verify the influence of the initial user's trust value.

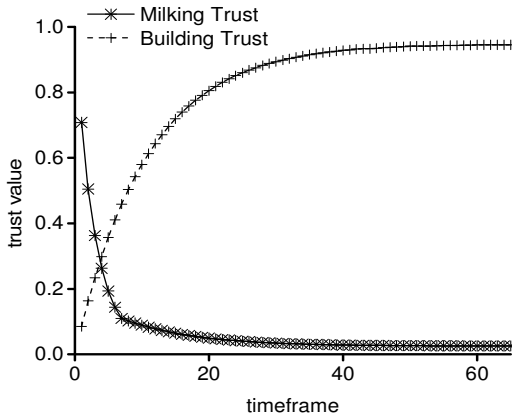

Figure 2. Building trust vs. milking trust

Fig.2 illustrates the trust building and milking process, which shows that the malicious user with initial trust value being 1.0 will lose 0.9 trust value only after 7 timeframes, while it costs about 20 timeframes to build up a user's trust value from 0.1 to 0.8. It indicates that the cost of building up ones reputation is much higher than that of milking it. On the other hand, when a user's trust value becomes relatively high or low, namely 0.92 or 0.03 in Fig.2, it may get steady, which shows that the model can achieve a convergence state.

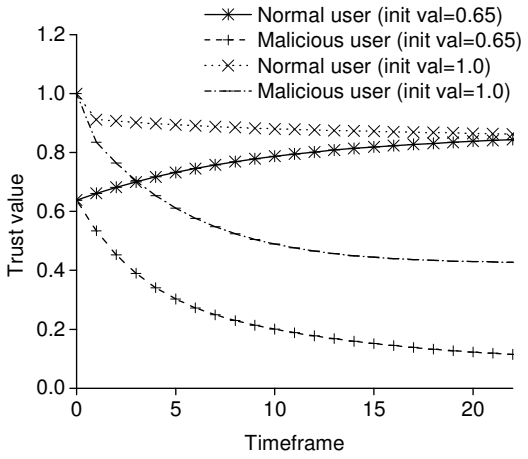

Figure 3. The influence of user's initial trust value

Fig.3 shows the influence of user's initial trust value. When the user's initial trust value is to 1.0, the normal user's trust value converges at 0.92 and that of the malicious one's decreases to 0.4 in around 20 timeframes. Similarly, when the initial trust value is set to 0.65, the normal user's trust value stays at 0.86 and that of the malicious user drops to 0.1 in around 20 timeframes. The results indicate that out trust model can successfully distinguish malicious behaviour and good behaviour in limited timeframes.

4.2. Sensitiveness to Strategically Altering Behaviours

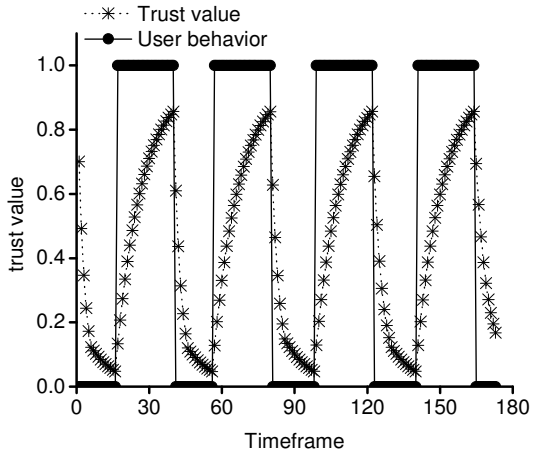

Figure 4. Strategically altering behaviours

Users may strategically change their behaviours to attack the trust model. An attacker can repeatedly behave well to increase their trust values, while carry out malicious behaviours when their trust values are high enough. To test the sensitiveness of our trust model in the experiments, we let the user behave well when its trust value is lower than 0.05, and be malicious once its trust value exceeds 0.85.

Fig.4 shows the variations of the trust values with the strategically altering behaviours. The line with circles means the current user behaviours status. When it stays at 1.0, it shows that the user performs well. While the line stays at 0, it means the user is behaving maliciously. The other line with stars represents the user's real trust value. The results show that it takes about 11 time frames to increase the user's trust value from 0.2 to 0.6 which is much more than that it takes to decrease from 0.6 to 0.2. We can conclude that our trust model is sensitive to realize strategically altering behaviours, and can defend against the threat effectively.

4.3. Effectiveness against Dishonest Feedback

A dishonest user may provide fake evaluations. One way is to spamming, which means an attacker gives out random evaluations. The other is to speaking irony, that an attacker provides adverse evaluations. The dishonest feedback may cause the system evaluations fail to reflect its real situations.

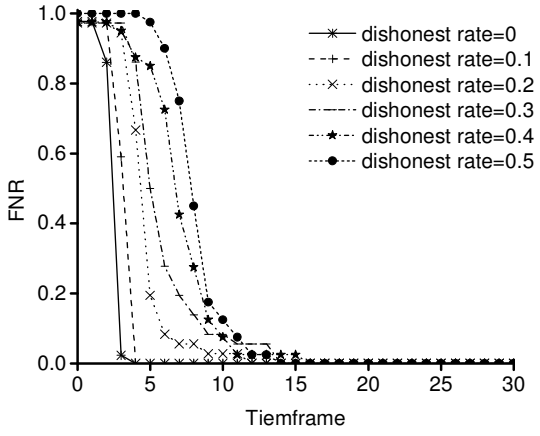

Figure 5. The FNR for different dishonest rate of the trust model (malice rate= 0.3)

We define malice ratio as the number of malicious users divided by the total number of users. Consequently, we define dishonest ratio as the number of dishonest users divided by the total number of users. Note that the behaviours of malice and lying are independent. In order to observe the trust model effectiveness against users' dishonest feedback, we define false positive rate (FPR) as the ratio between the number of non-malicious users evaluated as malicious and the total number of non-malicious users. At the same time, we define false negative rate (FNR) as the ratio between the number of malicious users evaluated as non-malicious and the total number of malicious users.

We set 30% number of users behave poorly in the experiments. When a user's trust value below the average value of the whole users, we treat it as malicious. Fig. 5 shows the FNR for different dishonest rate ranging from 0.0 to 0.5, the increasing step is 0.1. The results show that the FNR drops directly from the initial largest value 0.95 to zero within few timeframes for each

dishonest rate. It is very interesting that the FPRs in the experiments were always zero. These results indicate that our trust model can effectively detect malicious users with the existence of dishonest feedbacks in relatively short time.

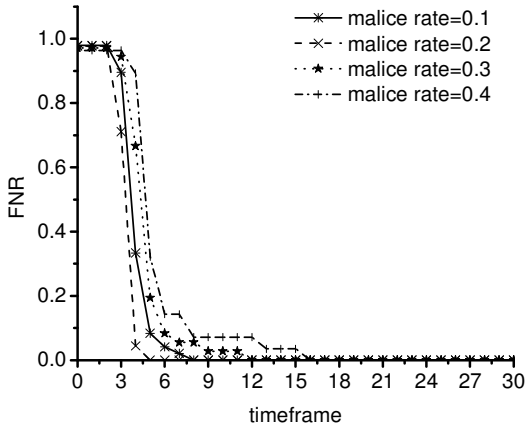

Figure 6. The FNR for different malice rate of the trust model (dishonest rate=0.2)

Fig.6 illustrates the FNR for different malice rate ranging from 0.1 to 0.4. The results show that the FNR converges to near zero within few timeframes, which indicate that the trust model can successfully detect malicious users with the existence of dishonest feedback, even under the condition that the malice ratio is 0.4. Combined with Fig.5, the experiments provide us the evidence that our trust model can effectively resist dishonest feedback attack.

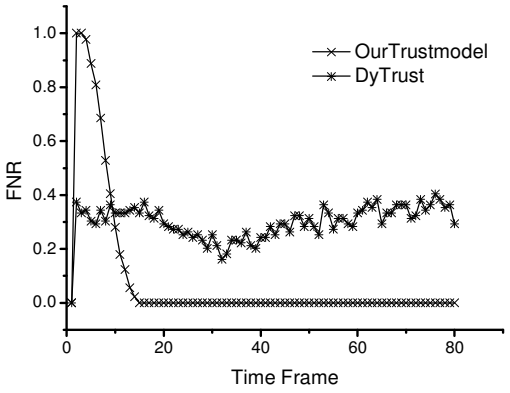

Figure 7. The FNR compare between our model and DyTrust model(dishonest rate=0.2, malice rate= 0.4)

Under the same scene, we do experiment to compare our trust model with other one to verify the model's performance to detect malicious. As our model introduce the Pearson correlation coefficient, global trust to improve the DyTrust model introduced in [6].for this experiment, we chose DyTrust model [6] as the target, set the dishonest ratio equals to 0.25, the malicious ratio equal to 0.4, then observer the malicious detection efficiency of each model. The figure 7 represents the detection FNR variation of two models. The figure 8 represents the detection FPR

variation of two models. The result denotes that our model have high FNR value at begin, but then decrease to 0, while DyTrust keep at a stable FNR value at about 0.4. The FPR of our model keep at 0, while the DyTrust model keeps at 0.6 for about 60 cycles. The main reason behind this can be concluded as: after a short initial time, the vector of direct evaluation can reflect service provider's real situation; so that the Pearson correlation coefficient works well to filter out the dishonest evaluation, hence, the final trust value can represents peer's real situation. In this way, the accuracy of malicious detection can be improved obviously. In conclusion our model has more accuracy detection efficiency than DyTrust model.

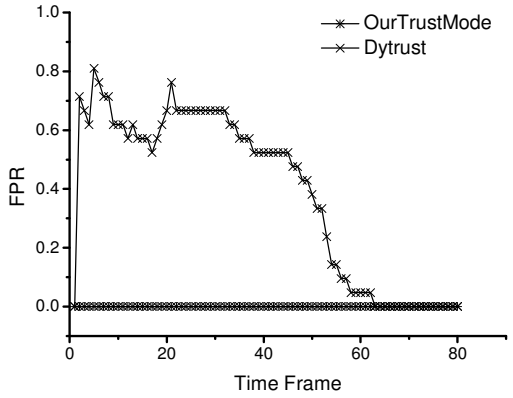

Figure 8. The FPR comparation between our model and DyTrust model (dishonest rate=0.2, malice rate= 0.4)

4.4. Loads Balance

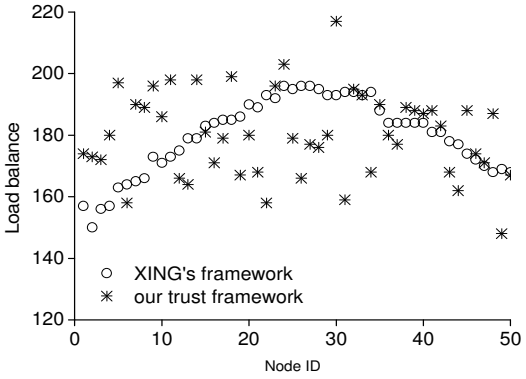

Figure 9. Loads distribution when the network is steady

We set the number of streaming users to 9000, and the number of upper nodes to 50. Parameters used in the load balance experiments are shown in table I SCE 5. Fig.9 compared the load balance of our trust framework and XING's framework [27]. The figure shows that for the XING's framework, the upper nodes' loads are light both on the beginning and ending of node IDs, but the central part of node IDs has heavier loads. The reason is that the monitoring overlay of XING's framework is based on tree graph, and the load redistribution is transmitted along the branches of the tree from root to leaf. On the contrary, the load distribution on upper nodes of our trust framework appears randomly distributed. Because the upper level overlay is based on KAD and no root exists, the load of each upper node is similar. The results indicate that the load balance of our framework seems better than tree based framework.

4.5. Complexity Analyse

We analyse the complexity of our trust model, and compared the computation cost and communication overhead with several other models, such as EigenTrust [2], DyTrust [6], PeerTrust [4], Xing's [27], in Table 2. As DyTrust has not introduced a network background, we use N/A denote not applicable. The analysis of communication overhead considers the message transitive and the route lookup cost. The result denotes that, ensuring system provides trust mechanism and the malicious detection ability, our model has lower computation cost and communication overhead among other four trust models.

Table 1. Complexity Comparison

Trust models	**Computation complexity**	**Communication overhead**
Xing's[27]	O(MN)	O(N)+O(N)
PeerTrust[4]	$O(N^3)$	O(N)+O(logN)
DyTrust[6]	O(N)	N/A
EigenTrust[2]	$O(N^2)$	O(MN)+O(logN)
Our's	$O(N^2)$	O(N)+O(logN)

.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel trust framework in distributed streaming systems. We introduced the evaluation feedback, history trust value, and group trust value into the calculation of a global trust value. The correlation coefficient detecting method is involved to filter out dishonest evaluation feedback, and the punishment mechanism is used to defend against strategic altering behaviours attacks. In the model verifying stage, we build an upper level overlay based on KAD. The upper level overlay plays the role to store and compute trust values. The simulation results show that our trust framework can effectively detect malicious users within around 10 timeframes. The loads of each upper node are evenly distributed.

Our future work in trust mechanism of streaming systems will focus on the detection of malicious users. We will consider the scoring system of feedback evaluation based on streaming system details, so that the evaluation value will have tighter correlation with the facts of streaming system. We will work toward introducing the incentive mechanism into trust framework, so that it can encourage the users to behaviour positively. Moreover, we will try to apply our trust framework into real streaming applications.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work is supported by a Safe Net Research Award, by key technology research and development program of Sichuan province under grant No. 2010FZ0101, and by important national science & technology specific projects under grant No. 2011ZX03002-002-03.

REFERENCES

[1] Cisco Visual Networking Index: Forecast and Methodology, 2010–2015.June 1,2011(white paper)

[2] Kamvar S, Schlosser M, Garcia-Molina H. The EigenTrust algorithm for reputation management in P2P networks,Budapest, Hungary; May 2003

[3] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Managing and sharing servents' reputations in p2p systems. IEEE Transactions on Data and Knowledge Engineering, 15(4):840-854, Jul./Aug. 2003

[4] Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)

[5] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems.Proceedings of First IEEE Symposium on Multi-Agent Security and Survivability, 2004

[6] J.Chang, H.Wang and Y.Gang, "A Dynamic Trust Metric for P2P System", Proceedings of the 5th International conference of Grid and Cooperative Computing Workshops, Changsha, China, 2006, pp.117-120.

[7] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the xor metric," in IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems. London, UK: Springer-Verlag, 2002, pp. 53–65.

[8] YouTube Homepage, [Online]. available: http://www.youtube.com

[9] Deshpande H, Bawa M, Garcia-Molina H. Streaming Live Media over a Peer-to-Peer Network, Stanford University. 2001,(8).

[10] D. A. Tran, K. A. Hua and T. Do. ZigZag: An Efficient Peer-to-Peer Scheme for Media Streaming. In: Proc. of IEEE INFOCOM'03, 2003:1283 - 1292.

[11] Castro M, Druschel P, and Kermarrec A. M. et al. Splitstream: High-bandwidth Content Distribution in a Cooperative Environment. Kaashoek F, Stoica I. eds. IPTPS 2003: 292 - 303. LNCS 2735.

[12] Nicolosi A, Annapureddy S. P2PCast: A Peer-to-Peer Multicast Scheme for Streaming Data. IRIS Student Workshop, MIT, 2003.1-13.

[13] X. Zhang, J. Liu, B. Li, and Y.-S. Yum. CoolStreaming/DONet: a data-driven overlay network for peer-to-peer livemedia streaming. In Proc. of the 24th IEEE Int. Conference on Computer Communications (INFOCOM'05), volume 3, pages 2102–2111, Mar. 2005.

[14] N. Magharei and R. Rejaie. PRIME: Peer-to-peer Receiver-drIven MEsh-based streaming. In Proc. of the 26th IEEE Int. Conference on Computer Communications (INFOCOM'07), pages 1415–1423. IEEE, 2007.

[15] Marsh Stephen．Formalising trust as acomputational concept：[PhD dissertation]．Scotland：University of Stirling，1994

[16] A Abdul-Rahman，S Hailes．Supporting trust in virtual communities．The 33rd Hawaii Int'1 Conf on System Sciences.Maui，Hawaii，2000

[17] Ion M., Danzi A., Koshutanski H., and Telesca L.."A Peer-to-Peer Multidimensional Trust Model for Digital Ecosystems," In:proceedings of the Second IEEE International Conference on Digital Ecosystems and Thecnologies (IEEE-DEST'08), Phitsanulok, Thailand,February 2008.

[18] Bajari, P. and Hortacsu, A. Winner's curse, reserve prices, and endogenous entry: Empirical insights from eBay auctions; see www.stanford.edu/~bajari/wp/auction/ebay.pdf.

[19] M. Ripeanu. Peer-to-peer architecture case study: Gnutella network. Technical report, University of Chicago, 2001.

[20] Yao Wang and Julita Vassileva. "Bayesian Network-Based Trust Model". Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on, pp. 372-378, 2003.

[21] H. Tian, S. Zou, W. Wang, S. Cheng, A group based reputation system for P2P networks, Autonomic and Trusted Computing. No. 4158 in LNCS. Third International Conference, ATC 2006, Springer, Wuhan, China, Sep. 2006, pp. 342–351.

[22] S. R. Subramanya and Byung K. Yi. Digital rights management. IEEE Potentials, 25(2):31–34, April 2006.

[23] Uichin Lee, Min Choi, Junghoo Cho, M. Y. Sanadidi, and Mario Gerla. Understanding pollution dynamics in p2p ¯le sharing. In Proc.IPTPS'06, Santa Barbara, CA, USA, Feb. 2006.

[24] Dhungel, P., X. Hei, and K. W. Ross et al. The Pollution Attack in P2P Live Video Streaming: Measurement Results and Defenses [A]. In: Sigcomm P2P-TV Workshop, Kyoto, 2007

[25] Dhungel, P., X. Hei, and K. W. Ross et al. The Pollution Attack in P2P Live Video Streaming: Measurement Results and Defenses [A]. In: Sigcomm P2P-TV Workshop, Kyoto, 2007

[26] M. Jelasity, G. P. Jesi, A. Montresor, S. Voulgaris, PeerSim S imulator , http://peersim.sourceforge.net/

[27] XING JIN, S.-H. GARY CHAN. Detecting Malicious Nodes in Peer-to-Peer Streaming by Peer-Based Monitoring. ACM Trans. Multimedia Comput. Commun. Appl. 6, 2, Article 9 (March 2010), 18 pages. 2010. ACM

[28] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In Proceedings of IPTPS02, Cambridge, USA, Mar. 2002

Authors

WANG Yong is an associate professor of University of Electronic Science and Technology of China (UESTC). His research interests include P2P network measurements and modelling, mobile social network privacy preserving.

BAI Yang is a graduate student of University of Electronic Science and Technology of China(UESTC). Her research interests include P2P network measurements, P2P streaming trust and reputation management and malicious detection.

HOU Jie is a graduate student of University of Electronic Science and Technology of China(UESTC). Her research interests include P2P network measurements And modelling, mobile social network privacy preserving.

TAN Yuan-wei is a graduate student of University of Electronic Science and Technology of China(UESTC). His major research direction is P2P network, privacy preserving and intersection computation.