Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 791560, 15 pages
http://dx.doi.org/10.1155/2013/791560

Research Article

Peer Selection Strategy Using Mobile Agent and Trust in Peer-to-Peer Streaming Media System

He Xu,[1,2,3] Han-Chen Huang,[4] Ruchuan Wang,[1,2,3] and Jun Dong[5]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education Jiangsu Province, Nanjing 210003, China*
[4] *Department of Leisure Management, Yu Da University, Miaoli 36143, Taiwan*
[5] *Institute of Intelligent Machines, Hefei Institute of Physical Science Chinese Academy of Sciences, Chinese Academy of Sciences, Hefei 230031, China*

Correspondence should be addressed to He Xu; xuhe@njupt.edu.cn

Received 12 August 2013; Accepted 26 September 2013

Academic Editor: Deguang Le

The aim of this study is to improve the quality of service (QoS) for peer-to-peer (P2P) streaming media system with the proposed peer selection strategy. P2P technology is promising for large scale streaming media files distribution over the Internet. As peers contribute their resources over multipeer transmission paths, it is particularly challenging to achieve consistently high streaming QoS such as low start-up delay, low jitter, low packet loss ratio, and high bandwidth assurance. Recently, there are a lot of researchers studying peer selection strategy in peer-to-peer streaming system. In this paper, a novel P2P streaming media peer selection strategy based on mobile agent and trust (MATPS) is proposed. The proposed approach is compared with bandwidth first peer selection strategy (BFPSS), GridMedia's peer selection (GridPS), trust-based peer selection strategy (TrBPS), and SmartPeerCast peer selection strategy (SPCPS), and the simulation results show that the proposed strategy improves the streaming QoS.

1. Introduction

In recent years, peer-to-peer (P2P) technology has become very popular and has drawn the attention of both academia and industry. According to [1], P2P file transfer traffic is occupying 86.7% of the total file transfer traffic. The basic principle of P2P technology is that the peer in P2P network acts as both a client and a server, which gets the resources from P2P network and also shares its own resources to other peers. P2P greatly reduces the server's load of the Client/Server (C/S) model and raises user satisfaction with the services on the network. At the beginning, applications of P2P networks were mostly focused on non-real-time data transmission such as file sharing. As the demand for multimedia services rapidly increases, especially as traditional centralized multimedia systems cannot scale well to accommodate such an explosive growth of requests, the highly scalable P2P streaming schemes naturally become a feasible solution to address the issues. Currently, Internet video broadcast and video on demand systems are mostly P2P-based architecture, such as PPLive, PPStreaming, and SopCast. Because of its advantages, P2P technology is widely used in the field of media streaming applications [2–5], having obtained a huge success. However, P2P networks are dynamic, given that any node in the network can join or leave at any time. This fact requires that special attention is devoted to the peer selection strategy, in order to minimize service failure. P2P systems build overlay networks on top of physical networks to facilitate peers' organization for content distribution. But the traditional Internet protocol (IP) is based on the best effort packet switching technology and it does not guarantee any quality of service (QoS), such as transfer delay, jitter, loss, and bandwidth [6]. P2P streaming systems require QoS performance guarantee in terms of bounded transfer delay and jitter, low packet loss, and bandwidth guarantee. This paper presents mobile agent based architecture and uses the trust mechanism to choose super-peer for peer selection in P2P streaming system. From the jitter rate, start-up delay, packet loss ratio, network load, and network scale, compared with Overcast's bandwidth first peer selection strategy (BFPSS) [7], GridMedia's peer selection (GridPS) [8], trust-based peer selection strategy (TrBPS, only using the trust scheme to select peer) [9], and SmartPeerCast's peer selection strategy (SPCPS) [10], the simulation results show that the proposed strategy improves the streaming QoS and also show that the proposed strategy is feasible to adapt to changes in network scale and has expansion and stability.

The remainder of the paper is structured as follows. In Section 2, we discuss the related work. In Section 3, firstly, we introduce the concepts of the mobile agent, and then we present the main modules of our proposed peer selection strategy based on mobile agent and super-peer selection strategy based on trust for P2P streaming media system. Section 4 is devoted to the description of the simulation environment with agent used for the assessment of the new peer selection strategy. In Section 5, simulation results are presented. Finally, the overhead of using MATPS is discussed, and concluding remarks and ideas for future work are drawn in Section 6.

2. Related Work

MATPS is related to some previous works in the context of using agent techniques in P2P network and trust-based peer selection scheme. Many research efforts were done to integrate multiagent with P2P architectures [6, 11–14]. Some recent works propose the use of P2P networks and agent technology together for content delivery network (CDN) to support large scale streaming transmission. The main idea is that as users in the same interests or regions often have the same content to get, thus those contents can be shared among users who have common interests. To support a large number of concurrent users, P2P network is used for easing the server loading, and agent is used for resource searching [11, 12]. Trust management is another critical issue in P2P system. How to build the trust relationship within P2P system is studied by many researchers [9, 15–18]. In [15], the authors divide the previous research work on building trust in P2P systems into two broad categories, that is, reputation based and trade based. Our proposed trust-based super-peer selection scheme is the combination of reputation based and trade based.

There is a huge amount of work that has been done in efficient peer selection in P2P system. A multiagents based P2P-SIP real time stream sharing system was proposed by Yang et al. [14], called MPSS. The peer selection strategy of this system does not decrease the backbone's overall throughput and is unstable for playback. Cluster based architecture was proposed in [19], where peers are clustered based on semantic routing in P2P networks. They present the scenarios under which k-means clustering is more useful than randomly selecting the cluster centroids, but the quality of the clusters formed is not guaranteed, and the effect of dimensionality reduction on query efficiency and result quality is not discussed. Huang et al. in [20] propose a network-aware P2P file sharing architecture, which has a resource provider selection algorithm that can select a new resource provider for mobile peers experiencing broken connections in wireless mobile networks, but this resource selection strategy cannot be used in the P2P streaming system. Sun et al. in [21] propose an optimization method using particle swarm optimization algorithm for neighbor selection in P2P networks. Yao et al. in [22] indicate that PPLive achieves high ISP level traffic locality spontaneously with its decentralized, latency based, and neighbor referral peer selection strategy, which provide some new insights for better understanding and optimizing the network- and user-level performance in practical P2P live streaming systems. Ciullo et al. in [23] assess what level of "network awareness" has been embedded in the most successful P2P-TV systems, namely, PPLive, SopCast, and TVAnts. They describe what parameters mainly drive the peer selection and data exchange and believe that next-generation P2P live streaming applications definitively need to improve the level of network awareness, to better localize the traffic in the network and thus increase their network friendliness as well.

Overcast [7] is designed to offer an on-demand delivery of noninteractive, bandwidth demanding, and video streaming service to a self-similar community of users through overlay network based multicast. The design goal of Overcast is to build single source multicast trees that maximize bandwidth availability from the root to each overlay node without knowing the details of the underlying network topology. Through a centralized lookup mechanism, a client wishing to receive a video streaming service finds a root node of a multicast tree that distributes a desired content. The client checks the availability of bandwidth from the root through each one of every descendant and determines which overlay nodes can offer the same amount of bandwidth as the nominal download bandwidth. The most distant node, in terms of tree hierarchy, from the root that satisfies the bandwidth requirement will be selected as the node to which the client will attach itself in the multicast tree. We named Overcast's peer selection strategy as BFPSS.

"GridMedia" is a well-know system offering P2P-based IPTV services [8]. "GridMedia" organizes the peers in unstructured overlay networks and implements a push-pull based approach to fetch the media contents from the neighbor peers [24]. It implements a block scheduling mechanism for efficient contents sharing among the peers. The neighbor peers are selected on the random basis. We named GridMedia's peer selection strategy as GridPS.

The peers of SmartPeerCast [10] are grouped into different clusters according to their uploading bandwidth. Three different ALM trees are generated based on the clusters to broadcast different quality streams to eliminate the bandwidth bottleneck between the heterogeneous peers. And the "transrating engine" is used between the peers in different ALM trees to fully utilize the leaf nodes' resources which are in the high quality ALM trees. SmatPeerCast uses the receiving peer to select the sending peer according to its uploading QoS records [10]. We named SmartPeerCast's peer selection strategy as SPCPS.

Another similar work is called P4P [25], which is a simple, light-weight framework to allow more effective cooperative traffic control between applications and network providers. Our proposed MATPS is a good application inspired by P4P. The focus in this paper is to explore peering strategy along with using mobile agent and trust. Our approach attempts to provide a scalable, robust, and reliable method ensuring QoS peer selection amongst peers in P2P streaming system. We propose a novel mobile agent based and trust value rank based peer selection in order to choose peers so as to improve stream QoS. From the jitter, start-up delay, packet loss ratio, network load, and network scale, compared with BFPSS, GridPS, TrBPS, and SPCPS, the results show that the proposed strategy is feasible.

Inspired by Raghuram and Munindar who proposed agent based service selection [26] and by Fortino and Russo who using P2P and agent technologies together for the development of content distribution networks [11], we introduce agent technology to the selection of peers for P2P streaming system. Aberer and Despotovic in [18] introduced that a simple, yet robust method that shows a solution to the problem of trust assessment based on reputation management in P2P environment. In the literature [18], they indicated that mutual trust allows agents to cooperate in a game-theoretic situation that corresponds to the repeated prisoners' dilemma and leads in the long term to an increased aggregated utility for the participating agents. And our proposed peer selection scheme is the combination of using agent and trust.

This paper makes the following main contributions for peer selection in P2P streaming media system. One, it shows how to use agent technology to support service for peer selection. Two, it proposed a novel trust-based super-peer selection strategy. Three, it proposes a novel simulation platform based on agent for evaluating peer selection. Four, it shows how the peer selections considered compare along different metrics of QoS requirements. The main benefits of our approach are as follows.

(i) Reducing the jitters of the receiving peers. By using mobile agent, we will be able to communicate to various peers to get required resources in local network. This will reduce the number of remote transmissions and interference, in turn decreasing the time needed for data transmissions in the system.

(ii) Guaranteeing the fairness of the peers' contribution by using trust value rank mechanism. Our trust scheme is the combination of reputation based and trade based. Peers that do not contribute/transmit stream enough will get low trust value, so they are periodically dropped in order to search/discover new potentially better peers. This will ensure that even with the lack of centralized information, if network environment changes, better new network topologies can be discovered.

3. P2P Streaming Peer Selection Model

P2P streaming system is required to encompass the key functions of object lookup, peer-based aggregated streaming, and dynamic adaptations to network and peer conditions. The quality of a peer depends on its availability, offered uplink bandwidth, and machine performance. We believe that a better peer selection for P2P streaming can offer a high streaming QoS. In this section, we will describe our proposed model for peer selection in P2P streaming system with using agent technology and trust mechanism.

3.1. Mobile Agent Based Peer Selection. Agent based computing is an effective paradigm for developing applications in complex domains as it supports the design and implementation of applications in terms of autonomous software entities, or "agents," which can achieve their goals flexibly by interacting with each another through high-level protocols and languages [11]. There is no accepted definition on what an agent is, but there is a set of characteristics which are commonly recognized by the community as endowing agents. There is a strong agreement that the following six characteristics form common features of agents [13].

(i) Autonomy: agents can operate without any direct human or any outside program intervention and incorporate some control over their actions and internal state.

(ii) Reactivity: agents perceive the environment and its changes and react to it as required by their goals.

(iii) Social ability: agents interact with other agents (possibly, with humans) by means of a communication language (either with the goal of cooperation and/or competition), giving rise to multiagent systems or community systems.

(iv) Proactivity: agents, in addition to the ability to react to the changing conditions in their environment, can also initiate actions in order to reach a specific goal.

(v) Adaptation/learning: agents learn by adapting their behaviors after a set of experiences for the purpose of improving the efficiency and efficacy of a given task and/or of performing new tasks which were hereto beyond their capabilities.

(vi) Mobility: agents have the ability to move from one to another node in a network.

Agents can be endowed with a mobility which enables them to roam across the network, thus potentially saving network resources, improving performance, and enabling dynamic configurability [27]. Agents are essentially autonomous programs that can take decisions based on their own history and the environment. In this paper, the mobile agent technology is exploited to make our peer selection mechanism adaptive to a dynamic P2P environment. Mobile agent has some benefits as follows [28, 29].

(1) A mobile agent can reduce the network load and latency by dispatching the mobile agents including the required services or data to remote nodes. Then, the services or data are locally executed at the remote nodes.

(2) A mobile agent can solve frequent and intermittent disconnection. Once a mobile agent is dispatched to

a destination node, it does not require direct connection with a user any more. Therefore, the mobile agent on behalf of a user is performed asynchronously and autonomously, even though a user (i.e., mobile device) is disconnected from the network.

(3) A mobile agent enables dynamic service customization and software deployment because it encapsulates some services or protocols into mobility entity.

(4) A mobile agent can adapt to heterogeneous environment and dynamic changes because it is computer- and transport-independent and also reacts autonomously according to its current execution environment.

In this paper, a framework by using mobile agent is proposed for peer selection. A peer selection architecture based on mobile agent is built and it is shown in Figure 1. Our proposal strategy not only keeps the simplicity and easy maintaining features of the P2P streaming system but also utilizes trust value rank to improve startup time delay and playback jitters performance of the P2P streaming system. As Figure 1 shows, the framework we propose in this paper is named as Mobile Agent and Trust based Peer Selection (MATPS), which is composed of three types of nodes: client, Mobile Agent Manage Server (MAMS), and peer. The client creates mobile agent to exchange information with MAMS. MAMS manages a number of peers' information (such as peer's trust value, file information). Besides the client node, the peer node also creates mobile agent to exchange information with other agents. In addition, our approach is based on a network of super-peers. A super-peer is a node in a peer-to-peer network that operates both as a server to a set of clients and as an equal in a network of super-peers. Super-peer based networks strike a balance between the inherent efficiency of centralized search and the autonomy, load balancing, and robustness to attacks provided by distributed search. Furthermore, they take advantage of the heterogeneity of capabilities (e.g., bandwidth, processing power) across peers. The basic steps of the proposed mechanism based on mobile agent are as follows.

(1) Client generates a control mobile agent (CA).

(2) CA generated in Step 1 can acquire query resources key from the client, also distribute a search agent (SA), and provide search key and search termination conditions (in case of resources are not found and the maximum number of queries sent to other peers).

(3) Then, SA generated in Step 2 migrates to remote management server MAMS. MAMS stores a large number of nodes information and keys. The selection of the MAMS can be made by the ISP or it can be automatically done using a dedicated super-peer selection. Super-peer selection strategy based on trust will be introduced in the following section.

(4) MAMS extracts the query key from SA and searches it in its own database. When finishing the inquiry, MAMS will create an information agent (IA) for the exchange of information with the SA. An IA returns the information included with the resources information, the trust value of peer information, and sorted list of peers (ranked peers), which contain attributes about whether they belong to local network.

(5) Then, SA will make a decision after receiving the information; if IA has the local peers, SA will return to this local peers; if IA returns no local peers, SA will move to ranked peers generated in Step 4.

(6) If Step 4 does not return the search peer information or reach the termination conditions, SA will return to its distributed client to announce that the resource was not found in this network.

In the P2P network, a fundamental distinction is made among unstructured and structured P2P systems for resource location. In unstructured P2P systems in principle peers are unaware of the resources that neighboring peers in the overlay networks maintain. Typically they resolve search requests by flooding techniques. In contrast, in structured P2P systems peers maintain information about what resources neighboring peers offer. Thus, queries can be directed and in consequence substantially fewer messages are needed. This comes at the cost of increased maintenance efforts during changes in the overlay network as a result of peers joining or leaving. The most prominent class of approaches to structured P2P systems is distributed hash tables (DHT) [30], for example, Chord. Figure 2 is the overlay structure of MAMS. All MAMS use a structured self-organizing P2P protocol to organize a structured P2P overlay network. Structured P2P uses DHT storage information (KeyID, PeersList), where KeyID is the identifier to find a shared resource; PeersList is the list of peers who share resources in P2P network. Request and response are the two messages between MAMS. When the index information of resources cannot be found on the local MAMS, MAMS will send a request message, while the request message format is Request (KeyID); while another MAMS receives this message, checklist is maintained by itself whether there is index information of resources needed; if it has, it will give a response information, the response message format is Reply (KeyID, PeersList), if not, put Request (KeyID) message forwarded to other MAMS; until it finds the resources needed or reaches the SA termination conditions, then it will return to its distributed client to announce that the resource was not found in this network.

There are some advantages to make use of mobile agents and also some problems to be resolved in our proposed peer selection strategy.

(1) A mobile agent can decrease the overhead of P2P system by performing peer selection, fault tolerance, and replication algorithms in a decentralized way. The peer selection mobile agents are distributed to groups. Then, they autonomously conduct a peer selection, fault tolerance, and replication algorithms in each super-peer group. Accordingly, the overhead system does not undergo the overhead any more.

(2) A mobile agent can adapt to a dynamical P2P environment. In a P2P environment, peers can join and

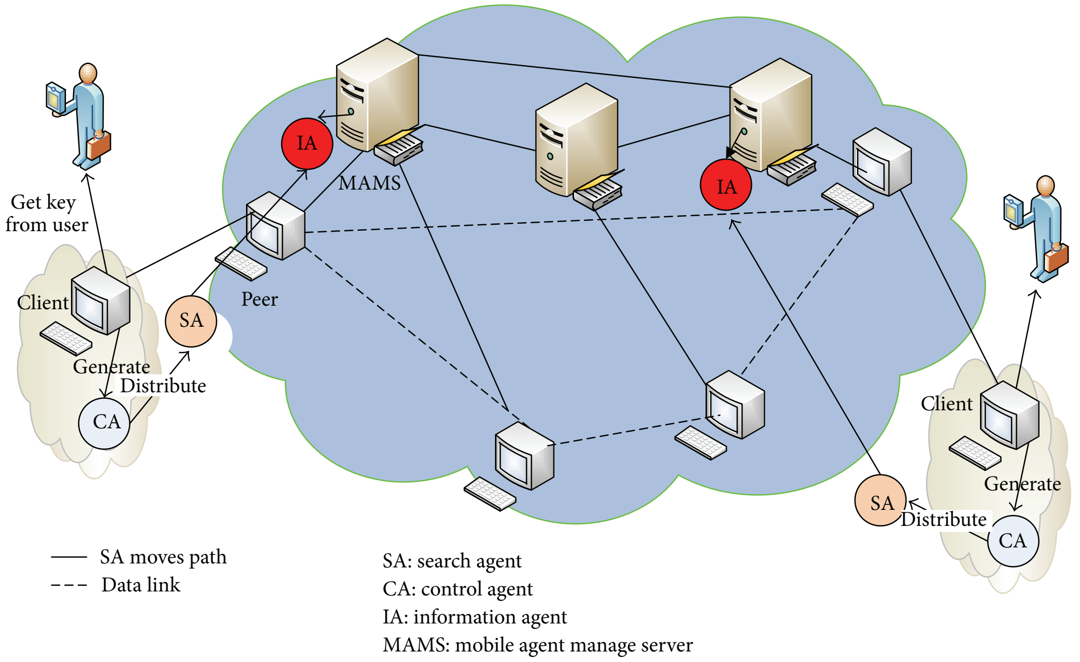


Figure 1: Mobile agent based peer selection architecture.

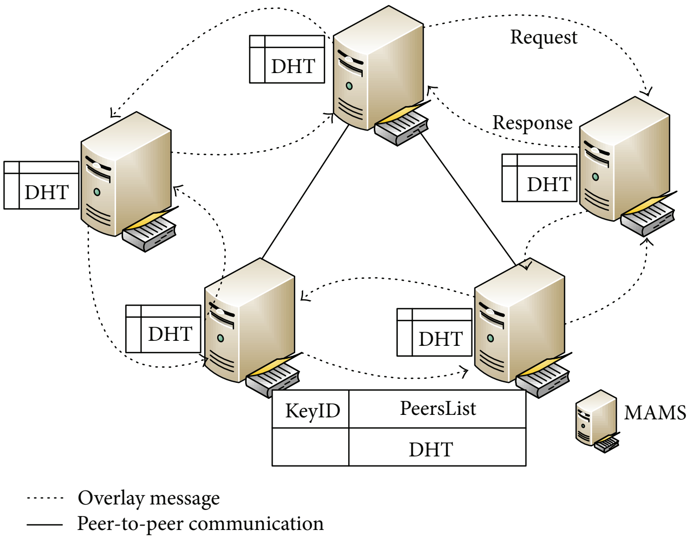


Figure 2: The architecture of MAMS's overlay.

leave at any time. In addition, they have heterogeneous properties such as capabilities (e.g., CPU, storage, or network bandwidth), location, availability, and credibility. These environmental properties change over time. A mobile agent can be performed asynchronously and autonomously coping with the changes. It can also tolerate the peers' autonomy failures by using migration and replication functionalities that a mobile agent itself provides.

(3) Various peers can be classified into different groups according to the properties of peers, and various peer selection mechanisms can be performed at a time in different groups. For example, these peers' selection mechanisms are implemented as mobile agents (e.g., peer selection mobile agents). After peers are classified into super-peer groups, the other peer selection mobile agent for a specific super-peer group is assigned according to its property. Existing P2P system, however, cannot apply various peer selection mechanisms at a time because only one peer selection mechanism is performed.

3.2. Trust Based Super-Peer Selection. Trust management is any mechanism that allows establishing mutual trust.

Reputation is a measure that is derived from direct or indirect knowledge on earlier interactions of agents and is used to assess the level of trust an agent puts into another agent. Thus, reputation-based trust management is one specific form of trust management. There are three issues that are important to address in any P2P reputation system. (1) The system should not assign any profit to newcomers. That is, reputation should be obtained by consistent good behavior through several transactions, and it should not be advantageous for malicious peers with poor reputations to continuously change their opaque identifiers to obtain newcomers status. (2) The system should have minimal overhead in terms of computation, infrastructure, storage, and message complexity. (3) The system should be robust to malicious collectives of peers who know one another and attempt to collectively subvert the system. An important example of successful reputation management is the online auction system eBay [31]. In eBay's reputation system, buyers and sellers can rate each other after each transaction, and the overall reputation of a participant is the sum of these ratings over the last 6 months. Inspired by eBay, we define that time is a factor influencing our trust value.

In the P2P streaming system, super-peer not only needs to transfer its own streaming media data but also to help other streaming media client to access resources and support information search and data forwarding, which require that super-peer has a high performance requirements, so the peer's performance (e.g., CPU computing power, memory size, disk space, available upload bandwidth, etc.) is a factor influencing our trust value in our proposed trust based strategy.

We propose a mechanism for super-peer selection strategy based on the trust management. In order to address above three problems in P2P system, we define that the trust value of each peer is calculated as follows:

$$Ti = C * \alpha T_{\text{time}} + \beta * \sum_{i,j \in N} Tj * Tji - \gamma D. \quad (1)$$

In the formula, Ti is the trust value of streaming media peer i, C is the peer's capacity points, including peer's CPU computing power, memory size, disk space, available upload bandwidth, and whether it has the public network IP. And C is divided into three grades, that is, excellent, middle, and poor, each level is divided into 1000 points, 500 points, and 100 points, separately. T_{time} is the peer's online time duration points; while T_{time} is longer, their trust value will be higher. In our experiment, we set that T_{time} will be added by 1 points when the peer is online at 1 hour. D is the penalty points; that is, when the resources peer denied service to other peers, D's points will be reduced. In our experiment, we set that D will be reduced by 10 points when the peer refuses to support service once. In a real system, we can cut down the D's points by using antivirus software to detect if the downloaded file is a virus. This mechanism ensures that the peer with less contributing to others will be punished fairly. $\sum_{i,j \in N} Ti * Tji$ is the recommended point, Tj indicates that the node j's trust value, and Tji indicates trust value that the node j trust the node i; when the node j support service for the node i once, the Tji will be added; in our experiment we set 1 point once. α and β are the factors for equity, and γ is the penalty factor. α and β are needed to be set to adequate values according to the strategy of trust and the characteristics of P2P streaming system. The Ti is updated in MAMS 1 hour once or when the stream is transferred completely between node i and j. The recommended points are obtained as follows: peer A establishes trust of peer C through peer B's recommended ($T_{AC} = T_B * T_{AB}$); while there are n recommenders, $T_{AC} = 1/n \sum_{i=1}^{n} T_B * T_{AB_i}$. The initial point of the recommended points is set as 1 and the initial point of the penalty points is set as 0.

Table 1: Parameter values description.

Parameter	Emphasis
α	While the online time is longer, the trust value of peer is bigger.
β	The bigger the value, the bigger the weight of the recommended trust among peers.
γ	When the peer denies serving other peers or fails to exchange resources, the peer's trust value will be reduced.

In the formula, for setting the size of different parameters, the trust measure metrics are different, as specified in Table 1. We chose the maximum trust value of peers as the best super-peer in the network, that is, as MAMS. Mobile agent based peer selection in Step 4 returns a list of sorted peers according to the trust value of peers, and the peer, which is standing in the front, is the priority to be chosen for resource links.

The benefits of using trust management to select super-peer are (1) the trust scheme is the combination of reputation based and trade based. Peers that do not contribute/transmit stream enough will get low trust value, so they are periodically dropped in order to search/discover new potentially better peers. This will ensure that even with the lack of centralized information, if network environment changes, better new network topologies can be discovered. (2) From the given formula (1), we can see that the trust value is difficult to get but easy to lose, which is guaranteeing the fairness of the peers' contribution by using trust value rank mechanism. The formula has no complicated iterative calculation and has good convergence of computing and scalability. (3) The trust mechanism is an incentive mechanism to make peer online longer and share their resources to other peers. When the online time is longer, the trust value of peer is bigger.

4. Experimental Environment

To prove and verify the MATPS performance, the OverSim [32] simulator is used to run and check the proposed MATPS protocol implementation. The experiment's target is to show that MATPS is a stable solution with good performance for improving QoS for P2P streaming system.

Peer-to-peer (P2P) systems are characterized by many annoying real life complexities, such as a high turnover of peers, download and connection failures, large numbers of stochastically behaving peers. A simulation is an attempt to model a system in order to study it scientifically [33]. Analytical approaches are being applied to P2P systems, but they often require simplifying assumptions that break down under real world conditions. In addition, the intention of many P2P systems is to scale to large numbers of peers; and testing that scalability in the actual system may be impractical. As a result P2P simulation studies for P2P network have been growing in numbers, many simulators have been developed for simulating P2P, such as PlantSim [34], PeerNS [35], PeerSim [36, 37], P2PAM [38], P2PStutio [39], SimGrid [40], ChunkSim [41], OPSS [42, 43], and OverSim [32]. We choose OverSim because it is an open-source overlay and P2P network simulation framework for the OMNeT++ [44] simulation environment. The simulator contains several models for structured (e.g., Chord, Kademlia, and Pastry) and unstructured (e.g., GIA) P2P systems and overlay protocols.

Experimental environment is set up in the server DELL business desktop PC OptiPlex 780 and the simulation runs on it. The server's configurations are as follows: CPU is Intel Core 2 Duo E7500 2.93 GHz, RAM is 2 GB, the operating system is Windows XP, using the OMNeT++, INET [45], and OverSim simulation platform. OMNeT++ is an object-oriented modular discrete event network simulation framework. It has a generic architecture, so it can be used in various problem domains. The INET underlay model is derived from the INET framework of OMNeT++, which includes simulation models of all network layers from the MAC layer. It is useful for simulations of complete backbone structures. The INET underlay model includes simulation models for all network layers. OverSim is a new simulation framework based on OMNeT++. It has been developed in the ScaleNet project at the University of Karlsruhe (Germany). It can simulate up to 100,000 nodes and its design is thoroughly documented. The main reasons for using this tool include the ease of use, a highly flexible and modular architecture, and an open-source code base. Both OMNeT++ and OverSim are built as hierarchical architectures. Simulation is done using the chord protocol of OverSim.

In order to verify the proposed peer selection strategy based on mobile agent and trust (MATPS), in our simulation experiment we had to extend P2P protocol based on mobile agent and the improvement module structure of OverSim based on agent is shown in Figure 3. In Figure 3, agent runtime environment is to support mobile agent running on OverSim; agent creation module is primarily responsible for creating the content distribution agent, content search agent, and content download agent; agent migration module will create the agent for moving to other peers; P2P protocol simulator layer is according to OverSim's P2P network protocol to organize peers and routing, and it is used as the underlying network; P2P application layer mainly provides peers login, resources release, and download functions.

The details of how peers distribute content and download content will be discussed in the following descriptions. Figure 4 is the process diagram of content distribution and Figure 5 is the process diagram of content download.

The steps of content distribution are as follows.

(1) Client creates a content distribution agent (DA).

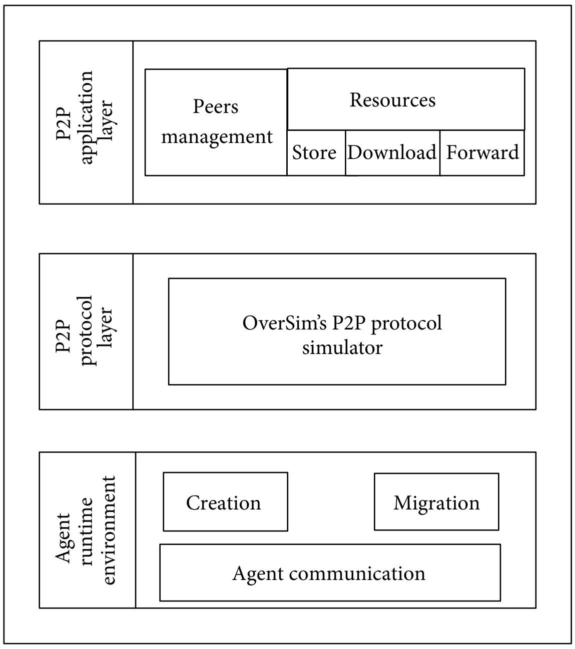


Figure 3: Mobile agent based on OverSim's P2P protocol layer.

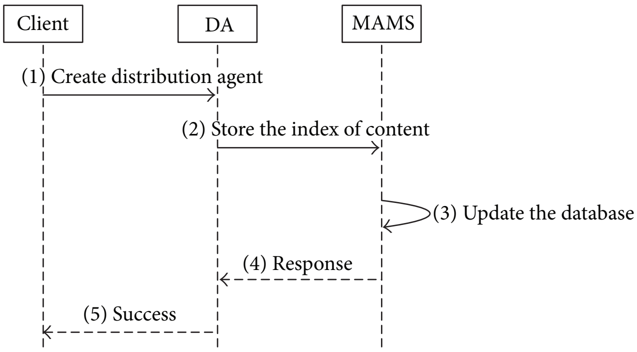


Figure 4: Diagram of content distribution.

(2) DA extracts the index information of shared resources, such as file name, file size, and file type.

(3) DA moves to MAMS. MAMS will create an information agent (IA) for the exchange of information with the DA. And IA will extract the index information from DA; if the index of content does not exist in database, the database will update it into database.

(4) IA will give a response message to inform DA whether the content is distributed or not.

(5) If the content is distributed, then DA will give a success message to its origin client.

The steps of content download are as follows.

(1) Client creates a content download agent (DownA).

(2) DownA moves to MAMS.

(3) MAMS will create an information agent (IA) for the exchange of information with the DownA. IA will

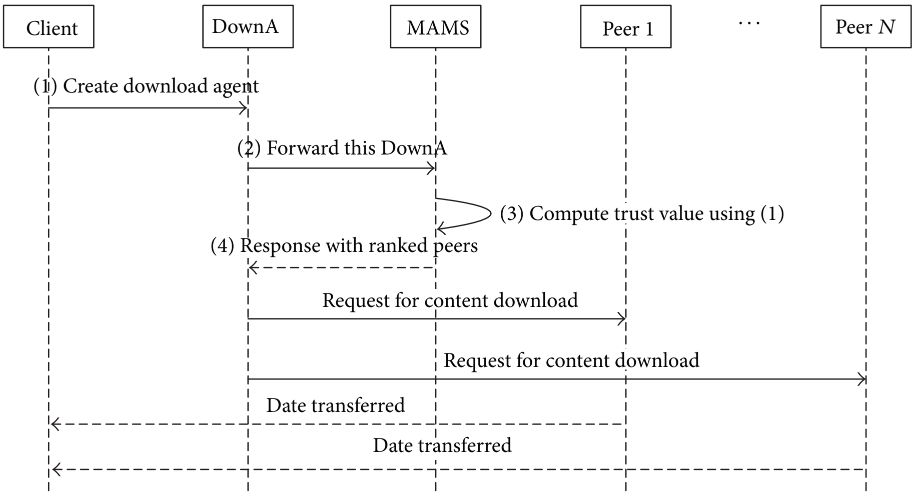

Client
DownA
MAMS
Peer 1
· · ·
Peer N
(1) Create download agent
(2) Forward this DownA
(3) Compute trust value using (1)
(4) Response with ranked peers
Request for content download
Request for content download
Date transferred
Date transferred

Figure 5: Diagram of content download.

extract the download file KeyID information from DownA, and then MAMS will search for it in its own database. When finishing the search, MAMS will give a list of peers who have the resource and compute each peer's trust value using (1). After that IA will respond by message to DownA with ranked peers list according each peer trust value, which also contains attributes about whether it belongs to local network;

(4) DownA receives the ranked peers and requests those peers for downloading content.

In the experiments, we have set the request issuing rate between 0 and 100 per minute, the nodes churn using the Pareto model of OverSim (OverSim provides several models for generating churn, which includes a lifetime based churn model supporting different distribution functions, such as Weibull, Pareto, or Exponential. Alternatively, a scenario or trace file containing join and leave events can be used to model churn behavior. It is possible to use more than one churn generator at the same time to simulate groups of nodes with different churn behaviors. For each churn generator different node configurations and overlay parameters can be specified, which allows to easily generate complex scenarios with heterogeneous node behavior.), download bandwidth of each node is 64 kbps–3 Mbps, the maximum upload bandwidth is 1 Mbps, the overall system has 500 media files, media playback rate is 256 kbps, the average length of each file is 1800 seconds, and each node has one of the 10~20 media files randomly. After the peers join the system, they can make player operation anytime. The number of nodes that are allowed to perform on-demand operations is controlled by a software script. Detailed simulation parameters are shown in Table 2. Simulation results and performance analysis are shown in the following section.

5. Performance Evaluation

The following section first finds the optimal values for α, β, and γ, then evaluates the effectiveness of the proposed method with BFPSS, GridPS, TrBPS, and SPCPS, and gives comparisons of the impact of their jitter, start-up delay, packet loss ratio, network load, and network scale.

Table 2: Simulation parameters.

Category	Value
Peers' download bandwidth	64 kbps~3 Mbps
Peers' upload bandwidth	0~1 Mbps
Number of media files	500
Play rate	256 kbps
The average length of media files	30 minute
Frequency of queries issued	0~100 query/minute
Peers	10000
Peers loss	Pareto
Simulation duration	500 minutes

5.1. Finding Optimal Values for α, β, and γ.

We have described earlier how to determine the best super-peer using (1). We conduct simulations for different values of α, β, and γ (ranges from 0.1 to 0.9) to compute the overall streaming media client's traffic throughput and the overall packet drop ratio for each set of values.

A brief description of the results for different sets of parameters is preset in Table 3. The network topology of simulation experiment for finding optimal α, β, and γ is based on Figure 6 with random nodes attached to super-peers. Link capacities are also chosen randomly on uplink and download links (uplink is from 0 to 1 Mbps, and download is from 64 kbps to 3 Mbps). There are five machines in the LAN, which are running a number of popular P2P applications, such as PPLive, Bittorrent, eMule, and Xunlei. Because OMNET++ is a discrete event simulation environment, in order to simulate large-scale P2P network, we use this small LAN to run OMNET++ simultaneously. In current network, we set up a flow collector, which uses NETMATE [46] software to capture packets and calculate the client's traffic

Table 3: Parameter values.

	$\alpha = 0.1$ $\beta = 0.1$ $\gamma = 0.8$	$\alpha = 0.2$ $\beta = 0.4$ $\gamma = 0.6$	$\alpha = 0.3$ $\beta = 0.5$ $\gamma = 0.7$	$\alpha = 0.4$ $\beta = 0.4$ $\gamma = 0.7$	$\alpha = 0.5$ $\beta = 0.7$ $\gamma = 0.6$	$\alpha = 0.6$ $\beta = 0.6$ $\gamma = 0.8$	$\alpha = 0.7$ $\beta = 0.8$ $\gamma = 0.9$	$\alpha = 0.8$ $\beta = 0.9$ $\gamma = 0.4$	$\alpha = 0.9$ $\beta = 0.9$ $\gamma = 0.7$
Traffic throughput (kbps)	923.57	901.83	1028.36	**1289.25**	1123.01	1127.63	999.34	972.93	1212.49
Packet drop ratio (%)	6.22	6.91	4.06	**3.25**	5.68	7.02	8.52	4.90	5.11

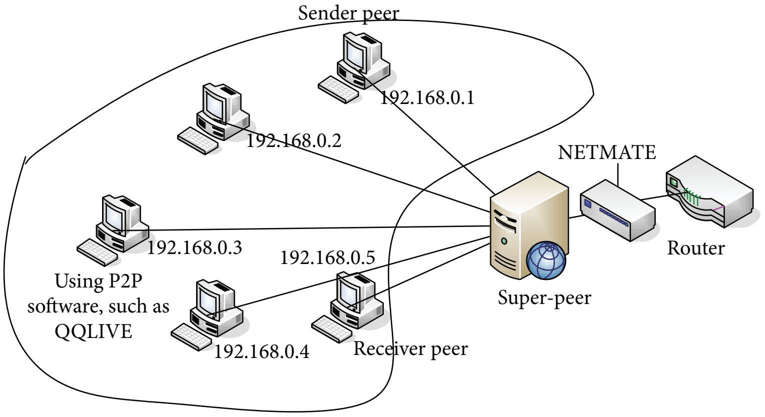


Figure 6: The network topology of experiment for finding optimal α, β, and γ.

throughput and the overall packet drop ratio according to its source address or destination address.

The results presented in the table are averages over 729 runs of simulations and only some of the results are presented due to the layout restrictions of the paper. We noticed that the streaming client (Receiver) receives maximum traffic throughput and low packet drop ratio for the set of values ($\alpha = \beta = 0.4$, and $\gamma = 0.7$) for calculating the trust value Ti. Thus, we give a weight of 0.4 to α and β, and 0.7 to γ.

5.2. The Jitter Performance Results. In the experiments, different P2P clients are used as separate MATPS, BFPSS, GRIDPS, TrBPS, and SPCPS simulator machines according to Figure 6. The jitter and start-up delay are the two performance parameters of the P2P streaming application. Jitter is introduced into the traffic as a result of unexpected delay experienced by the flow on network routers [47]. Assuming that the packet arrives at the receiver node at time t, if the expected arrival time of packet is e, $t - e$ is the jitter. Figure 7 shows the jitter caused by the node failure rate changes in the case of the request rate of 60 (query/min). From Figure 7, we can see that in the beginning of the system start-up stage, the five strategy's jitter rates are unstable, and after 10 minutes, the jitter rates are stable, but the jitter of MATPS and SPCPS strategies remains at a low level, the jitter of BFPSS, GridPS, and TrBPS strategies changes severely. MATPS strategy jitter rate is 0.45% lower than SPCPS strategy. As BFPSS strategy is the bandwidth first used by the selection strategy and does not take into account the stability of the node, GridPS uses random peer selection, TrBPS uses only trust based peer selection, while the MATPS strategy is comprehensive in consideration of the node bandwidth, performance, and so forth and select online time as a standard for peer choice, so the longer time the node in the network takes, the probability of being selected to support service for other peers is increasing. So the average jitter rate of the MATPS is lower than that of the other peer selection strategies.

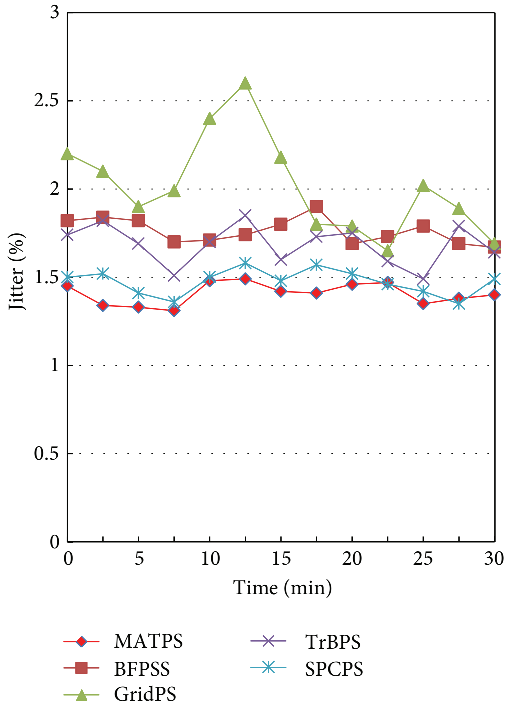


Figure 7: The jitter rates change over time.

5.3. The Start-Up Delay Performance Results. The start-up delays incurred by the clients and the number of channels

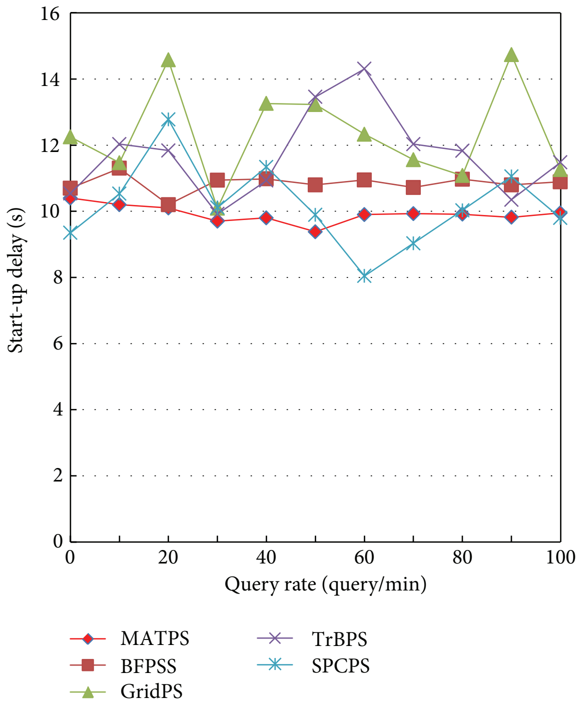


Figure 8: The start-up delay of different request rates.

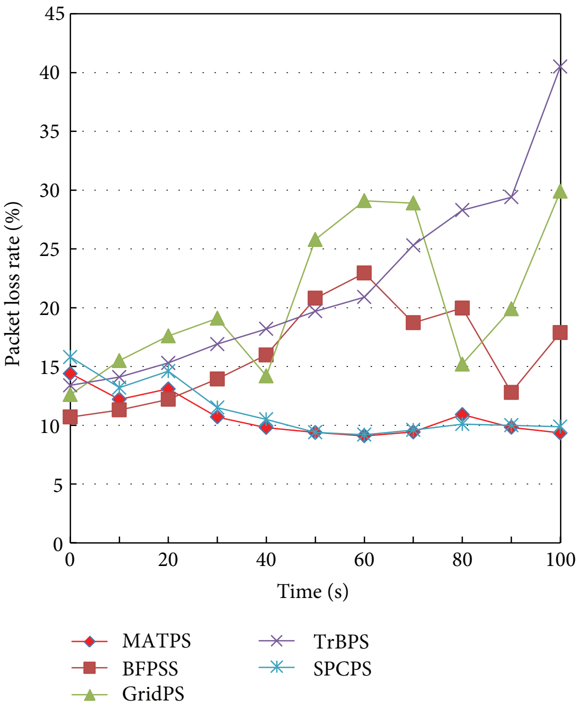


Figure 9: The packet loss ratio performance.

(bandwidth) needed to serve all the requests [48]. In P2P streaming media system, resource query delay and streaming media transmission delay are the main factor of affecting the client start-up delay. Figure 8 is the client start-up delay as the node request rate variation diagram, while the simulation platform to run for some time and the number of nodes in P2P network is stability. From Figure 8, we can see that at the beginning of the playing among 10 seconds, because resources are distributed on each node irregularity, and the number of nodes join P2P network is not enough, the query requests sent but most of resources are not find, which led to a larger search resources delay and influences the client to play. With the increasing number of requests and the stabilized network size, a large number of requests by the network are responded. At the same time, the proposed MATPS selects high bandwidth and local node as a resource provider node, which greatly reduces the network transmission time of media resources; thus, the MATPS has the better start-up delay time performance.

5.4. The Packet Loss Ratio Performance Results. The packet loss ratio is another key performance metric besides the start-up delay in the P2P streaming application [10]. The packet loss will cause the receiving peer to have playback jitters. The playback jitter becomes worse with more packets lost in the receiving peer when necessary stream frames are dropped. In our experiment, (2) as defined below is used to measure the packet loss ratio at the receiving peer. Equation (2) calculates the receiving peer's packet loss ratio by dividing n with N:

$$\text{packet loss ratio} = \frac{n}{N}. \tag{2}$$

In (2), n is defined as the number of the total dropped packets, and N is defined as the total expected receiving packets in the duration of a stream connection session. The following gives the value of N:

$$N = \frac{T_w}{T_p}. \tag{3}$$

In (3), T_w presents the total time that the receiving peer is watching and staying with the stream, and T_p is the playback duration for a packet. In our experiments, T_w is a fixed number in the receiving peer and it always equals thirty minutes in every experiment. T_p is also a constant value in our experiments because it is assumed that the stream is encoded in CBR and the stream is divided to packets with the fixed playback duration in the source node. The packets have variable data size in different quality streams, but they always have the same playback duration. According to the definition of T_w and T_p, it is concluded that N is also a constant value. Thus, the packet loss ratio defined in (3) depends on the value of n.

Figure 9 shows the packet loss experiment results for the five peer selection strategies. MATPS has smaller packet loss ratio than that of the other four strategies. Because the results are based on PC simulation, in a real deployed system the MATPS still has the chance to get lower pack loss ratio performance than the results shown in Figure 9 by increasing the receiving peer's buffer size.

5.5. Network Load. From Figure 10, we can see that with the rate request increasing the MATPS and SPCPS strategy backbone utilization decreased, while the other three

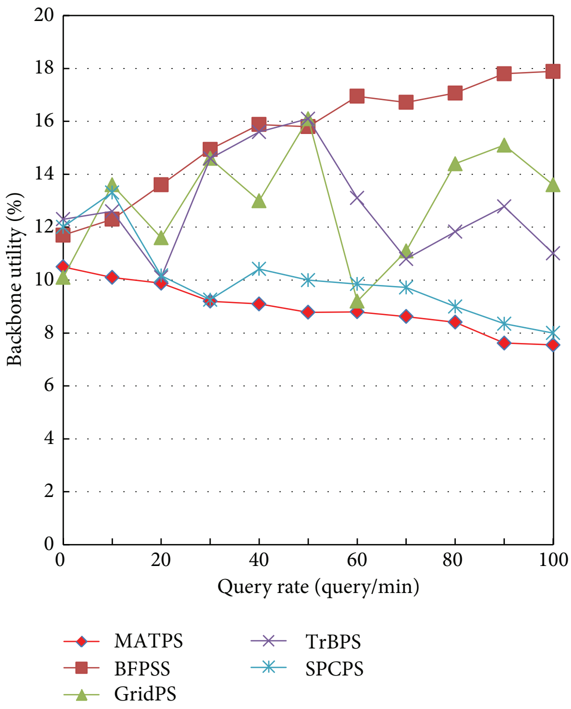


Figure 10: The backbone utilization rates of different requests.

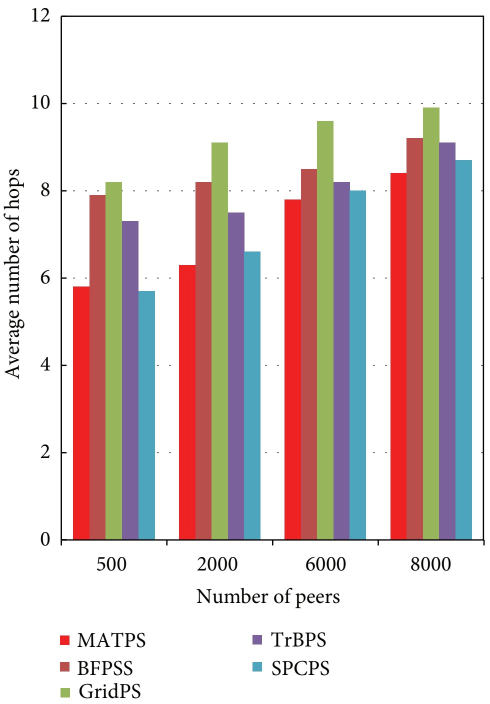


Figure 11: The average number of hops compared with network scale.

strategies' backbone usage increased gradually. In the request rate of 100 (query/min) the backbone utilization of MATPS strategy has fallen to 7.5%, while the backbone utilization rate of BFPSS, GridPS, and TrBPS strategy has increased. Due to the number of requests increasing, more and more traffic of the media data needs to be transferred through the upper network in BFPSS strategy, GridPS, and TrBPS, which makes backbone network usage increasing. While in MATPS strategy, because it uses the way of the local node based on trust preferred, when a growing number of local node requests the same content, which also means more and more nodes become nodes of local and follow-up requests for resource providers, so the more the requester, the more the local resources, and thus reduce the local nodes to exchange resources through the upper network, continuously reduce the backbone network usage. The MATPS makes lower backbone utility than SPCPS.

5.6. The Impact of Network Scale. The maximum number of hops can specify how many times a query is allowed to be transferred in P2P streaming system, which determines how much the network will be flooded by a single query. Generally, the larger the network is, and the greater the cost is. By measuring and comparing the number of hops a query sent to different nodes, we can analyze the performance of the selection algorithm. The network scale is carried out in four conditions: 500, 2000, 6000, and 8000. The average number of hops results is shown in Figure 11. From Figure 11, with the network size increasing, in MATPS node selection algorithm, the hops change little, while in the other four strategies the average number of hops rises. This is because MATPS node selection strategy is based on trust mechanism, and it takes into account local resources, network bandwidth, node performance, and other factors, which can be a good adaptive to change in network size. With the increasing of the network scale, the adaptive choice peers of optimized node information make inquiries in localization, so the average hop remains more stable than other strategies.

Figure 12 shows the average jitter rates of the five strategies with the increase of the network scale. In general, the average jitter rates of the five strategies decrease with the increase of the peers. Figure 13 shows the average start-up delays of the five strategies with the increase of the network scale. In general, the average start-up delays of the five strategies decrease with the increase of the peers. Figure 14 shows the average packet loss rates of the five strategies with the increase of the network scale. In general, the average packet loss rates of the five strategies increase with the increase of the peers. As shown in Figure 14, the average packet loss rates of BFPSS, GridPS and TrBPS increase very quickly. On the contrary, the average packet loss rates of MATPS and SPCPS go up slowly. The average packet loss rate of MATPS is 1 s lower than that of SPCPS strategy. Figure 15 shows the average backbone utilities of the five strategies with the increase of the network scale. In general, the average backbone utilities of the five strategies increase with the increase of the peers. As shown in Figure 15, the average backbone utilities of BFPSS, GridPS and TrBPS increase very quickly, while the average backbone utilities of MATPS and SPCPS go up slowly. The average backbone utility of MATPS is 1% lower than that of SPCPS strategy.

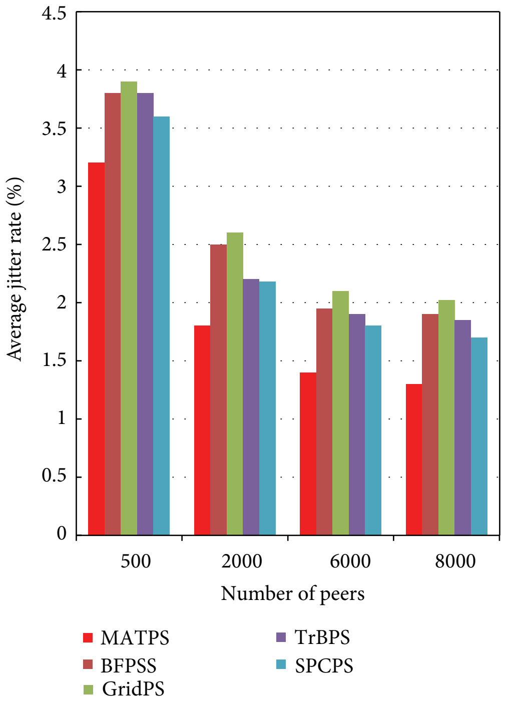


Figure 12: The average jitter rate compared with network scale.

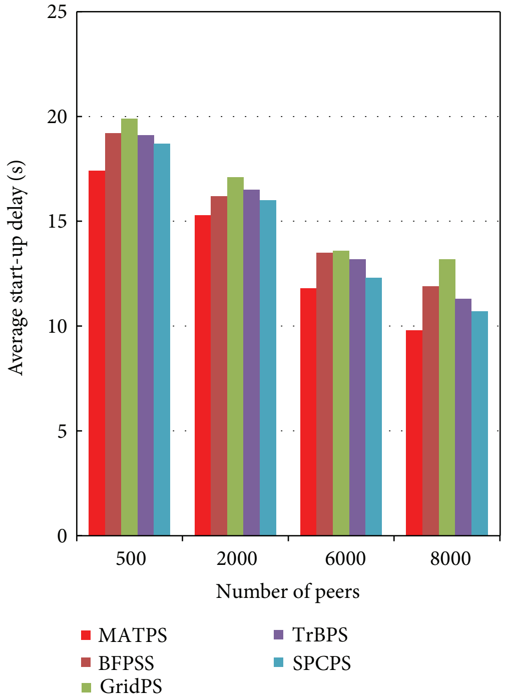


Figure 13: The average start-up delay compared with network scale.

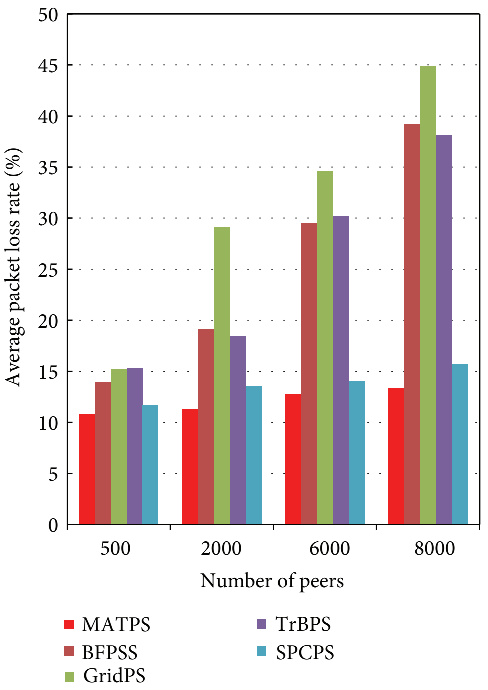


Figure 14: The average packet loss rate compared with network scale.

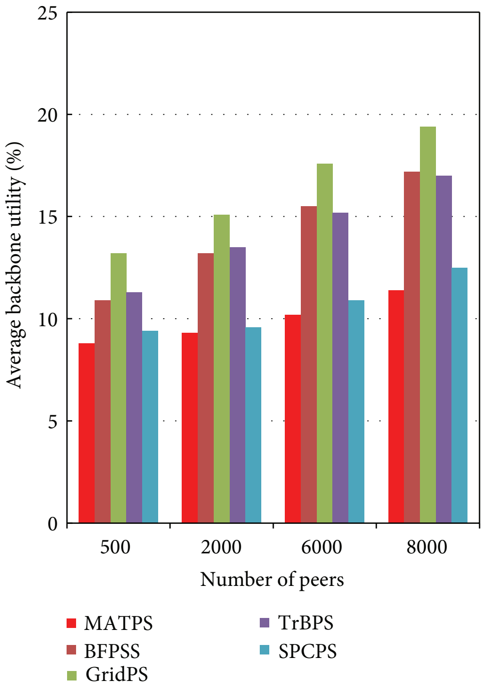


Figure 15: The average backbone utility compared with network scale.

From Figures 11–15, we can find that MATPS strategy has better average indicators than other four strategies, and we can make a conclusion that the proposed MATPS strategy can improve the streaming QoS effectively. In addition, we also select some other simulation parameters to evaluate the performance, and then we found that the performance indicators of other parameters' values of alfa, beta, and gamma are worse than those of optimal parameters, and because we have already done more experiments for finding the optimal values for α, β, and γ in Section 5.1, we have not given the results of the other performance indicators with different simulations parameters of alfa, beta, and gamma.

Table 4: Message's overhead of using MATPS.

	Mobile agent		Trust value	
	Use	No	Use	No
Total packets sent	134235	129894	145069	142872
Message's overhead	3.34%		1.54%	

6. Discussion, Conclusions, and Future Work

6.1. Discussion

Message's Overhead of Using MATPS. When peers login or quit, and the trust value of each peer is updating, refresh messages are distributed through creating agents. As for the message's overheads with using mobile agent and trust or not, from Table 4 we can see that in our simulation experiment, the message's overhead of using mobile agent is negligible, that is, only a percentage of 3.34 for using mobile agent and 1.54 for updating trust value.

6.2. Conclusions. In this paper, we have presented a peer selection strategy for improving the QoS requirements of P2P streaming system. The basic idea is to use agent to select peer and to choose peer as super-peer which has high trust value. Also a new architecture to simulate P2P network is presented, which uses agent technology to OverSim platform. The proposed MATPS scheme aims at improving the QoS of P2P streaming system. Simulation results show that our scheme can effectively select good peer to support sources. Our experiments results also show that the Internet backbone utilization of the P2P streaming system has been decreased and the QoS requirements of P2P streaming system can be improved comparing with other peer selection strategies.

6.3. Future Work. Future research can proceed in the following directions. First, we are developing the P2P streaming system implementing MATPS. How to design the system architecture of such a system and how to enhance the current available streaming protocol to support MATPS are both challenging work. Second, in this paper we point out that the value of α, β, and γ is optimized set through experiment, but if these parameters are really optimal in all use cases we need to do more experiments. A technique for selecting these parameters automatically by the system is also worth further study. Third, in real life, some contents are typically very popular and some are not; the cache technology is needed to be integrated with MATPS in the future. Finally, the client behind the firewall may not be able to allow the outside clients to get the stream, and this paper did not consider the security issues of mobile agent. We will study mechanisms to overcome these questions in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The subject is sponsored by the National Natural Science Foundation of P. R. China (nos. 61170065, 61103195, 61203217, 61202355), the Natural Science Foundation of Jiangsu Province (BK2011755, BK2012436, and BK20130882), Scientific & Technological Support Project of Jiangsu Province (nos. BE2011189, BE2012183, and BE2012755), Scientific Research & Industry Promotion Project for Higher Education Institutions (JHB2012-7), Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institutions (12KJB520009), Doctoral Fund of Ministry of Education of China (20103223120007 and 20123223120006), A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), and Talent Introduction Foundation of Nanjing University of Posts and Telecommunications (NY213034). The authors would like to thank the editors and the anonymous reviewers, who provided insightful and constructive comments for improving this paper.

References

[1] "Cisco Systems, Inc., Cisco Visual Networking Index: Forecast and Methodology, 2012–2016," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf.

[2] A. Bikfalvi, J. García-Reinoso, I. Vidal, and F. Valera, "A peer-to-peer IPTV service architecture for the IP Multimedia Subsystem," *International Journal of Communication Systems*, vol. 23, no. 6-7, pp. 780–801, 2010.

[3] N. Efthymiopoulos, S. L. Tompros, A. Christakidis, K. Koutsopoulos, and S. Denazis, "Enabling live video streaming services realization in telecommunication networks using P2P technology," *International Journal of Communication Systems*, vol. 24, no. 10, pp. 1354–1374, 2011.

[4] C. Lin, "Enhancing P2P live streaming performance by balancing description distribution and available forwarding bandwidth in P2P streaming network," *International Journal of Communication Systems*, vol. 24, no. 5, pp. 568–585, 2011.

[5] M. Amad, A. Meddahi, D. Aïssani, and G. Vanwormhoudt, "GPM: a generic and scalable P2P model that optimizes tree depth for multicast communications," *International Journal of Communication Systems*, vol. 25, no. 4, pp. 491–514, 2012.

[6] A. Radu and G. Baudoin, "Intelligent agents for quality of service evaluation in multimedia services," in *Proceedings of the IADIS International Conference (WWW/Internet '03)*, pp. 1219–1224, Algarve, Portugal, 2003.

[7] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole Jr., *Overcast: Reliable Multicasting with an Overlay Network*, USENIX, Berkeley, Calif, USA, 2000.

[8] L. Zhao, J. Luo, J. Luo et al., "Gridmedia: a practical peer-to-peer based live video streaming system," in *Proceeding of IEEE 7th Workshop on Multimedia Signal Processing (MMSP '05)*, November 2005.

[9] S. Marti and H. Garcia-Molina, "Taxonomy of trust: categorizing P2P reputation systems," *Computer Networks*, vol. 50, no. 4, pp. 472–484, 2006.

[10] W. Y. Wang and Y. W. Chen, "SmartPeerCast: a Smart QoS driven P2P live streaming framework," *Multimedia Tools and Applications*, vol. 54, no. 2, pp. 445–471, 2011.

[11] G. Fortino and W. Russo, "Using P2P, GRID and Agent technologies for the development of content distribution networks," *Future Generation Computer Systems*, vol. 24, no. 3, pp. 180–190, 2008.

[12] V. Gorodetskiy, O. Karsaev, V. Samoilov, and S. Serebryakov, "Agent-based service-oriented intelligent P2P networks for distributed classification," in *Proceedings of the International Conference on Hybrid Information Technology (ICHIT '06)*, pp. 224–233, November 2006.

[13] G. Moro, A. M. Ouksel, and C. Sartori, "Agents and Peer-to-Peer computing: a promising combination of paradigms," in *Proceedings of the 1st International Workshop: Agents and Peer-to-Peer Computing (AP2PC '02)*, Lecture Notes in Artificial Intelligence, pp. 1–14, Springer, Berlin, Germany, July 2002.

[14] D. G. Yang, H. Wang, C. R. Wang, and Y. Gao, "MPSS: A multi-agents based P2P-SIP real time stream sharing system," in *Proceedings of the 9th Pacific Rim International Workshop on Multi-Agents (PRIMA '06)*, vol. 4088 of *Lecture Notes in Computer Science*, pp. 398–408, Springer, Berlin, Germany, 2006.

[15] B. Zhu, S. Jajodia, and M. S. Kankanhalli, "Building trust in peer-to-peer systems: a review," *International Journal of Security and Networks*, vol. 1, no. 1-2, pp. 103–112, 2006.

[16] X. Jin, Q. Xia, and S.-H. G. Chan, "Building a monitoring overlay for peer-to-peer streaming," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, December 2006.

[17] L. Xiong and L. Liu, "PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[18] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM '01)*, pp. 310–317, Atlanta, Ga, USA, November 2001.

[19] M. Eisenhardt, W. Müller, A. Henrich, D. Blank, and S. El Allali, "Clustering-based source selection for efficient image retrieval in peer-to-peer networks," in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM '06)*, pp. 823–828, San Diego, Calif, USA, December 2006.

[20] C. M. Huang, T. H. Hsu, and M. F. Hsu, "Network-aware P2P file sharing over the wireless mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 1, pp. 204–210, 2007.

[21] S. Sun, A. Abraham, G. Zhang, and H. Liu, "A particle swarm optimization algorithm for neighbor selection in peer-to-peer networks," in *Proceedings of the 6th International Conference on Computer Information Systems and Industrial Management Applications (CISIM '07)*, pp. 166–170, Elk, Poland, June 2007.

[22] L. Yao, G. Lei, L. Fei, and C. Songqing, "A case study of traffic locality in Internet P2P live streaming systems," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS '09)*, pp. 423–432, Montreal, Canada, June 2009.

[23] D. Ciullo, M. A. Garcia, A. Horvath et al., "Network awareness of P2P live streaming applications: a measurement study," *IEEE Transactions on Multimedia*, vol. 12, no. 1, pp. 54–63, 2010.

[24] M. Zhang, J. G. Luo, L. Zhao, and S. Yang, "A Peer-to-Peer network for live media streaming- using a push-pull approach," in *Proceedings of the 13th Annual ACM International Conference on Multimedia (MULTIMEDIA '05)*, pp. 287–290, 2005.

[25] H. Y. Xie, A. Krishnamurthy, and L. Popkin, "P4P: provider portal for (P2P) applications," in *Proceedings of the ACM SIGCOMM Conference*, 2008.

[26] R. M. Sreenath and M. P. Singh, "Agent-based service selection," *Web Semantics*, vol. 1, no. 3, pp. 261–279, 2004.

[27] M. Luck, P. McBurney, and C. Preist, "A manifesto for agent technology: towards next generation computing," *Autonomous Agents and Multi-Agent Systems*, vol. 9, no. 3, pp. 203–252, 2004.

[28] D. Wong, N. Paciorek, and D. Moore, "Java-based mobile agents," *Communications of the ACM*, vol. 42, no. 3, pp. 92–102, 1999.

[29] S. Choi, M. Baik, and C. Hwang, "Location management & message delivery protocol in multi-region mobile agent computing environment," in *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS '04)*, pp. 476–483, March 2004.

[30] "Kademlia: a Peer-to-Peer information system based on the XOR metric," in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds., vol. 2429 of *Lecture Notes in Computer Science*, pp. 53–65, 2002.

[31] eBay, http://www.ebay.com/.

[32] I. Baumgart, B. Heep, and S. Krause, "OverSim: a flexible overlay network simulation framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI '07)*, pp. 79–84, Anchorage, Alaska, USA, May 2007.

[33] A. M. Law, W. D. Kelton, and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw-Hill, New York, NY, USA, 1991.

[34] J. Pujol-Ahulló and P. García-López, "PlanetSim: an extensible simulation tool for peer-to-peer networks and services," in *Proceedings of the IEEE 9th International Conference on Peer-to-Peer Computing (P2P '09)*, pp. 85–86, Piscataway, NJ, USA, September 2009.

[35] S. K. Dhurandher, S. Misra, M. S. Obaidat, I. Singh, R. Agarwal, and B. Bhambhani, "Simulating peer-to-peer networks," in *Proceedings of the 7th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '09)*, pp. 336–341, Piscataway, NJ, USA, May 2009.

[36] M. Jelasity, A. Montresor, G. P. Jesi et al., "The Peersim simulator," 2010, http://peersim.sourceforge.net/.

[37] E. J. L. Lu, Y. T. Jang, and G. W. Hwang, "PeerSim Cooker: a guide for PeerSim," in *Proceedings of the International Computer Symposium (ICS '08)*, pp. 267–272, Tamkang University, New Taipei, Taiwan, 2008.

[38] M. Agosti, F. Zanichelli, M. Amoretti et al., "P2PAM: a framework for peer-to-peer architectural modeling based on PeerSim," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Commuications, Networks and Systems & Workshops*, pp. 1–10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, 2008.

[39] N. Kotilainen, M. Vapa, A. Auvinen, M. Weber, and J. Vuori, "P2PStudio-monitoring, controlling and visualization tool for peer-to-peer networks research," in *Proceedings of the ACM international workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks (PM2HW2N '06)*, pp. 9–12, ACM, Cologne, Germany, October 2006.

[40] H. Casanova, A. Legrand, and M. Quinson, "SimGrid: a generic framework for large-scale distributed experiments," in *Proceedings of the 10th International Conference on Computer*

Modelling and Simulation (EUROSIM/UKSim '08), pp. 126–131, IEEE, Piscataway, NJ, USA, April 2008.

[41] J. Kangasharju, U. Schmidt, D. Bradler et al., "ChunkSim: simulating peer-to-peer content distribution," in *Proceedings of the 2007 Spring Simulaiton Multiconference*, pp. 25–32, Society for Computer Simulation International, San Diego, Calif, USA, 2007.

[42] L. Bracciale, F. L. Piccolo, D. Luzzi et al., "OPSS: an overlay Peer-to-peer streaming simulator for large-scale networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 3, pp. 25–27, 2007.

[43] L. Bracciale, F. L. Piccolo, S. Salsano et al., "Simulation of Peer-to-peer streaming over large-scale networks using OPSS," in *Proceedings of the 2nd International Conference on Performance Evaluation Methodologies and Tools*, pp. 1–10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, 2007.

[44] "OMNet++," http://www.omnetpp.org/.

[45] "INET," http://inet.omnetpp.org/.

[46] "NETMATE 0.9.4," http://www.ip-measurement.org/tools/netmate/.

[47] J. Khan and R. Zaghal, "Jitter and delay reduction for time sensitive elastic traffic for TCP-interactive based world wide video streaming over ABone," in *Proceedings of the 12th IEEE International Conference on Computer Communications and Networks (ICCCN '03)*, pp. 311–318, Dallas, Tex, USA, 2003.

[48] A. Bar-Noy, J. Goshi, and R. E. Ladner, "Off-line and on-line guaranteed start-up delay for Media-on-Demand with stream merging," *Journal of Discrete Algorithms*, vol. 4, no. 1, pp. 72–105, 2006.

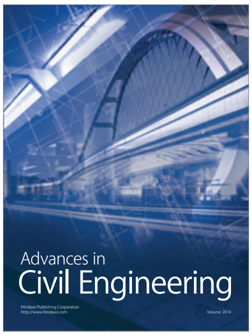
Journal of
Engineering

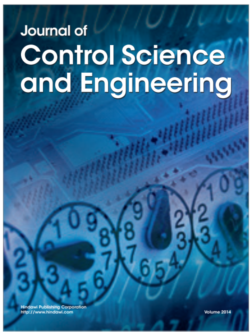
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

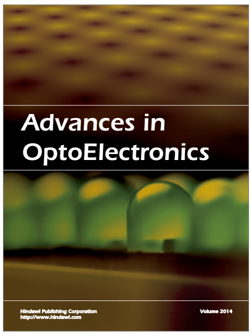
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

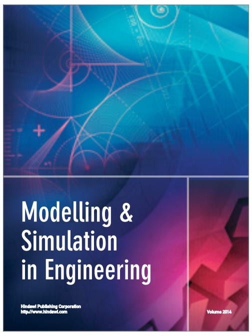
Journal of
Robotics

Journal of
Electrical and Computer Engineering

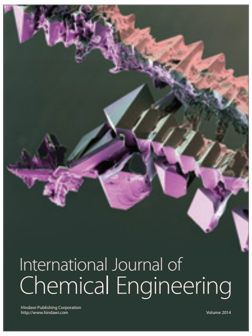

Advances in
OptoElectronics

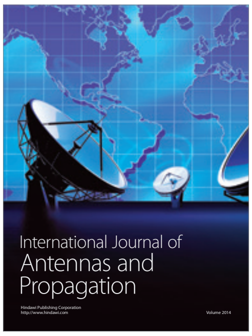
VLSI Design

International Journal of
Navigation and Observation

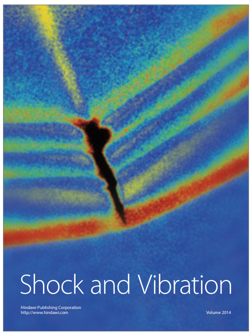

Modelling & Simulation in Engineering

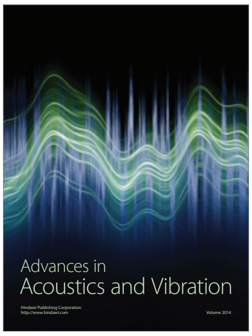
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration