Journal of Engineering Science and Technology Review 10 (3) (2017) 73-80

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Cloud-Assisted Cooperative Mechanism for File Download in Mobile Peer-to-Peer Networks

Liu Yuan-ni[1], Zou Yang[1], Guan Xin[2], GILANI Syed Mashhad Mustuzhar[1, 3] and Zhao Guo-feng[1,*]

[1] *School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China*
[2] *Yuzhong District Branch of Chongqing Municipal Public Security Bureau, 400010, China*
[3] *University Institute of Information Technology, Pir Mehr Ali Shah Arid Agriculture University Rawalpindi, Punjab, 46000, Pakistan*

Received 9 March 2017; Accepted 5 July 2017

Abstract

Complex computational tasks, such as topology construction and neighbor selection, which affect the file download efficiency in mobile peer-to-peer (MP2P) networks, are difficult to perform given the limited performance of mobile nodes. Thus, a cloud-assisted cooperative mechanism (CACM) for file download was proposed in this study to address this problem. The CACM aimed to construct a stable network topology based on the improved ant colony optimization algorithm and select appropriate neighbor nodes by adopting multiple strategies in the download process. Moreover, the main computational tasks were offloaded from the mobile nodes to the cloud server. Simulation results indicate that the file download efficiency of the CACM is close to that of BitTorrent in the first download stage, and CACM performs better at the second stage. In addition, the average download time of the CACM is reduced by at least 5% compared with BitTorrent, and its performance remains unchanged with the expansion of the network scale. The finding in this study proves that the research of CACM is crucial in improving the file download efficiency in MP2P networks.

Keywords: Cloud-assisted, Cooperative, Mobile Peer-to-Peer, Efficiency

1. Introduction

Mobile peer-to-peer (MP2P) has been extensively used in data distribution with the rapid development of mobile network and the peer-to-peer (P2P) technology. Research on improving the efficiency of file download in MP2P networks has attracted considerable attention because high-quality service requires an efficient network. Therefore, designing a file download mechanism in MP2P networks is an interesting and significant research topic. The previous studies [1, 2] revealed that cloud computing can be applied to the download mechanism for MP2P networks, and the cloud server has computing, network bandwidth, and storage capacity advantages.

In the existing literature, the file download mechanism's design mainly focuses on network topology construction and neighbor selection. The construction of a network topology is an essential step in the design of the download mechanism for MP2P networks. This process has become a significant challenge to mobile nodes because of their different computation capability. It has always been assumed that each node have the same attributions in the network. However, the performance of each node is different [3]. Nodes that have high performance cannot be fully utilized, while nodes with low performance would be overloaded under the unrealistic assumption. The difference in computation capability leads to load imbalance among nodes [4], which results in network instability and low file download efficiency. Thus, constructing a stable network topology requires the cloud server to evaluate the performance of nodes and coordinate the computational tasks during file distribution. The selection of neighbor nodes based on a more complex strategy is difficult because of the limited computing capability of the mobile node. Thus, the computation capability of nodes should be fully utilized by seeking extra assistance from the cloud server.

Based on the above analysis, this study examines a core problem in designing the download mechanism for MP2P networks. Subsequently, the effects of network size and load are considered.

2. State of the art

Many research institutions and scholars currently concentrate on investigating the download mechanism of MP2P networks. Their works mainly focus on network architecture, neighbor selection, resource scheduling, and upload bandwidth allocation.

2.1 Network architecture

The cloud server can configure resources and improve equipment utilization better than common mobile terminals. Many researchers have proposed the idea of a cloud-assisted

[*] E-mail address: zhaogf@caupt.edu.cn

ISSN: 1791-2377 © 2017 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.
doi:10.25103/jestr.103.10

MP2P architecture. Liu et al. [5] presented a new cloud-assisted architecture model for the mobile social network to reduce the traffic on the core network. In addition, a cloud-assisted MP2P network architecture was presented in [6], in which the cloud server adopted a novel content distribution mechanism to reduce data transmission and energy consumption. Rocha et al. [7] proposed a hybrid architecture, which included cloud and P2P layers, for solving the multimedia information retrieval problem. Riad et al. [8] suggested a novel cloud framework of the P2P video-on-demand system, which reduced the load on the server. Kavalionak et al. [9] argued that combining cloud computing and P2P would be one of the milestones for next generation distributed P2P-based architectures.

2.2 Neighbor selection

A performance evaluation model, which selects appropriate neighbor nodes with high performance during file distribution, was proposed in [10] to improve the file download efficiency in MP2P networks. In [11], the node in MP2P network selected its neighbors with maximum reputation value to forward the query walker during the search process, which solved the low resource search efficiency. Zhang et al. [12] highlighted that each peer should select the closest neighbor node to reduce network power consumption. In addition, a new neighbor selection strategy presented in [13] constructed a hierarchical overlay, which reduced the cost by two-thirds compared with random overlay.

2.3 Resource scheduling

Mobile terminal users are data consumers and providers in MP2P networks, enabling upload services to others after obtaining file pieces. Nodes that own more file pieces are more likely to provide upload service if the download request is stable, then, the download rate of the system would be faster. Considerable studies have been conducted on the design of resource scheduling to raise the resource utilization in file distribution systems. An incentive mechanism based on the credit and balance of resources was presented in [14]. The incentive mechanism highlighted the dynamic price of resources and reasonable penalties for free riders. Harjula et al. [15] proposed an approach that distributed the download tasks to mobile terminals with low power consumption, thereby extending the battery life of the mobile terminal. Wang et al. [16] constructed a three-phase scheduling mechanism for load balance in cloud computing networks. In the first phase, the cloud server sorted the task requirements to determine the best task order. In the second phase, the cloud server allocated each task to a suitable service manager based on task property. In the third phase, the cloud server selected the best service node to ensure that the tasks were accomplished within the minimum time.

2.4 Allocation of the upload bandwidth

The allocation of upload bandwidth for mobile terminal users is a popular research topic. Satsiou et al. [17] established a reputation system in the content distribution network, in which the system allocated the upload bandwidth to the mobile nodes with high reputation. Kapadnis et al. [18] considered that the upload bandwidth of the system should be allocated to the nodes with great contributions. Cong et al. [19] defined a utility function to describe the users' demand of file and designed a distributed bandwidth allocation scheme to maximize the overall utilization of the network. Huang et al. [20] presented an upload bandwidth allocation algorithm based on the bandwidth demand model to enhance the bandwidth utilization of nodes. Zheng et al. [21] proposed a push mechanism to ensure the nodes' upload bandwidth can be utilized efficiently.

Existing studies confirmed the significance of improving the file download process in MP2P networks. However, certain limitations should be addressed. (1) The capability difference among nodes, which has not been explored thoroughly, would lead to load imbalance. (2) A single strategy is not always optimal for nodes during file download when download tasks increase. Different strategies should be adopted in the different download stages. This study proposes a cloud-assisted cooperative mechanism (CACM) to address these problems. First, the cloud server collects the node status information and evaluates the computation capability of each node. Second, the improved ant colony optimization algorithm is used to construct the network topology to ensure load balance in the network. Third, the file download process is divided into two stages in which different download strategies are adopted for file distribution and neighbor selection.

The rest of this study is organized as follows. Section 3 describes the architecture of the cloud-assisted MP2P network and discusses the mathematical model based on the proposed architecture. The performance of CACM is evaluated in Section 4. Finally, conclusions are drawn in Section 5.

3. Methodology

This section presents the architecture of the cloud-assisted MP2P network. Then, we model the construction of file distribution topology and the download process based on this architecture.

3.1 Cloud-assisted MP2P architecture

This study presents a three-layer architecture. Fig. 1 shows that the cloud-assisted MP2P network architecture consists of the Internet, user, and cloud layers.

The function of each layer explained as follows:

(1) The Internet layer is the content provider in this architecture. It provides file resources to the cloud and user layers. Moreover, the Internet layer supports the network manager in maintaining the stability of the architecture.

(2) The user layer supports the users in downloading files they require. In this layer, the end users construct a mobile network, and each mobile user can send requests to the Internet layer. The file pieces owned by users come from the cloud layer or neighbors.

(3) The cloud layer acts as an intermediary layer between the user and the Internet layers. In this layer, the cloud server collects basic user information, including CPU, battery capacity, access method, and bandwidth. In addition, users can send download requests to the cloud server, and the cloud server can leverage its powerful computation capability to download files and store them in a server. Then, the files are split into pieces, and the cloud server distributes them among mobile terminals according to their computation capability. On one hand, the cloud layer reduces the traffic cost between the user and the Internet layers. On the other hand, most of the computational tasks

are offloaded to the cloud layer, thereby reducing the burden on each mobile terminal.

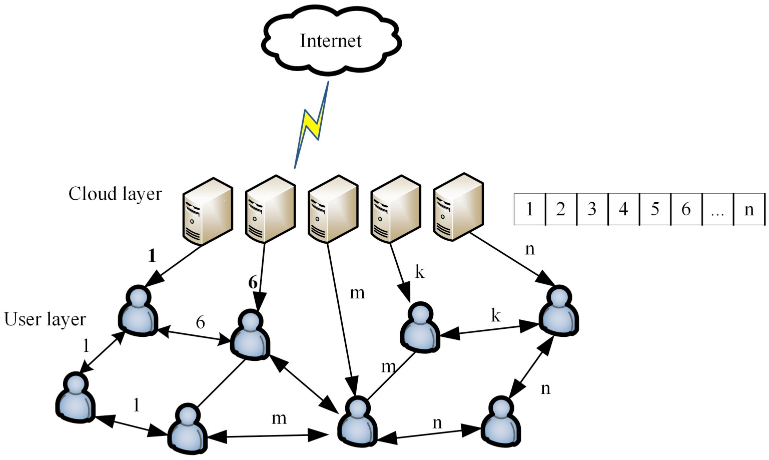


Fig. 1. Architecture of a cloud-assisted MP2P network

According to the architecture described above, we introduce the cooperative download process as follows. First, the number of nodes in the network is initialized at time instant t . Second, the cloud layer evaluates the computation capability of nodes according to the status information. Third, the file distribution topology is constructed based on the improved ant colony optimization algorithm. Finally, the nodes join the download process. In the download process, the number of file pieces owned by node i is fewer than 90% of an entire file at the initial state. Then, the cloud layer adopts the random and rare-first strategies [22] for file distribution. Otherwise, the node can select the neighbor node with the same number of file pieces and download the rest of the file pieces from its neighbors.

3.2 Mathematical model

The file download process was modeled based on the architecture described in Section 3.1. This model was established in the context of three major assumptions. (1) The joining process of nodes follows a monotonic decreasing Poisson distribution. (2) The nodes are randomly scattered in the network and would not leave until they obtain all the pieces of an entire file. (3) The upload bandwidth of a node is lower than its download bandwidth.

3.2.1 Network initialization

Because the joining process of nodes follows the Poisson distribution given the stability, generality, and non-aftereffect property, the number of nodes in the network should be initialized before the download process begins. The number of nodes at time instant t is represented by $N(t)$, which can be expressed as Eq. (1).

$$N(t) = N_0 e^{-\lambda t} \frac{\lambda^k}{k!} \tag{1}$$

Where N_0 is the number of nodes at initial state, λ is the average arrival rate of nodes, k is the number of times the node joins the network.

3.2.2 Performance assessment of nodes

The cloud layer evaluates the computation capability of nodes based on their status information, such as CPU, memory size, access bandwidth, and battery capacity. ω_1, ω_2, ω_3 and ω_4 represent the weighting factors. To evaluate the computation capability of nodes, we formulate the equations as follows:

$$\phi = \omega_1\alpha + \omega_2\beta + \omega_3\gamma + \omega_4\theta \tag{2}$$

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1 \tag{3}$$

In Eq. (2), ϕ is the computation capability of nodes. α denotes the available CPU ratio. β is the available memory size ratio, γ is the available access bandwidth ratio, θ is the available battery capacity ratio.

To determine the values of weighting factors, the analytic hierarchy process (AHP) method was used in this study. The AHP can decompose the decision problem into different hierarchies. Moreover, it calculates the priority weight of each element to another in the last hierarchy by creating a judgment matrix. The last step of AHP calculates the final weight of the alternative solution to the general goal. Then, the maximum weighted solution is optimal. The details of the computation method are as follows:

(1) Establishing the hierarchical structure model

First, we build a multi-objective hierarchical structure model based on practical application, as illustrated in Fig. 2. The model is a three-layer structure, which includes the target, criterion, and program layers. The goal of the target layer is to select the node with the highest computation capability. The criteria considered in this evaluation are CPU, memory size, bandwidth, and battery capacity. Finally, the program layer proposes an optimal solution.

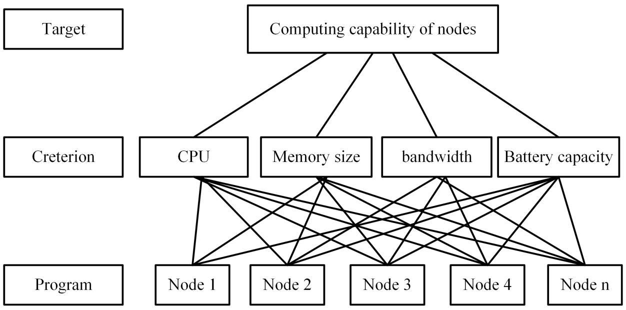


Fig. 2. Hierarchy structure

(2) Creating the judgment matrix

We compare the CPU, memory size, bandwidth, and battery capacity in terms of their importance in reaching the goal. The results of the pairwise comparison are organized in the judgment matrix $\boldsymbol{A}$.

$$\boldsymbol{A}=\begin{bmatrix} 1 & 3 & 1 & 5 \\ 1/3 & 1 & 1/2 & 1/2 \\ 1 & 2 & 1 & 2 \\ 1/5 & 2 & 1/2 & 1 \end{bmatrix} \tag{4}$$

The AHP method is used to determine the appropriate values of weighting factors ω_1 , ω_2 , ω_3 and ω_4 , and the result is obtained as follows:

$$\boldsymbol{\omega}=(\omega_1,\omega_2,\omega_3,\omega_4)^T=(0.42,0.12,0.30,0.16)^T \tag{5}$$

Where $\boldsymbol{\omega}$ is a feature vector of the matrix in Eq. (4), the superscript T is the transpose symbol.

Then, the following expressions can be derived by substituting Eq. (5) to Eq. (2):

$$\phi=0.42\alpha+0.12\beta+0.30\gamma+0.16\theta \tag{6}$$

3.2.3 File distribution topology

To improve the efficiency of the file download and achieve load balance in the MP2P network, we employs the improved ant colony algorithm to construct the file distribution topology. The ant denotes the file piece in this study, the computation capability of nodes is determined as the initial factor for ant to search the path. The ant updates the pheromone content when it finds one node with the same computation capability. Then, the nodes with similar computation capability become the neighbor nodes of each other. We construct the network topology based on the computation capability of nodes. The basic idea of the improved ant colony optimization algorithm is described as follows.

First, according to the weighting factors, the cloud layer estimates the computation capability of each node by collecting its status information, such as CPU, memory size, access bandwidth, and battery capacity. Then, the cloud layer sorts the nodes according to their computation capability to establish a balance tree, where the nodes with similar computation capability are selected as neighbors. Supposing that the node computation capability ranges from 0 to 1 and we select the values 0.3, 0.6, and 0.9, then the balance tree is as shown in Fig. 3.

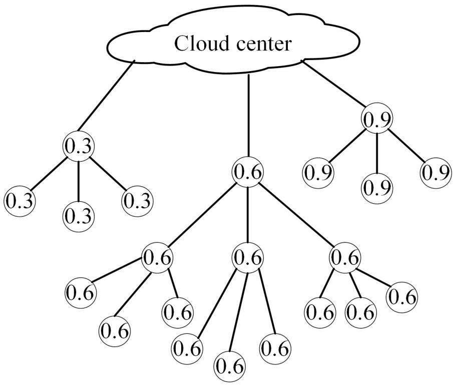


Fig. 3. Topology of file distribution

3.2.4 File download process of nodes

At the first stage, the file pieces owned by node i are less than 90% of an entire file. a_{ij} is the number of the j th file piece that node i owns, and n is the total number of file pieces belonging to an entire file. The network state at the first stage described as follows:

$$\sum_{j=1}^{n} a_{ij} \le 90\% \times n \tag{7}$$

At the beginning of the file distribution, one mobile node cannot provide any pieces to other nodes, and it must obtain all the file pieces as soon as possible. The first file piece distributed from the cloud layer to the node is randomly selected, and then the cloud layer adopts the rare-first strategy for file distribution.

The model is built based on the following assumptions:

(1) Each mobile node has equal opportunity to download file pieces.

(2) The environmental factors that affect the bandwidth of a node can be ignored.

(3) Except for the cloud server, none of the other mobile terminal nodes can be regarded as a seed node.

(4) The model can be simplified based on the assumption that none of the nodes leaves the network until it has successfully downloaded all pieces of an entire file.

In the cloud-assisted MP2P network, we denote the shared file as SF . The number of nodes that participate in the download process is N . Each node connects with m neighbor nodes. The shared file is split into n pieces, SF_i and each piece has its copies in the network. A node can download the file successfully only when has collected all the pieces of SF . We use the probability that the file is successfully downloaded to quantify the validity of shared file.

The probability that a node obtains the i th file piece of SF can be expressed as P_{SF_i} , which calculated as follows:

$$P_{SF_i}=X_i/N \tag{8}$$

Where SF_i is the i th piece of shared file, X_i is the number of SF_i and its copies that cloud layer distributed to the network, N is the total number of nodes in the network.

We denotes the probability that all neighbor nodes have no file piece SF_i as $\overline{P_{SF_i}}$, which can be calculated as follows:

$$\overline{P_{SF_i}}=(1-P_{SF_i})^m=(1-X_i/N)^m \tag{9}$$

Where m is the number of neighbors of each node.

The probability that at least one of the m neighbor nodes owns the file piece SF_i is denoted by P'_{SF_i} :

$$P'_{SF_I}=\left(1-\overline{P_{SF_i}}\right)=1-(1-X_i/N)^m \tag{10}$$

The validity of the shared file can be obtained as follows:

$$F = \prod_{i=1}^{n} P'_{SF_i} = \prod_{i=1}^{n}\left(1-\left(1-X_i/N\right)^m\right) \tag{11}$$

Where F is the validity of shared file, n is the number of pieces of an entire shared file.

According to the previous definition, the shared file with better validity is more likely to be downloaded successfully. Therefore, the above problem can be transformed into a constrained optimization problem:

$$\begin{cases} \min : F = -\prod_{i=1}^{n}\left(1-\left(1-X_i/N\right)^m\right) \\ s.t: H = Sum - \sum_{i=1}^{n} X_i = 0 \\ X_i \geq 0, i = 1,2,\cdots,n \\ \end{cases} \tag{12}$$

Where $\min : F$ is the objective function of the optimization problem, $s.t: H$ is an objective function constraint, and Sum is the total number of file pieces, including the copies of each piece of shared file. Such a constrained optimization problem can be solved using Kuhn-Tucker conditions. We construct a Lagrange function as follows:

$$L(X,v) = -\prod_{i=1}^{n}\left(1-\left(1-X_i/N\right)^m\right) - v\left(Sum - \sum_{i=1}^{n} X_i\right) \tag{13}$$

Where v is the Lagrange multiplier. The partial derivatives of each X is obtained in Eq. (14):

$$\frac{\partial L}{\partial X_i} = -\frac{m}{N}\left(1-\frac{X_i}{N}\right)^{m-1}\prod_{k=1,k\neq i}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right) + v \tag{14}$$

Where L is the Lagrange function in Eq. (13), and X_k is the number of the k th file piece and its copies in network. The partial derivatives of v is obtained as follows:

$$\frac{\partial L}{\partial v} = \sum_{i=1}^{n} X_i - Sum \tag{15}$$

Then, the $\nabla L(X,v)$ is defined as the gradient of function $L(X,v)$, which can be expressed as follows:

$$\nabla L(X,v) = \begin{pmatrix} -\frac{m}{N}\left(1-\frac{X_i}{N}\right)^{m-1}\prod_{k=1,k\neq i}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right)+v, \\ \sum_{i=1}^{n} X_i - Sum \end{pmatrix} \tag{16}$$

According to the conditions of Kuhn-Tucker, we list the formulas as follows:

$$\begin{cases} -\frac{m}{N}\left(1-\frac{X_i}{N}\right)^{m-1}\prod_{k=1,k\neq i}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right)+v = 0 \\ \sum_{i=1}^{n} X_i - Sum = 0 \end{cases} \tag{17}$$

The non-negative v can be expressed as Eq. (18):

$$v = \frac{m}{N}\left(1-\frac{X_i}{N}\right)^{m-1}\prod_{k=1,k\neq i}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right) \tag{18}$$

Then, we list the formula as follows:

$$\begin{aligned} &\frac{m}{N}\left(1-\frac{X_i}{N}\right)^{m-1}\prod_{k=1,k\neq i}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right) \\ &= \frac{m}{N}\left(1-\frac{X_j}{N}\right)^{m-1}\prod_{k=1,k\neq j}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right) \end{aligned} \tag{19}$$

The solutions of above equations can be obtained as follows:

$$X_1 = X_2 = X_3 = \cdots = X_n \tag{20}$$

Then, the local optimal solution $\boldsymbol{X}^*$ can be obtained as follows:

$$\boldsymbol{X}^* = \left(X_1^*, X_2^*, X_3^*, \cdots, X_n^*\right) = \left(\frac{Sum}{n}, \frac{Sum}{n}, \frac{Sum}{n}, \cdots, \frac{Sum}{n}\right) \tag{21}$$

Where X_i^* is the optimal valve of X_i, Sum is the total number of all file pieces and their copies, n is the number of file pieces that a shared file owned.

The second derivative of the function F can be expressed as follows:

$$\frac{\partial^2 F}{X_i^2} = \frac{m(m-1)}{N^2}\left(1-\frac{X_i}{N}\right)^{m-2}\prod_{k=1,k\neq j}^{n}\left(1-\left(1-\frac{X_k}{N}\right)^m\right) > 0 \tag{22}$$

According to Eq. (22), F is an increasing function, and the constrained optimization problem is also a convex programming problem. Therefore, $\boldsymbol{X}^*$ is the global optimal solution. The optimal solution indicates that the number of SF_i is equal to the average number of total file pieces in the cloud-assisted mobile P2P network. Therefore, only when the cloud layer adopts the rare-first strategy to distribute file pieces evenly in the first stage can the proposed model obtain the optimal solution.

At the second stage of file download process, the number of file pieces owned by node i is more than 90% of an entire shared file. The network state can be expressed as follows:

$$\sum_{j=1}^{n} a_{ij} \geq 90\% \times n \tag{23}$$

Where a_{ij} is the number of the j th file piece that node i owns, and n is the number of file pieces owned by an entire file.

The efficiency of file download in the network cannot be improved when the node selects its neighbor nodes with more file pieces. The nodes with fewer file pieces are less likely to be selected as neighbors; thus, the loads of other nodes increase.

Similarly, the node has less interest in its neighbors with fewer file pieces. Then, the node would fail to obtain the desired file pieces from its neighbors, thus resulting in low file download efficiency.

To maintain high file download efficiency at the second stage, we propose a new strategy for neighbor selection. In this strategy, the node with the same number of requests is selected as the neighbor node. The success rate of exchanging pieces between nodes with the same number of pieces is higher. To evaluate the file download efficiency in MP2P networks, the average download time (ADT) should be considered. The ADT can be calculated by Eq. (24):

$$ADT = \frac{\sum_{i-1}^{N} T_i}{N} \tag{24}$$

Where N is the total number of nodes, and T_i is the download time of node i. The shorter average download time indicates a higher efficiency of file download.

4 Result analysis and discussion

This section first discusses the simulation environment and parameters. More analyses based on the simulation results are discussed in the following section. In the simulation, to evaluate the performance of CACM, we compare it with existing algorithms, such as Random [23], BitTorrent [24], Random+LoadBalance, and BitTorrent+LoadBalance [25].

4.1 Experimental environment

We simulate and evaluate the efficiency of file download in PeerSim [26]. The experimental platform and the required hardware/software are shown in Table 1. To simulate the actual network features, we list the related parameters in Table 2.

Table 1. Experimental platform

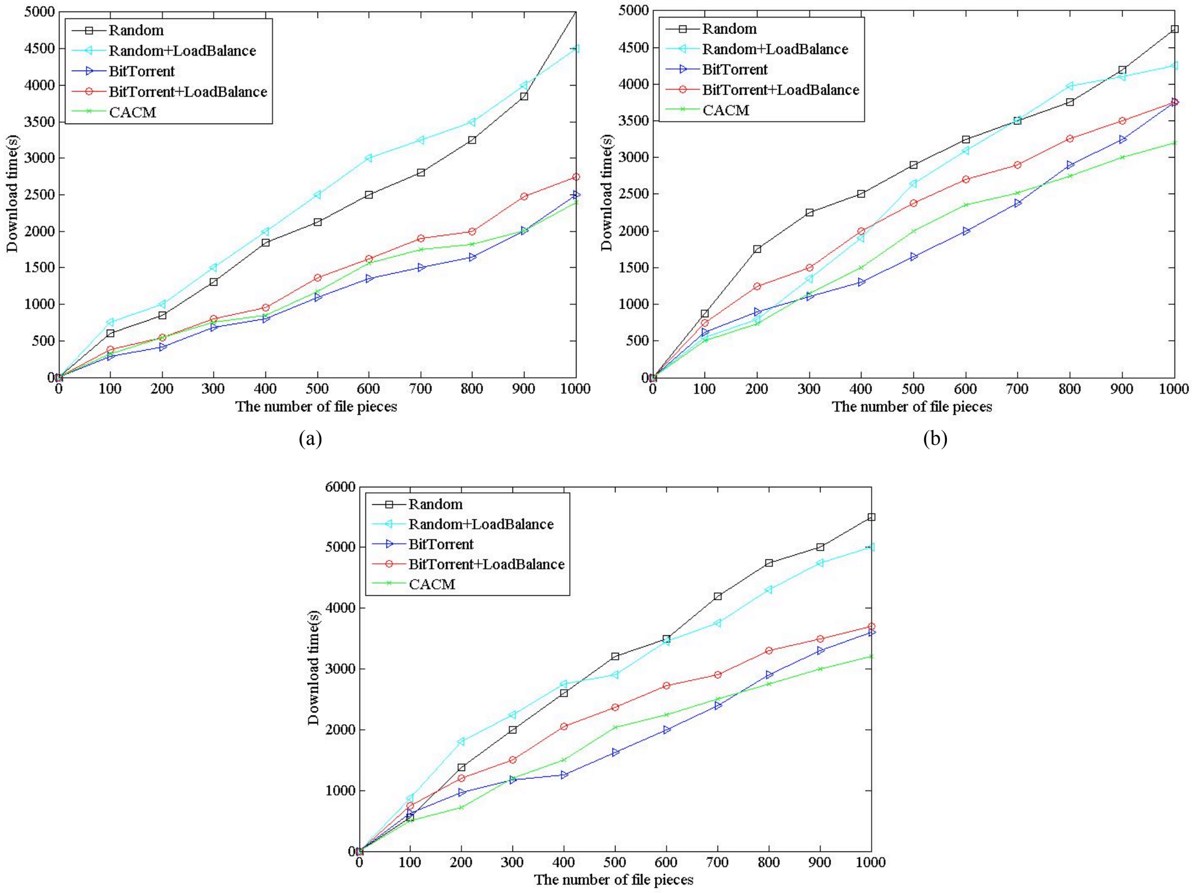
GPU(GHz)	2.3
Memory (GB)	8
Hard disk (GB)	320
Operate system	Windows
Development environment	Eclipse
Simulation software	PeerSim-1.0.5

Table 2. Experimental parameters

Parameters types	**Parameters settings**
File size (MB)	8
File piece size (KB)	320
The download bandwidth of node (kbps)	1024
The upload bandwidth of node (kbps)	512
The number of initial nodes	100, 200, 300
Maximum number of neighbor nodes	5
Minimum number of neighbor nodes	1
The rate of addition/departure	5%

4.2 Simulation results and analysis

This study compares the performance of the CACM with Random, Random+LoadBalance, BitTorrent+LoadBalance, and BitTorrent. In the simulation, a file (256 MB) is split into 1000 pieces, and the size of each piece is 256 KB. The number of nodes that participate in the download process is 100, 200, and 300.



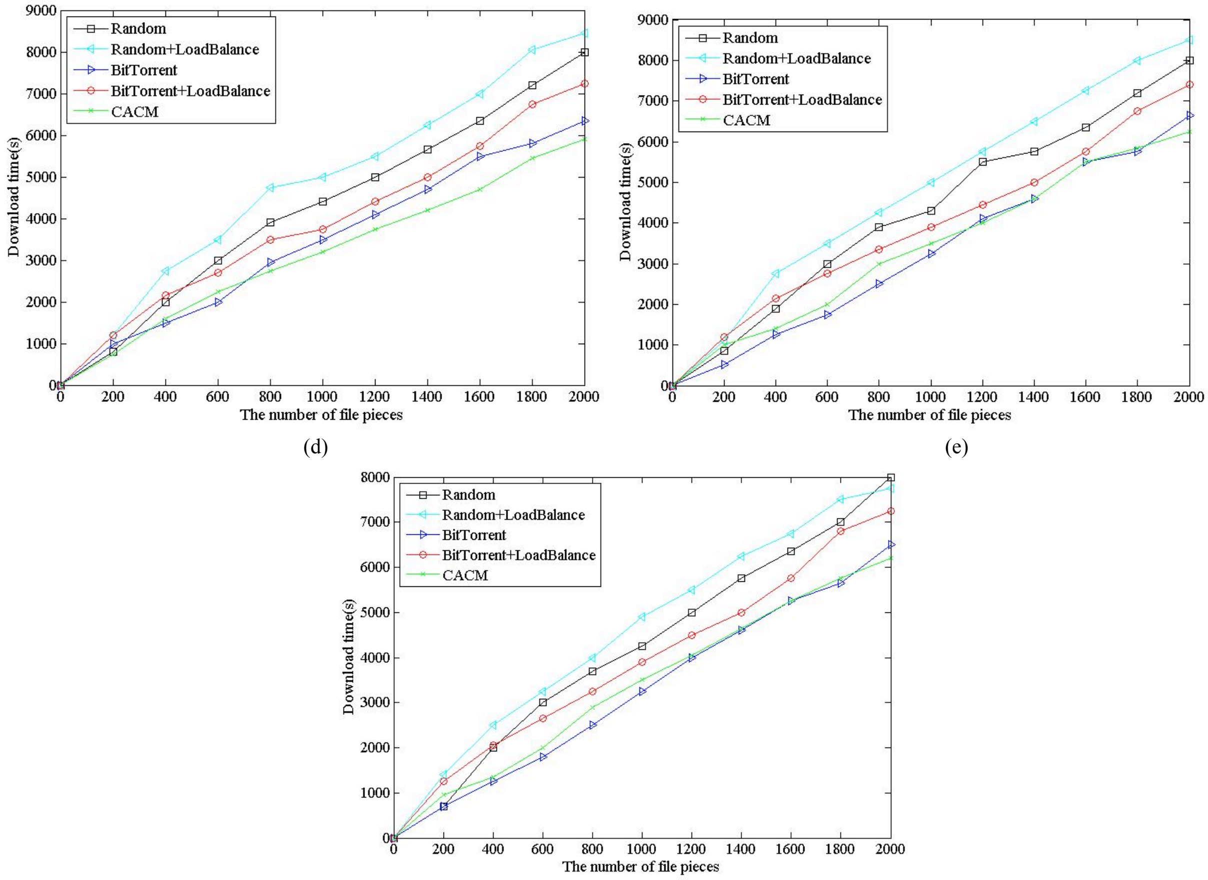
(a)

(b)

(c)

Fig. 4. File download time of nodes in MP2P networks. (The file size is 256 MB. (a) 100 nodes; (b) 200 nodes; (c) 300 nodes)

As shown in Fig. 4, the horizontal axis represents the number of downloaded file pieces in the system, and the vertical axis represents the total download time. Five curves represent the different algorithms. Fig. 4 shows that, at the first stage, the download time of the CACM is longer than that of the original BitTorrent. However, it performs better than the original BitTorrent at the second stage. When file download is completed, the average download time of the CACM is less than those of BitTorrent and the other algorithms. These results remain unchanged with the increasing node number of the network. From the perspective of average download time, the CACM improves the efficiency of file download in MP2P networks.



(d)

(e)

(f)

Fig. 5. File download time of nodes in MP2P networks (The file size is 512 MB. (d) 100 nodes; (e) 200 nodes; (f) 300 nodes)

To illustrate the suitability of the CACM for downloading larger files, we set the file size to 512 MB, and the results are shown in Fig. 5. When the number of nodes increases from 100 to 200 and 300, the average download time of BitTorrent is nearly in its original state. In comparison, the average download time of the CACM is closer to BitTorrent at the first stage. However, the total download time of the CACM is less than that of BitTorrent. Therefore, the CACM also improves the efficiency of file download from other nodes in MP2P network when it used to download large file.

5. Conclusions

To improve the efficiency of file download in MP2P networks, a file download mechanism referred to as the CACM was proposed in this study. The mechanism was used to construct a stable network topology and to select neighbor nodes by adopting multiple strategies during file download. The simulation results were analyzed to evaluate the CACM performance. The following conclusions were drawn:

(1) Among the five algorithms (CACM, Random, Random+LoadBalance, BitTorrent+LoadBalance, and BitTorrent), the file download efficiency of the CACM is close to that of BitTorrent in the first stage, and it performs better at the second stage. The file download efficiency of the other three algorithms is lower than that of CACM.

(2) The comparison result does not change significantly with the expansion of network size.

(3) The CACM performance is also superior to those of the other algorithms when downloading large files. Moreover, the average download time of CACM is at least 5% lower than that of BitTorrent.

The results demonstrate that the CACM can improve the file download efficiency in MP2P networks. However, many deficiencies still exist in our mechanism. For instance, the simulation results show that the CACM only performs better at the second stage. In addition, the CACM proposed in this study was only compared with other algorithms with respect to the download time. Thus, further analysis is required to analyze the time complexity of these algorithms. In future work, more factors are necessary to be considered to improve the CACM performance.

Acknowledgements
This work was partially supported by NSFC China (No. 61501075), Scientific and Technological Research Program of Chongqing Municipal Education Commission, KJ1400419.

References

1. Larosa, Y. T., Chen, J. L., Deng, D. J., Chao, H. C., "Mobile cloud computing service based on heterogeneous wireless and mobile P2P networks". In: *2013 9th International Wireless Communications and Mobile Computing Conference*, University of Cagliari Sardinia, Italy: IEEE, 2013, pp. 661-665.
2. Dinh, H. T., Lee, C., Niyato, D., Wang, P., "A survey of mobile cloud computing: architecture, applications, and approaches". *Wireless communications and mobile computing*, 13(18), 2013, pp.1587-1611.
3. Wang, Y., Li, Z. K., "A multicast advanced algorithm based on node heterogeneity for P2P MMOG". In: *2010 2nd International Conference on Future Computer and Communication*, Wuhan, China: IEEE, 2010, pp.165-168.
4. Xia, L., Duan, H. C., Zhou, X., Zhifeng-Zhao., "Heterogeneity and load balance in structured P2P system". In: *International Conference on Communications, Circuits and Systems*, Chengdu, China: IEEE, 2010, pp.245-248.
5. Liu, Y. N., Tang, H., Zhao, G. F., "CAMPSNA: A cloud assisted mobile peer to peer social network architecture". *Journal of Digital Information Management*, 12(2), 2014, pp.126-132.
6. Liu, Yuanni., Chen, Hongyan., Syed Mushhad, M.Gilani., "A cloud-assisted architecture for content distribution in mobile peer to peer networks". *Journal of Digtal Information Management*, 14(6), 2016, pp.403-412.
7. Rocha, V., Kon, F., Cobe, R., Wassermann, R., "A hybrid cloud-P2P architecture for multimedia information retrieval on VoD services". *Computing*, 98(1-2), 2016, pp.73-92.
8. Riad, A. M., Elmogy, M., Shehab, A. I., "A framework for cloud P2P VoD system based on user's behavior analysis". *International Journal of Computer Applications*, 76(6), 2013, pp.20-26.
9. Kavalionak, H., "*Autonomous resource management for cloud-assisted peer-to-peer based services*". Doctoral thesis of University of Trento, University of Trento, Italy, 2013.
10. Sun, Z., Shen, J., "A high performance peer to cloud and peer model augmented with hierarchical secure communications". *Journal of Systems and Software*, 86(7), 2013, pp.1790-1796.
11. Ge, J., Shi, H., Fang, Y., "Improved resource search strategy using random walk based on node reputation for unstructured mobile P2P network". In: *First International Conference on Information Science and Electronic Technology*, Wuhan, China: Atlantis Press, 2015, pp.9-12.
12. Zhang, X., Hassanein, H., "On network utilization of peer-to-peer video live streaming on the internet". In: *IEEE International Conference on Communications*, Kyoto, Japan: IEEE, 2011, pp.1-5.
13. Zhang, X., Hassanein, H., "A neighbouring strategy for ISP-Friendly peer-to-peer video live streaming". In: *IEEE International Conference on Communications*, Kyoto, Japan: IEEE, 2011, pp. 1-5.
14. Rui, L. L., Xiong, Y. L., Huang, H. Q., Qiu, X. S., "Mobile P2P resource sharing mechanism". *Journal of Beijing University of Posts and Telecommunications*. 39(1), 2015, pp.46-49.
15. Harjula, E., Gurtov, A., Koskela, T., Ojala, T., Ylianttila, M., "Energy-aware load monitoring for improving battery life of mobile peer-to-peer nodes". *Sustainable Computing: Informatics and Systems*, 12, 2016, pp.43-54.
16. Wang, S. C., Yan, K. Q., Wang, S. S., Chen, C. W., "A three-phases scheduling in a hierarchical cloud computing network". In: *International Conference on Communications and Mobile Computing*, Qingdao, China: IEEE, 2011, pp. 114-117.
17. Satsiou, A., Tassiulas, L. "Reputation-based resource allocation in P2P systems of rational users". In: *IEEE Transactions on Parallel and Distributed Systems*, 21(4), 2010, pp.466-479.
18. Kapadnis Sapana., Prof. Ranjana Dahake., "Time efficient distributed file storage and sharing using P2P network in cloud". *International Journal of Scientific Engneering and Applied Sicence*, 1(3), 2015, pp.537-543.
19. Cong, X., Shuang, K., Su, S., Yang, F. C. "An efficient server bandwidth costs decreased mechanism towards mobile devices in cloud-assisted P2P-VoD system." *Peer-to-Peer Networking and Applications*, 7(2), 2014, pp.175-187.
20. Huang, G., Yu, T., Liu, P., Zhang, Y., "An upload bandwidth allocation algorithm in data scheduling of P2P VoD system". In: *2014 5th IEEE International Conference on Software Engineering and Service Science*, Beijing, China: IEEE, 2014, pp.435-438.
21. Zheng, Y., Lin, F., Yang, Y., Gan, T., "Adaptive resource scheduling mechanism in P2P file sharing system". *Peer-to-Peer Networking and Applications*, 9(6), 2016, pp.1089-1100.
22. Legout, A., Urvoy-Keller, G., Michiardi, P., "Rarest first and choke algorithms are enough". In: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, New York, USA: ACM, 2006, pp.203-216.
23. Spritson, A., Sadeghi, P., "A randomized algorithm and performance bounds for coded cooperative data exchange". In: *IEEE International Symposium on Information Theory Proceedings*, Austin, USA: IEEE, 2010, pp.1888-1892.
24. Pouwelse, J., Garbacki, P., Epema, D., Sips, H., "The bittorrent P2P file-sharing system: measurements and analysis". In: *International Workshop on Peer-to-Peer Systems*, New York, USA: Springer, 2005, pp.205-216.
25. Xu, Z., Bhuyan, L., "Effective load balancing in P2P systems". In: *2006 IEEE International Cluster Computing and the Grid*, Singapore Management University, Singpore: IEEE, 2006, pp.81-88.
26. Montresor A., Jelasity M., "PeerSim: A scalable P2P simulator". In: *IEEE Ninth International Conference on Peer-to-Peer Computing*, Seattle, USA: IEEE, 2009, pp.99-100.