

# OMICC: an overlay multicast infrastructure based on cloud computing for streaming media data distribution

Wei Zhang<sup>a,b,\*</sup>, Meihong Yang<sup>b</sup>, Xinchang Zhang<sup>b</sup>, Huiling Shi<sup>b</sup>

<sup>a</sup> College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao 266510 China

<sup>b</sup> Shandong Key Laboratory of Computer Networks, Shandong Computer Science Centre (National Supercomputer Centre in Jinan), Jinan 250000, China

\*Corresponding author, e-mail: wzhang@sdas.org

Received 31 Aug 2014

Accepted 20 Jul 2016

**ABSTRACT:** With the rapid development of information technology, cloud computing is becoming known and used by more and more people. Many application layer multicast protocols, however, do not take the emergence of cloud computing into account. Cloud computing centres have many good properties, such as sufficient network bandwidth and more stability and so they are more suitable for streaming media data distribution applications. This paper presents a new overlay multicast infrastructure based on cloud computing for streaming media data distribution called OMICC. OMICC has a two-tier multicast structure. In the higher tier, there is an overlay tree which is organized by the set of Cloud Computing Centres, called CCCs. In the lower tier, each CCC is a set of roots of multicast trees.

**KEYWORDS:** application layer multicast, two-tier multicast structure, virtual machine, multicast tree

## INTRODUCTION

The rapid development of information technology is changing the way people obtain information. Before, they obtained information through PC. Now, more and more people obtain information from data centres via the internet. Computing services that are provided by data centres over the internet are now commonly referred to as cloud computing. Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The Cloud Computing model offers the promise of massive cost savings combined with increased IT agility. Recently, three service models, software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) are used by more and more people<sup>1</sup>. To provide a good quality of service through the internet, sufficient network bandwidth is the basis for each cloud computing centre. It is necessary therefore to use the bandwidth of the cloud centre effectively by improving existing networks and internet technologies with cloud computing.

Although invented in the early days of the internet, its design is to be scalable to the size and

the dynamics of the internet in present days. The emergence of cloud computing was however not foreseen, and many network protocol did not take that into account. On the other hand, in currently existing network protocol, the use of efficient and scalable multicast mechanism is essential to the success of large-scale group communication applications. These applications, such as network conferences or online video streaming, need to distribute the data effectively<sup>2</sup>. Multicast services allow the information to be sent from one host to many receivers, which makes applications more scalable and leads to more efficient use of system resources. The early multicast protocols called IP Multicast need the support of network layer equipment, and have not been widely deployed in the internet. The lack of network layer support for multicast however makes it necessary to obtain multicast service at a higher level. To do so, application layer multicast (ALM) was proposed. In these approaches, certain nodes form a virtual network, and multicast delivery structures are constructed on top of this virtual network. To improve the stability and scalability of the multicast transmission, researchers proposed an agent-based application layer multicast or an overlay multicast. These approaches uses strategically deployed overlay proxy nodes (sometimes referred to as service nodes) besides end hosts to

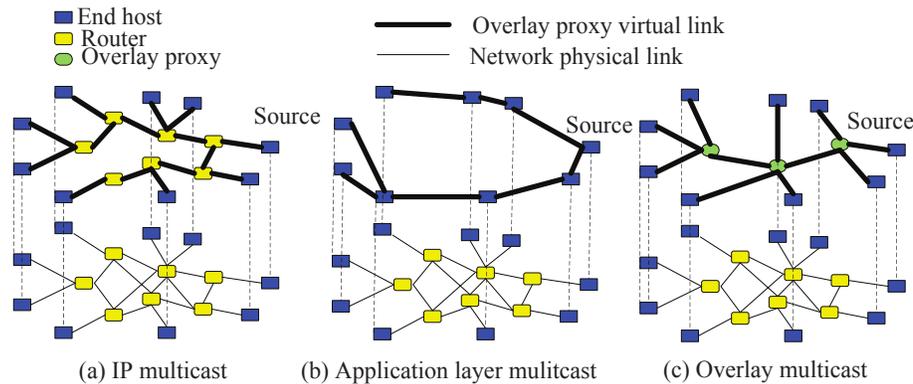


Fig. 1 Three types of multicast.

facilitate the management of group membership and multicast delivery structures. All the three types of multicast can be seen in Fig. 1.

So far, there are many ALM protocols to distribute streaming media data which address how to use a tree structure to deliver the multicast data from the source to all the receivers<sup>3-9</sup>. Depending on the sequence of constructing overlays and building delivery trees, existing application layer multicast solutions are classified into three approaches: mesh-first approach (e.g., NARADA<sup>10</sup>), tree-first approach (e.g., ALMI<sup>3</sup>), and implicit approach (e.g., ZIGZAG<sup>11</sup>).

Recently the data distribution based on a combination of cloud computing and mesh-pull P2P has gradually becoming interesting to the research community<sup>12-16</sup>. Trajkovska et al<sup>12</sup> presented a use case of the cloud infrastructure by introducing architecture for P2P multimedia streaming in both CS and P2P style. Jin and Kwok<sup>14</sup> offered QoS API functions implemented in a web service of the cloud streaming service provider, which enables users to decide about the contract type to establish with the service provider or go for P2P streaming as a possible solution. Islam and Gregoire proposed a dichotomy between the user and cloud-based services by using an edge based computation, coordination, and storage facility<sup>15</sup>. This extension is used to support multimedia applications, but it seems to be difficult to widely deploy data in a real network. Yu and Lai<sup>16</sup> presented a P2P resource search mechanism which adopts P2P networking technologies to orchestrate all the computing resources.

To our knowledge, there is scant literature discussing combination of cloud computing. Fouquet et al<sup>17</sup> discussed the possible combination of cloud computing and tree-base ALM. They did not discuss, however, how to implement an effective combina-

tion, and did not discuss how to use cloud computing techniques to construct an overlay network.

All the protocols discussed above, however, do not take into account the emergence of cloud computing. As we known, Cloud Centre can offer many services. It is possible that a network service like the proxy nodes be offered by a cloud centre. The purpose of this paper is to consider an overlay multicast infrastructure based on cloud computing, called OMICC, which can effectively use the bandwidth of the cloud centre by improving the existing overlay multicast. The infrastructure consists of a little set of cloud computing centres (called CCCs) distributed in the internet and provides efficient data distribution services to a large set of end users.

## PROBLEM DESCRIPTION AND MODEL OVERVIEW

In this section, we describe our problem in detail, and give the overview of the OMICC model.

This paper focuses on the problem of providing practical solutions for large-scale streaming media data distribution applications with the Cloud Computing Centres. For large-scale data distributions, such as live web-casts, we assume that there is a single source. In a normal application layer multicast which every node is the same end-host, each end user can only accept limited nodes as its child nodes. The Cloud Computing Centres have sufficient bandwidth, and can accept much more nodes as their child nodes<sup>17</sup>. First, we describe the properties of CCCs according to the actual situations.

*Property 1:* the number of CCCs ( $N_{\text{CCCs}}$ ) is far less than the number of end users ( $N_u$ ), i.e.,  $N_{\text{CCCs}} \ll N_u$ .

In the current circumstances, the Cloud Computing Centres are constructed by a few large enterprises or governments. From the global point

of view, there are only less than ten famous Cloud Computing Providers like Amazon, Google, etc. For a data distribution application, the number of CCCs that can be used is much few. However, the number of end users who want to accept data is more than millions. Thus we can say that the number of CCCs is far less than the number of end users.

*Property 2:* The CCCs are more stable than the end users.

As a service provider, Cloud Computing Centre must ensure its continuous stability. With the support of large enterprises or governments, Cloud Computing Centres have a higher level of technology and capital to ensure their stable service. From the perspective of data distribution service, the end user itself has relatively low stability. The end users can be as the data distribution nodes only when they need to accept the data and is involved in the multicast service. However, they also have greater uncertainty, such as power outages or other special events to make them leave. In essence, the end users are free to join or leave the multicast. So the CCCs are more stable than the end users.

To provide a good quality of service through internet, sufficient bandwidth is the basis for each Cloud Computing Centre. The CCC can serve millions of users, and its bandwidth must be sufficient, which can be supported by the government or enterprises. However, the bandwidth of each end user is different from each other.

According to these properties above, we can propose an infrastructure which consists of a set of CCCs as the super nodes distributed in the internet and provides efficient data distribution services to a set of end user. So our problem can be described that given a set of CCCs with access bandwidth constraints distributed in the network, construct a multicast data delivery backbone such that the overlay latency to the end users set is minimized. Another problem is how the CCC distributes the data to each end user when it has received the data.

According to the above description of the problem, we present an overview of our OMICC. There are three different parties in OMICC: the source node (SN), the CCC nodes, and the end user nodes. OMICC is the two-tier architecture. It is an overlay network formed among a set of few overlay CCCs, on top of which multicast distribution trees are built for data delivery based on multicast routing protocols. For the differences from the previous overlay architectures, every CCC is not only the root of one delivery tree, but the set of roots of several trees.

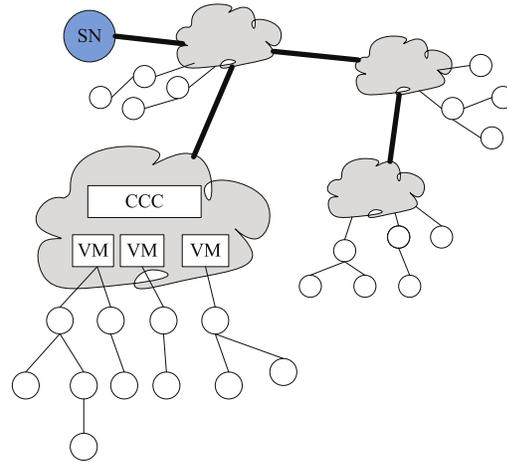


Fig. 2 The architecture of OMICC.

Fig. 2 gives a main description of the components and architecture of OMICC. There are two-tier in the architecture. Because of good properties of the CCCs discussed above, they can support a variety of group communication applications and be service providers for many end users. Besides, the overlay tree is relatively stable to be the multicast data delivery backbone. The construction of the overlay tree is done by the SN. Since the number of the CCCs is small, the SN can construct the tree and put forward the routing table to each of the CCCs before SN sending data. More details on the construction of the overlay tree can be seen in the next section.

For the lower tier, there are groups of the deliver multi-trees. In one group, the root node of every tree is a virtual machine in the cloud. The multi-tree structure has several advantages. This structure has better robustness and bandwidth utilization than the tree. For example, in a  $k$ -quad tree with the height of  $h$ , if the node in the  $i$ th layer is a failure, it will affect all its child nodes. The number  $N_i$  of nodes which are affected is given by the equation:

$$N_i = \frac{k^{h-i} - 1}{k - 1} - 1, \quad i = 0, 1, 2, \dots, h - 1. \quad (1)$$

The number of all nodes in this tree can be given by equation:

$$N_{\text{all}} = \frac{k^h - 1}{k - 1}, \quad k > 1. \quad (2)$$

The number of all leaf nodes is  $k^{h-1}$ , so the ratio  $\delta$  of the number of all leaf nodes and the number of

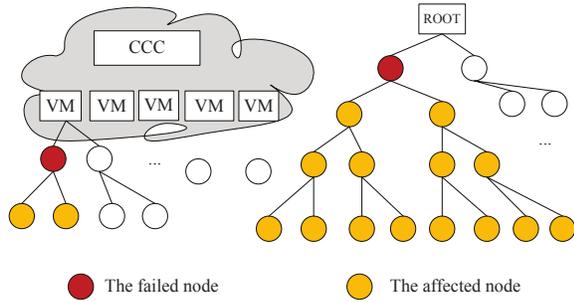


Fig. 3 Comparison of the two structures.

all nodes are as the follow equation:

$$\delta = \frac{(k-1)k^{h-1}}{k^h - 1}, \quad k > 1. \quad (3)$$

According to (1), for a binary tree with the height of 5, if one of the child nodes of root is a failure, 14 nodes would be affected. However, for a multi-tree with 5 roots, if one of the child nodes of one root is a failure, only 2 nodes will be affected (Fig. 3). In addition, the robustness of a tree is proportional to the ratio  $\delta$ , because only the nodes which have child nodes can affect their child nodes, and the leaf nodes do not have any child node. According to (3), when the height  $h$  is lower, the ratio  $\delta$  is bigger, that the main tree is more robust.

We give the proof that the function  $\delta(h)$  is monotone decreasing with  $h$ , for  $h > 1$ . Suppose that  $h_1 < h_2$ , we calculate  $\delta(h_2) - \delta(h_1)$  from (3):

$$\delta(h_2) - \delta(h_1) = \frac{-(k-1)k^{h_1-1}(k^{h_2-h_1} - 1)}{(k^{h_2} - 1)(k^{h_1} - 1)} < 0,$$

which shows that the function  $\delta(h)$  is monotone decreasing.

When an end user wants to join the structure, it can receive the CCC information from SN. After finding a CCC, the end user sends its join request to the CCC, which will subscribe to the multicast group inside CCC on behalf of this member. More details on the construction of the lower-tier multicast trees can be seen in the next section.

### THE HIGHER-TIER OVERLAY TREE

In this section, we describe the higher-tier overlay tree in our proposed OMICC in detail.

For the higher tier, there is an overlay tree, which each peer is CCC and is uniquely identified by a tuple of several attributes, like: (IP address, port number, network coordinate, capacity). In a typical streaming, data distribution is scheduled to commence at a specific time. Prior to this instant, the

CCCs and the SN organize themselves into an initial data delivery tree. Because the Cloud Computing Centres provide services for all the application, we assume that the SN and every CCC can obtain all information of each CCC, including the ID of CCC, the IP address, etc.

The multicast overlay network is the network induced by the CCCs. It can be modelled as a complete directed graph, denoted by  $G = (V, E)$ , where  $V$  is the set of vertices and  $E \subseteq V \times V$  is the set of edges  $e$ . Each vertex  $v$  in  $V$  represents a CCC or the SN. The directed edge  $e = \langle i, j \rangle$  from node  $i$  to node  $j$  in  $G$  represents the unicast path from  $CCC_i$  to  $CCC_j$  in the physical topology. Because of the fast data processing capabilities and plenty of bandwidth of the CCCs, we ignore the data processing time in each CCC node, and we do not have to consider the degree of each CCC node. The latency of an edge  $e$  corresponds to the unicast path latency from  $CCC_i$  to  $CCC_j$ , denoted by  $c(i, j)$ . The outgoing access link capacity of  $CCC_i$  is denoted by  $p_i$ , the source injects traffic at the rate of  $P$  per second. Every  $CCC_i$  can send data to at most  $d_i = \lfloor p_i/P \rfloor$  other CCCs. This imposes an out-degree bound at  $CCC_i$  on the  $G$ . Thus the problem is to find a multicast scheme which minimizes the delay by which all the CCCs receive data from SN. Such scheme corresponds to an ordered directed tree  $T = (V, E')$ , rooted at SN, and the degree of the node  $v$  in the tree is denoted by  $\deg_T[v]$ , (note that  $\deg_T[v] \leq \max_i d_i[v]$ ). The algorithm is given below.

#### Algorithm 1 Build the overlay tree.

Input:  $G = (V, E)$ ,  $c(i, j)$ , SN,  $T = \emptyset$

Output:  $T = (V, E')$

Step 1: set SN to be the root of  $T$ ;

Step 2: for every  $v \in V$ , do calculate the  $R(v)$

$$R(v) = \begin{cases} c(\text{SN}, v), & \langle \text{SN}, v \rangle \in E, \\ \infty, & \langle \text{SN}, v \rangle \notin E; \end{cases}$$

if  $R(v) = \min_{w \in V \setminus T} R(w)$  set  $v \in T$  and  $\langle \text{SN}, v \rangle \in E'$ ;

Step 3: for  $v \in V \setminus T$ , do update  $R(v)$

$$R(v) = \min_{w \in T} [R(w) + c(w, v)];$$

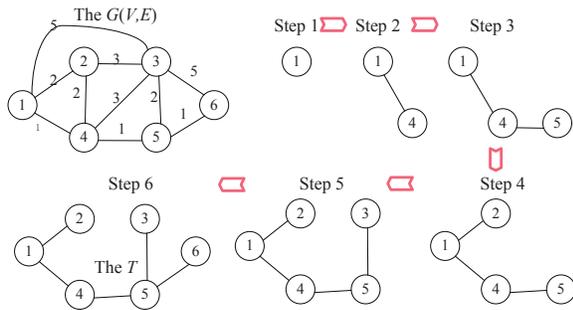
add  $v \in T$  and  $\langle w, v \rangle \in E'$  if  $R(v)$  is minimum;

Step 4: Repeat 3 until all node  $v \in V$  has been added to  $T$ ;

Step 5: Output  $T$ .

**Table 1** End user ( $e$ ) information.

Field	Description
CCC ID	The identification of $e$ 's current CCC
VM ID	The identification of $e$ 's current VM
Parent node ID	The identification of $e$ 's current parent node
Distance info	Distances between $e$ 's candidate parent nodes and $e$
Parent starting time	The moment when the core cloud receives any parent message from $e$
Children list	The list of $e$ 's current children



**Fig. 4** The example of building the overlay tree.

For example in Fig. 4, the first graph describes a completely connected graph with weight, denoted by  $G = (V, E)$ . There are five nodes in  $G$  with SN as the node 1. The latency of an edge corresponding to the unicast path latency is denoted as weight in the graph. We can obtain the multicast tree  $T$  which minimizes the delay by which all the nodes receive data from SN using Algorithm 1. Fig. 4 also gives the steps to construct the tree  $T = (V, E')$ .

As we know, Cloud Computing Centre must ensure its continuous stability for their service. With the support of the large enterprises or governments, Cloud Computing Centres can have the higher level of technology and capital to ensure their stable service. So the probability of each CCC failure as a node in the overlay tree is very low. We do not need to present the recovery method of the overlay tree structure under the condition of one CCC node failure.

**THE LOWER-TIER STRUCTURE**

In this section, we describe the lower-tier structure in our proposed OMICC in detail. This model contains two types of nodes, i.e., cloud virtual machine and end host. Each end host belongs to some CCC service domain, which is led by a virtual machine (VM) in the clouds. The data source delivers the multicast data to CCC heads on demand or in advance, and these heads each buffer all the received data packets. Note that CCC heads usually are in the

clouds at different locations.

OMICC collects and saves the information on each end user (described in Table 1). In most existing ALM protocols, the newcomer measures the distances from the candidate parent nodes to itself during its joining procedure. Note that the newcomer parent node is chosen among newcomer candidate parent nodes. In this paper, we use  $d(e_i, e_j)$  to denote the distance from  $e_i$  to  $e_j$ . In the application layer multicast, dynamic host nodes forward the received data packets to their downstream nodes. Hence a pair of neighbour nodes needs to know the aliveness of each other. Additionally, OMICC needs to know when end users leave the multicast session, to compute the contributions of the end users and optimize the multicast tree structure. OMICC meets the above requirements through a periodical aliveness detection process. In the process, an end user stops forwarding received data packets to its child node if it finds that it is not alive. An end user seeks for new parent node if it finds that its parent node is not alive. Note that an end user actively sends detection message to its parent node if it receives no aliveness detection messages from its parent node during a designed time interval.

The messages related to the end user management are as following:

**Login(SN,  $e_i, F$ ):** when an end user  $e_i$  wants to join the structure, it first logs in (as a registered user or an anonymous user) the system by sending the message to the SN. In the message,  $F$  means the local information of the user. Note that when SN receives this message, it will find the closest CCC, and send the CCC information to the end user  $e_i$ .

**Login(CCC,  $e_i, F$ ):** when an end user  $e_i$  receives the message from SN, it will find the CCC and join the CCC domain. The CCC will send the message which VM can serve it to  $e_i$ .

**Myparent(VM,  $e_i, p$ ):** when an end user  $e_i$  finds its parent node (including new parent node)  $p$ , it sends the message to VM. And the VM updates the related information when it receives the message.

**Alivenessask( $e_i, e_j$ ):** an end user  $e_i$  sends the

message to other user  $e_j$  for deciding whether  $e_j$  is alive or not.

Alivenesses( $e_i, e_j$ ): when receiving Alivenessask( $e_i, e_j$ ),  $e_j$  sends the message to  $e_i$  to claim that it is alive.

Algorithm 2 describes the lower-tier multicast structure building algorithm. Note that the procedure ignores the joining procedure of relay cloud nodes. From the algorithm, we can notice that the end users first attempt to obtain the multicast data from the VM nodes. Since the VM nodes are stable, the above feature can effectively improve the multicast performance.

**Algorithm 2** The newcomer  $e$  joins the lower-tier multicast structure.

Input:  $T, f(e)$   
Output:  $T \cup \{e\}$

- Step 1:  $e$  sends join(SN,  $e$ ) to the SN;
- Step 2:  $e$  receives the information of the CCC and VM;
- Step 3: if  $f(e) = 0$ , then  $e$  joins VM, and becomes a leaf node of  $T \cup \{e\}$ ;
- Step 4: Return;
- Step 5: if  $|\text{VM}(\text{SN})| \geq 1$ , then returns  $R(|\text{VM}(\text{SN})|)$  to  $e$ ; ( $R$  is the set of node which is not the node SN or newcomer in  $T$ )
- Step 6: for  $p \in R(|\text{VM}(\text{SN})|)$  do measure the distance  $d(p, e)$ ;
- Step 7: Find the closest node  $c$  in  $T$  such that  $d(c, e) = \min\{d(k, e) \mid k \in R(|\text{VM}(\text{SN})|)\}$
- Step 8:  $e$  sends join( $c, e$ ) to  $c$  and becomes the child node of  $c$ ;
- Step 9: else  $e$  joins new VM;
- Step 10:  $T \leftarrow T \cup \{e\}$ ;

**EXPERIMENTAL RESULTS**

In this section, we evaluate OMICC using NS2<sup>18</sup> as the simulator. We use the GT-ITM Generator<sup>3</sup> to create a 5000-node transit-stub graph as our underlying network topology. The cloud nodes are fixed at different stub-domain nodes, and 600 end users are located randomly among the other stub-domain nodes. In our experiments, the fanouts of VM and end user nodes are valued by a random integer from 2–5.

We first investigate the data distribution delays in different groups. Fig. 5 gives the experiment results. We can notice that the data distribution delay rapidly increases with the growth of group size from 100–600.

We introduce three definitions, the peer transfer failure probability  $P$ , latency optimizing ratio  $L$  and

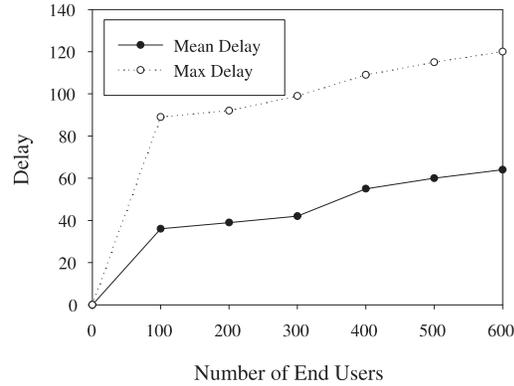


Fig. 5 Distribution delay in different groups of end users.

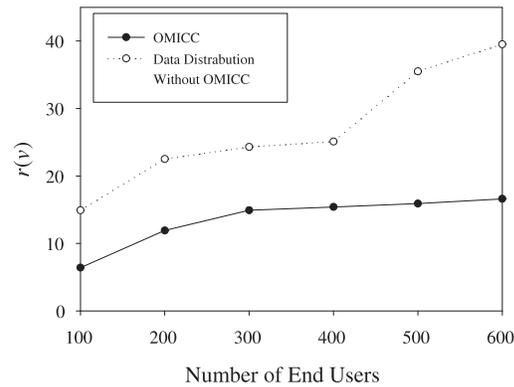


Fig. 6 Transfer time ratios in different groups of end users.

transfer time ratio  $r(v)$ . The peer transfer failure probability means the probability that an end user fails to offer transfer data distribution when it is scheduled to do it. By default, the peer transfer failure probability is  $P = 0.05$ . The latency optimizing ratio denotes the ratio of the mean latency of a multicast tree to that optimized by OMICC. The transfer time ratio is defined as

$$r(v) = \frac{t_{\text{OMICC}'(v)}}{t_{\text{OMICC}(v)}}$$

where  $v$  means the event that the parent of an end user  $e_v$  becomes invalid node in the multicast session,  $t_{\text{OMICC}'(v)}$  and  $t_{\text{OMICC}(v)}$  denote the duration times from the moment that  $v$  happens to the moment that  $e_v$  re-receives the data packets in terms of OMICC and the main data distribution approach of OMICC, respectively.

Fig. 6 shows the transfer time ratios in 9 different multicast sessions. With the growth of the group size, the transfer time ratio tends to increase. The main reason of the above phenomenon can be

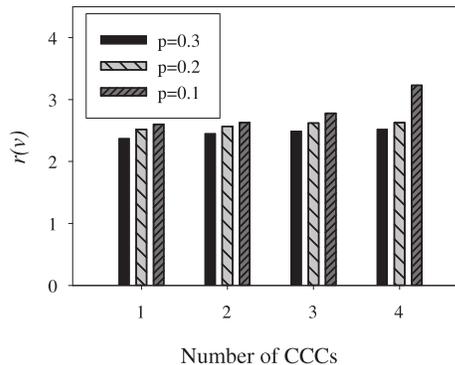


Fig. 7 Transfer data distributions with different  $P$  values.

explained as follows. As the group size increases, more and more transfer data distributions are performed by end users, and the end user spends longer time to reconnect to the multicast tree in the worst case. The minimum transfer time ratio is not less than 1, which occurs when some end user close to the core cloud reconnects to the multicast tree or some scheduled end user cannot offer transfer data distribution. In the case where end users have high reliability of offering transfer data distributions, the difference between the transfer data distributions provided by the end user and relay cloud is not obvious. Hence the transfer time ratio is not significantly improved with the growth of the number of clouds. When end users have low reliability of offering transfer data distribution, more relay clouds can effectively improve the performance of the transfer data distributions (Fig. 7).

## CONCLUSIONS AND FUTURE WORK

This paper presents a new overlay multicast infrastructure based on cloud computing for streaming media data distribution, called OMICC. OMICC has two-tier multicast structure. In the higher-tier, there is an overlay tree which is organized by the set of Cloud Computing Centres, called CCCs. In the lower-tier, each CCC is a set of roots of multicast trees. Because of the good properties of CCCs, the innumerable CCCs can effectively serve many end users for streaming media data distribution applications. OMICC introduces a new approach for optimizing the performance of the overlay multicast, i.e., building the logical multicast tree with the cloud computing centres and improving the overlay multicast performance in terms of the optimization of the logical multicast structure.

In our future work, we plan to deploy OMICC in Shandong Cloud Computing Platform (developed

and operated by Shandong Computer Science Centre) in China to further evaluate its performance through wide trial applications.

**Acknowledgements:** This work was supported by Shandong Provincial Natural Science Foundation of China (Grant no. ZR2015YL019). This work was also supported by the National Natural Science Foundation of China (Grant no. 61272433 and no. 61472230) and International Science and Technology Cooperation Programme of China (Grant no. 2012DFR10500).

## REFERENCES

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin A, Stoica I, Zaharia M (2009) Above the clouds: a Berkeley view of cloud computing. EECSS Department, Univ of California, Berkeley, Rep. UCB/EECS-2009-28.
2. Bonald T, Massoulié L, Mathieu F, Perino D, Twigg A (2008) Epidemic live streaming: optimal performance trade-offs. *Perform Eval Rev* **36**, 325–36.
3. Zegura EW, Calvert KL, Bhattacharjee S (1996) How to model an internetwork. In: *Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies: Networking the Next Generation (INFOCOM'96)*, pp 594–602.
4. Pendarakis D, Shi S, Verma D, Waldvogel M (2001) ALMI: an Application Level Multicast Infrastructure. In: *Proceedings of the 3rd Conference on USENIX Symposium on Internet Technologies and Systems (USITS'01)* vol 3, p 5.
5. Banerjee S, Kommareddy C, Kar K, Bhattacharjee B, Khuller S (2006) OMNI: an efficient overlay multicast infrastructure for real-time applications. *Comput Netw* **50**, 826–41.
6. Francis P (1999) Yoid: extending the internet multicast architecture, [www.icir.org/yoid/docs/ycHtmLL/htmlRoot.html](http://www.icir.org/yoid/docs/ycHtmLL/htmlRoot.html).
7. Li X, Striegel AD (2007) A case for passive application layer multicast. *Comput Netw* **51**, 3157–71.
8. Strufe T, Schafer G, Chang A (2006) BCBS: an efficient load balancing strategy for cooperative overlay live-streaming. In: *Proceedings of the 2006 IEEE International Conference on Communications (ICC'06)*, pp 304–9.
9. Zhang B, Jamin S, Zhang L (2002) Host multicast: a framework for delivering multicast to end users. In: *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, pp 1366–75.
10. Chu YH, Rao SG, Seshan S, Zhang H (2002) A case for end system multicast. *IEEE J Sel Area Comm* **20**, 1456–71.
11. Tran DA, Hua KA, Do T (2003) Zigzag: an efficient peer-to-peer scheme for media streaming. In: *Proceedings of the Twenty-Second Annual Joint Conference*

- of the *IEEE Computer and Communications Societies (INFOCOM 2003)*, pp 1283–92.
12. Trajkovska I, Salvachua Rodriguez J, Mozo Velasco A (2010) A novel P2P and cloud computing hybrid architecture for multimedia streaming with QoS cost functions. In: *Proceedings of the 18th ACM International Conference on Multimedia (MM'10)*, pp 1227–30.
  13. Payberah AH, Kavalionak H, Kumaresan V, Montresor A, Haridi S (2012) CLive: Cloud-assisted P2P live streaming. In: *Proceedings of the 2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*, pp 79–90.
  14. Jin X, Kwok YK (2010) Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers. In: *Proceedings of the 2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, pp 800–5.
  15. Islam S, Grégoire JC (2012) Giving users an edge: a flexible Cloud model and its application for multimedia. *Future Generat Comput Syst* **28**, 823–32.
  16. Yu YF, Lai KC (2010) A semi-structured overlay for multi-attribute range queries in Cloud computing. In: *Proceedings of the 2010 IEEE 13th International Conference on Computational Science and Engineering (CSE)*, pp 88–95.
  17. Fouquet M, Niedermayer H, Carle G (2009) Cloud computing for the masses. In: *Proceedings of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities (U-NET'09)*, pp 31–6.
  18. Fall K, Varadhan K (2007) The network simulator NS2, [www.isi.edu/nsnam/ns](http://www.isi.edu/nsnam/ns).