Ant-inspired Mini-Community-based Solution for Video-On-Demand Services in Wireless Mobile Networks

Changqiao Xu, *Member, IEEE,* Shijie Jia, Lujie Zhong, Hongke Zhang and
Gabriel-Miro Muntean, *Member, IEEE*

***Abstract*—Highly efficient distribution and management of media resources and fast content discovery are key determinants for mobile peer-to-peer video-on-demand solutions, especially in wireless mobile networks. Virtual communities making use of users' common characteristics such as interest and interaction to describe the boundary of sharing content and objects are a promising avenue for high-efficiency resource sharing. In this paper, we propose a novel Ant-inspired Mini-Community-based Video sharing solution for on-demand streaming services in wireless mobile networks (AMCV). AMCV relies on a newly designed two layer architecture and on an algorithm inspired from the indirect communication between ants via pheromone trails which enables them to discover and use shortest paths. The architecture is composed of a mini-community network layer and a community member layer. The ant-inspired algorithm enables finding the common interest of users in video content within large amounts of pseudo disorderly interactive behavior data. AMCV proposes an Ant Colony Optimization (ACO)-based community communication strategy that dynamically bridges communities to support fast search for resources. AMCV achieves high scalability by making use of a designed community maintenance mechanism to uniformly distribute the maintenance cost of members and resources in the community, according to various member roles. Simulation-based testing shows how AMCV outperforms another state of the art solution in terms of a wide set of performance metrics.**

***Index Terms*—Virtual community, mobile peer-to-peer, Video-on-Demand, interactivity, ant colony optimization**

I. Introduction

LATELY, it has become highly popular to access rich-content multimedia services via the Internet from mobile devices with significant computation and communication resources [1]-[4]. The increasing communication capabilities in wireless mobile networks provide the high-bandwidth required by these multimedia services. Video-on-Demand (VoD) supports stream interactivity for the users and has become the most popular multimedia service. Yet, the highly dynamic characteristics of wireless mobile networks and user viewing behavior, including topology, the use of multi-hop communications and random access of video content, affect the content sharing performance [5]-[7]. Different from Internet-based P2P technologies [8]-[13], mobile peer-to-peer (MP2P) networking focuses on improving the performance of large-scale mobile VoD in wireless mobile networks [14]-[20]. However, enabling efficient MP2P-based video content sharing with user interactivity, while providing high user quality of experience (QoE) is very challenging in the resource-constraint wireless mobile environments.

The *virtual community* concept is a promising avenue for resource sharing; it makes use of innovative algorithms, which based on user common characteristics such as interest and interaction with the content, describe the boundary between shared resources, enhance sharing efficiency and reduce the unnecessary energy consumption. For instance, *Tu et al.* [1] proposed an architecture of collaborative content fetching for a group of mobile users which are geographically close for a period of time and fetch the same resources. These subscribers form a small community and collaboratively fetch and efficiently share the resources. *Chen et al.* [20] proposed a community-based P2P file sharing system in disconnected mobile ad-hoc networks (SPOON). The nodes with common interest for content and which frequently interact with each other are grouped into a community, and SPOON achieves fast location of available resources and reduces the duration of the search process. However, the existing community-based resource sharing solutions are unsuitable for constructing a VoD-oriented Virtual Community (VVC). This is because their resource sharing approaches not only cannot handle the dynamic user playback behavior, but also are associated with high maintenance cost for the community members.

Ant Colony Optimization (ACO)-based solutions rely on a relatively new concept which is inspired from closely observing the foraging behavior of ants. Ants use a chemical substance called pheromone for indirect communications between them, indicating the path towards the food, which then attracts high number of individuals [21]-[23]. The foraging behavior of ants and user interactivity are very similar, namely the viewers get attracted to the already highly popular video clips. By

This work was supported in part by the National Key Basic Research Program of China (973 Program) under Grant No. 2013CB329102; the National Natural Science Foundation of China (NSFC) under Grant Nos. 61372112, 61232017; the Beijing Natural Science Foundation of China under Grant No. 4142037, the Fundamental Research Funds for the Central Universities and in part by Lero under Grant no. 10/CE/I1855.

C. Xu is with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China. (e-mail: cqxu@bupt.edu.cn).

S. Jia is with the Academy of Information Technology, Luoyang Normal University, Luoyang, Henan, China. (e-mail: shjjia@gmail.com).

L. Zhong is with the Information Engineering College, Capital Normal University, Beijing, China. (zljict@gmail.com).

H. Zhang is with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing, China. (e-mail: hkzhang@bjtu.edu.cn).

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Network Innovations Centre, Rince Institute, Dublin City University, Dublin, Ireland. (e-mail: gabriel.muntean@dcu.ie).

making use of ACO, the common characteristics of members from swarm behavior can be analysed and the common user interest from pseudo disorderly interactive behavior can be extracted. Then by clearly describing the boundary of user behavior, the efficiency of resource sharing can be improved.

This paper proposes **a novel Ant-inspired Mini-Community-based Video sharing solution (AMCV)** for on-demand services in wireless mobile networks. AMCV relies on a two layer architecture which consists of a lower community member layer and an upper mini-community network layer. AMCV constructs multiple mini-communities based on the video content to distribute the resource maintenance costs and enhance the overall solution scalability. A novel ACO-based community communication strategy that dynamically bridges communities according to the movement behavior of community members is proposed. This strategy reduces both the data traffic and delay of the resource search. A novel community maintenance mechanism that assigns the role and task of members in terms of the interest level in the video content is introduced. This mechanism balances the load across nodes and enhances the utilization efficiency of resources. Simulation results show how AMCV achieves lower average number of hops for request delivery, lower average delay for resource search, higher average lookup success rate, lower server stress, higher average playback continuity and lower node load in comparison with another state of the art solution.

II. Related Work

There have been numerous studies on P2P-based video resource sharing solutions supporting user interactivity for VoD services in recent years [24]-[33].

The traditional solutions focus on the construction of a content distribution topology. For instance, we have previously proposed a multi-way tree structure (SDNet) [10] and balanced binary tree-based strategy (BBTU) [33] for unstructured media distribution in P2P networks along with novel algorithms to address VCR operations-related issues. By making use of additional prefetching buffers, video segments are prefetched and stored in the nodes along a tree in distributed manner in order to support interactivity. An unstructured network is established to support streaming between nodes by using the buffer overlapping mechanism. However, any increase in the node churn, which is often associated with mobile wireless networks, requires reconstruction of the tree structure and determines an increase in the overhead. In VMesh [9], videos are divided into chunks and stored in local storage at the nodes in distributed manner. In order to eliminate the impact of user interactivity such as random forward/backward seek, pause and restart during playback and ensure playback continuity, VMesh utilizes the total storage capacity of peers and a DHT-based network solution to improve the supply of video segments and support the interactive demands in a scalable manner. These structured P2P-based solutions result in high efficiency resource search and support continuous playback. Unfortunately, the increase in the numbers of nodes in these structures determines a high increase in the cost of maintaining the structures, which becomes the bottleneck in system's scalability.

Some scholars have proposed mesh-based solutions with an unstructured topology to solve the system scalability problem. For instance, by adopting a mesh-based topology, Chang *et al.* [11] introduced a interleaved video frame distribution scheme (IVFD) to handle full VCR operations. However, by making use of gossip messages to search for resources, the mesh-based topology cannot support fast supplier discovery. The low performance of resource lookup reduces the QoE of user interactivity which requires fast location of the suppliers for the new video clips.

Recently, the community-based file sharing schemes have attracted increasing research interests from various researchers, which have used them to balance the system scalability and resource search performance. *C5* [1] proposed an architecture of collaborative content fetching by grouping the mobile users which fetch the same content and are near to each other. The nodes in close vicinity form a group and use a high-speed WLAN to share local resources with other requesting members by making use of multicast at MAC layer. *C5* makes use of idle WLAN interfaces for in-community communication to speed up the fetching rate for community members. However, *C5* relies on the premise that a number of mobile subscribers are close to each other for a period of time and request the same content, which is not true most of the time in mobile environments. SPOON [20] includes an interest extraction algorithm to refine node interests from the content-based file search process and groups the nodes which have common interests and frequent interaction into communities. SPOON formulated a role for the community member in terms of handling the file search both intra-community and inter-community. Moreover, SPOON proposed an interest-oriented file searching scheme for high-efficiency resource search. However, the similarity between files stored by nodes cannot accurately estimate the similarity level of nodes' interest and clearly define the community boundary, so this has a negative influence on file search efficiency. Importantly SPOON neglects the mobility of nodes and therefire the increasing geographical distance between community members may reduce the content delivery performance. Moreover, SPOON's static community construction strategy is unsuitable for coping with the real time dynamicaly changing characteristics of VoD systems. Additionally, these solutions which use static resource sharing strategies cannot adapt to the dynamic resource requests of users and real-time delivery requirements of video data. Moreover, the increasing community scale increases the maintenance overhead of members, making the solution inefficient.

Some group-based video sharing solutions like mini-community structure which cluster the nodes carrying similar resources together were proposed. These solutions make use of the advantages of a group structure to achieve high video sharing efficiency and low maintenance cost. In SURFNet [13], stable peers are used to construct an AVL tree to provide video superchunk-level data availability information. Other peers storing the same superchunk data are grouped into a holder-chain. The holder-chain is then attached to the stable node in the AVL tree, which is the head of the corresponding holder-chain. By using this structure, SURFNet can support nearly-constant and logarithmic search time for seeking within a video stream and jumping to a different video content, respectively. However, in SURFNet, the stability of the AVL tree highly depends on the premise that the tree consists of stable nodes, so the flexibility of the VoD delivery system is highly restricted. Additionally, with the increase in scale, the tree becomes larger, and the maintenance of the tree structure also increases very much. VOVO [26] groups the peers which request the same video segment into a session and uses batching and patching to serve asynchronous peers in each session. VOVO makes use of the peer groups to improve the hybrid resource caching and reduce the waiting latencies of the requesting peers. However, VOVO relies on batching and patching to handle the movement between sessions and so introduces high start delays. VOVO neglects the management mechanism of resources and construction of community network, resulting in low utilization efficiency of resources and low resource search performance. We have previously proposed a solution for QoE-driven user-centric VoD services in urban vehicular network environments (QUVoD) [14]. QUVoD designed a novel grouping-based storage strategy which distributes uniformly the video segments along a Chord overlay. The peers which store similar video segments in the Chord overlay form a group, reducing segment seeking traffic while also enabling load-balancing. However, the increasing number of nodes in the Chord overlay also becomes the bottleneck in QUVoD's scalability. SURFNet and QUVoD make use of the tree and Chord structure to achieve fast location of resources, but the high maintenance cost brought by these structures impacts on the system's scalability.

III. AMCV Architecture Overview

AMCV focuses on constructing an MP2P-based video content sharing solution in wireless mobile networks, which supports user interactivity and efficient management and search for resources. AMCV relies on grouping nodes in MP2P networks into communities and formulating the communication approach between communities to address the problems of high maintenance cost of MP2P networks and low efficiency of resource search. The AMCV architecture, illustrated in Fig. 1, organizes a media server and multiple nodes in a structure with two layers: a mini-community network layer and a community member layer. AMCV defines the community structure and logical links between communities in the mini-community network layer, which can achieve fast resource search and low-cost link maintenance. AMCV defines the roles and tasks of the community members in the community member layer, in charge with their economic management. AMCV balances the maintenance cost and efficiency of search for resources to support large-scale VoD services and high QoE for the users in wireless mobile networks.

The media server is the original repository for all video content, is well-known to all nodes and provides delivery services when the video resource sharing between nodes is unavailable for some reason. The video content saved at the media server is normally divided into n chunks with equal length, namely $video \Leftrightarrow (chunk_1, chunk_2, ..., chunk_n)$. The

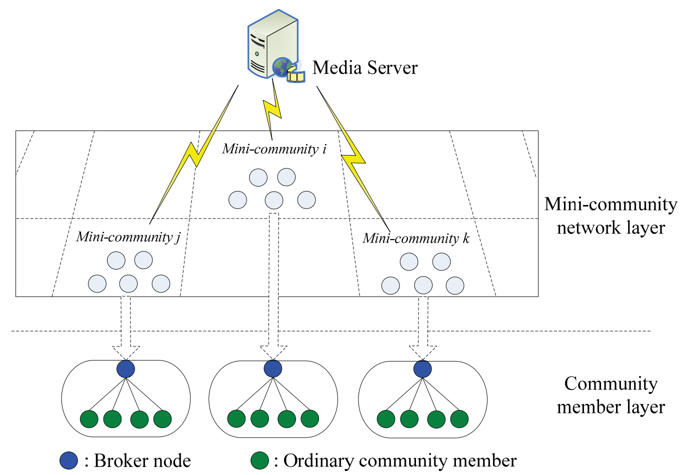


Fig. 1. AMCV two-layers architecture.

server maintains a node queue to record the nodes' initial information (i.e. time of entering the system and playback position). For AMCV's scalability purposes, the server does not maintain any real-time state of nodes in the queue. Instead, it plays the role of an agent between the mobile nodes and the overlay network to assists mobile nodes join into the corresponding communities.

In the **mini-community network layer**, the nodes playing the same video chunk form a mini-community and share resources (chunks of video content) with each other. By making use of this structure, AMCV replaces the need for complex resource management and large overhead involving nodes from the whole overlay network with the maintenance of multiple mini-communities including relatively low number of nodes, enabling also flexibility and scalability of the solution. User interactivity drives the movement of community members between communities so that communities contact each other to reduce the movement cost (i.e. start latency). Maintaining real-time connections between all communities not only consumes large network resources, but also has high maintenance cost. Inspired by ACO, by extracting the common characteristics from the movement behavior of community members, AMCV communities which have tight contact with each other maintain their inter-connection, while the others do not, saving resources. The close related communities dynamically update the connected objects according to the interest variation of members for the video content.

In the **community member layer**, the members of each community form a P2P overlay network which achieves autonomous management in terms of assigned roles and tasks. A member in a community which has higher interest level in the video content than other community members is selected as the broker node; this avoids frequent replacements of the broker node. The broker node is responsible for collecting the information and movement behavior of members for high-efficiency maintenance of resources in the community and discovery of interests in real time. It manages the contact with other community broker nodes and maintains real-time connections with corresponding communities. In order to

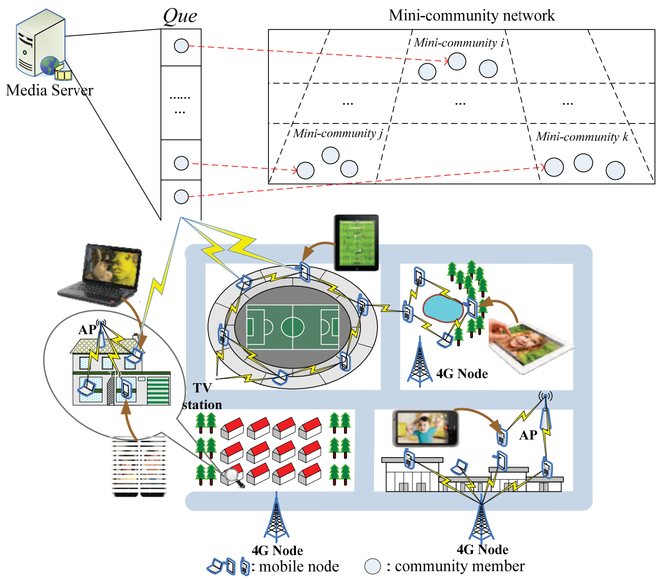


Fig. 2. Initial agent function of server.

balance the maintenance load of communities, the ordinary nodes participate in community management, as it will be described in the next section.

IV. AMCV Detailed Design

A. Media server

Table I defines several notations which are used in this section and the following ones. The media server maintains a list with the nodes which have joined the AMCV system $Que \Leftrightarrow \{n_1, n_2, ..., n_v\}$ for the purpose of providing initial supplier for future new joining nodes, as Fig. 2 shows. When a mobile node N_i joins the system, the server adds N_i to Que. Any item n_c in Que can be defined as a 3-tuple $n_c \Leftrightarrow (ID_c, et_c, pt_c)$ where ID_c is the n_c's ID, et_c and pt_c are the joining system time and the required playback position from within the requested video for n_c. The server does not maintain real-time playback state of any item in Que. Each n_c in Que has one of the two states: *available* or *unavailable*, defined in terms of the video resource duration as indicated in eq. (1).

$$\begin{cases} available, & scTime - et_c < n \times len - pt_c \\ unavailable, & scTime - et_c \geq n \times len - pt_c \end{cases} \tag{1}$$

where $n \times len$ denotes the total length of the required video and $scTime$ is the system current time. $scTime - et_c < n \times len - pt_c$ indicates that the n_c's playback position is still within the video duration, and therefore node n_c can be used to provide video delivery services to other nodes. $scTime - et_c \geq n \times len - pt_c$ indicates that node n_c has left the system and should be removed from Que. The server considers that all items in Que perform sequential playback as it cannot do a more detailed real-time maintenance of the playback status of each item for scalability reasons. Once the N_i joins the system, the server's first concern is to search for an initial supplier which matches the requested playback position of the new mobile node (i.e. falls into the supplier's playback buffer range). This check is performed using the Buffer Overlapping Mechanism (BOM) described in [33]. BOM verifies the degree with which the playback buffers of the streaming service requester and supplier overlap and if so they have close to synchronous playback situations or not.

TABLE I
NOTATIONS USED BY AMCV

Notations	Descriptions
N_i	a member i in a mini-community
C_x	a community x in mini-community network
n	total number of video chunks
$chunk_i$	a video chunk i
len	length of a video chunk(seconds)
NL_x	nodes list in C_x
CL_x	connection list in C_x
$esTime_i$	time of N_i joining the system
$egTime_{ix}$	time of N_i joining C_x
$lgTime_{ix}$	time of N_i leaving C_x
$scTime$	current time of system at server side
PT_b	threshold of a community building dynamic connection
PT_r	threshold of a community removing dynamic connection
UT	period time of community updating dynamic connection
Que	A node queue maintained by the media server
$count()$	function which calculates the number of elements in a set

Algorithm 1 Search for an initial content supplier when N_i joins AMCV

```
let rsSet be an empty result set
for(c = 0; c < count(Que); c++)
    n_c = Que[c];
    if (n_c meets N_i's request playback point by BOM)
        add n_c to rsSet;
    end if
end for c
if (rsSet == NULL)
    the server delivers directly the video data to N_i;
    goto step 2 with new request playback point of N_i;
else
    let n_j from rsSet be N_i's initial supplier;
    N_i connects to n_j and gets data and broker node ID;
end if
```

In order to enhance the lookup success rate of supplier, the server selects several items from Que as potential suppliers according to BOM and returns these items to N_i. If the server does not provide any available suppliers or the supplier(s) provided by the server rejects the request for data due to change in its playback position, the server provides the video streaming service to N_i while it continues to search for a new supplier for N_i. Algorithm 1 describes the search for an initial content supplier process when N_i joins AMCV. The server searches for supplier candidates in Que which have synchronous playback situations with N_i according to BOM. If the server cannot search for the appropriate supplier candidates, it provides requested video content to N_i and continues to search for the available supplier candidates in Que. Otherwise, if the server finds the supplier candidates, it returns the information of these candidates to N_i. N_i selects the appropriate supplier n_j from the candidates provided by the server according to the supplier selection approach described in our previous work [18]. When N_i connects successfully with a supplier n_j, N_i receives both the streaming data and ID of the broker node in the corresponding community from n_j and joins that community. N_i's supplier reports the information of N_i to the broker node directly or indirectly, as detailed in sub-section C of this section.

B. Mini-community network layer

As mentioned, the nodes which request the same video chunk are grouped into a mini-community. This community-based approach enables the nodes in the overlay network decentralize information storage, control and distribution, enhancing the scalability and flexibility of the system and uniformly distributing the maintenance load. As the video playback position changes dynamically, N_i gets associated with a community or another. The community needs to build the contact with other communities for fast movement of members. We propose an ACO-based community communication strategy which formulates the construction and regulation approaches of connection between communities to balance the maintenance cost of community connection and movement rate of members between communities.

1. Connection classification:

Let we consider community C_x, whose members access $chunk_x$, which is the x^{th} video chunk in a video stream. We use a 2-tuple to denote the community relationship, as follows: $C_x \Leftrightarrow (chunk_x, NL_x)$ where $NL_x = (n_1, n_2, ..., n_k)$ is the list of members associated with the community C_x and any item n_c is expressed as 1-tuple $n_c \Leftrightarrow (ID_c)$ containing the ID of the node. C_x is maintained by the broker node of community. We define two movement behaviors for the members in terms of distance between current and target communities. The distance of communities is defined as in eq. (2):

$$dis(N_i) = CurrentComm(N_i) - TargetComm(N_i) \quad (2)$$

where $CurrentComm$ and $TargetComm$ return the ID of video chunks corresponding to current and target communities of N_i, respectively.

(1) ***near-end movement***. If $|dis|$ is 0 or 1, the movement of N_i is considered to be a near-end movement. In this situation N_i moves within the current community or to the next/previous community considered in sequential video playback order.

(2) ***far-end movement***. If $|dis| \geq 2$, the movement of N_i is considered to be far-end.

C_x cannot build connections with all other $n-1$ communities due to the cost of maintenance of these connections which increases with the increasing value of n. Consequently, we define two kinds of connections in terms of the movement behaviors of members: static and dynamic connections.

(1) ***Static connection***. Any community needs to build static connections with its next and previous communities, as Fig. 3 shows. Similar to a double-linked circular chain, the communities have contact with each other via static connections. The motivation of building static connections is to support near-end movement of members and ensure the members can access any community by forwarding request messages via these static connections. We rule that static connections cannot be removed.

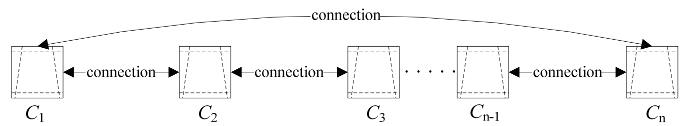


Fig. 3. Static connection of communities.

(2) ***Dynamic connection***. Dynamic connections are built between the current community and the other $n-3$ communities, (i.e. its next and previous communities are excluded). The motivation of building dynamic connections is to support fast supplier discovery for members' far-end movement. This is due to the fact that by using static connections to support far-end movement the delays of the forwarding request messages and request responses increase significantly. We rule that any dynamic connection can be created and removed.

C_x needs also to store information about both static and dynamic connections and therefore its structure is extended from a 2-tuple to a 3-tuple, as follows: $C_x \Leftrightarrow (chunk_x, NL_x, CL_x)$, where $CL_x \Leftrightarrow (inf_1, inf_2, ..., inf_v)$ is the connection list whose items inf_j indicate connections to C_j and are defined as 2-tuples $inf_j \Leftrightarrow (ID_j, type_j)$. ID_j is the ID of broker node of C_j and $type_j$ is a flag associated with the connection; if $type_j = 0$, inf_j is a static connection, if $type_j = 1$, the connection is dynamic. The broker node in a community acts as the connection manager to contact with the broker nodes of other communities, ensuring the direct communication with these communities.

2. Dynamic connection construction:

The process of users' viewing video is similar to the behavior of ants foraging for food. The movement of users' playback point from the current community C_x to another community C_y can be considered similar to that when ants are going through a path $path_{xy}$ using C_x and C_y as vertices and depositing a pheromone on $path_{xy}$. This path selection model for searching for the most interesting content is used in the ACO problem for optimal solution finding, as illustrated in Fig. 4. By analyzing the interactive behaviors of users, AMCV

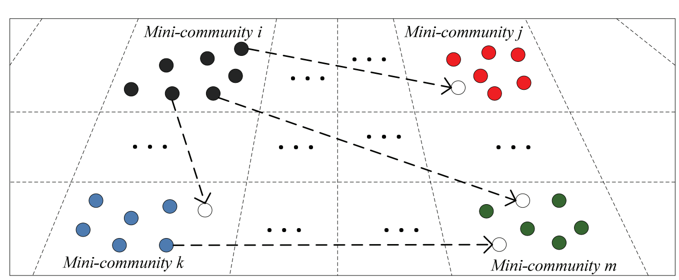


Fig. 4. Far-end movement of members in the community.

can extract the common interest of community members to accurately predict the future destination community.

The heuristic value and pheromones released by ants are key factors in the ACO model. The higher the number of ants passing on the same path, the stronger the pheromone generated is. This results in having higher probability for other ants to pass on the path with strong pheromones, namely the probability of an ant to select that path is high. By collecting the pheromone and heuristic values generated by the members performing the far-end movement in current community, the broker node builds and removes dynamic connections in terms of the probabilities of using these paths from the current community to the other $n-3$ communities.

1) **Determining Pheromone Level**. The pheromone levels are determined by the strength of the trace generated by far-end movement which includes three inherent factors: distance and direction of movement and popularity values of skipped content. This reflects the relevance of the link between video content chunks. Eq. (2) defines the approach for movement distance estimation. The direction includes two factors: "movement backward" and "movement forward". The "movement backward" refers to users viewing already watched content, indicating a high interest in the video content. The "movement forward" means that users skip content, denoting a low interest in the video content. The pheromone generated by "movement backward" should be greater than that of a "movement forward" due to the difference in viewer interest. When a member N_i performs a far-end movement from its community C_x to community C_y, the pheromone $\tau_{path_{xy}}$ generated by N_i is computed based on distances and direction of movement and popularity values of the skipped video chunks, according to eq. (3).

$$\tau_{path_{xy}} = \begin{cases} \operatorname{arccot}\left(\frac{1}{1-\sum_{k=a}^{|dis|} pop_k}\right), dis > 0 \\ \arctan\left(\frac{1}{1-\sum_{k=a}^{|dis|} pop_k}\right), dis < 0 \end{cases} \tag{3}$$

where $dis > 0$ and $dis < 0$ denote that the member performs "movement forward" or "movement backward", respectively. $pop_k \in [0,1]$ is the popularity of chunk k ($\sum_{k=1}^{n} pop_k = 1$) and $|dis|$ is the number of chunks neglected. Additionally, we neglect the condition with "$dis = 0$" which denotes that the member performs a movement in the current community. A long distance "movement forward" denotes that the user skips much video content. The more chunks with high popularity values are, the lower the interest in the video content is, and the weaker the pheromone generated is. A long distance "movement backward" indicates that the user views again much of the content. The more re-watched chunks with high popularity values are, the higher the interest in the video content is, and the stronger the pheromone generated is.

2) **Determining Heuristic Value**. The number of times a video sequence is viewed is one of the very important parameters for estimating users' interest level in the content. We investigate two time-based parameters generated by each member N_i performing far-end movement: average stay time per community $\overline{T_{ix}}$ before entering a community C_x and stay time $\overline{ST_{ix}}$ in C_x. By making use of the Information Theory model [34] to calculate the information entropy $I_x(N_i)$ and $E_x(N_i)$ generated by $\overline{T_{ix}}$ and $\overline{ST_{ix}}$, we use the information content $Gain_x(N_i)$ to denote the heuristic value brought by N_i at $path_{xy}$. For example, we assume that the playback trace of N_i includes L_i video chunks before playing $chunk_x$ (joining C_x) and the stay time in C_x of N_i is $\overline{ST_{ix}}$. Whereafter, N_i performs the far-end movement from C_x to C_y. $\overline{T_{ix}}$ can be calculated according to eq. (4).

$$\overline{T_{ix}} = \frac{egTime_{ix} - esTime_i}{L_i} \tag{4}$$

where $esTime_i$ and $egTime_{ix}$ are the time of N_i joining the system and community C_x, respectively and should be maintained by N_i. Consequently N_i-related information is extended from a 1-tuple to a 3-tuple by storing $esTime_i$ and $egTime_{ix}$, as follows: $N_i \Leftrightarrow (ID_i, esTime_i, egTime_{ix})$. $esTime_i$ is a constant and is set when N_i enters the system, only. $egTime_{ix}$ is variable, refers to the current community the member belongs to and requires update every time when N_i joins another community. "$egTime_{ix} - esTime_i$" is the online time of N_i before joining into C_x. For convenience, $\overline{T_{ix}}$ needs to be normalized according to eq. (5).

$$\overline{P_{ix}} = \begin{cases} \overline{P'_{ix}}, & 0 < \overline{P'_{ix}} < 1 \\ 1, & \overline{P'_{ix}} \geq 1 \end{cases} \quad , \quad \overline{P'_{ix}} = \frac{\overline{T_{ix}}}{len} \tag{5}$$

where len is the length of a video chunk (they are assumed to have equal size). $\overline{P'_{ix}} \geq 1$ means that "pause operation" or reviewing the content already watched lead to the time of N_i experiencing L_i video chunks be greater than $L_i \times len$. $\overline{P_{ix}}$ is the average playback time ratio per community of N_i playing L_i chunks. The information entropy $I_x(N_i)$ generated by $\overline{P_{ix}}$ is obtained as in eq. (6).

$$I_x(N_i) = \begin{cases} -\overline{P_{ix}} \log_2(\overline{P_{ix}}), & 0 < \overline{P_{ix}} < 1 \\ 1, & \overline{P_{ix}} = 1 \end{cases} \tag{6}$$

The approach of obtaining the information entropy generated by $\overline{ST_{ix}}$ is similar to $\overline{T_{ix}}$. $\overline{ST_{ix}}$ can be calculated according to eq. (7).

$$\overline{ST_{ix}} = lgTime_{ix} - egTime_{ix} \tag{7}$$

where $egTime_{ix}$ and $lgTime_{ix}$ is the time of N_i joining and leaving C_x, respectively. "$lgTime_{ix} - egTime_{ix}$" indicates the stay time of N_i in C_x. $\overline{ST_{ix}}$ also needs to be normalized according to eq. (8).

$$H_{ix} = \begin{cases} H'_{ix}, & 0 < H'_{ix} < 1 \\ 1, & H'_{ix} \geq 1 \end{cases} \quad , \quad H'_{ix} = \frac{\overline{ST_{ix}}}{len} \tag{8}$$

where H_{ix} is the playback time ratio of N_i in C_x. Similarly, $H'_{ix} \geq 1$ means that "pause operation" or reviewing the content watched caused by movement backward inside $chunk_x$ lead to the online time of N_i in C_x greater than len. The information entropy $E_x(N_i)$ generated by $\overline{H_{ix}}$ is obtained as in eq. (9).

$$E_x(N_i) = \begin{cases} -H_{ix}\log_2 H_{ix}, & 0 < H_{ix} < 1 \\ 1, & H_{ix} = 1 \end{cases} \tag{9}$$

The information content $Gain_x(N_i)$ calculated by $E_x(N_i)$ and $I_x(N_i)$ can be obtained as described by eq. (10).

$$Gain_x(N_i) = E_x(N_i) - I_x(N_i), Gain_x(N_i) \in (-1, 1) \tag{10}$$

"$Gain_x(N_i) \in (-1, 0)$" denotes that the level of interest of N_i to $chunk_x$ is lower than other L_i chunks, namely the level of interest of N_i to $chunk_x$ is falling. "$Gain_x(N_i) \in (0, 1)$" denotes that the level of interest of N_i to $chunk_x$ is increasing. The heuristic value $\eta_{path_{xy}}(N_i)$ at $path_{xy}$ can be calculated according to eq. (11).

$$\eta_{path_{xy}}(N_i) = \arctan(Gain_x(N_i)) + \frac{\pi}{2} \tag{11}$$

We assume that m members in C_x perform the far-end movement from C_x to other communities in a time interval where the $k(0 < k < m)$ members move from C_x to C_y. We obtain the total pheromone $\hat{\tau}_{path_{xy}}$ and heuristic value $\hat{\eta}_{path_{xy}}$ generated by k members at $path_{xy}$ according to eq. (12).

$$\begin{cases} \hat{\tau}_{path_{xy}} = \sum\limits_{c=1}^{k} \tau_{path_{xy}}(N_c) \\ \hat{\eta}_{path_{xy}} = \sum\limits_{c=1}^{k} \eta_{path_{xy}}(N_c) \end{cases} \tag{12}$$

By making use of an ACO-based probability model, the probability of any member moving from C_x to C_y, $PB_{path_{xy}}$ can be obtained according to eq. (13).

$$PB_{path_{xy}} = \frac{\hat{\tau}^{\alpha}_{path_{xy}} \times \hat{\eta}^{\beta}_{path_{xy}}}{\sum\limits_{j=1}^{n-3} \hat{\tau}^{\alpha}_{path_{xj}} \times \hat{\eta}^{\beta}_{path_{xj}}} \tag{13}$$
$$PB_{path_{xy}} \in (0, 1], \quad \alpha, \beta \in (0, 1)$$

where α, β are weight factors to influence pheromone and heuristic values at each path. We set a threshold UT for the time interval of updating the dynamic connections. C_x extends its 3-tuple to a 4-tuple by storing the update time $utime_x$, as follows: $C_x \Leftrightarrow (chunk_x, NL_x, CL_x, utime_x)$. Moreover, we formulate the update rule for the dynamic connections by Definition 1.

Definition 1: Set PT_b and PT_r as thresholds for building and removing dynamic connections, respectively. After updating the probability $PB_{path_{xy}}$ between C_x and C_y, if there is no dynamic connection between C_x and C_y and $PB_{path_{xy}}$ is greater than the threshold PT_b, the connection should be built. If there is a dynamic connection between C_x and C_y and $PB_{path_{xy}}$ is less than the threshold PT_r, the connection should be removed.

When C_x needs to build a dynamic connection with the far-end community C_y, the broker node of C_x sends a request message for building the dynamic connection to the broker node of C_y. As there is a need to fast deliver the request message to C_y, we propose a search algorithm for the nearest connection to enable fast message delivery. The main idea of the algorithm is that the broker node of C_x sends the request message to a connection manager inf_{min} in CL_x which has the minimum distance $|dis|$ to C_y according to eq. (2).

Algorithm 2 Search for the nearest connection inf_{min}

```
/* getCID() returns current community ID; */
let TC indicate target community;
CID = 0; /* community ID (chunk ID) */
minimum = ∞;
for(i = 0; i < count(CL_x); i + +)
    dis = |getCID(CL_x[i]) − getCID(TC)|;
    if (dis < minimum)
        minimum = dis;
        CID = getCID(CL_x[i]);
    end if
end for i
returns CID;
```

Algorithm 2 describes the process of searching for inf_{min}. When inf_{min} receives the request, it re-performs Algorithm 2 until the request is sent to the broker node of C_y. This algorithm reduces the number of hops the request message takes to destination in comparison with the case when the message is forwarded over the static connections only. When the broker node of C_y receives the request from C_x, it inserts the information about the broker node of C_x into its CL_y and returns a response message containing its broker node information. C_x and C_y maintain the dynamic connection by exchanging information about any change in the broker nodes. C_y cannot remove the connection to C_x other than when C_x sends a remove message to C_y. The update of the dynamic connection is performed when periodically the probability with which the members in C_x perform far-end movement to different communities is re-calculated. The periodic adjustments of dynamic connections reflect in real-time the changes of user interest in video content and reduce the maintenance cost of the dynamic connections and the number of hops when forwarding request messages. Algorithm 3 describes the update strategy for the dynamic connections of C_x.

C. Community member layer

The broker node and ordinary members form a community and based on this structure, high performance benefits are obtained in terms of high-efficiency management of members and resources and fast response to resource requests.

The broker node is responsible for maintaining connections within community, collecting the information of members and handling the messages for resource requests.

1) **Maintenance of community connections**.

By exchanging the change message with CL_x items, the broker node n_k in C_x achieves the real-time maintenance of CL_x. When n_k leaves C_x, it informs the CL_x items about the change of broker node of C_x. If the items in CL_x vary, n_k updates the CL_x items after receiving the change messages.

The broker node cannot shoulder the calculation of probability PB due to limited capacity of computation and energy in mobile device side. n_k periodically sends the far-end

Algorithm 3 C_x dynamic connections update

/* $nodeSet$ is the node set requesting far-end movement; */
/* $PB[]$ is an array of length $n-3$ used to store the probabilities of member movement from C_x to other communities; */
/* m is size of the $nodeSet$; The initialized value of μ is 0; */
for($i = 1$; $i \leq n$; $i++$)
  if($i \in (x, x+1, x-1)$)
    continue;
  end if
  for($k = 0$; $k < m$; $k++$)
    if($i == TargetComm(nodeSet[k])$)
      $PB[i].\tau = PB[i].\tau + \tau(nodeSet[k])$;
      $PB[i].\eta = PB[i].\eta + \eta(nodeSet[k])$;
    end if
  end for k
  $\mu = \mu + (PB[i].\tau)^{\alpha} \times (PB[i].\eta)^{\beta}$;
end for i
for($i = 1$; $i \leq n$; $i++$)
  if($i \in (x, x+1, x-1)$)
    continue;
  end if
  $PB[i] = \frac{(PB[i].\tau)^{\alpha} \times (PB[i].\eta)^{\beta}}{\mu}$ by eq. (13);
  if(C_x has no dynamic connection with C_i && $PB[i] > PT_b$)
    build the dynamic connection;
  else if(C_x has dynamic connection with $C_i \&\& PB[i] < PT_r$)
    remove the dynamic connection;
  end if
end for i

movement information collected and NL_x to the server. The period time can be set to $\lambda \times len, \lambda = 1, 2, ..., n$, reducing the interaction frequency. The server handles the calculation of PB and updates Que in terms of NL_x received to keep the availability of items information in Que. n_k makes use of the returned PB to maintain the items in CL_x according to ***Definition 1***.

2) **Maintenance of community members**.

n_k cannot maintain the real-time status of members due to the limited capacity. It relies on the several members to assist collecting the information of members. We propose a "collaborator" selection strategy to balance the maintenance load of members to each member in community. Each member periodically reports the information of receivers which obtain the video data from it to its supplier (The report period time can be set to $\psi \times len, \psi \in (0, 1)$). By repeatedly performing the above process, the members whose supplier is the server are considered as collaborators and report the information of members received to n_k. The load of member maintenance is distributed to multiple members, enhancing the system's scalability.

3) **Handling resource request**.

The broker node n_k also cannot take on a mass of load of handling the resource request messages caused by the far-end movement of current community members. If a member moves to other community, it disconnects the logical link with its supplier and receiver of video data and re-obtain new suppliers by enquiring the broker node; otherwise, a member always relies on the logical link to obtain the video data from its supplier or provide the video data to its receiver. The members which keep the sequential playback do not frequently change their logical link, reducing the load of broker node. If a member N_i performs the far-end movement or near-end backward movement, it sends the request message to n_k. n_k delivers the request message of N_i to its inf_{min} by performing Algorithm 2 and removes N_i from NL_x. When the broker node of the target community finally receives the request message, it returns some supplier candidates for N_i in terms of BOM and inserts N_i into its NL. Moreover, n_k sends the information of connections in CL_x to the collaborators and requires the collaborators to assist disseminating the information of community connections, so as to support the movement of members. The collaborators can append the information of connections into streaming data sent to its receivers. With the sent streaming data, all receivers of collaborators can know the information of connections. Similarly, these receivers can disseminate the information of connections by making use of delivering streaming data. All members can obtain the information of connections and directly send the request message to inf_{min} by performing Algorithm 2.

In order to further reduce the number of resource search message, each ordinary member N_i needs to pre-fetch the future playback content in terms of the speculation-based pre-fetching strategy in our previous work [14] and share the information (chunk ID and supplier) of content pre-fetched with its receivers. The dissemination of resource information can reduce the interaction between broker node and ordinary members, so as to keep the low load for the broker nodes.

Replacement of broker node:

When the broker node n_k leaves C_x, it selects a member in NL_x as its successor and delivers it the community information. To prevent frequent replacements of the broker node, n_k needs to estimate the potential for becoming a broker node for items in NL_x. Unlike SURFNet which estimates node stability based on node's lifespan online, which is a metric highly imprecise. AMCV considers some interest-related crucial factors - the average movement weight ratio, average playback time ratio (see eq. (5)) and the online time ratio in community as estimation parameters for content.

1) **Average movement weight ratio**. The movement distance dis of N_i can be obtained from eq. (2) considering the case when a member performs a far-end movement. When N_i performs one movement operation, the movement weight ratio generated by this movement can be defined in eq. (14).

$$wr_i^{'} = 1 - \frac{|dis|}{n} \tag{14}$$

A lower value of $wr_i^{'}$ indicates that the movement distance of the member is larger, namely the member has the little interest in the video. If N_i performs a k movement before joining into C_x, the average movement weight ratio of N_i can be obtained according to eq. (15).

$$wr_i = \begin{cases} \overline{wr_i}, & k > 0 \\ 1, & k = 0 \end{cases}, \quad \overline{wr_i} = \frac{\sum_{c=1}^{k} (wr_i^{'})_c}{k} \tag{15}$$

If $k = 0$, $wr_i = 1$, it denotes that N_i performs a sequential playback and has the highest interest for the video content.

2) **Online time ratio in current community**. If N_i just joined C_x, the time of it staying in the current community may be longer and N_i is an appropriate candidate for broker. We use st_{ix} to indicate the stay time ratio of N_i in C_x, defined in eq. (16).

$$st_{ix} = 1 - \frac{cTime_i - egTime_{ix}}{len} \tag{16}$$

where $cTime_i$ is current time of N_i. We use the product between wr_i , st_{ix} and $\overline{P_{ix}}$ to estimate the each N_i potential performance as a broker in C_x according to eq. (17).

$$sw_{ix} = wr_i \times \overline{P_{ix}} \times st_{ix} \tag{17}$$

where $\overline{P_{ix}}$ is the average playback time ratio of N_i before joining into C_x by eq. (5). The broker node selects a member whose sw_{ix} is the highest from the NL_x as a broker candidate.

V. Performance Analysis and Evaluation

A. AMCV Overhead Analysis

This section analyzes the load related to the use of AMCV in terms of three aspects: the source media server stress, load of network and maintenance overhead of nodes.

1) The slight stress added by AMCV to the server is due to maintaining the node list Que and the output video streaming to the system members. The server does not maintain the state of items of Que in real time, so the maintenance of Que is very light. Pre-fetching video content into the local buffer can further reduce the number of request messages and smooth user playback experience. Moreover, AMCV groups all nodes in MP2P networks into communities to integrate the available resources in networks. The efficient utilization of resources also reduces the dependency level for the original video resources stored in the server. Therefore, AMCV enables a small number of output video streams to support high system scalability.

2) In AMCV, the request message of member which moves from current community to another community is forwarded among the brokers node of several communities to achieve supplier lookup. The forwarding message number is less than n. In SURFNet, the supplier lookup of node performing VCR operations by gossip protocol results in huge search messages dissemination in networks. The high traffic load increases the stress of network.

3) AMCV uses a self-organization topology in overlay network so that the maintenance cost of the topology is embodied in the cost of the broker node maintaining the community-related information, which will be detailed next.

The load of the broker node mainly includes two aspects: handling the request for resource messages and maintaining the community connections and information of members.

(A) The number of request messages depends on the P2P-VoD system scale and popularity of the current video chunk. The larger the system scale is or the higher the popularity of current video chunk is, the more the number of members which join the community is. Then a broker node will handle more messages and perform more searches for suppliers. However, the members pre-fetch the future playback content and share the information of supplier which carries the pre-fetching content, so as to reduce the number of messages from the nodes which request joining community. Moreover, the broker node does not need to handle the request message from members inside the community due to disseminate information of community connections with the help of collaborators. Therefore the load of broker nodes is kept in the low level.

(B) The maintenance of community-related information mainly includes NL and CL management. The maintenance of NL has low load due to the fact that the broker node only updates NL after it receives the message containing the information of members from collaborators. The maintenance of CL is dependent on the threshold PT_b and PT_r which determine the number of items in CL (the maximal length of CL is $\left\lfloor \frac{1-PT_b}{PT_r} \right\rfloor + 1$). The dynamic connections reflect the level of interest and behavior similarity of the members and our purpose is to make sure most nodes achieve fast access supplier, so the number of items in CL is limited to a value tolerable by the broker node. The maintenance of CL does not overload the broker.

B. Performance Evaluation

The performance of AMCV is compared with that of SURFNet [13], a state of the art solution which uses a node group strategy similar to that of AMCV supporting the VoD services. Both AMCV and SURFNet solutions are deployed in the same wireless mobile network environment. After the test parameter description, we will discuss the set up of a common simulation environment for these two solutions.

1) Test Parameter Description: This section discusses how the parameters used by AMCV affect its performance. α and β as weight factors affect and adjust the pheromone and heuristic value at each path. The higher the value of α is, the larger the influence brought by pheromone for the movement probability at path is. The high value of α denotes that the relevance between video content reflected by movement distance and direction plays a decisive role for the estimation of probability at path. This is based on the premise that each viewer has the adequate interest for the video content so that the content relevance relationship from movement behavior is authentic. However, the "noise" in these movement behaviors (such as the random movement caused by browse without the interest) reduces the quality of estimating movement probability. Similarly, the high value of β denotes that the variation of interest reflected by the comparison between two online time parameters $\overline{T_{ix}}$ and $\overline{ST_{ix}}$ has the large ratio for estimating movement probability. The variation of interest can reduce the influence from "noise" in the movement behaviors, but the relevance between video content is neglected. UT determines the time interval between two dynamic connection updates. In the simulations, the setting of UT value should consider several factors including the number of nodes in AMCV. If each community needs to obtain more accurately movement status of its members (level of interest change), the value of UT should be decreased. PT_b and PT_r determine the

TABLE II
SIMULATION PARAMETER SETTING FOR WIRELESS MOBILE NETWORK

Parameters	Values
Area	$800{\times}800(m^2)$
Bandwidth of Each Mobile Node	2 Mb/s
Bandwidth of Media Server	10 Mb/s
Channel	Channel/WirelessChannel
Default Distance between Server and Nodes	6 hops
802.11 Data Transmission Rate	2 Mb/s
802.11 Basic Transmission Rate	1 Mb/s
Interface Queue Type	CMUPriQueue
Length of Video Chunk	30 s
MAC Interface	MAC/802_11
Movement Direction of Each Mobile Node	Random
Network Interface	Phy/WirelessPhy
UT	60 s
PT_b	0.2
PT_r	0.12
Number of Mobile Nodes	400
Number of Video Chunks	20
Mobility Speed Range	[10, 30] m/s
Pause time of Each Mobile Node	0 s
Rate of Streaming Data	180 kb/s
Routing Protocol	DSR
Simulation Time	600 s
Size of Request Messages	2 KB
Transmission Protocol of Video Data	UDP
Transmission Protocol of Control Messages	TCP
Wireless Signal Range	200 m
α	0.5
β	0.5

number of dynamic connections. If PT_b and PT_r are small, the number of dynamic connections is more, so that the maintenance cost of the connections will increase. Otherwise, if PT_b is large, the number of dynamic connections is low and so is the movement cost of the members.

2) Testing Topology and Scenarios and Parameter Settings: A self-organization wireless mobile network topology is built by making use of the Network Simulator (NS-2.35) [35]. Table II lists some important NS2 simulation parameters of the wireless mobile network. The popularity values of video chunks follow a Zipf distribution. We created 200 synthetic user viewing log entries in terms of the interactive actions, measurements and statistics from [36]. 200 mobile nodes join the system every 1 second and follow the 200 viewing log entries to play video content.

3) Simulation Results: The performance of AMCV is compared with that of SURFNet in terms of average hops of forwarding request message, average delay for responding request, average lookup success rate, server stress, average playback continuity and node load, respectively.

(1) **Average hops of forwarding request message during far-end movement of community members**. The number of intermediate nodes which a request message goes through from the requesting node to the supplier with the targeted resource is used to indicate this number of hops. The total hops divided by the number of request messages is used to indicate the average hops.

In Fig. 5, we illustrate the average number of hops variation during the time of the simulation, when SURFNet and AMCV are used in turn. It can be seen how SURFNet's average number of hops is maintained at a relatively high, but stable value of 4 with moderate fluctuations. AMCV's average number of hops curve reaches a peak value of 7, after which decreases to roughly 2, twice lower than that of SURFNet, after t = 60 s. The results shown by Fig. 5, indicate initial worse values for AMCV than for SURFNet as the former has static connections and a little dynamic connections at that stage, only. With multiple dynamic connections are created, after t = 60 s AMCV's performance benefits become evident.

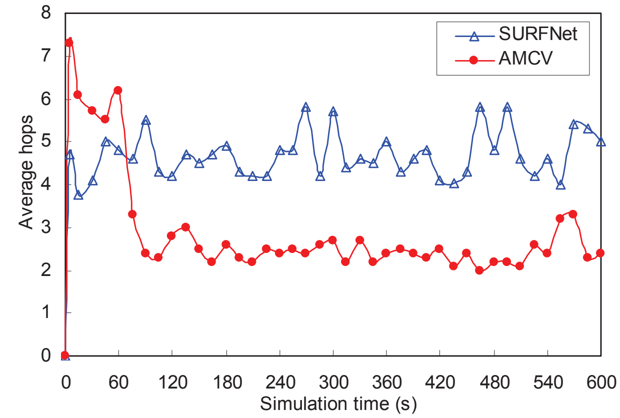


Fig. 5. Average number of hops variation during the simulation time.

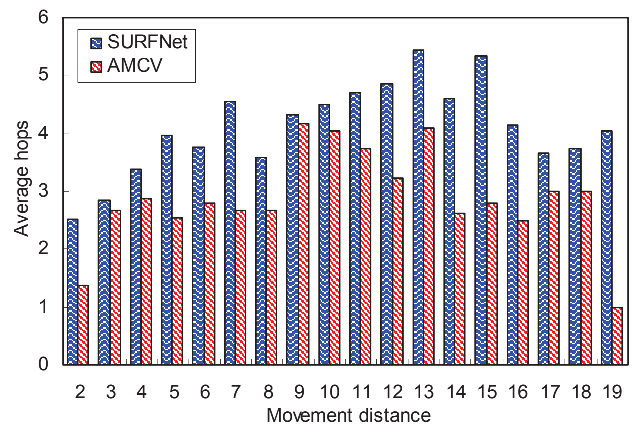


Fig. 6. Average number of hops with increasing the movement distance.

Fig. 6 shows how the average number of hops varies with an increase in the movement distance, as described by eq. (2). The blue histograms corresponding to the SURFNet's results have both higher values and larger fluctuation with the increase in the movement distance (the values are between 2.5 and 5.5 hops). The AMCV results, illustrated with red bars have values between 1 and 4, with lower variations than SURFNet's results. On average AMCV's results are better than those of SURFNet. Each community in AMCV has in the dynamic connections a fast direct access path, reducing the number of intermediate nodes which a request message experiences. The continuous self-optimizing characteristic of AMCV (periodic updated its dynamic connections) makes sure the performance

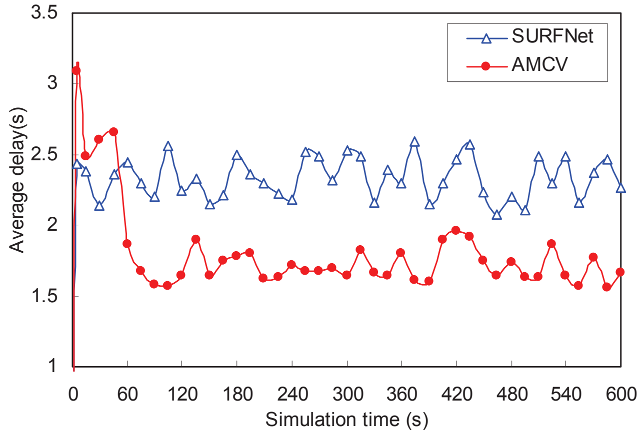


Fig. 7. Average delay during the simulation time.

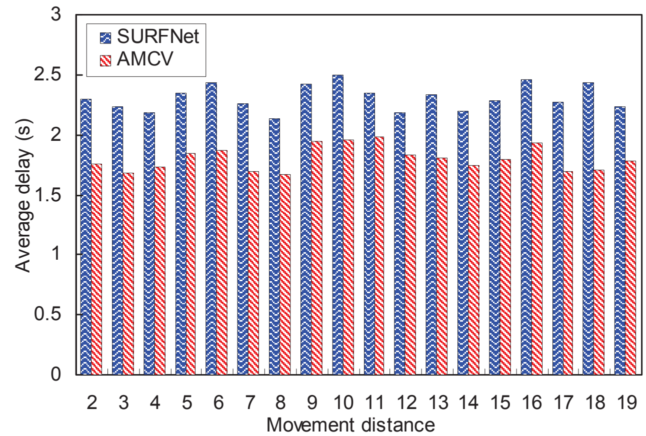


Fig. 8. Average delay with increasing the movement distance.

advantage in favor of AMCV is maintained with the increase in the movement distance. Consequently, the average number of hops and average delay of AMCV are less than those of SURFNet, which does not consider the behavior of the nodes and therefore cannot improve its mechanisms for resource search.

(2) **Average delay for responding to far-end movement of community members**. The delay which a request message experiences from the requested node to the supplier with the targeted resource is used to indicate the delay for responding to far-end movement of nodes. The total delay divided by the number of request messages is used to indicate the average delay.

Fig. 7 illustrates the average delay when receiving request messages at the application layer during 15 s long intervals. SURFNet's delay curve experiences very high jitter with a range of delays between 2 s and 2.5 s. AMCV's delay first increases to above 3.0 s and then decreases to around 1.6 s with low jitter from t = 60 s to end. AMCV makes use of static connections and a little dynamic connections to deliver the request message between communities from 0 s to 60 s, leading to the higher delay for AMCV than for SURFNet. With increasing behavior of node's movement, the dynamic connections between communities reduce the number of forwarding the request message so that the average response delay of AMCV keeps the lower level than that of SURFNet from t = 60 s to end.

Fig. 8 presents the average delay variation with the increase in the movement distance. SURFNet's blue delay bars vary between 1.35 s and 1.75 s, whereas AMCV's red bars vary from 1.1 s to 1.55 s, with the increase in the movement distance. Overall AMCV values are 20% better than those of SURFNet, clearly indicating its performance-related benefit. This is due to the fact that AMCV collects and analyzes the behavior of nodes' far-end movement and builds the dynamic connections with other community. The dynamic connections ensure request messages arrive fast at the broker node of the target community, so that both average number of the delay is reduced. During the simulation time, AMCV periodically updates the dynamic connections in terms of the behavior of nodes, so the average number of the delay of AMCV are kept low and lower than those of SURFNet. In SURFNet, the performance is limited due to the fixed search for resources mechanism.

(3) **Average lookup success rate (ALSR)**. ALSR of video chunk describes the system capability for supporting the smooth playback service. The ratio between the number of success lookup times and the total number of search message indicates the ALSR (the message which is sent to the server is neglected). The larger the value of ALSR is, the higher the user experience is. Fig. 9 shows the ALSR variation: ALSRs of AMCV and SURFNet increase with the increase in the number of nodes from 60 s to 600 s. The increment and level of curve corresponding to AMCV are greater than those of SURFNet. The red curve quickly rises from 60 s to 180 s and keeps the stable increase from 180 s to 600 s. The quick rise range for blue curve is [60, 240] s and this curve keep a slight rise trend in the remaining time. ALSR of AMCV are about 30% higher than that of SURFNet.

SURFNet relies on the AVL tree structure and chain list to search and share video resources. When the system scale is small, the scarce resources in overlay network cannot meet the demand of users. The relatively finite resources at the server side also cannot handle the numerous unexpected request. ALSR of SURFNet keeps the low level. The increasing number of nodes in overlay network indicates the incremental available resources. ALSR of SURFNet has a quick rise trend. However, the static node structure and low-efficiency resource distribution in SURFNet cannot adapt the dynamic playback behavior of users so as to result in the relatively low ALSR. The broker nodes in AMCV can implement the high-efficiency and flexible maintenance for video resources. Pre-fetching resource at the node side can optimize the resource configuration in overlay network. The balanced distribution of resources can improve the ALSR of AMCV. Therefore AMCV can obtain the high level ALSR.

(4) **Server stress** and (5) **Average playback continuity**. The number of streams serving nodes at media server side indicates the server stress. The server stress is used to measure the scalability of the system (the lower the server stress, the higher the scalability of the system is).

Fig. 10 illustrates the average server stress every 60 s. We

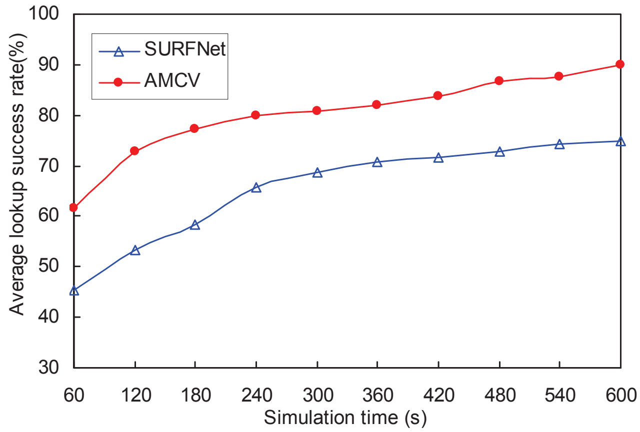


Fig. 9. Average lookup success rate.

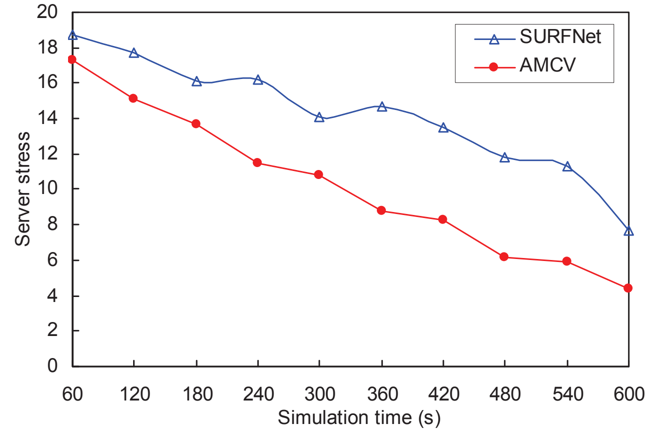


Fig. 10. Server stress.

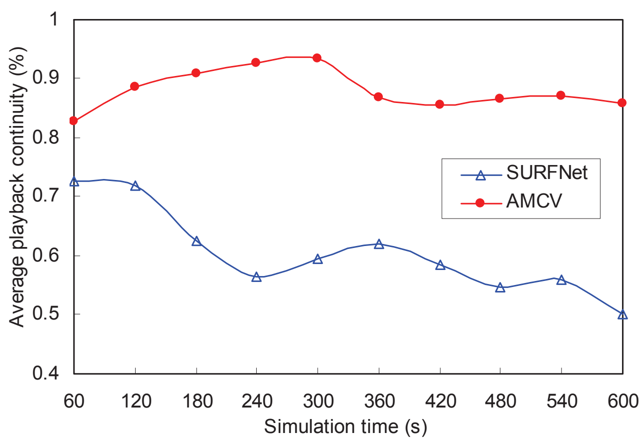


Fig. 11. Average playback continuity.

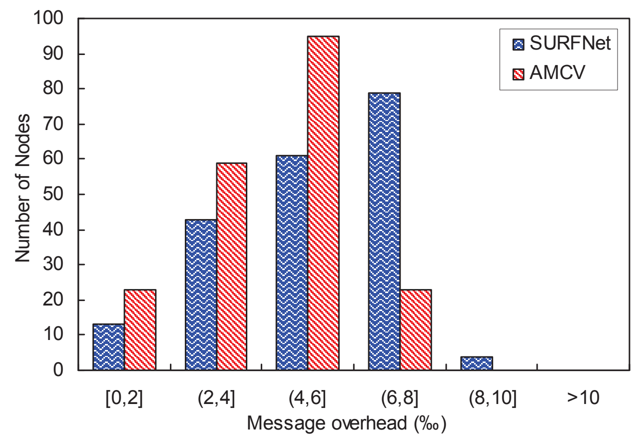


Fig. 12. Node load.

can see in the figure how the media server stress in SURFNet and AMCV decreases with the increase in the simulation time, namely the curves of SURFNet and AMCV maintain a relative stable low level after having high initial server stress. However, AMCV server stress values are 35% better than those of SURFNet, clearly indicating its performance-related benefit.

The ratio between the number of nodes which continuously obtain the played video data (seeking delay is 0 s) and total number of nodes in current overlay network every 60 s indicate the average playback continuity. Fig. 11 shows the average playback continuity. The curve corresponding to SURFNet results keep a fast fall trend with severe jitter. The curve to indicate AMCV results presents a rise from 60 s to 300 s and stable fall from 300 s to end, more than 50% lower than that of SURFNet.

This lightweight load and high playback continuity of AMCV is due to the accurate pre-fetching of video chunk and success in sharing resources. The pre-fetching of chunks not only enables the nodes download the future playback content in advance to local buffer to ensure the smooth playback, but also optimize the distribution of resources to reduce the output number of streams at the media server.

(6) **Node load**. The number of message processed at application layer indicates the message overhead of nodes. Fig. 12 illustrates the comparison statistics about the number of nodes in different ranges of message rate between AMCV and SURFNet. The message rate is defined as:

$$MR = \frac{M_i}{N^{(t)}} \tag{18}$$

where M_i denotes the number of message processed by N_i and $N^{(t)}$ is the total number of message processed by all nodes. The black bars corresponding to the SURFNet results mainly distribute in the ranges of message number (2, 4], (4, 6] and (6, 8]. The white bars indicating the AMCV results are in the ranges [0, 2], (2, 4] and (4, 6]. Fig. 12 clearly shows how AMCV outperforms SURFNet in terms of performance.

The main reason for this difference between the two solutions is that SURFNet needs to maintain an AVL tree structure and its derivative chain list so that the number of message needed to handle increase with the increasing simulation time. In AMCV, the broker nodes relies on the collaborators and members in community to collect the information of members and disseminate the information of community connections, so that the maintenance load of community uniformly is distributed to each member in community. Therefore, the distribution of node load in AMCV is more balanced than that of SURFNet.

VI. Conclusion

This paper proposes a novel Ant-inspired Mini-Community-based Video sharing solution (AMCV) for VoD services in wireless mobile networks. AMCV bases its efficiency on a two-layer architecture which includes: the mini-community network layer - which constructs the static and dynamic connections between communities in terms of user interactivity to support fast content discovery and access via innovative algorithms and the community member layer - in charge with the management of the members and resources in the community to support high-efficiency resource sharing and low maintenance cost. AMCV's performance was assessed in comparison with that of a state of the art alternative solution SURFNet via extensive simulations. The results show how AMCV ensures fast content discovery between requester and holder of resource and achieves lower average number of hops for request delivery, lower average delay of resource search, higher average lookup success rate, lower server stress, higher average playback continuity and lower node load.

References

[1] L. Tu and C.-M. Huang, "Collaborative Content Fetching Using MAC Layer Multicast in Wireless Mobile Networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 3, pp. 695-706, Sept. 2011.

[2] S.-E. Hong and M. Kim, "Design and Analysis of a Wireless Switched Digital Video Scheme for Mobile TV Services Over WiMAX Networks," *IEEE Trans. on Broadcasting*, vol. 59, no. 2, pp.328-339, June 2013.

[3] L. Zhou, M. Chen, Y. Qian, and H.-H. Chen, "Fairness Resource Allocation in Blind Wireless Multimedia Communications," *IEEE Trans. on Multimedia*, vol. 15, no. 4, pp. 946-956, June 2013.

[4] X. Wang, M. Chen, T. T. Kwon, LaurenceT. Yang, and V. C. M. Leung, "AMES-Cloud: A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Sharing in the Clouds," *IEEE Trans. on Multimedia*, vol. 15, no. 4, pp. 811-820, June 2013.

[5] C. Xu, E. Fallon, Y. Qiao, L. Zhong and G.-M. Muntean, "Performance Evaluation of Multimedia Content Distribution over Multi-Homed Wireless Networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp.204-215, June 2011.

[6] L. Zhou, X. Wang, W. Tu, G. Muntean and B. Geller, "Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 3, pp. 409-419, Apr. 2010.

[7] C. Xu, T. Liu, J. Guan, H. Zhang and G. Muntean, "CMT-QA: Quality-aware Adaptive Concurrent Multipath Data Transfer in Heterogeneous Wireless Networks," *IEEE Trans. on Mobile Computing*, vol. 12, no. 11, pp.2193-2205, Nov. 2013.

[8] M. Mu, J. Ishmael, W. Knowles, M. Rouncefield, N. Race, M. Stuart, and G. Wright, "P2P-Based IPTV Services: Design, Deployment, and QoE Measurement," *IEEE Trans. on Multimedia*, vol. 14, no. 6, Dec. 2012.

[9] S. H. Gary Chan and W. P. Ken Yiu, "Distributed storage to support user interactivity in peer-to-peer video," *US Patent*, number 7925781, April 2011.

[10] C. Xu, G.-M. Muntean, E. Fallon and A. Hanley, "Distributed storage-assisted data-driven overlay network for P2P VoD services," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp.1-10, March 2009.

[11] C. L. Chang and S. P. Huang, "The interleaved video frame distribution for P2P-based VoD system with VCR functionality," *Computer Networks*, vol. 56, no.5, pp.1525-1537, March 2012.

[12] W. Wu and J. C.S. Lui, "Exploring the Optimal Replication Strategy in P2P-VoD Systems: Characterization and Evaluation," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1492-1503, Aug. 2012.

[13] D. Wang and C. K. Yeo, "Superchunk-Based Efficient Search in P2P-VoD System Multimedia," *IEEE Transactions on Multimedia*, vol. 13, no. 2, pp.376-387, April 2011.

[14] C. Xu, F. Zhao, J. Guan, H. Zhang and G.-M. Muntean, "QoE-Driven User-Centric VoD Services in Urban Multihomed P2P-Based Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp.2273-2289, June 2013.

[15] S. Jia, C. Xu, G.-M. Muntean, J. Guan and H. Zhang, "Cross-Layer and One-Hop Neighbour-Assisted Video Sharing Solution in Mobile Ad Hoc Networks," *China Communications*, vol. 10, no. 6, pp. 111-126, June 2013.

[16] L. Zhou, Y. Zhang, K. Song, W. Jing and A. V. Vasilakos, "Distributed Media Services in P2P-Based Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 692-703, Feb. 2011.

[17] M. Fanelli, L. Foschini,A. Corradi and A. Boukerche, "Self-Adaptive Context Data Distribution with Quality Guarantees in Mobile P2P Networks," *IEEE Journal on Selected Areas in Communications/Supplement*, vol. 31, no. 9, pp. 115-131, Sept. 2013.

[18] S. Jia, C. Xu, A. V. Vasilakos, J. Guan, H. Zhang and G.-M. Muntean, "Reliability-oriented Ant Colony Optimization-based Mobile Peer-topeer VoD Solution in MANETs," *ACM/Springer Wireless Networks*, vol. 19, no. 8, Published Online, Nov. 2013.

[19] S. Wang, M. Liu, X. Cheng, Z. Li, J. Huang and B. Chen, "Opportunistic Routing in Intermittently Connected Mobile P2P Networks," *IEEE Journal on Selected Areas in Communications/Supplement*, vol. 31, no. 9, pp. 369-378, Sept. 2013.

[20] K. Chen, H. Shen and H. Zhang, "Leveraging Social Networks for P2P Content-based File Sharing in Disconnected MANETs," *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 235-249, Feb. 2014.

[21] C.-H. Hsu and C.-F. Juang, "Evolutionary Robot Wall-Following Control Using Type-2 Fuzzy Controller With Species-DE-Activated Continuous ACO," *IEEE Trans. on Fuzzy Systems*, vol. 21, no. 1, pp.100-112, Feb. 2013.

[22] Y. Lin, J. Zhang, H.S.-H. Chung, W. H. Ip, Y. Li and Y.-H. Shi, "An Ant Colony Optimization Approach for Maximizing the Lifetime of Heterogeneous Wireless Sensor Networks," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 3, pp.408-420, May 2012.

[23] M. Farah, C. Hicham, S. Hichem, Y. Farouk, A. Lionel and R. Cedric, "Controlled Mobility Sensor Networks for Target Tracking Using Ant Colony Optimization," *IEEE Trans. on Mobile Computing*, vol. 11, no. 8, pp.1261-1273 , Aug. 2012.

[24] Y. Ding and L. Xiao, "Video On-Demand Streaming in Cognitive Wireless Mesh Networks," *IEEE Trans. on Mobile Computing*, vol. 12, no. 3, pp.412-423, Mar. 2013.

[25] Y. He and L. Guan, "Solving Streaming Capacity Problems in P2P VoD Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp.1638-1642, Nov. 2010.

[26] Y. He and Y. Liu, "VOVO: VCR-Oriented Video-on-Demand in Large-Scale Peer-to-Peer Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, no. 4, pp. 528-539, April 2009.

[27] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, "On Replication Algorithm in P2P VoD," *IEEE/ACM Trans. on Networking*, vol. 21, no. 1, pp.233-243, Feb. 2013.

[28] Q. Yu, B. Ye, S.Lu and D. Chen, "Optimal data scheduling for P2P video-on-demand streaming systems," *IET Communications*, vol. 6, no. 12, pp.1625-1631, Dec. 2012.

[29] W. Wu, J. C.S. Lui and R. T.B. Ma, "On incentivizing upload capacity in P2P-VoD systems: Design, analysis and evaluation," *Computer Networks*, vol. 57, no. 7, pp.1674-1688, May 2013.

[30] S. Gramatikov, F. Jaureguizar, J. Cabrera and N. Garca, "Stochastic modelling of peer-assisted VoD streaming in managed networks," *Computer Networks*, vol. 57, no. 9, pp.2058-2074, June 2013.

[31] L. Acunto, N. Chiluka, T. Vink and H. Sips, "BitTorrent-like P2P approaches for VoD: A comparative study," *Computer Networks*, vol. 57, no. 5, pp.1253-1276, April 2013.

[32] Y. Zhou, T. Z.J. Fu and D. M. Chiu, "Server-assisted adaptive video replication for P2P VoD," *Signal Processing: Image Communication*, vol. 27, no. 5, pp.484-495, May 2012.

[33] C. Xu, G.-M. Muntean, E. Fallon and A.Hanley, "A Balanced Tree-based Strategy for Unstructured Media Distribution in P2P Networks," *in Proc. IEEE ICC*, May 2008.

[34] Z. Cernekova, "Information theory-based shot cut/ fade detection and video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 82-91, Jan 2006.

[35] UC Berkeley, LBL, USC/ISI and Xerox Parc, NS-2 documentation and software, v. 2.35, Available: www.isi.edu/nsnam/ns, Nov. 2011.

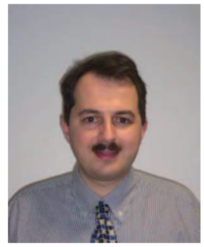
[36] A. Brampton, A. MacQuire, I. A. Rai, N. J. P. Race, L. Mathy, and M. Fry, "Characterising user interactivity for sports video-on-demand," *in Proc. ACM NOSSDAV*, Apr. 2007.

Changqiao Xu received his Ph.D. degree from Institute of Software, Chinese Academy of Sciences (ISCAS) in Jan 2009. He was an Assistant Research Fellow in ISCAS from 2002 to 2007, where he was research and development project manager in the area communication networks. During 2007-2009, he worked as a researcher in the Software Research Institute at Athlone Institute of Technology, Ireland. He joined Beijing University of Posts and Telecommunications (BUPT) in Dec. 2009, and was an Assistant Professor from 2009 to 2011. He is currently an Associate Professor with the Institute of Network Technology, and vice-Director of the Next Generation Internet Technology Research Center at BUPT. He has published more than 90 technical papers in prestigious international journals and conferences, such as IEEE TMC, IEEE TVT, IEEE TBC and Proc. ACM Multimedia. His research interests include wireless networking, multimedia communications and next generation Internet technology.

Shijie Jia received his Ph.D. degrees in communications and information system from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in March 2014. He is a lecturer in the Academy of Information Technology, Luoyang Normal University, Luoyang, Henan, China. His research interests include next generation Internet technology, wireless communications and peer-to-peer networks.

Lujie Zhong received his Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences (ICT) in July 2013. She is a lecturer in the Information Engineering College, Capital Normal University, China. Her research interests are in the areas of communication networks, computer system and architecture.

Hongke Zhang received his Ph.D. degrees in electrical and communication systems from the University of Electronic Science and Technology of China in 1992. From 1992 to 1994, he was a postdoctoral research associate at Beijing Jiaotong University (BJTU), and in July 1994, he became a professor there. He has published more than 150 research papers in the areas of communications, computer networks, and information theory. He is the author of eight books written in Chinese and the holder of more than 40 patents. He is the chief scientist of a National Basic Research Program ("973" program). He is Director of the Next Generation Internet Technology Research Center at Beijing University of Posts and Telecommunications (BUPT) and Director of the National Engineering Laboratory for Next Generation Internet Interconnection Devices at BJTU.



Gabriel-Miro Muntean received the Ph.D. degree from Dublin City University, Ireland for research in the area of quality-oriented adaptive multimedia streaming in 2003. He is a Senior Lecturer with the School of Electronic Engineering at Dublin City University, Ireland. Dr. Muntean is Co-Director of the DCU Performance Engineering Laboratory, Director of the Network Innovations Centre, part of the Rince Institute Ireland and Consultant Professor with Beijing University of Posts and Telecommunications (BUPT), China. His research interests include quality-oriented and performance-related issues of adaptive multimedia delivery, performance of wired and wireless communications, energy-aware networking and personalised e-learning. Dr. Muntean has published over 180 papers in prestigious international journals and conferences, has authored three books and fifteen book chapters and has edited six other books. Dr. Muntean is Associate Editor of the IEEE Transactions on Broadcasting, Associate Editor of the IEEE Communications Surveys and Tutorials and reviewer for other important international journals, conferences and funding agencies.