Cross-layer and one-hop neighbor-assisted video sharing solution in MANETs

Item Type	Article
Authors	Shijie, Jia;Changqiao, Xu;Muntean, Gabriel-Miro;Jianfeng, Guan;Hongke, Zhang
Citation	China Communications;10(6) pp. 111-126
Publisher	IEEE Computer Society
Download date	2026-04-20 15:35:17
Item License	https://creativecommons.org/licenses/by-nc-sa/1.0/
Link to Item	https://hdl.handle.net/10344/3453

Cross-Layer and One-Hop Neighbor-Assisted Video Sharing Solution in MANETs

JIA Shijie[1], **XU Changqiao**[1,2], **MUNTEAN Gabriel-Miro**[3], **GUAN Jianfeng**[1], **ZHANG Hongke**[1]
[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Wuxi 214135, China
[3]Performance Engineering Laboratory, School of Electronic Engineering, Network Innovations Centre, Rince Research Institute, Dublin City University, Dublin, Ireland

Abstract: Fast resources discovery and high QoS guarantee are key determinants for efficient Mobile P2P (MP2P) video sharing. In this paper, we propose a Cross-Layer and one-Hop Neighbor-assisted Video Sharing solution (CNVS) in MANETs. By making use of cross-layer approaches to bridge the overlay and MAC layer and with the help of dissemination assisted by one-hop neighbors, CNVS intelligently builds the resource-centric self-organization node cluster group. In order to meet the QoS requirement, by making use of video resources access cost model, each peer can disconnect less efficient connection with original supplier and choose the peer which provides the low access cost as new supplier. Simulation results also show how CNVS achieves lower average end-to-end delay, less average number of hops for video data delivery, lower routing overhead and packet loss rate, and higher network throughput in comparison with another state of the art solution.

Key words: cross-layer; video sharing; clustering; assist

I. INTRODUCTION

A mobile ad-hoc network (MANET) is a collection of mobile nodes which does not rely on any infrastructure for communication [1]. Applications of MANETs can be useful in many areas, including disaster relief, military, inter-vehicle communications, road traffic management, business and entertainment. In many of these domains, multimedia data exchange is becoming increasingly popular despite the resource-constraint wireless environments [2, 3]. Research has demonstrated that Peer-to-Peer (P2P) networks, with their distributed self-organization characteristics, are a successful solution for large scale multimedia distribution over the Internet [4]-[16]. Inspired by the success of the Internet-based P2P technology, Mobile Peer-to-Peer (MP2P) networks have emerged as a state-of-the-art technology for video resource exchange in MANETs [17]-[19].

The Internet-based P2P multimedia delivery systems can be broadly classified based on the architecture of their content distribution topology into three categories: tree-based [4]-[8], Chord-based [9]-[11] and mesh-based [12]-[16]. The tree-based and Chord-based architectures as structured topologies are known for the efficiency of resource searching and are widely researched. A structured topology can have high performance for any search for resources, but the P2P multimedia system pays the high price to maintain it. The nodes in the overlay can randomly join or leave at will. Once this happens, the system needs to reconstruct the current architecture. Along with the increase in the scale of the system, the overhead caused by this reconstruction becomes the bottleneck of the system, severely affecting its performance. The mesh-based architecture with an unstructured topology solves this problem as it does not need to construct and maintain such a complex structure, in particular a system with mesh-based architecture does not need to frequently define or change the father/child or precursor/successor roles for each node. However, the low efficient seeking resource restricts the performance of resource sharing in the mesh-based architecture.

Recently, cross-layer solutions [20]-[23] have attracted great interest of academic research. Generally speaking, cross-layer design refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains [20]-[22]. Among the cross-layer-based solutions proposed, one which creates new interfaces between the layers opens an interesting avenue. The new interfaces are used for information sharing between the layers at runtime to improve performance. Cross-layer design has been demonstrated to be a useful way to achieve highly-improved video performance for real-time wireless multimedia transmissions [23].

In this paper, we propose a novel Cross-layer and one-hop Neighbor-assisted Video Sharing solution (CNVS) for live media streaming in MANETs. CNVS uses two layers architecture to build the relationship between the geographical location of the mobile nodes and the quality of their communication channel and video content distribution. By making use of the cross-layer approach to append the information of video resource into the one-hop multicast message at the MAC layer and with the help of dissemination assisted by the one-hop neighbors, the carriers of video resource and mobile nodes close to them form the resource-centric self-organization cluster structure. Each node use the cluster to fast discover the available optimal service source in the low cost so that the overlay nodes can switch between the low and high efficiency service source in terms of their demand of QoS. Extensive tests show how CNVS achieves lower average end-to-end delay, less average hop count for video data delivery, lower routing overhead, and higher network throughput in comparison with other state of the art solutions.

II. RELATED WORK

There have been numerous studies on P2P topology architecture in recent years. One of the most extensively discussed proposals is the tree-based approach. In such an approach, peers are organized into a tree structure for delivering data, with each data packet being disseminated using the same structure. Nodes in the structure have well-defined relationships-"parent-child" as encountered in trees. A *tree-based solution* called SURFNet for P2P VoD services was proposed in [4]. In SURFNet, stable peers are used to construct an AVL tree to provide 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, respectively. In SURFNet, the stability of the AVL tree highly depends on the premise that the tree consists of stable nodes. The Chord-based architecture is a famous P2P distribution topology, also widely researched [24]. Nodes in Chord also have embedded the relationships-"precursor-successor". By associating a key with each data item and storing the key/data item pair at the node to which the key maps, the data location can be implemented on top of Chord. For example, a *Chord-based interactive VoD system* named VMesh was introduced in [9, 10]. It utilizes the total storage capacity of peers and a Distributed Hash Tables (DHT)-based network to improve the supply of video segments and support large interactive demands in a scalable manner. Obviously, in VMesh, with the increase of node numbers in the Chord structure, the cost of maintaining the structure will become the bottleneck of system's scalability.

Over the years, many tree-based or Chord-based P2P multimedia distribution solutions have been proposed and were investigated in academia, achieving some success. However, they seldom took off commercially. On the other hand, mesh-based systems have been successfully and widely deployed in multimedia commercial Internet, as part of solutions such as Cool-Streaming [12], PPLive [16], etc. The advantages of a mesh-based topology are the simple design principle and inherent robustness, particularly desirable for the highly dynamic P2P environment. For instance, the authors of [14] proposed *a mesh-based fluid model-based* P2P streaming solution. Each peer contacts the nodes selected according to different policies defined by the system as its neighbors. A random graph as the overlay topology is composed by mutual contact between these nodes. Fluid captures the dominant dynamics of the video chunk distribution process over several families of random graphs. The fluid models can utilize the connectivity of peers with large available bandwidth to create a cluster of large-bandwidth peers. However, the aforementioned traditional mesh-based overlay networks lack the high efficient seeking resource strategy.

The above Internet-based P2P video streaming solutions' deployment does not mention the issues such as node mobility and limited bandwidth. These issues have increasingly attracted researchers' attention. Recent papers [25]-[28] have focused on the P2P resource sharing in MANETs. QUVoD in our previous work [25] proposed a novel grouping-based storage strategy which distributes uniformly the video segments along the Chord overlay, reducing segment seeking traffic and balancing the service load. QUVoD makes use of the multi-homed hierarchical P2P and vehicular ad-hoc network (VANET) architecture, namely vehicles construct a low layer VANET via WAVE interfaces and also form an upper layer P2P Chord overlay on top of a cellular network via 4G interfaces. By making use of the segment seeking and multipath delivery scheme, QUVoD achieves high lookup success rate and very good video data delivery efficiency. Moreover, QUVoD employs the speculation-based pre-fetching strategy, which analyses users' interactive viewing behavior by estimating video segment playback order. MESHCHORD [26] enables location-aware ID assignment to peers to improve the basic Chord design and exploit the MAC cross-layer technique to speed up the efficiency of resource lookup in wireless mesh networks. However, the maintenance overhead of overlay in MESHCHORD caused by the large number of messages can result in performance degradation. The bottlenecks at the network and source media server are key factors which affect the scalability and performance of any MP2P multimedia system. Therefore exchange messages produced for neighborhood maintenance may lead to too much routing overhead, increasing the network load. PatchPeer [27] addresses the scalability issue associated with the original Patching technique in a traditional wireless network for supporting the video-on-demand. However, the load at the server side results in low PatchPeer's scalability due to the peers request the server to obtain the entire video when they cannot obtain the patch from their one-hop neighbors or regular stream from other peers. For instance, frequently failing seeking the patching stream increases the load of server. PatchPeer cannot handle the mobility of peer. For instance, obtaining patching stream from the one-hop neighbors of requesting node and using the Closest Peer to select the patching peer neglect other peers due to the mobility of peer lets these non-one-hop neighbors become the one-hop neighbor of requesting node in the next time period. Especially, if the requesting peer cannot obtain the regular and patch stream from one-hop neighbors and queue in the waiting queue at server side, the long start delay is intolerability for user.

III. CNVS ARCHITECTURE OVERVIEW

The CNVS architecture, illustrated in Fig. 1, organizes a media server and multiple mobile nodes in a structure with two layers: neighborhood layer and cluster layer.

(1) ***Neighborhood Layer.*** For each mobile node, the neighborhood layer is composed of those neighboring nodes with which there is good communication quality in terms of support for multimedia data delivery. The node maintains the neighboring layer in form of a neighborhood list. The neighborhood list includes those nodes which are selected by a ***neighboring node selection algorithm*** from all current nodes' next hop nodes.

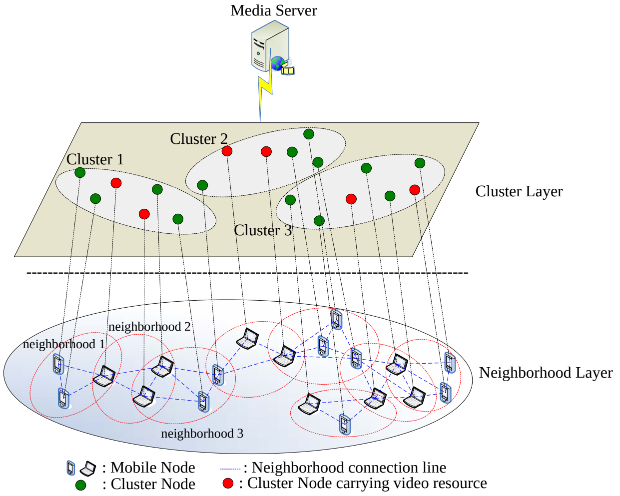

Fig. 1 *CNVS two-layers architecture*

(2) ***Cluster Layer.*** The cluster is a resource-centric self-organization node group without the intervention from the server. Each mobile node in the network is assigned the cluster identification (cluster ID); the mobile nodes with the same cluster ID form a cluster. The mobile nodes can make the decision whether joining or leaving a cluster in terms of the access cost provided by the cluster. The cluster becomes the node group of high node and resource density by cluster mergence. The members in the cluster can fast obtain the desired video resource from the optimal service source in the same cluster.

(3) ***Media Server.*** As the original owner of video resources, the media server is well-known to all mobile nodes and provides the streaming service for the mobile nodes. When the mobile node requesting the streaming service cannot obtain the suited video resource from the overlay networks, the media server needs to provide the initial streaming. When the nodes playing the video content can perceive the watching quality cannot meet their demand, they re-seek and connect with the new media service source (other nodes carrying video resource or server).

IV. CNVS DETAIL DESIGN

4.1 Neighborhood Layer

Each node $node_i$ considers the one hop neighbor in its wireless signal range as neighborhood candidates. $node_i$ obtains the list of nodes geographically located in its neighborhood by making use of a "location-aware" solution as described in [26]. According to ***Definition 1***, some of these nodes are selected as neighborhood candidates of $node_i$ and form a node set denoted as $locL_i$.

Definition 1 Set *dis* as $node_i$'s wireless signal range. If the geographical distance between $node_i$ and $node_j$ is less than *dis*, $node_j$ is considered a neighborhood candidate of $node_i$.

The selection of neighborhood for $node_i$ relies on two factors: signal strength and available bandwidth which are used to evaluate the communication quality. The high communication quality can ensure the high-efficiency video data transmission. The subset of nodes from $locL_i$ with which $node_i$ can communicate at higher quality will be selected. By making use of the bandwidth estimation approach in our previous work [25], the bandwidth estimation value between $node_i$ and $node_j$ can be defined as:

$$AB_{ij} = \frac{const}{RTT_{ij} \times \sqrt{PLR}} \quad (1)$$

Let sig_{ij} denote a quality value which is computed based on the signal strength between the two nodes. In terms of Grey Relational Analysis (GRA) [29], the estimation attributions - signal strength and bandwidth of items in $locL_i$ are normalized according to eq. (2).

$$x_{ij}^{*}(att) = \frac{x_{ij}(att) - lower_{att}}{upper_{att} - lower_{att}}, x_{ij}^{*}(att) \in [0,1] \quad (2)$$

where *att* denotes the attribution of evaluation (signal strength and bandwidth to $node_i$). $lower_{att}$ and $upper_{att}$ are minimum and maximum corresponding to current attribution *att* for neighborhood candidates, respectively. $x_{ij}(att)$ is the value of attribution *att* of $node_j$. The Grey Relational Coefficient (GRC) of IN nodes can be obtained according to eq. (3).

$$GRC_{ij} = \frac{1}{\sum w_{att} \,|\, x_{ij}^{*}(att) - 1 \,|\, + 1} \quad (3)$$

where w_{att} is the weight value of $x_{ij}^{*}(att)$ and each attribution has the different value of w_{att}. For instance, we focus on the bandwidth to $node_i$ so as to set the higher value of w_{att} than signal strength.

The members in $locL_i$ whose *GRC* is greater than the threshold *S* become members of the neighborhood list of $node_i$, denoted as $neinodeL_i$, $neinodeL_i \in locL_i$. Any $node_i$ in the network can be associated a 3-tuple $node_i = (NID_i, locL_i, neinodeL_i)$. NID_i is the ID of $node_i$, $locL_i$ is next hop node set of $node_i$ and $neinodeL_i$ is the neighborhood list of $node_i$. For $node_i$, the neighborhood construction algorithm is described as follows:

Algorithm 1: neighborhood node selection for $node_i$

```
for each node from next hop nodes set $locL_i$ of $node_i$;
//count($locL_i$) is size of set $locL_i$;
for($j$ = 0; $j$ < count($locL_i$); $j$++ )
    get $AB_{ij}$ and $sig_{ij}$ between $node_i$ and $node_j$ of $locL_i$;
    normalizes $AB_{ij}$ and $sig_{ij}$ by eq. (2);
    computes $GRC_{ij}$ of $node_j$ by eq. (3);
    if $GRC_{ij} > S$ then
        put $node_j$ into $neinodeL_i$;
    end if
end for $j$
```

Along with the movement of mobile nodes, the spatial position relationship between the neighborhood nodes dynamically changes, and therefore the $locL_i$ set. Additionally the signal strengths and bandwidth vary in time; they should be periodically measured, the quality of communication metric recomputed and the $neinodeL_i$ updated. A solution is to set a time interval *T*, and all the nodes $node_i$ should update their 3-tuple every time *T*, including their node sets $locL_i$ and $neinodeL_i$. Note that the message used to detect node's next hop nodes should be one-hop multicast message. This localizes the discovery procedure and increases its performance.

4.2 Cluster Layer

The cluster is the resource-centric node group, which means that one or multiple mobile node(s) carrying video resource and several ordinary mobile nodes form a cluster. We consider each mobile node playing video content as the video resource carrier. Initially, in terms of cross-layer approach, each mobile node $node_i$ playing video resource res_x appends the information of res_x into the one-hop multicast message sent to its neighbors so that each neighbor of $node_i$ is aware of accessing res_x in one-hop. $node_i$ and its neighbors form a cluster. $node_i$ can use the hash value $H(i)$ of node ID of all neighbors as the uniqueness cluster ID and assign to each neighbor. The members in the cluster $H(i)$ invite their neighbors to join $H(i)$ by making use of appending the information of resources in cluster and cluster ID into the detection message. As any mobile node $node_j$ receives the message containing information of invitation, it makes the decision whether joining cluster $H(i)$. If the higher cost obtaining the streaming service from *H*(*i*) than the lower bound *C*(*j*) of $node_j$'s QoS, $node_j$ rejects the invitation from sender. If the invitation sender perceives the decrease in the accessing cost and $node_j$ still is its neighbor, it re-invites $node_j$ to join current cluster. When $node_j$ accepts the access cost provided by *H*(*i*), it becomes a new member of cluster *H*(*i*). The dissemination of resource information assisted by one-hop neighbors not only enables the mobile nodes discover the available streaming service source, but also the overlay nodes discover other nodes playing the same video content to achieve the service source switchover which will be detailed next. Eq. (4) describes the access cost of each mobile node relative to res_x in cluster *H*(*i*).

$$C(H(i)) = w_{ij} \times \overline{AB_{ij}}, C(H(i)) > 0 \qquad (4)$$

Where w_{ij} is an impact factor and $\overline{AB_{ij}}$ is the average bandwidth between $node_i$ and $node_j$ according to multiple detection periods $TP=(t_1,t_2,\ldots,t_Y)$ and defined as:

$$\overline{AB_{ij}} = \frac{\sum_{c=1}^{Y} AB_{ij}(t_c)}{Y} \qquad (5)$$

where $AB_{ij}(t_c)$ denotes the bandwidth value between $node_i$ and $node_j$ at t_c. *Y* is the total number of detection at $node_i$ side. The variance σ_{ij} can be defined as:

$$\sigma_{ij} = \sqrt{\frac{\sum_{c=1}^{Y} (AB_{ij}(t_c) - \overline{AB_{ij}})^2}{Y}} \qquad (6)$$

where σ_{ij} is used to indicate the range of bandwidth variation. The estimation range of bandwidth can be defined as: $R_{ij} = [\overline{AB_{ij}} - \sigma_{ij}, \overline{AB_{ij}} + \sigma_{ij}]$. We need to re-divide *TP* into multiple sub-periods in order to investigate the bandwidth variation level. Let $AB_{ij}(t_{k-h}) \in R_{ij} \to AB_{ij}(t_k) \notin R_{ij} \to AB_{ij}(t_{k+v}) \in R_{ij}$ denote an equilibrium event, namely the bandwidth value re-belongs to R_{ij} after experiencing $v+h$ detection periods. Then $v+h$ is an equilibrium period ($v+h$ also can be considered as resilience period of communication quality). Moreover, we consider the variation level of number of intermediate node between $node_i$ and $node_j$. As we know, the increase in the number of intermediate node in the path of accessing res_x results in the long transmission delay, high packet loss rate and probable link break. Therefore we investigate the distribution composed of the equilibrium period and average hop between $node_i$ and $node_j$ according to eq. (7).

$$D = \{(p_1, \hat{h}_1), (p_2, \hat{h}_2), \ldots, (p_u, \hat{h}_u)\}, u \le Y \qquad (7)$$

By making use of Least Square Method (LSM) [30] to implement LRF for *D*, we use the correlation coefficient to indicate the w_{ij} according to eq. (8)

$$w_{ij} = \frac{\sum_{c=1}^{u} |p_c - \bar{p}|\,|\hat{h}_c - \bar{\hat{h}}|}{\sqrt{\sum_{c=1}^{u} |p_c - \bar{p}|^2} \times \sqrt{\sum_{c=1}^{u} |\hat{h}_c - \bar{\hat{h}}|^2}}, w_{ij} \in [0,1] \qquad (8)$$

where $\bar{\hat{h}}$ and $\bar{p}$ are the mean value of average equilibrium period and the related average hop, respectively and their values can be obtained according to eq. (9).

$$\bar{p} = \frac{\sum_{c=1}^{u} p_c}{u}, \bar{\hat{h}} = \frac{\sum_{c=1}^{u} \hat{h}_c}{u}, \bar{p}, \bar{\hat{h}} \in [0,1] \qquad (9)$$

The mobility of node leads to the dynamic *C*(*H*(*i*)) so that the members in the cluster *H*(*i*) can remove own cluster ID (leave *H*(*i*)) at any moment. Along with the movement of each cluster *C*(*H*(*i*)) member $node_x$, it may receive the inviting message from the members of another cluster *H*(*a*). If the access cost provided by cluster *H*(*a*) cannot meet the demand of $node_x$, $node_x$ keeps the original cluster ID and rejects the invitation. Otherwise, $node_x$ needs to make a decision to join *H*(*i*) or *H*(*a*). Eq. (10) describes the membership value of $node_x$ belonging to *H*(*i*).

$$p_x^{(H(i))} = \frac{|neinodeL_x^{(H(i))}|}{|neinodeL_x|} \qquad (10)$$

where $|neinodeL_x|$ is the number of node in $node_x$'s neighbor set $neinodeL_x$ and $|neinodeL_x^{(H(i))}|$ is the number of the node belonging to cluster *H*(*i*) in $neinodeL_x$. $node_x$ can select the cluster with maximum of membership value and join it according to eq. (11).

$$P_{\max} = MAX[p_x^{(H(a))}, p_x^{(H(i))}, \ldots, p_x^{(H(v))}] \qquad (11)$$

The high node density in the cluster can ensure the high reliability of delivering streaming data. The mobility can influence the connection stability with the resource supplier for the cluster with low node density. By making use of eq. (10), the cluster having high node density can merge the cluster low node density. The carrier of res_x also changes its cluster ID and joins other cluster by exploiting eq. (11).

Any cluster *H*(*i*) may include multiple mobile nodes carrying video resource due to cluster mergence. We transform eq. (4) to eq. (12) to support calculating the access cost of multiple resources in cluster.

$$\hat{C}(H(i))=\sqrt{\sum_{e=1}^{s}C_e(H(i))^2}\,,\hat{C}(H(i))>0 \quad (12)$$

where s is the number of video resource in cluster $H(i)$. If $\hat{C}(H(i))\geq\sqrt{s}C(j)$, $node_j$ rejects joining cluster $H(i)$, where $\hat{C}(H(i))$ is the Euclidean distance from origin to point mapped by corresponding access cost of each resource in s-dimension space and $\sqrt{s}C(j)$ is $node_j$'s lower bound of QoS. Otherwise, if $\hat{C}(H(i))<\sqrt{s}C(j)$, $node_j$ becomes the member of cluster $H(i)$. We also transform eq. (10) to eq. (13) to support the cluster mergence in the multiple resources condition.

$$p_x^{(H(i))}=\frac{|neinodeL_x^{(H(i))}|}{|neinodeL_x|}\arctan(1+s_{H(i)}) \quad (13)$$

where $\arctan(1+s_{H(i)})$ is the impact factor to reflect the influence from the number of resource in the cluster. By making use of eq. (12) and (13) to calculate $P_{\max}$ in the multiple resources condition, $node_x$ can join the suited cluster. Each member of cluster can share the cost of accessing each video resource in the cluster with its neighbor node having the same cluster ID in the period time μT ($0<\mu<1$).

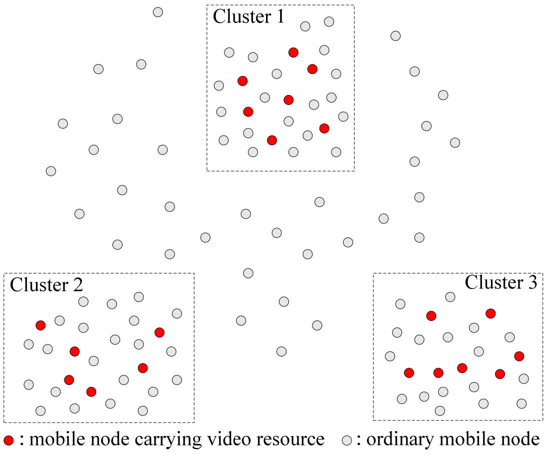

Fig.2 *Example of cluster*

As Fig. 2 shows, along with the movement of mobile nodes, the cluster can be the node set having high node and resource density after going through continuous cluster mergence.

4.3 Media Server

The media server stores original video resources to provide the streaming service for the requesting nodes when the overlay network lacks the corresponding resources requested or the mobile nodes cannot search the suited resource which can meet the demand of their QoS. If the mobile nodes do not join the any cluster, they need to send the requesting message containing the resource ID to the server. The server provides the initial streaming service for the requesting nodes. Otherwise, if the requesting nodes are the member of cluster, they take precedence of seeking desired resource from the nodes in the same cluster. This is the fact that the low hop can ensure the high efficiency of delivering streaming data and the number of interactive message with the server can be reduced. If these members in the cluster cannot obtain the resource requested from other members of cluster, it needs to obtain the streaming service from the server. Moreover, we propose the service source switchover mechanism to ensure the overlay nodes always connect with the optimal service source as follows.

(1) Each mobile node $node_i$ connecting with the server enters the cluster range. If the member $node_j$ in the cluster can provide the streaming service for $node_i$, $node_i$ disconnects with the server and contact the service source $node_j$ in the cluster. Otherwise, if there is no the video resource needed by $node_i$ in the cluster, $node_i$ keeps the connection with the server and acts as the service source for other nodes in the cluster.

(2) $node_i$ obtains the streaming service from other member in the same cluster. If $node_i$ leaving the cluster range leads to the decrease in the efficiency of delivering streaming data, $node_i$ needs to disconnect with the service source in the cluster and requests the streaming service from server.

(3) If $node_i$ perceives the decrease in the transmission efficiency of streaming data from the member in the same cluster so that $node_i$'s demand cannot be met, $node_i$ re-seeks and connect with the new service source from other members in the same cluster. If the members in the cluster do not include the resource requested, $node_i$ requests the streaming service from server.

V. PERFORMANCE EVALUATION

5.1 Simulation Settings

(1) ***Parameter Settings:*** A mesh-based topology is built in MANET by making use of NS-2. CNVS was modeled and implemented in NS-2, as described in the previous sections. The values of specific CNVS parameters are set following the parameter analysis described in the previous paragraph. *dis* is set to 200

m as the threshold. An ideal situation for the simulation is considered: the signal between mobile nodes does not weaken in coverage, so the selection of neighbor nodes relies on the available bandwidth. Important factors in the mobile scenario such as node's speed, node's number and node's signal coverage, etc. are considered. Through repeated experiments, T is set to 5 *s* as the updating time interval for one-hop neighbors of a node. μ is set to 0.6.

(2) ***Testing Topology and Scenarios:*** The following content discusses the setting of a common simulation environment for the two solutions.

The lower level architecture includes 200 mobile nodes. The nodes are located in a range of x=1500 *m* and y=1500 *m*. The mobile speed range of nodes is between 10 and 30 *m*/*s*. The direction of each node is randomly assigned and the nodes' pause time is 0. The signal range of nodes is set to 200 *m*. The wireless routing protocol used is DSR. The default distance is set to 6 hops between the server and any node. These nodes uniformly join the P2P overlay every 1 second from 0 *s* until 60 *s*. Fifteen nodes which are receiving streaming media data may randomly quit the multimedia streaming system in the time interval from 30 *s* to 60 *s*. The simulation time is 80 *s*. The bandwidth of the media server and each node is 10 *Mb*/*s* and 2 *Mb*/*s,* respectively. The rate and transport protocol of streaming data sent by any serving node or media server is 480 *kb*/*s* and UDP, respectively. In CNVS, the size of exchange message between nodes and server such as detection and requesting resource message is set to 2 *KB* payload. Messages are sent over TCP. MESHCHORD uses the same message size like CNVS. Finally we set 6 mesh routers for MESHCHORD to cover the whole network.

5.2 Performance Evaluation

The performance of CNVS is compared with that of MESHCHORD in terms of end-to-end delay, network throughout, average hop of streaming data delivery, packets loss rate, routing overhead and traffic packets, respectively.

(1) ***End-to-end delay:*** We calculate the total delay time of receiving data at the application layer during intervals 1 *s* long. The total delay time divided by the amount of data received is used to indicate the average end-to-end delay. In addition, by observing the simulation results of the two systems, the largest delay of MESHCHORD and CNVS is less than 5 *s* and there is some packet loss. In order to allow graphical representation of the average delay, if the number of received packets is 0 during an interval, the average delay is not considered and rule that its value is 6 *s* (6 *s* corresponds to an infinite delay).

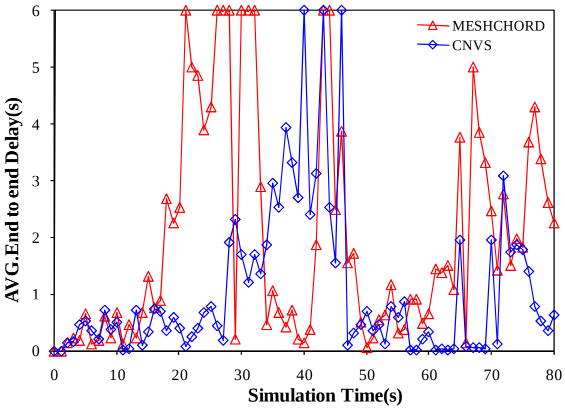

Fig. 3 *Comparison of average end-to-end delay*

TABLE I
COMPARISON OF AVERAGE END-TO-END DELAY BETWEEN CNVS AND MESHCHORD

Time(*s*)	**Average end-to-end delay(*s*)**	
	CNVS	**MESHCHORD**
5	0.523954	0.644693
10	0.494766	0.679028
17	0.691806	1.877763
20	0.393104	2.53812
28	1.903318	∞
30	1.688635	∞
35	2.965031	1.064102
40	∞	0.136526
48	0. 303343	1.72229
50	0.703343	0.057738
55	0.577268	0.300432
60	0.322552	0.650339
65	1.951922	3.759304
70	1.946799	2.464349
75	1.786409	1.81839
80	0.629865	2.24511

Fig. 3 and Table I illustrate the comparison between MESHCHORD and CNVS in terms of delay. Through the results displayed in Table I to observe two curves' change process in Fig. 3, MESHCHORD's delay curve experiences a slight jitter before time t=17 *s*, fast rises at t=17 *s*, enters first congestion from t=17 *s* to t=48 *s* (31 *s* long) and finally is affected by the second congestion from t=50 *s* to t=80 *s*. CNVS's delay curve has the same shape, but the two congestion periods start at t=28 *s* and t=70 *s* and are shorter: 18 *s* and 9 *s*, respectively. The video data rate reaches the peak and is maintained stable after the sixtieth node joining the overlay. If no node leaves, the congestion continues to be recorded after 80 *s*. Next we discuss the difference between CNVS and MESHCHORD results, respectively.

As Fig. 3 shows, the start time of CNVS's congestions are later and congestion periods are shorter than those experienced by MESHCHORD. The degree of congestion (as measured by the average delay) is lower for CNVS than for MESHCHORD. For example the maximum delay experienced by MESHCHORD is 5 *s*, with 25% higher than that of CNVS (close to 4 *s*). This is mostly due to the fact that the nodes in CNVS search for the source node within close physical distance. With node movement, the distance between data source and receiver can change, but also nodes can perceive the change of distance and switch to a new source node. The shorter the physical distance between data source node and receiver, the number of hops required for forwarding content data and control messages decreases. This reduces the time the data travels through the network, alleviating congestion and therefore the helps achieve lower overall end-to-end delays.

In MESHCHORD, the nodes search for the serving source nodes in terms of the coverage range of the mesh router. As the mesh router helps find source nodes with low physical distance to the resource requesting nodes, at the beginning low delay are achieved. However, with node movement, the distance between the requesting and source nodes may increase, communication quality between them may decrease, and communication delay is likely to increase. This is as the requesting nodes in MESHCHORD cannot obtain real-time location information and therefore cannot update their source nodes in a timely manner. It is therefore natural to see significant differences of up to 33% in the delay between MESHCHORD and CNVS, in favor of the CNVS. Although CNVS cannot avoid the congestion, it delays it a bit and lessens its effect through the intelligent mechanism for automatic switching the service nodes.

(2) ***Network throughput:*** In order to assess the performance in terms of throughput, the total number of packets received in the overlay during a certain period of time is divided to the length of that period of time. Network throughput comparison results between CNVS and MESHCHORD are shown in Fig. 4 and Table II.

As Fig. 4 and Table II illustrate, there is fast increase in throughput for both CNVS and MESHCHORD from time t=0 *s* to t=17 *s*, with only slightly better performance in favor of CNVS. MESHCHORD throughput experiences a gradual decrease from t=17 *s* to t=35 *s* due to the effect of the first congestion period, it increases slowly again from t=36 *s* to t=69 *s* (before the second congestion), and falls from t=70 *s* to the end. Unlike MESHCHORD, CNVS's throughput continues to increase from the beginning to t=17 *s* and through to t=29 *s*, well into the first congestion period. The throughput slowly decreases due to the congestion from t=30 *s* to t=41 *s*, but remains almost twice higher than that of MESHCHORD. For both solutions, there is another rise in throughput from t=42 *s* to t=60 *s*, followed by a slight decrease during the second congestion period from t=61 *s* to the end. Looking at Fig. 4, it can be clearly seen how CNVS outperforms MESHCHORD in all situations, congestion affected or not.

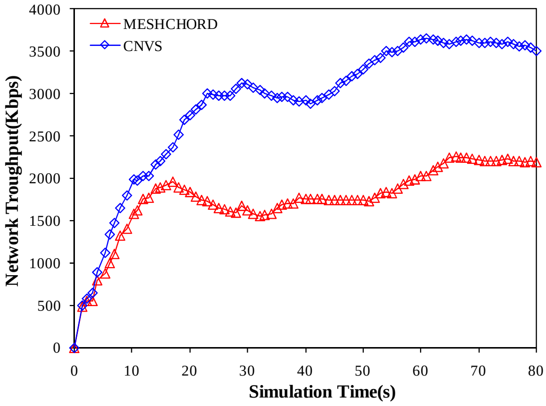

Fig. 4 *Comparison of network throughput*

TABLE II
COMPARISON OF NETWORK THROUGHPUT BETWEEN CNVS AND MESHCHORD

Time(*s*)	**Network throughput(*Kbps*)**	
	CNVS	**MESHCHORD**
5.32	1118.741	873.310
10.23	1988.861	1588.663
17.14	2361.495	1963.310
20.14	2748.360	1835.088
25.37	2976.803	1647.975
29.36	3123.058	1673.246
36.12	2967.867	1691.785
41.09	2887.021	1754.673
45.06	3027.039	1746.627
50.29	3294.965	1741.516
55.04	3497.034	1829.739
61.06	3656.413	2031.757
67.03	3634.164	2249.611
69.09	3627.737	2231.890
75.04	3613.684	2225.890
80	3501.684	2194.890

When sixty nodes enter the overlay in succession, increasingly large amount of data is exchanged, be it content or control messages. The total volume of data in the network should fast increase from t=1 *s* to t=60 *s* (period in which the nodes enter the overlay from the first one to the sixtieth one) and be maintained constant from t=60 *s* until the nodes start leaving the network, when the amount of data should decrease.

However, the results illustrated by Fig. 4 indicate that the throughput of both MESHCHORD and CNVS is also affected by congestion, namely between *t*=17 *s* and *t*=35 *s* and *t*=17 *s* and *t*=29 *s*, respectively. As fifteen nodes randomly leave the overlay from *t*=30 *s* to *t*=60 *s*, the volume of data exchanged is directly reduced and congestion is alleviated. However new nodes continue to join the overlay resulting in additional amounts of data. After a relative slow rise, the throughput curves fall again due to appearance of the second congestion. If the effect of congestion can be mitigated, the throughput may continue to rise before falling due to nodes leaving the network. This is not the case for MESHCHORD, whose throughput reaches the maximum at *t*=67 *s* in terms of the result displayed by Table II, much less than the theoretical peak (12000~14521 *kbps*). Unlike that, CNVS's throughput increases further as it deploys the dynamic algorithm which improves the physical distance between data source and receiver during data delivery. However even CNVS's throughput achieves saturation and finally gradually decreases.

As Fig. 4 and Table V show, the throughput of CNVS is higher than that of MESHCHORD, exhibiting better increment rates and longer growth time periods. As we know, in CNVS when the requesting node perceives the existence of a higher priority source node, it switches its delivery from this node. The switchover ensures closer physical distance and better communication quality between source and receiver so that network traffic in CNVS is reduced. Furthermore, the packets with higher number of forwarding hops have higher probability to be dropped during congestion. Conversely, the packets with lower number of forwarding hops have lower loss rate and therefore CNVS has better throughput. Despite of the fact that MESHCHORD selects very good sources for data delivery to its nodes, as it does not adapt to the changing delivery conditions, including to node movement as CNVS does, MESHCHORD's throughput is almost half that of CNVS.

(3) ***Average number of hops and packet loss rate:*** Content data delivery accounts for most of the network traffic. Delivery performance is highly influenced by the number of hops (number of intermediate forwarding nodes) data goes through and the node packet loss rate (node throughput) which is defined as the ratio between the number of packets loss and the total number of packets sent at the application layer. Statistical results are collected and shown every 10 *s* in order to be more convenient for the analysis. Fig. 5, Fig. 6 and Table III illustrate the comparison between CNVS and MESHCHORD results for the average hop count and packet loss rate, respectively.

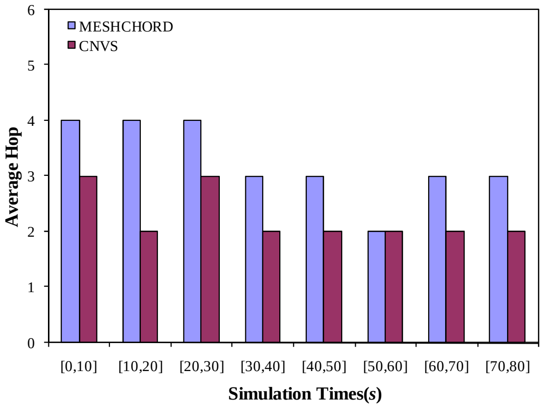

Fig. 5 *Average hop of streaming data delivery*

TABLE III
COMPARISON OF AVERAGE HOP OF STREAMING AND PACKET LOSS RATE BETWEEN CNVS AND MESHCHORD

Time(*s*)	**Average hop of streaming data delivery**		**Packet loss rate**	
	CNVS	**MESHCHORD**	**CNVS**	**MESHCHORD**
0~10	4	4	0.225	0.372
10~20	3	4	0.569	0.693
20~30	3	4	0.705	0.852
30~40	3	3	0.742	0.869
40~50	2	3	0.687	0.882
50~60	2	2	0.712	0.872
60~70	2	3	0.807	0.868
70~80	3	3	0.758	0.875

In Fig. 5 and Table III, MESHCHORD's average number of hops starts at 4, is invariant until *t*=30 *s*, decreases to 2 from *t*=30 *s* to 60 *s* and finally rises to 3 from *t*=60 *s* to the end. CNVS's average number of hops falls from *t*=0 *s* to *t*=70 *s* from 4 to 2 and rises to 3 in the last 10 *s*. It can be clearly seen how CNVS's overall performance in terms of the average number of hops is better than that of MESHCHORD. Next the reasons behind this positive result are discussed.

In both MESHCHORD and CNVS data sources include the server and source nodes. In the first 10s most of the data is sent by the server which stores the original data. As there is an average of 6 hops from the server to any node, both solutions have a high average number of hops at the beginning. As more nodes join the overlay, data traffic increases in network and the amount of data served directly by the server reduces. As source nodes close to the requesting nodes are found, the average number of hops for data delivery decreases. In fact the first 10 *s* could be considered transitory and could be removed from the overall result analysis. The increase in the number of nodes and in the volume of data exchange between nodes reduces the average hop count. However the physical distances between nodes change due to node movement and therefore the hop count varies. CNVS which has an intelligent algorithm which reacts to node movement and dynamically re-assigns source nodes closer to the requesting node, experiences a decrease in the average hop count between t=10 *s* and t=40 *s*, whereas MESHCHORD's average number of hops is maintained roughly constant. Moreover, the congestion caused by the increase in the amount of data in the network leads to the data needing to be excessively forwarded has higher loss probability. The increase in loss rate is a key factor in the average hop count reduction during congestions. However the hop count value is even lower for CNVS than that of MESHCHORD as the former searches for optimal source nodes to minimize the negative effect of node movement. An excessive number of hops can be an early indication of congestion and result in higher delays.

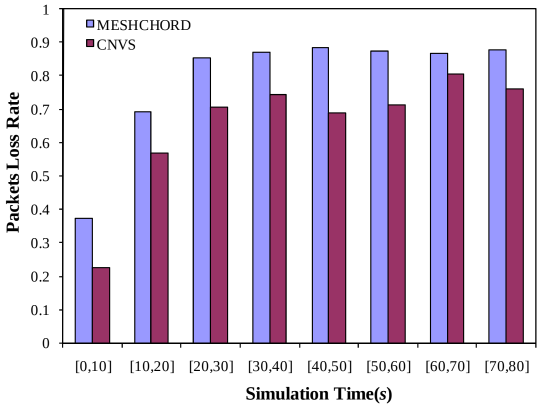

Fig. 6 *Packet loss rate*

In Fig.6 and Table III, where CNVS and MESHCHORD are compared in terms of the packet loss rate, it can be seen how the rate rapidly increases after the transitory period of 10 *s*. With increasing number of nodes joining the system, the increasing amount of traffic determines buffer overflow in several nodes in the transmission paths. This increase in packet loss rate caused by congestion determines a rise in the average loss rate for both solutions. However by dynamically switching the source nodes to ones closer to the requesting nodes in terms of physical location is very effective in reducing the number of hops and loss probability and therefore improving packet loss rate. Therefore it can be seen clearly how CNVS has lower loss rate than MESHCHORD during the whole simulation.

(4) ***Routing overhead and total number of traffic packets:*** Next the performance of CNVS and MESHCHORD are compared in terms of routing overhead and total number of traffic packets exchanged. The number of routing messages exchanged in the network layer is used to indicate the routing overhead. Routing overhead is counted every 10 *s*. The traffic packets are comprehensively considered both routing overhead and number of application data in the process of node entering overlay. Fig. 7, Fig. 8, Table IV and Table V illustrate the results.

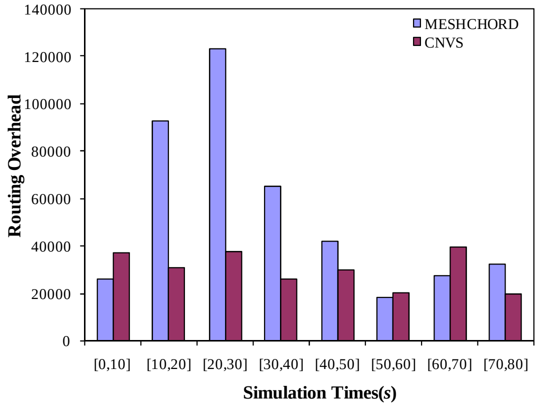

Fig. 7 *Routing overhead*

TABLE IV
COMPARISON OF ROUTING OVERHEAD BETWEEN CNVS AND MESHCHORD

Time(*s*)	**Routing overhead(number of messages)**	
	CNVS	**MESHCHORD**
0~10	37098	25874
10~20	30826	92880
20~30	37450	123025
30~40	25922	65091
40~50	30048	41985
50~60	20045	18349
60~70	39528	27327
70~80	19956	32415

In Fig. 7 and Table IV it can be seen how MESHCHORD overhead experiences dramatic changes with the increase in number of nodes. For example, in terms of the results in Table IV, if the number of messages exchanged reaches 25874 in the first 10 *s*, this number nearly becomes six times higher in the third interval (reaches 123025). Unlike MESHCHORD, CNVS has low routing overhead and most importantly is maintained almost constant regardless of the variation in the number of nodes in the system, exhibiting extremely good scalability.

Next a CNVS-MESHCHORD comparison-based analysis of the results is performed.

Initially, CNVS performs the process of cluster generation. The number of messages exchanged in CNVS leads to larger routing overhead and packet number than those of MESHCHORD. Additionally, whenever the number of clusters in network changes (for example when nodes are moving or leaving), the nodes change their cluster IDs and an increase in the number of messages is encountered.

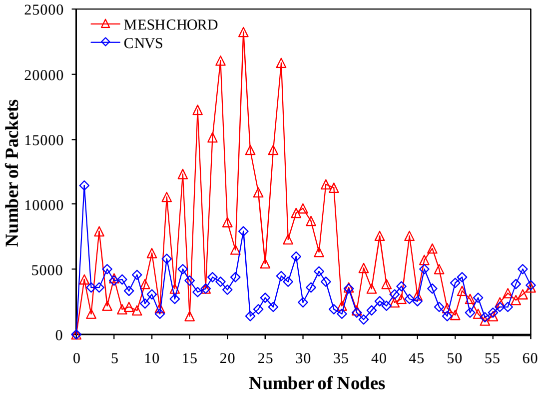

Fig. 8 *Total number of traffic packets*

TABLE V

COMPARISON OF TOTAL NUMBER OF TRAFFIC PACKETS BETWEEN CNVS AND MESHCHORD

Number of nodes	**Total number of traffic packets**	
	CNVS	**MESHCHORD**
5	4156	4361
10	3128	6292
15	4213	1484
20	3519	8666
25	2860	5493
30	2475	9752
35	1624	2257
40	2595	7598
45	2572	3017
50	3974	1534
55	1703	1434
60	3863	3624

The small number of nodes in the overlay is responsible for low routing overhead in the early simulation time. With the continuous increase in the number of nodes requiring video resources, the number of data packets and therefore the number of routing messages also increase until congestion appears. During congestion, many packets are discarded and therefore the number of routing messages reduces. Furthermore, as fifteen nodes leave, indirectly the routing overhead and data packets number decreases.

The routing overhead of MESHCHORD increases sharply from t=0 s to 30 s due to the fact that node movement increases the physical distance between the statically selected source-receiver nodes pair. The routing overhead decreases from t=30 s to t=60 s due to some of the nodes leaving the system and increase loss rate due to congestion. After t=60 s, the routing overhead rises again. Unlike MESHCHORD, CNVS maintains a relative stable routing overhead due to the fact that its source-receiver node pairing algorithm ensures low physical distance between the nodes at any moment. The only slight increase in the overhead is due to the change of clusters-related messages which is experienced by CNVS from t=50 s to t=70 s.

VI. CONCLUSION

This paper proposes a Mesh-based Mobile Peer-to-Peer network (CNVS) for efficient video resource sharing in MANETs. CNVS bases its efficiency on a two layers architecture which includes neighborhood and cluster layers – reflecting the close geographical location and communication quality between the mobile nodes and video content distribution. By making use of cross-layer approach and cooperation principle to dynamically clustering mobile nodes, CNVS addresses the mobility of node and discovery of video resource distributed in the mobile nodes so that the mobile nodes can perceive the close service source carrying the desired resource in the geographical distance. The service source switchover mechanism ensures the overlay node obtain the continuous optimal streaming service.

CNVS's performance was assessed in comparison with that of a state of the art alternative solution MESHCHORD via simulations. The results show how CNVS ensures efficient video delivery between mobile nodes and achieves lower average end-to-end delay, lower average number of hops for data delivery, lower routing overhead and packet loss rate, less packet traffic and up to six times higher network throughput. Future work will involve deploying VCR-like functions in conjunction with CNVS and optimizing CNVS for high speed mobility wireless networks.

ACKNOWLEDGEMENT

This work was supported in part by the National High-Tech Research and Development Program of China (863) under Grant no. 2011AA010701, in part by the National Basic Research Program of China (973 Program) under Grant 2013CB329102, in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61001122, 61003283, and 61232017, in part by the Jiangsu Natural Science Foundation of China u nder Grant no. BK2011171, in part by the Jiangxi Natural Science Foundation of China under Grant no. 20122BAB201042, in part by the Fundamental Research Funds for the Central Universities under Grant No. 2012RC0603, and in part by Lero under Grant no.10/CE/I1855.

References

[1] GUAN Quansheng, YU F. R., JIANG S. M., *et al*. Topology Control in Mobile Ad Hoc Networks with Cooperative Communications[J]. IEEE Wireless Communications, 2012, 19(2): 74-79.

[2] XU Changqiao, FALLON E., QIAO Yuansong, *et al*. Performance Evaluation of Multimedia Content Distribution over Multi-Homed Wireless Networks[J]. IEEE Transactions on Broadcasting, 2011, 57(2): 204-215.

[3] XU Changqiao, LIU Tianjiao, GUAN Jianfeng, *et al*. CMT-QA: Quality-Aware Adaptive Concurrent Multipath Data Transfer in Heterogeneous Wireless Networks[J]. IEEE Transactions on Mobile Computing, 2012, pp(99):1.

[4] WANG Danqi, YEO C.-K. K. Superchunk-Based Efficient Search in P2P-VoD System Multimedia[J]. IEEE Transactions on Multimedia, 2011, 13(2): 376-387.

[5] LUSS H. Optimal Content Distribution in Video-on-Demand Tree Networks[J]. IEEE Transactions on Systems, Man and Cybernetics, 2010, 40(1): 68-75.

[6] XU Changqiao, MUNTEAN G.-M. M., FALLON E., *et al*. A Balanced Tree-based Strategy for Unstructured Media Distribution in P2P Networks[C]// Proceedings of IEEE International Conference on Communications (ICC'08). Beijing, China: IEEE Press, 2008: 1797-1801.

[7] XU Tianyin, CHEN Jianzhong, LI Wenzhong, *et al*. HAMDI M. Supporting VCR-like Operations in Derivative Tree-Based P2P Streaming Systems[C]// Proceedings of IEEE International Conference on Communications (ICC'09). Dresden, Germany: IEEE Press, 2009: 1-5.

[8] XU Changqiao, MUNTEAN G.-M. M., FALLON E., *et al*. Distributed Storage-Assisted Datadriven Overlay Network for P2P VoD Services[J]. IEEE Transactions on Broadcasting, 2009, 55(1): 1-10.

[9] YIU W.-P K., JIN Xing, CHAN S.-H.-H G. VMesh: Distributed Segment Storage for Peer-to-Peer Interactive Video Streaming[J]. IEEE Journal on Selected Areas in Communications, 2007, 25(9): 1717-1731.

[10] CHAN S.-H.-H G, YIU W.-P K. Distributed Storage to Support User Interactivity in Peer-to-Peer Video, United States Patent 7,925,781 [P]. Hong Kong University of Science and Technology, April, 2011.

[11] LIU C.-L., WANG C.-Y., WEI H.-Y. Cross-Layer Mobile Chord P2P Protocol Design For VANET[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2010, 6(3): 150-163.

[12] ZHANG Xinyan Yan, LIU Jiangchuan, BO Li, *et al*. CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming[C]// Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05). Florida, USA: IEEE Press, 2005: 2102-2111.

[13] OH H. R., SONG H. J. An Effective Mesh-Pull-Based P2P Video Streaming System Using Fountain Codes with Variable Symbol Sizes[J]. Computer Networks, 2011, 55(12): 2746-2759.

[14] DA S., ANA P.C., LEONARDI E., *et al*. Chunk Distribution in Mesh-Based Large-Scale P2P Streaming Systems: A Fluid Approach[J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22(3): 451-463.

[15] FOUDA M. M., TALEB T., GUIZANI M. On Supporting P2P-Based VoD Services over Mesh Overlay Networks[C]// Proceedings of IEEE International Conference on Global Telecommunications (GLOBECOM'09). Hawaii, USA: IEEE Press, 2009: 1-6.

[16] HEI Xiaojun, LIANG Chao, LIANG Jian, *et al*. A Measurement Study of a Large-Scale P2P IPTV System[J]. IEEE Transactions on Multimedia, 2007, 9(8): 1672-1687.

[17] ANDREA P. A Survey on Content-Centric Technologies for The Current Internet: CDN and P2P Solutions[J]. Computer Communications, 2012, 35(1): 1-32.

[18] LI Jingwei, CHAN S.-H.-H G. Optimizing Segment Caching for Mobile Peer-to-Peer Interactive Streaming[C]// Proceedings of IEEE International Conference on Communications (ICC'10). Cape Town, South Africa: IEEE Press, 2010: 1-5.

[19] SI Pengbo, YU R.R., JI Hong, *et al*. Distributed Sender Scheduling for Multimedia Transmission in Wireless Mobile Peer-to-Peer Networks[J]. IEEE Transactions on Wireless Communications, 2009, 8(9): 4594-4603.

[20] BERRY R.A., YEH E.M. Cross-layer wireless resource allocation[J]. IEEE Signal Processing Magazine, 2004, 21(5): 59-68.

[21] CONTI M., GREGORI E., TURI G. A Cross-Layer Optimization of Gnutella for Mobile Ad Hoc Networks[C]// Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing. Illinois, USA: ACM Press, 2005: 343-354.

[22] MELODIA T., AKYILDIZ I.F. Cross-layer QoS-Aware Communication for Ultra Wide Band Wireless Multimedia Sensor Networks[J]. IEEE Journal on Selected Areas in Communications, 2010, 28(5): 653-663.

[23] MILANI S., CALVAGNO G. A Low-Complexity Cross-Layer Optimization Algorithm for Video Communication Over Wireless Networks[J]. IEEE Transactions on Multimedia, 2009, 11(5): 810-821.

[24] STOICA I., MORRIS R., KARGER D., *et al*. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications[C]// Proceedings of ACM SIGCOMM. California, USA: ACM Press, 2001: 149-160.

[25] XU Changqiao, ZHAO Futao, GUAN Jianfeng, *et al.* QoE-driven User-Centric VoD Services in Urban Multi-homed P2P-based Vehicular Networks[J]. IEEE Transactions on Vehicular Technology, 2012, pp(99):1.

[26] CANALI C., RENDA M. E., SANTI P., *et al.* Enabling Efficient Peer-to-Peer Resource Sharing in Wireless Mesh Networks[J]. IEEE Transactions on Mobile Computing, 2010, 9(3): 333-347.

[27] DO T. T., HUA K. A., JIANG Ning, *et al.* PatchPeer: A Scalable Video-on-Demand Streaming System in Hybrid Wireless Mobile Peer-to-Peer Networks[J]. Peer-to-Peer Networking and Applications, 2009, 2(3): 182-201.

[28] DORIA S. S., SPOHN M. A. A Multicast Approach for Peer-to-Peer Content Distribution in Mobile Ad Hoc networks[C]// Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'09). Budapest, Hungary: IEEE Press, 2009: 1-6.

[29] RAZZAQ A., MEHAOUA A. Layered Video Transmission Using Wireless Path Diversity Based on Grey Relational Analysis[C]// Proceedings of IEEE International Conference on Communications (ICC'11). Kyoto, Japan: IEEE Press, 2011: 1-6.

[30] BO Y., BALANIS C.A. Least Square Method to Optimize The Coefficients of Complex Finite-Difference Space Stencils[J]. IEEE Antennas and Wireless Propagation Letters, 2006, 5(1): 450-453.

Biographies

JIA Shijie, received the M.S. degree from Kunming University of Science and Technology, Kunming, China in 2009. He is currently working toward the Ph.D. degree in the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include peer-to-peer networks, wireless communications and cognitive networks.

XU Changqiao, 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 at BUPT. His research interests include wireless networking, multimedia communications and next generation Internet technology. Email: cqxu@bupt.edu.cn

MUNTEAN Gabriel-Miro, 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 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 120 papers in prestigious international journals and conferences, has authored a book and eleven book chapters and has edited four 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.

GUAN Jianfen, received his Ph.D. degrees in communications and information system from the Beijing Jiaotong University, Beijing, China, in Jan. 2010. He is a lecturer in the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His main research interests focus around mobile IP, mobile multicast and next generation Internet technology.

ZHANG Hongke, 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 was the chief scientist of a National Basic Research Program (“973” program). He is now the Head of the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Director of the Next Generation Internet Technology Research Center at BUPT, and Director of the National Engineering Laboratory for Next Generation Internet Interconnection Devices at BJTU.