Performance-Aware Mobile Community-based VoD Streaming over Vehicular Ad Hoc Networks

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

***Abstract*—Leveraging the development of mobile communication technologies and increased capabilities of mobile devices, mobile multimedia services have set new trends. In order to support high-quality Video-on-Demand (VoD) in mobile wireless networks, using virtual communities-based approaches to balance the efficiency of content sharing and maintenance cost of the performance-aware solutions has attracted increasing research interests. In this paper, we propose a novel Performance-aware Mobile Community-based VoD streaming solution over vehicular ad hoc networks (PMCV). PMCV relies on a newly designed mobile community detection scheme and an innovative community member management mechanism. The former employs a novel fuzzy ant-inspired clustering algorithm and an innovative mobility similarity estimation model to group together the mobile users with similar behavior in terms of playback and movement into mobile communities. The latter introduces the role and task of members, member join and leave, collaborative store and search for resources and replacement of broker member. Simulations-based testing shows how PMCV outperforms another state of the art solution in terms of performance.**

***Index Terms*—VoD streaming, mobile community, vehicular ad hoc networks.**

I. Introduction

THE latest advancements in vehicular wireless technologies foster a number of new applications in vehicular ad-hoc networks (VANETs) [1]-[3], including rich media-based [4]-[8]. The extended features of the new protocols such as the IEEE 802.11p and development of mobile multimedia technologies enable provision of the high-bandwidth required by these multimedia services for drivers and passengers in various complex road scenarios [9]. In particular, Video-on-Demand (VoD) services also require support for interactivity in order to enhance viewer quality of experience levels [10]-[15]. Yet, the provision of high-quality VoD services in VANETs is very challenging. The dynamic pseudo-random user behavior when interacting with the content leads to fragmentation, high maintenance cost and low sharing efficiency of video resources. Moreover, the variation of the geographical distance between vehicular nodes caused by their high mobility severely affects the delivery efficiency of video data [16]. Therefore, a hybrid solution based on the playback behavior of users, which integrates distributed video resources, supports high efficiency search for video content and addresses the high mobility of vehicular nodes should be considered for VoD in VANETs.

Recently, numerous researchers have shown great interest in making use of virtual communities for management of resources [17]-[27] in order to improve content sharing performance and reduce the unnecessary energy consumption by gathering users with common characteristics. For instance, SPOON [22], a community-based peer-to-peer (P2P) content file sharing solution in disconnected mobile ad-hoc networks, groups the users which have common interest in shared files and frequent interactions as a community. SPOON achieves fast location of available resources and reduces the search for resources effort. In *C5* [24], the mobile users which are geographically close for a period of time and require the same resources are clustered as a group and collaboratively fetch the content, enhancing the efficiency of resource searching and reducing the number of search messages.

The traditional community identification approaches are classified in graph partitioning, hierarchical clustering and partitional clustering [28]. However, their inherent drawbacks lead to fragile and imprecise community structures. For instance, graph partitioning relies on preliminary knowledge about the size of community so that the rough partition component size negatively influences the partitioning accuracy level. Hierarchical clustering cannot accurately describe the difference level between communities due to its reliance on the measurement method of similarity between objects, so it introduces numerous errors related to classifying objects. Partitional clustering has the advantage that it does not need to provide any community size or similarity measurement method, but the predefined number and centers of communities also affect clustering accuracy. Constructing a VoD-oriented mobile community needs not only to describe accurately the dynamic playback behavior of users to meet the demand of

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61372112 and Grant 61232017, by the Fundamental Research Funds for the Central Universities, by the National Key Basic Research Program of China (973 Program) under Grant 2013CB329102, by the Beijing Natural Science Foundation under Grant 4142037, by the Science Foundation Ireland under the International Strategic Cooperation Award Grant SFI/13/ISCA/2845.

C. Xu and M. Wang are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China (e-mail: {cqxu, wangmu}@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 (e-mail: 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).

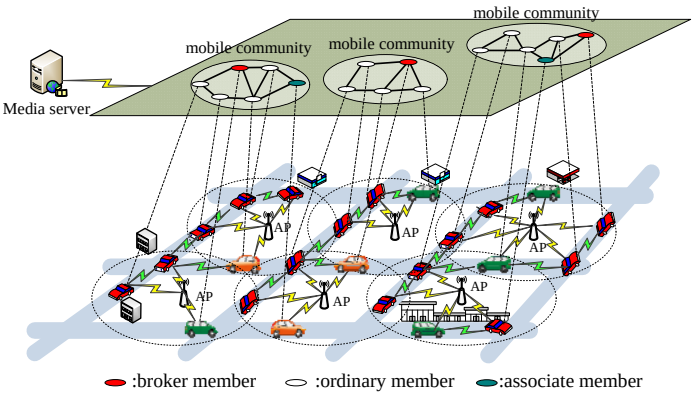


Fig. 1. Overall System Architecture

smooth playback and streaming interactivity, but also address the mobility problem of mobile users, enhancing the video delivery efficiency.

In this paper, we propose **a novel Performance-Aware Mobile Community-based VoD streaming solution over vehicular ad hoc networks (PMCV)**. PMCV employs a newly designed mobile community detection scheme to group together the mobile users with similar behavior in terms of playback and movement. The mobile community detection scheme includes a newly designed *Fuzzy Ant Clustering algorithm* (FAC) and a *mobility similarity measurement model* (MSMM). FAC makes use of a novel *ant colony clustering algorithm* and a fuzzy C-Means to refine the popular patterns for finding the mobile users with similar playback behavior. By making use of a Markov process to analyze the trajectory and state of movement of mobile users, MSMM finds the mobile users with similar mobility. As Fig. 1 shows, the vehicles which have similar playback behavior and mobility are grouped into a mobile community, enhancing the transmission efficiency of video data.

Furthermore, a *mobile community management mechanism* (MCMM) which introduces the role and tasks of members, member joining and moving between communities, collaborative storage and search for resources and replacement of broker members is proposed. Extensive tests show how PMCV achieves better results in comparison with other state of the art solutions in terms of seek latency, packet loss rate, throughput, video quality, server stress, overhead and lookup success rate.

II. Related Work

Recently, there has been increasing research interest in proposing community-based P2P resource sharing solutions, which can balance the maintenance costs and resource sharing efficiency. For instance, *Wang et al*. [18] have examined the performance of P2P file sharing in Twitter communities. The results show that the online patterns of users with common interest are very similar so that there is a great opportunity for improving the file sharing capacity by using innovative solutions. *Iamnitchi et al.* [19] have proposed an interest-sharing graph structure which captures common user interests in accessed data and have designed an information dissemination system based on the small-world interest-sharing graphs. However, the structure depends on predefined threshold to define edge weight in graphs, which leads to inaccurate estimation of user interest and reduction of the file sharing performance. The authors of SPOON [22] have designed a community detection algorithm which measures interest similarity in terms of file content stored by mobile nodes and interaction frequency. This algorithm is used to group the mobile nodes in a community and results in high file sharing efficiency in intra-community and inter-community. However, because the community detection performance depends on the similarity between the stored file content and interaction frequency, SPOON cannot accurately estimate the similarity level of the node interest and does not clearly define the community boundary, negatively influencing the file search efficiency. Also, as SPOON does not consider the mobility of mobile nodes, the increasing geographical distance between community members in case of high mobility situations such is the case in VANETs leads to low-efficiency file delivery.

Doulkeridis et al. [25] have proposed a self-organizing P2P system using hierarchical clustering to identify peers with similar content and deliberately assign them to the same super-peer. The resources are indexed by few super-peers only to improve the performance of query processing. However, the proposed hierarchical clustering algorithm also uses the similarity between stored content as a metric for clustering users and so the accuracy of the clustering cannot be ensured. *Datta et al*. [26] have proposed two algorithms: centralized and decentralized K-means clustering approaches in order to ensure communication efficiency by uniformly distribing the data and computing resources in a P2P overlay and by considering peer costs. However, the local centralized K-means clustering faces the local optimal solution problem, the difference between sample nodes also reduces the uniform distribution effect, and making use of threshold to formulate the boundary of clusters can reduce the accuracy of clustering results. Moreover, the decentralized K-means clustering depends on the assumption that the network is static. HP2PC [27] introduces a hierarchy of P2P neighborhoods by partitioning the clustering problem across neighborhoods and by making use of distributed K-means. However, the hierarchical clustering depends on the measurement method of similarity between objects in terms of the interaction between peers in P2P networks. The boundary between clusters cannot clearly be explained so this reduces the clustering accuracy.

These community-based P2P file sharing solutions cannot handle the dynamic playback behavior of users, so they are unsuitable for VoD services. Inspired by the community concept, some P2P-VoD solutions make use of clustering the nodes which store similar video resources in order to improve the resource sharing efficiency and reduce the maintenance costs. For instance, SURFNet [15] groups some nodes which store the same superchunk data into a holder-chain and attaches the chain to a stable node with long online time in an AVL tree. This solution supports streaming interactivity and obtain a logarithmic search time for seeking within a video stream. However, the maintenance cost of the AVL tree highly depends

TABLE I
DIFFERENCE BETWEEN EXISTING SOLUTIONS AND PMCV

	VoD	Mobile Networks	Community Detection	Scalability
"The Small World " [18]	No	No	Graph Partitioning	High
SPOON [21]	No	Yes	Partitional Clustering	High
"Content Similarity" [24]	No	No	Hierarchical Clustering	High
"K-Means Clustering " [25]	No	No	Partitional Clustering	High
HP2PC [26]	No	No	Hierarchical Clustering	High
SURFNet [15]	Yes	No	No	Low
SocialTube [19]	No	No	Hierarchical Clustering	High
QUVoD [4]	Yes	Yes	Partitional Clustering	Low
PMCV	Yes	Yes	Partitional Clustering	High

on the number and state stability of the members in the tree. Additionally, with the increase in scale, the tree and chain have become larger, so the scalability of SURFNet is limited. SocialTube [20] investigates the social relationship and interest similarity between users to construct group-based P2P overlays by analyzing playback trace of users. However, SocialTube makes use of the video categories to partition user interest groups so that the accuracy of the partitioned results cannot be ensured, namely there is high interest difference level between in-group members. Our previous work QUVoD [4] constructs a group-based P2P overlay over 4G networks. The nodes having similar video resources form a group in order to distribute uniformly the video chunks along the P2P overlay, reducing chunk seeking traffic. However, the maintenance overhead of the Chord overlay limits the scalability of QUVoD. As QUVoD does not consider the mobility of vehicles, the increasing geographical distance between vehicles reduces the transmission efficiency in vehicle-to-vehicle (V2V) mode. Additionally, the increasing traffic results in higher costs for users as a higher traffic share is offloaded from V2V to 4G.

Table I lists the major differences between existing solutions and the PMCV, proposed in this paper. The solutions in [19], [22], [25], [26] and [27] employ community detection algorithms to cluster the nodes in P2P networks and design community-based file sharing schemes. These solutions do not support VoD services, but their community structures enable high system scalability. Moreover, SPOON [22] also supports file sharing in a mobile network environment. SURFNet [15] provides streaming interactivity service, but it does not include a community detection algorithm and its tree-based P2P overlay has low scalability. SocialTube [20] is oriented to the video content sharing and has high scalability, but it does not include an approach related to streaming interactivity. SURFNet and SocialTube cannot be deployed in mobile networks. QUVoD [4] clusters nodes in terms of the similarity of caching video content, but its Chord structure limits system scalability. It supports streaming interactivity and can be deployed in VANETs.

III. PERFORMANCE-AWARE MOBILE COMMUNITY-BASED VOD STREAMING OVER VEHICULAR AD HOC NETWORKS (PMCV) OVERVIEW

For convenience, Table II defines several notations which are used in this paper. PMCV focuses on constructing a community-based P2P VoD solution in VANETs, which supports efficient distribution, management and search for video content. The performance of the community-based P2P VoD solution in VANETs depends on the accuracy of clustering nodes and efficiency of managing video resources. The in-community members which have common characteristics can cooperatively store and distribute video resources, improving sharing efficiency of video content. Therefore, PMCV relies on FAC to accurately find common behavior patterns between nodes, reducing the negative effects caused by random access of video content such as frequent replacement of stored video content, high start-up delay and low resource search success rate. The vehicular nodes' mobility also affects negatively the video content delivery. Therefore, PMCV makes use of MSMM to address the performance problem introduced by the mobility of the vehicular nodes by estimating the similarity of node mobility in the future. Further, PMCV employs MCMM to balance the management cost and search efficiency for resources. This supports large-scale VoD services and results in high quality of experience (QoE) levels for the users in VANETs.

As Fig. 1 shows, the media server, which originally stores all video resources, provides support for the streaming service to all vehicular nodes. We assume that all vehicular nodes request the same video content $Video$ (note the proposed solution is applied similarly to any number of video streams). The vehicular nodes which join the PMCV system find partners for video content sharing from the system members, namely those which have similar playback behavior and mobility; these members build and maintain logical links between them to form mobile communities.

The PMCV community members can have one of three roles: *broker*, *ordinary* member and *associate* member, respectively, in order to achieve autonomous community management and distribute local video resources. PMCV makes

TABLE II
NOTATIONS USED BY PMCV

Notations	Descriptions
c_a	video chunk a
S_{tr}	historical playback trace library stored in media server
tr_i	playback trace i in S_{tr}
n	total number of video chunks
V_i	vehicular node i in VANETs
AP_a	access point a visited by V_i
s_a	state of V_i which enters the coverage area of AP_a
mt_i	movement trajectory of V_i
TH_m	threshold of mobility similarity
C_a	mobile community
n_k	broker member in C_a
n_j	member in C_a

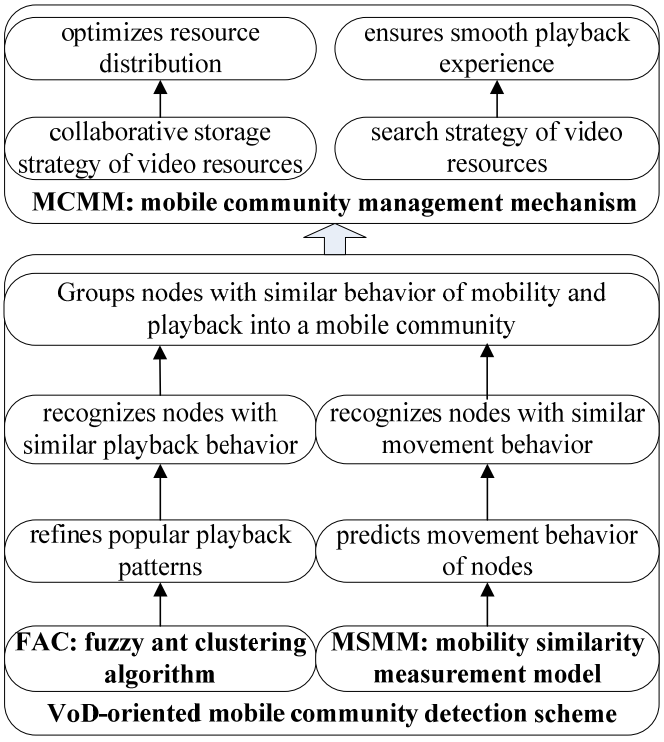


Fig. 2. PMCV Components

use of mobile community structure to integrate the distributed video data carried by vehicular nodes in terms of their common characteristics, support high-efficiency resource sharing and enhance system scalability.

As Fig. 2 shows, PMCV relies on newly designed mobile community detection and management mechanisms. The mobile community detection scheme includes a novel FAC and a MSMM which supports vehicular nodes in finding the community members with similar behavior in terms of playback and mobility. MCMM formulates the role and task of community members based on FAC and MSMM to support the community self-management and high-efficiency distribution of video content.

(1) FAC employs newly designed ant colony clustering algorithm to pre-cluster the historical playback trace library for further refining the popular playback patterns. FAC makes use of fuzzy C-Means to partition and reconstruct the pre-clustered results into multiple clusters based on similar playback content. The playback patterns are extracted from these clusters and are used to recognize the nodes with similar playback behavior.

(2) MSMM makes use of a Markov process to predict the movement behavior of vehicular nodes in terms of movement trajectory. The predicted results and movement trajectory are used to estimate the similarity of mobility of vehicular nodes. The nodes with similar behavior of mobility and playback form the mobile communities.

(3) MCMM manages the process of members joining and leaving communities, the role assignment of members and switchover between roles. Moreover, it includes the collaborative storage strategy of video resources in the communities and search strategy of video resources in order to optimize the resource distribution and smooth the playback experience.

IV. VoD-oriented Mobile Community Detection

A. User Playback Pattern

Information from the playback traces is considered for assessment of user experience and to provide semantic description for the video content. By clustering the similar items in historical playback trace library, we extract the popular user playback behavior, which is considered as user playback pattern. The user playback pattern plays an important role in the VoD community detection, which is responsible for recognizing users with similar playback behavior. As users with similar playback behavior may need similar or the same video content in the future, they are grouped into a community whose members can collaboratively fetch the video content and efficiently share local resources. An effective and robust clustering method can accurately describe the playback characteristics, so as to find the available user playback patterns.

The discrete items in the playback trace library stored in the media server are grouped in terms of the ant colony clustering algorithm in FAC. Different from traditional ant clustering algorithms, FAC considers two interest factors reflected by traces: **playback time** and **continuity** to construct the initial trace heaps and select the heap centers. The traces are preprocessed to determine several raw trace heaps whose items have close interest levels. The traces in different interest-based heaps have different playback content. However, the traces in the same interest-based heap may include different viewing content. Further, FAC makes use of a *fuzzy C-Means* solution to partition the interest-based heaps and refine the new heap centers into user playback patterns.

1) Construction of the initial trace heaps:

The purpose of this stage is to initially cluster the traces in order to ensure high cluster result accuracy and reduce the "noise" in clustering results (i.e. traces which have not been clustered). FAC makes use of the playback time and continuity to describe the traces and map them into a closed space on a plane. Each ant in FAC searches an unknown region based on a Hilbert curve and intelligently carries the data in terms of exploratory results.

Any video resource $Video$ saved in the media server is normally divided into n chunks with equal length: $Video \Leftrightarrow (c_1, c_2, ..., c_n)$. Each trace is defined as $tr_i = (c_a, c_b, ..., c_v), tr_i \in S_{tr}, v \leq n$ where S_{tr} denotes the historical playback trace library, a, b, v denote the ID of chunks watched by user i and n is the total length of the video.

The number of items in tr_i indicates the playback time of user i, which is defined as $|tr_i|$ (tr_i's length). The chunks which are sequentially included in tr_i form a substring. For instance, c_1, c_2 and c_3 form sequentially occurred video chunk subset in $tr_j = (c_1, c_2, c_3, c_5, \ldots c_n)$ and are considered as a substring. The longer the length of a substring in tr_i is, the higher the interest level of user i for the current content is. tr_i's

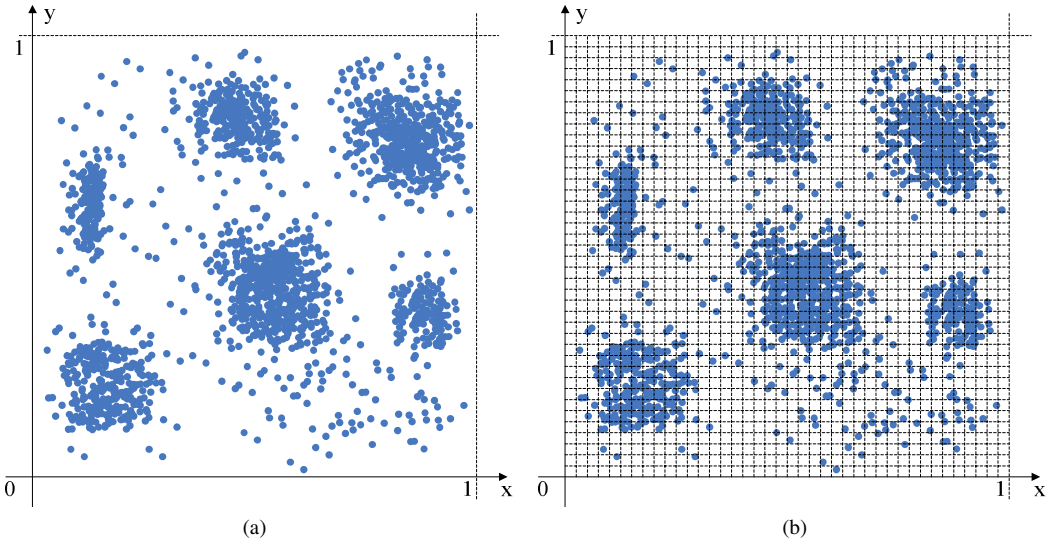


Fig. 3. Data in two-dimensional coordinate system

playback continuity can be defined in terms of the substring lengths as in eq. (1):

$$u_{tr_i} = \sum_{c=1}^{h} |str_c|, str_c \in tr_i \tag{1}$$

where str_c denotes a substring in tr_i, $|str_c|$ is the length of str_c and h is the number of substrings in tr_i, respectively. For convenience, $|tr_i|$ and u_{tr_i} are normalized according to eq. (2).

$$L_{tr_i} = \begin{cases} \frac{|tr_i|}{n}, 0 \leq \frac{|tr_i|}{n} \leq 1 \\ 1 \quad , \frac{|tr_i|}{n} > 1 \end{cases}, \overline{u}_{tr_i} = \begin{cases} \frac{u_{tr_i}}{n}, 0 \leq \frac{u_{tr_i}}{n} \leq 1 \\ 1 \quad , \frac{u_{tr_i}}{n} > 1 \end{cases} \tag{2}$$

As Fig. 3 (a) shows, all items in S_{tr} can be mapped into a two-dimensional coordinate system according to the values of L_{tr_i} and $\overline{u}_{tr_i}$. Any 1×1 region can be divided into $m \times m$ cells for the purpose of ant clustering as illustrated in Fig. 3 (b) where m can be set to $2\sqrt{|S_{tr}|}$ ($|S_{tr}|$ is the number of items in S_{tr}). If the data in the region divided is located at cell edges, it needs to be placed in the adjacent cells. The data on an edge is preferentially placed into the adjacent cell which already has data in order to form heaps or merged into existing heaps. If more of the adjacent cells include data, the data located on an edge should be placed in the cell which has most data. If all these adjacent cells are empty, the data is moved to a random adjacent cell. This preprocessing achieves the construction of a mapping relationship between data and cells and makes the required preparations for ant clustering.

The ants in FAC have a specified search behavior which is different from the random movement in a traditional ant clustering algorithm. As Fig. 4 shows, they consider their origin as a starting point and explore the unknown cells in terms of a $2^2 \times 2^2$ Hilbert curve, where each $2^2 \times 2^2$ cell region is considered as a subdomain. In the process of searching the unknown cells in the region, the ant picks up the data in the current cell and records the number of cells visited so that the ant possibly carries multiple data. When an ant has searched in an unknown subdomain, it calculates the data density according to eq. (3).

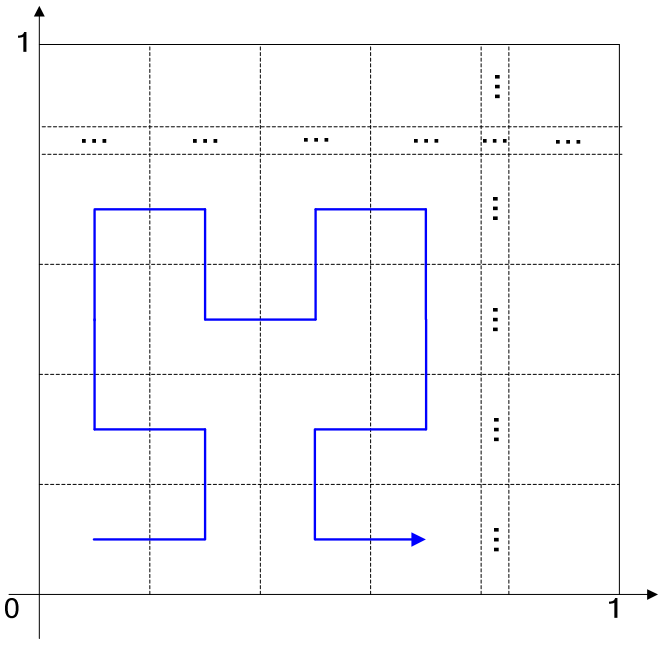


Fig. 4. $2^2 \times 2^2$ Hilbert curve

$$den = \frac{N_{data}}{N_{cell}} \tag{3}$$

In eq. (3) N_{data} denotes the number of data items carried by an ant and N_{cell} is the number of cells searched by the ant. After the ant has searched in an entire unknown subdomain, it evaluates the variation of data density in the subdomains searched. The criterion of stopping the search process and dropping the data gathered by the ant is defined as $den^{(k)} > den^{(k+1)}$, namely the data density in the $k+1^{th}$ subdomain

Algorithm 1 Ant A_i search process

```
A_i picks up and carries data in two subdomains;
k = 1;
q = k + 1;
calculates data density den^(k) and den^(q) by eq. (3);
while (den^(k) < den^(q))
    k = q;
    q++;
    A_i picks up and carries data in q^th subdomain;
    A_i updates the data amount collected N_data;
    A_i records the number of cells searched N_cell;
    calculates data density den^(q) by eq. (3);
end while
```

searched is less than that in the k^{th}. The convergence condition for the search indicates that the ant goes from a subdomain with dense data to one with sparse data. The ant marks the cells searched, indicating that the data in these cells has been processed. When an ant drops the data carried, it leaves the region. Any subsequent ant neglects the labeled cells and continues to search in new cells. The pseudo-code of the above process is detailed in Algorithm 1.

After the above-described initial clustering, the region contains several small and large data heaps and single data cells. In order to clear up these single data cells and merge the small-scale heaps into larger ones, there is a need for further processing. Each heap needs to have a heap center which has the smallest difference to the other in-heap members. The interest distance between heap member d_i and d_j indicates their difference level and is obtained according to eq. (4).

$$D_{ij} = \sqrt{\left(L_{tr_i} - L_{tr_j}\right)^2 + \left(\overline{u}_{tr_i} - \overline{u}_{tr_j}\right)^2} \tag{4}$$

The lower the difference level between d_i and d_j is, the more similar their interest-related characteristics are. We calculate the interest distance between d_i and other in-heap members according to eq. (4) and further obtain the mean value of these interest distances, according to eq. (5).

$$\overline{D_i} = \frac{\sum_{c=1}^{|HP_k|} D_{ic}}{|HP_k|} \tag{5}$$

where $|HP_k|$ is the amount of data in the heap HP_k. If $\overline{D_i}$ is less than the mean value of interest distance of other in-heap members, d_i is considered as heap center. The merge process between two heaps relies on the interest distance between their centers and centers' mean value, as follows.

Rule 1: if the interest distance D_{ij} between centers d_i and d_j of two heaps HP_k and HP_h is less than either $\overline{D_i}$ or $\overline{D_j}$, HP_k and HP_h will form a new heap and the merged heap will select a new heap center.

After the merging process according to **Rule 1**, the small-scale heaps which have short interest distances between them in the same region form large-scale heaps. The method of handling single data is similar to the heap merge, as follows.

Rule 2: if the interest distance D_{ab} between a single data item d_a and center d_b of a heap HP_p is less than that to other heap centers, d_a becomes a new member of HP_p.

Ant clustering makes use of interest-based characteristics to group the traces into several large-scale heaps in which the in-heap traces have low interest distance and high interest distance with the traces in other heaps. Let S_{HP} denote a heap set to store these existing heaps, namely $S_{HP} \Leftrightarrow (HP_1, HP_2, ..., HP_v)$. The number of items in the trace library S_{tr} determines the search range of ants, namely the convergence rate of construction algorithm of the initial interest heaps relies on the number of cells searched by the ants. Therefore, the computational complexity of the algorithm is $O(|S_{tr}|^2)$.

2) Generation of playback patterns:

As we know, the larger the interest distance between two traces is, the higher the difference level of their playback pattern is. However, shorter interest distance cannot ensure that the playback patterns of traces are similar. In order to enhance the clustering accuracy, we measure the pattern similarity between traces.

Two traces tr_i and tr_j are converted to binary string BS_i and BS_j with length n, where each bit corresponds to a video chunk in proper sequence. The values of the corresponding bits for the chunks included in the trace are set to 1 and the remaining bits are set to 0. In terms of the Hamming Distance [29], the pattern distance between tr_i and tr_j is defined as in eq. (6):

$$D_{play_{ij}} = D_{\text{Hamming}}(BS_i, BS_j) \tag{6}$$

We reselect the heap center for each interest-based heap clustered. Similar to the selection method of interest-based heap center, the pattern-based center should have the minimum mean value of the pattern distance to other in-heap data. The mean value is defined as in eq. (7):

$$\overline{D}_{play_i} = \frac{\sum_{c=1}^{|HP_k|} D_{play_{ic}}}{|HP_k|} \tag{7}$$

Based on the heaps resulted from the clustering process and their pattern-based centers, we make use of Fuzzy C-Means [30] to refine all items in S_{HP} and create the new pattern-based heaps, as follows.

Step 1 The first step calculates the objective function value $J_m^T(W, V)$ according to eq. (8).

$$J_m^T(W,V) = \sum_{i=1}^{|S_{tr}|}\sum_{j=1}^{c} w_{ij}^m D_{play_{ij}}, \sum_{j=1}^{c} w_{ij} = 1, w_{ij} \geq 0 \tag{8}$$

where m is the degree of fuzziness, $D_{play_{ij}}$ denotes the pattern distance between any in-heap member tr_i and pattern-based center tr_j, c is the number of existing heaps and T is the number of iterations. w_{ij} is the membership of the i^{th} data item in the j^{th} heap. Let L_{remove} denote a list which is used to store the removed members from the heap.

Step 2 If S_{HP} is a nonempty set, any heap $HP_k \in S_{HP}$ is selected as the refinement object and S_{HP} removes HP_k. Otherwise, if S_{HP} is an empty set, all items in S_{HP} are refined into pattern-based heaps and this iteration ends.

Step 3 HP_k removes a member d_u which has the maximum pattern distance with the center of HP_k. HP_k reselects a new pattern-based center and obtains $J_m^{T+1}(W, V)$.

Step 4 If $J_m^{T+1}(W,V) < J_m^T(W,V)$, d_u is a noise member and pushed into L_{remove} and the algorithm returns to Step 3.

Step 5 If $J_m^{T+1}(W,V) \geq J_m^T(W,V)$, all members in the current HP_k form a pattern-based heap. If L_{remove} is a nonempty set, all items in L_{remove} form a new interest-based heap HP_h. HP_h is pushed into S_{HP} and the algorithm returns to Step 2.

After the above algorithm has ended, all interest-based heaps become pattern-based heaps. The heap centers in the pattern-based heaps are considered as user playback patterns, namely $S_{pattern} \Leftrightarrow (p_1, p_2, ..., p_k)$, $S_{pattern} \in S_{tr}$. The convergence rate of the construction algorithm of pattern heaps relies on the number of removed noise members from the initial interest heaps, so the computational complexity of the algorithm is $O(|S_{tr}|)$.

B. User Mobility Similarity Estimation

Vehicle movements determine dynamic transmission distances between the communication parties, which results in negative effects in terms of the delivery quality of video data, mostly due to long delays and high packet loss rates. Sharing video resources between vehicles which have similar movement trajectories are beneficial for the quality of delivery. This is as similar mobility of vehicles ensures maintaining relative close inter-vehicle transmission distances. In the process of movement of vehicles, they go through multiple access points (AP). Each vehicle V_i can record the list of encountered APs and residence time in each AP range (i.e. time difference between time of entering an AP range and time of leaving the AP range). Let $mt_i = (AP_a, AP_b, \ldots, AP_v)$ be the movement trajectory of V_i, where each item is an AP beside which V_i has passed. Estimating the mobility between vehicles by making use of current vehicle movement trajectory cannot ensure the similarity of their movement trajectory in the future. Therefore, the proposed MSMM employs a Markov process to consider the similarity of future mobility of vehicular nodes and also includes vehicle residence time in each AP range to weight the estimation of mobility similarity.

Each vehicular node V_i uses a 4-tuple $V_i = (ID_i, ST_i, RT_i, A_i)$ to store the mobility information, where ID_i and A_i are the ID of the node and the AP, respectively and ST_i and RT_i are the residence time in A_i range and timestamp associated with the of recording, respectively. All items in mt_i are ranked in terms of their timestamp. The movement from an AP to another AP in VANETs is considered as a transition of vehicle state. For instance, let s_a denote a state of V_i entering in the coverage area of AP_a. The movement from AP_a to AP_b is equivalent with V_i's state transfer from s_a to s_b, which can be described as a Markov process [31] $\{X_n, n > 0\}$, $X_n \in S$. X_n denotes the n^{th} transition state and $S = \{s_1, s_2, ..., s_v\}$ denotes a state set whose members are corresponding to all items in mt_i. Let T_n ($T_0 = 0$) be n^{th} state transition time and $t_{s_a}^{(e)}$ be an event - the duration time of s_a is t (the residence time of V_i in AP_a). The occurrence probability of $t_{s_a}^{(e)}$ in the state transition process $s_a \rightarrow s_b$ at T_n can be defined as in eq. (9):

$$\begin{aligned} T_{ab} &= \Pr\{X_{n+1} = b, T_{n+1} - T_n \leq t | X_n = a\} \\ &= \Pr\{X_{n+1} = b | X_n = a\} \times \\ &\quad \Pr\{T_{n+1} - T_n \leq t | X_{n+1} = b, X_n = a\} \\ &= p_{ab} H_{ab} \end{aligned} \tag{9}$$

where $H_{ab} = \Pr\{T_{n+1} - T_n \leq t | X_{n+1} = b, X_n = a\}$ denotes the occurrence probability of $t_{s_a}^{(e)}$ before $s_a \rightarrow s_b$ is triggered and p_{ab} is the probability of state transition $s_a \rightarrow s_b$ at T_n. Let t_{s_a} be the duration time of s_a based on the condition of ignoring the next transition state. According to eq. (1), the occurrence probability of t_{s_a} can be defined as:

$$D_a(t_i) = \sum_{c=1}^{v} T_{ac} \tag{10}$$

$D_a(t)$ is considered as the occurrence probability of event - the residence time of V_i in the coverage area of AP_a is t. If another vehicular node V_j goes through the coverage area of AP_a, $D_a(t_j)$ can be obtained according to eq. (10). We calculate the similarity value of residence time of V_i and V_j in AP_a according to eq. (11) in order to estimate the similarity level of movement speed.

$$RS_{ij} = \operatorname{arc\,cot}(\left| E(D_a(t))_i - E(D_a(t))_j \right|) \times \frac{2}{\pi} \tag{11}$$

where $E(D_a(t_i))$ and $E(D_a(t_j))$ are the expectation value of $D_a(t_i)$ and $D_a(t_j)$, respectively. The smaller the difference between $E(D_a(t_i))$ and $E(D_a(t_j))$ is, the more similar the movement speeds of V_i and V_j when passing the coverage area of AP_a are. In terms of the Markov process, we construct the $v \times v$ transition probability matrix $P = \begin{pmatrix} p_{11} & p_{12} & \ldots & p_{1v} \\ p_{21} & p_{22} & \ldots & p_{2v} \\ \ldots & \ldots & \ldots & \ldots \\ p_{v1} & p_{v2} & \ldots & p_{vv} \end{pmatrix}$ where p_{ab} is the transition probability of $s_a \rightarrow s_b$ and v is the length of the state set S. The measurement of movement trajectory similarity relies on two factors: the historical movement trajectory and movement state predicted by the matrix P for the future. The movement trajectory similarity between V_i and V_j is defined as in eq. (12):

$$sim(V_i, V_j) = \frac{\sum_{c=1}^{v} p_{ac}^i \cdot p_{ac}^j}{\sqrt{\sum_{c=1}^{v} (p_{ac}^i)^2} \cdot \sqrt{\sum_{c=1}^{v} \left(p_{ac}^j\right)^2}} \tag{12}$$

We make use of the similarity of residence time from eq.(11) to weight the movement trajectory similarity, so as to obtain the mobility similarity between V_i and V_j according to eq.(13).

$$MS_{ij} = sim(V_i, V_j) \times \sum_{c=1}^{|mt_i \cap mt_j|} RS_c(V_i, V_j) \tag{13}$$

Where $|mt_i \cap mt_j|$ returns the number of items of the intersection of mt_i and mt_j and MS_{ij} denotes the mobility similarity between V_i and V_j. Let TH_m be the threshold to evaluate the mobility similarity between vehicular nodes. If MS_{ij} is greater than a threshold TH_m, the mobility pattern

of the two nodes V_i and V_j is similar. The measurement of mobility similarity not only considers the static character - resident state similarity, but also investigates the historical movement trajectory and future movement state, ensuring the accuracy of measured results. The number of states in the transition probability matrix and the number of access points encountered by the vehicular nodes determine the computational complexity of MSMM, which is $O(v^2)$.

V. Mobile Community Management Mechanism

The media server provides original video content to all vehicular nodes and does not interfere with the autonomous management of mobile communities. However, the server needs to make use of FAC to calculate popular playback patterns used to construct mobile communities. Moreover, it maintains a local community list by periodically receiving information about the community members from the broker members of all communities and returns the updated list to each broker member, supporting in this way member movement between communities. The nodes make use of the received playback patterns from the media server to find the nodes with similar playback behavior by exchanging messages containing playback traces. If the nodes which belong to the same playback pattern have similar mobility (i.e. their mobility similarity values are greater than the threshold TH_m), they form a mobile community. Because the members in the same community have the same playback pattern, they can collaboratively obtain and store certain video chunks in order to meet the demand of chunk search of in-community members and reduce the number of inter-community requests.

The members in a mobile community are assigned one of three possible roles: broker, associate or ordinary member; any community has one broker member and multiple ordinary and associate members. The *broker member* is a particular member responsible for maintaining the contact with broker members in other geographically adjacent communities and for distributing resources to and from the ordinary members of its community. The *ordinary members* in a community have the same playback pattern and have similar mobility patterns. The nodes which join a community but do not meet the requirements of mobility similarity with all members in current community are considered as *associate members*.

Each member n_j stores a local member list $S_{member_{n_j}} \Leftrightarrow (n_1, n_2, ..., n_h)$ whose items are community members n_c which are represented via a 3-tuple as follows: $n_c = (c_t, T_c^P, F_c)$. c_t is the video chunk stored in n_c's static buffer, T_c^P denotes the n_c's current playback point within the video and F_c is a flag bit to mark the member role. $F_c = 1$ denotes that n_c is the broker, $F_c = 0$ indicates that n_c is an ordinary member and $F_c = 2$ indicates that n_c is an associate member.

Each community member n_j has both a static storage buffer and a dynamic playback buffer. n_j collaboratively stores video chunks in the static buffer, trying to reduce the communication with other communities or server and ensure high lookup success rate for chunk search. The replacement strategy of content in the static buffer of the members is based on the distribution of resources in the community, which will be detailed next. n_j maintains the state of other members by periodical exchanging state information with them. The dynamic buffer is used for the current video playback purposes only. Next the major operations involving community will be detailed.

A. Member Join A Community

When a vehicle V_i joins the system, it broadcasts a request message containing the required *chunk ID* in the coverage area of the local AP. The nodes which have the requested chunk or find members storing the requested chunk in their maintained member list return a response message containing their member list and a movement trajectory to V_i. These response nodes may belong to different communities. Upon receiving these messages, V_i calculates the mobility similarity with the response nodes and then joins the community C_a corresponding to the response node with the highest mobility similarity with V_i, also higher than TH_m. If joining the community C_a, V_i becomes an ordinary member. Otherwise, if the mobility similarity between V_i and the response nodes are less than TH_m, V_i joins the community whose number of node members in the response is the largest in comparison with other communities. As V_i and the response nodes are located in the same AP, the selected community can provide more resources to V_i. In this situation V_i is an associate member in C_a. After V_i updates and stores the member list, it sends a request message to the members carrying the desired resource. If V_i receives the chunk from the static buffer of a supplier, it needs to re-search for supplier node with the next chunk in playback order before it consumes the whole current chunk. It is preferred that V_i selects the members which have the playback point synchronized with V_i as suppliers, and fetches the content from their dynamic buffer as the playback is performed, reducing therefore the number of request messages.

Assume V_i is a new member in C_a and needs to store a chunk in its static buffer. Let n_k be the broker in C_a. n_k will not distribute the stored chunk to all the members due to the high message overhead. PMCV employs a *popularity-based balanced chunk distribution algorithm* to achieve the collaborative storage of video chunks for in-community members. The broker n_k periodically collects the information and access frequency of the chunks stored in static buffers at all members and calculates the popularity of each chunk c_i according to eq. (14).

$$P_{c_i} = \frac{H_{c_i} + f_{c_i}}{H_{total} + f_{total}} \tag{14}$$

where f_{c_i} is the access frequency of c_i in the current period and H_{c_i} is the total access frequency of c_i. H_{total} is the total access frequency of all n chunks in C_a during the whole period and f_{total} is the total access frequency of the n chunks in C_a during the current period. $P_{c_i} \times N_{C_a}$ is the number of members which need to store c_i where N_{C_a} is the total number of members in C_a, which can achieve load balance. Moreover, we define a storage priority for each member based on the Hamming distance, defined in eq. (6): the lower the distance between the traces of members and playback pattern corresponding to C_a is, the higher the popularity of the chunks

stored is. This is because the members with high Hamming distance have an unstable state so that they may very likely move from C_a to other communities. The important resource carried by the stable members can ensure the resource security within the community.

In order to reduce n_k's load, we propose a "reporter"-based collection and dissemination strategy. Each member n_j in C_a periodically exchanges T_j^P, mt_j and the information on access frequency of the chunks in its static buffer with n_j's supplier n_s, which provides the video data to n_j. In turn, n_s interacts with its own supplier and exchanges messages including information about n_s and its receiver nodes (e.g. n_j), etc. If n_s finds that its own supplier is the media server or a member in other communities, n_s is considered a "reporter" and sends the received information to n_k. After n_k receives the information from all the members, it updates the information about the items in the member list, calculates the popularity of chunks and disseminates the updated list and chunk popularity to the reporters. In turn, these reporters will distribute this information to their receivers. When all members receive the information, they update the local member list and regulate the content in their static buffer. The period time of collecting member information can be defined as $T_c = \lambda \times len, \lambda = 1, 2, ..., n$, where len is the chunk size. In order to reduce the replacement frequency of content in the static buffer, the replacement period of time can be set to $T_r = \theta \times T_c, \theta \in (0, n)$. After V_i joins C_a, it can receive the popularity from its supplier and download the corresponding chunk from other members into the static buffer. The associate members store the chunks with the lowest popularity only.

If V_i cannot obtain a response message from the current AP, it sends a request message to the server. After the server receives the message from V_i, it returns a response message containing the community subset whose items include the requested resource to V_i. V_i preferentially joins the community whose items have the highest mobility similarity, higher than TH_m. If these communities cannot meet the demand of V_i's mobility, V_i joins the community which has the highest number of chunks in the corresponding playback pattern among all the communities. In this way V_i obtains a more stable source of resources and reduces the supplier churn. If no community has V_i's requested resource, the server directly transmits the video data to V_i. V_i makes use of the member list to search for a desired resource when it changes the current playback point location. When V_i cannot obtain the available supplier from the members, V_i sends a request message to the broker member in the current community. The broker member searches the supplier candidates with the required resource in the local communities list and returns these candidates to V_i. V_i exchanges location information with the candidates and selects the candidate which has the closest geographical distance with V_i as the supplier.

B. Member Movement Between Communities

With increasing number of items in the playback trace tr_{V_i} of V_i, V_i calculates the Hamming distance between tr_{V_i} and all items in $S_{pattern}$. V_i joins the community which has the minimum Hamming distance with tr_{V_i}. As we know, the members in the same community have very similar playback behavior. However, the movements between communities ensure stability of the logical link between V_i and its suppliers and availability of reliable resources. When V_i needs to move from the current community to another community, it sends a request message containing the corresponding playback pattern to the broker member n_k. n_k returns a node list whose items belong to the requested playback pattern and have the resource corresponding to current playback point of V_i. n_k removes V_i from the local member list. After V_i receives the response message, by exchanging the message containing movement trajectory with the items in this list, V_i calculates the mobility similarity with these items. If the requirement of mobility similarity still cannot be met, V_i selects the node with the closer geographical distance with V_i as its supplier and joins its community.

Additionally, when the mobility similarity between associate members in the same community is greater than TH_m, these associate members leave the current community and form a new community. Moreover, in the process of resource search, if an associate member finds a supplier whose mobility similarity with the associate member meets the requirement in terms of movement trajectory, it joins the community corresponding to the supplier.

C. Broker Node Replacement

When a broker member n_k in the community C_a finishes the playback or moves to another community, it selects another member as the new broker member. In order to avoid frequent replacements of the broker member, n_k needs to estimate the broker suitability for all members (not including the associate members) in terms of their online time and Hamming distance between trace and pattern. The estimate value of the normalized online time of any member n_j can be obtained according to eq. (15).

$$t_j = \frac{|tr_j|}{p_{C_a}} \tag{15}$$

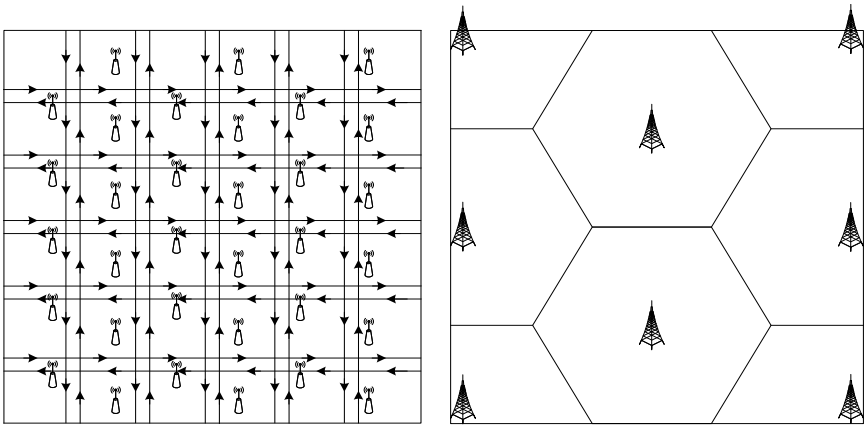
where $|tr_j|$ returns the number of chunks in n_j's playback trace tr_j and p_{C_a} is the playback pattern corresponding to C_a. t_j is denoted as the *online time ratio* and estimates the time before n_j leaves the system. The lower the value of t_j is, the longer the time of n_j serving other members is. The normalized Hamming distance between tr_j and s_{C_a} is defined as in eq. (16):

$$\overline{D}_j = \frac{D_{\text{Hamming}}(tr_j, s_{C_a})}{n} \tag{16}$$

$\overline{D}_j$ is the similarity level between playback trace of n_j and playback pattern of C_a and alongside with t_j are the-smaller-the-better parameters. We use t_j and $\overline{D}_j$ as estimate values of stability for broker member candidates according to eq. (17).

$$sw_j = \frac{1}{\alpha\overline{D}_j + (1-\alpha)\, t_j + 1}, \alpha \in (0, 1) \tag{17}$$

where α is a weight factor. n_k makes use of the above approach to obtain the estimate stability values of other



(a) (b)

Fig. 5. VANET and 4G simulation scenarios. (a) VANET architecture and mobility model of vehicular nodes. (b) 4G WiMAX network architecture.

members, namely $(sw_1, sw_2, \ldots, sw_k)$. The broker member selects as the new broker, the member whose estimate stability value is the highest in comparison with the other members.

VI. Performance Evaluation

A. Simulation Settings

We compare the performance of the proposed solution PMCV with that of QUVoD [4], both deployed in a VANET-based environment by making use of the Network Simulator (NS2).

(1) Testing Topology and Scenarios:

Table III lists some important NS2 simulation parameters of the VANET and 4G WiMAX networks, respectively. There are 1000 mobile nodes in VANET. The nodes are deployed in a square area of x = 2000 *m* by y = 2000 *m*. An urban topology, which consists of five horizontal and five vertical streets is considered, as illustrated in Fig. 5 (a). Every street has two lanes in each direction. The vehicular movement behavior follows the Manhattan mobility model [32]. When a vehicle reaches the specified destination, it is assigned a random residence time and restarts its movement with a new assigned speed and destination. Thirty-three APs equipped with IEEE 802.11p WAVE interfaces are distributed in VANET as illustrated. Fig. 5 (b) shows the 4G WiMAX architecture of QUVoD, which is composed of 8 hexagonal cells, each with a 578 m cell radius.

PMCV uses the IEEE 802.11p WAVE network interface only to support wireless communications between the vehicle nodes for the exchange of control and state and transmission of video data. QUVoD uses two wireless communication interfaces for each of the vehicle nodes: the IEEE 802.11p WAVE network interface and the 4G WiMAX interface. The vehicular nodes in QUVoD dynamically regulate the usage of network interfaces (switchover between G-path and V-path) in terms of communication quality of transmission path. For QUVoD, the value of AB_{thres} is set to 128 kb/s, which is equal to the streaming rate. AB_{value} is 24 kb/s, 48 kb/s and 96 kb/s, respectively. For example, if the value of V-Path's detection bandwidth is less than AB_{thres} - AB_{value} = 104 kb/s and G-Path's detection bandwidth is greater than 128 kb/s, the packets will transfer over the G-Path. Otherwise, the V-Path is used to deliver the video chunks. Video content is being exchanged by the vehicular nodes with an average video streaming rate of 128 kb/s. The simulations consider a 3600 *s*-long video that is divided into $n = 90$ chunks. Each chunk is 40 second long and about 640 KB in size. The simulation time is set to 2400 *s*. The movement trajectories of mobile nodes are generated from 0 *s* to 600 *s* and the nodes start to join the system starting from 600 *s* to the end. We created 20,200 synthetic user viewing log entries based on the interactive actions, measurements and statistics from [33] where 20,000 log entries are considered as historical playback trace library. 200 mobile nodes join the system in terms of a Poisson distribution before 1500 s and follow the remaining

TABLE III
Simulator Parameter Setting

Parameters		Values
VANET	Area	$2000 \times 2000(m^2)$
	Channel	Channel/Wireless Channel
	Network Interface	Phy/WirelessPhyExt
	MAC Interface	Mac/802_11Ext
	Bandwidth	27 Mbps
	Frequency	5.9 GHz
	Multiple Access	OFDM
	Transmission Power	33 dBm
	Wireless Transmission Range	250 *m*
	Interface Queue Type	Queue/DropTail/PriQueue
	Interface Queue Length	50 packets
	Antenna Type	Antenna/OmniAntenna
	Routing Protocol	DSR
	Peak Mobility Speed	30 *m/s*
	Transport-Layer Protocol	UDP
AP	AP Bandwidth	625 kb/s
	AP Transmission Range	250 m/s
	AP Number	33
4G WiMAX	Operating Frequency	2.5 GHz
	Channel Bandwidth	10 MHz
	Channel Capacity	40 Mbps
	Radio Propagation Model	MPropagation/MFreeSpace
	Frame Duration	5 *ms*
	Multiple Access	OFDM
	Duplexing Mode	TDD
	Interface Queue Length	50 packets
	Cell Layout (BS Number)	8 hexagonal cells
	Cell Radius (Transmission Range)	578 *m*
	Transport-Layer Protocol	UDP

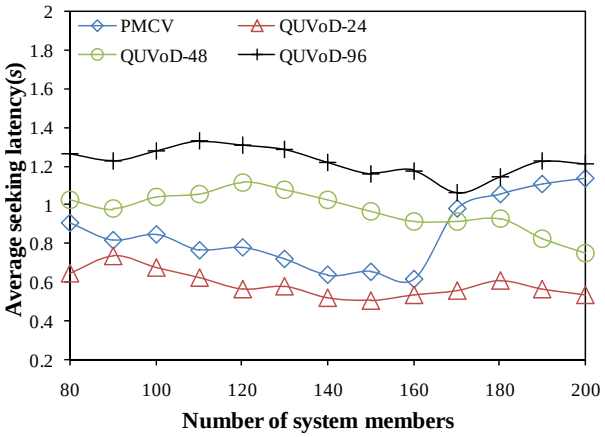


Fig. 6. Average seeking latency versus the number of system members

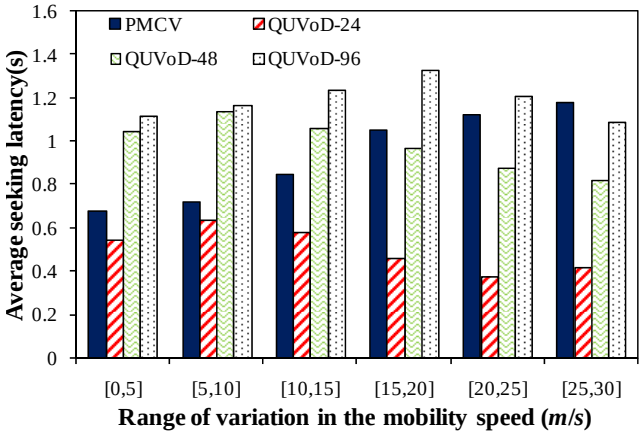


Fig. 7. Average seeking latency versus mobility speed of mobile nodes

200 viewing log entries to play video content. The mobile nodes which finish the playback of the whole video quit the system. The bandwidth at the media server side is set to 10 Mb/s.

(2) Parameter Settings:

The value of the threshold TH_m is set to 0.5 following repeated testing results. λ is the impact factor of the period of time for collecting member information. The larger the value of λ is, the longer the period of time is. If the load of members in the community is kept at low level, the member state cannot be updated in real time. In order to balance the load and timeliness, the value of λ is set to 60. θ determines the period of time for the replacement of content in the static buffer. The larger the value of θ is, the lower the frequency of replacement is. The period time for replacement is normally one or multiple video lengths. The variation of access frequency of a chunk in a community during short periods of time is very low. The frequent update of chunk access frequency wastes large amounts of storage and bandwidth resources. The period of time of exchanging information between server and broker members is set to $T_s = \mu \times T_c, \mu \in (0, n)$ where μ determines the frequency of interaction between broker members and server. In order to balance the real-time community state and load of broker members and server, μ is set to 1. m is the degree of fuzziness and is set to 2 [34]. In order to balance the online time ratio and membership level of community, α is set to 0.5.

PMCV and QUVoD are deployed in the same VANET. However, as the mobile nodes in QUVoD are equipped with two network interfaces: the IEEE 802.11p WAVE network interface and the 4G WiMAX interface, the simulation architecture of QUVoD includes an additional 4G WiMAX network for communication (G-path). The parameter settings of the VANET and 4G WiMAX network are described in Table III. Moreover, the mobile nodes in two solutions have the same behaviors when viewing video and join/leave the system.

B. Performance Evaluation

Average seek latency (ASL): The difference between the time when the node has received the video data and the time when the node has sent the request message is denoted as the *seek latency*. ASL is the average value of the seek latency readings. ASL mainly includes the latency of requested resource locating and the response latency of supplier. The smaller the value of ASL is, the higher the user QoE level is.

Fig. 6 plots the comparison results when PMCV and QUVoD-24, QUVoD-48 and QUVoD-96 are used in turn with increasing number of system members (mobile nodes in the system), when the mobility speed varied in the [0, 30] m/s range. QUVoD-24, QUVoD-48 and QUVoD-96 represent various QUVoD versions with different values of AB_{value}, respectively. The curve corresponding to QUVoD-24 has a decreasing trend with some fluctuations, but the QUVoD-24 results are better than those of QUVoD-48, QUVoD-96 and PMCV. PMCV's ASL curve has a decreasing trend with increasing the number of nodes and reaches a minimum value of 0.624 s for 160 system members, 50% and 80% lower than that of QUVoD-48 and QUVoD-96. Further increasing the number of nodes, PMCV's ASL increases to 1.146 s.

Fig. 7 illustrates ASL comparison results for different ranges of mobility speeds. As the figure illustrates, QUVoD-24 outperforms QUVoD-48, QUVoD-96 and PMCV. However the blue histograms corresponding to PMCV's results are better than both QUVoD-48 and QUVoD-96 for a wide range of speeds. For instance for speeds in the [10-15] m/s range, PMCV's ASL is 35% and 40% lower than ASL of QUVoD-48 and QUVoD-96, respectively. However, at very high mobile speeds, the ASL of PMCV also increases.

The mobile nodes in QUVoD make use of a path over the 4G network to search for video chunks, so the high efficiency and reliability of the 4G network and excellent search performance of the Chord structure ensures fast location of the requested chunks. As QUVoD does not consider the mobility of the nodes, the close geographical distance between the requesting sender and supplier cannot be ensured in case of mobility, so often a multi-hop delivery of video data results in high response latency. AB_{value} determines the switchover between the VANET path and the 4G path. For low AB_{value} , the data transmission path switches sooner from the VANET path to the 4G path. This reduces the response latency, but the increased usage of the 4G network path results in higher user costs, 4G network load and energy consumption. When the value of

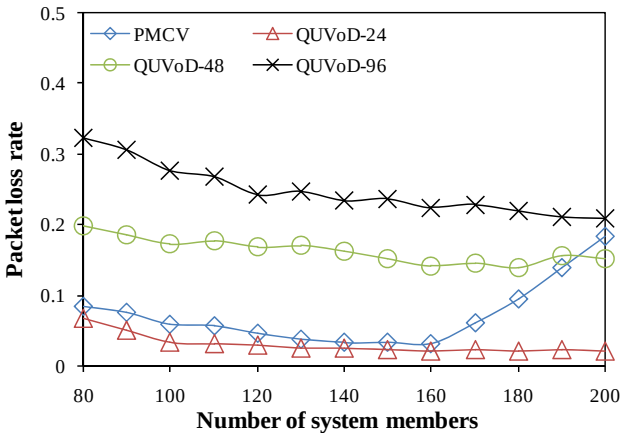


Fig. 8. Packet loss rate versus the number of system members

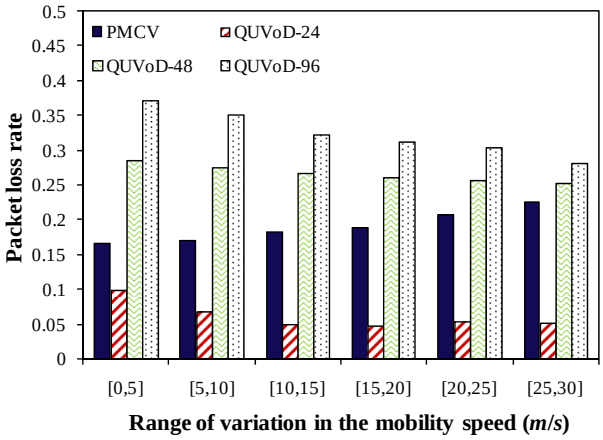


Fig. 9. Packet loss rate versus mobility speed of mobile nodes

AB_{value} is high, the VANET path is mostly used and there is a resulting increase in latency with a negative influence on user QoE. The same increase in the number of system members results in network congestion and QUVoD switches to the 4G path. This increases the performance, but affects user costs, 4G network load and node energy consumption. Further, after the mobile nodes in QUVoD have acquired and played a chunk, they need to search again for a new supplier with the requested resources and this leads to high message overhead and large start delays.

The nodes in PMCV make use of the extracted popular playback patterns to form multiple communities. The members in the same community use a member list to maintain the state and resources stored by other members. The increasing number of mobile nodes, results in increasing resources in the community and this reduces the cross-community resource search. PMCV ensures increased efficiency of resource search, and the logical one hop relationship between community members reduces the latency of resource location. Moreover, in PMCV, when the logical link between the supplier and receiver of video data is disconnected, the receiver seeks a new supplier in the same community, reducing the overhead. The synchronization of playback point between supplier and receiver also reduces the number of messages for resource search. The network congestion caused by the increase in the number of system members and the increasing mobility of nodes leads to some growth of the PMCV's ASL, but ASL values are still maintained at a low level.

Packet loss rate (PLR): The ratio between the number of packets lost in the process of video data transmission and the total number of packets of video data sent is referred to as PLR.

As Fig. 8 shows, the curves corresponding to QUVoD-24, QUVoD-48 and QUVoD-96 exhibit a decreasing trend with the increasing number of system nodes. The QUVoD-24 results are lower than those of PMCV, QUVoD-48 and QUVoD-96. PMCV PLR has very low values, close to those of QUVoD-24 and roughly 110% and 200% better than the values associated with QUVoD-48, and QUVoD-96. PLR increases when the system node number grows above 160, indicating increased network load.

Fig. 9 shows how PLR varies with the variation in mobility speed in VANETs. The red histograms corresponding to the QUVoD-24 results have both low values and slight decrease from [0, 5] to [25, 30] (the values are between 0.02 and 0.07), lower than those of PMCV, QUVoD-48 and QUVoD-96. The PMCV results, illustrated with blue bars, have values between 0.17 and 0.23, lower than the results of both QUVoD-48 and QUVoD-96.

The mobile nodes in QUVoD make use of the 4G network path to compensate for the decreased performance of the transmission of the VANET path. The lower the value of AB_{value} is, the higher the utilization frequency of the 4G path is. The increasing number of nodes and mobility speed do not have a negative influence on the QUVoD-24's PLR due to the high performance of the 4G network, but this comes at a price in terms of additional cost, cellular network load and energy consumption. PMCV outperforms both QUVoD-48 and QUVoD-96 which cannot easily switch to the 4G network to deliver the video data due to the high value of AB_{value}.

PMCV instead makes use of the mobile communities to enhance the efficiency of resource sharing and reduce the unsupervised search. This is as the members of each community have similar movement trajectory so that they maintain close geographical distance between them. This ensures high transmission performance - low delay and reduced PLR of PMCV. When the number of nodes is 160, the PLR of PMCV reaches the minimum value of 0.031. However, further increase in the number of system members results in more resource requests and many video transmissions lead to overall network traffic increase and certain degree of the network congestion due to limited VANET bandwidth. This affects PLR of PMCV as this uses a single network approach.

Throughput: The ratio between the total number of packets received in the overlay during a certain period of time and the length of this period of time is defined as the average throughput.

Fig. 10 shows the total system throughput variation of PMCV, QUVoD-24, QUVoD-48 and QUVoD-96 during the simulation time. The curves corresponding to the four solutions considered have similar shapes, experiencing a fast rise from the start at t = 600 s until simulation time 1500 s, then

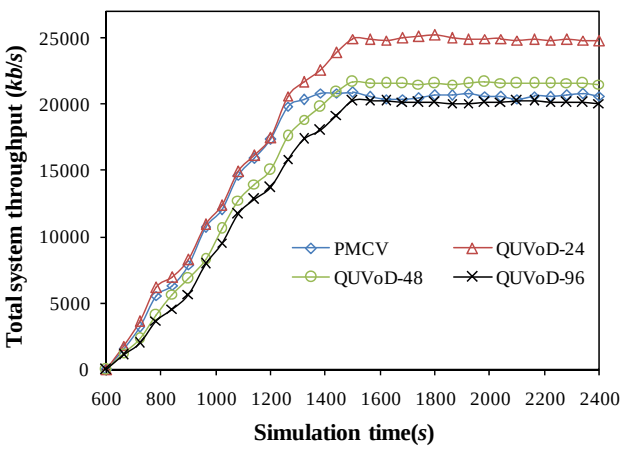


Fig. 10. Total system throughput versus simulation time

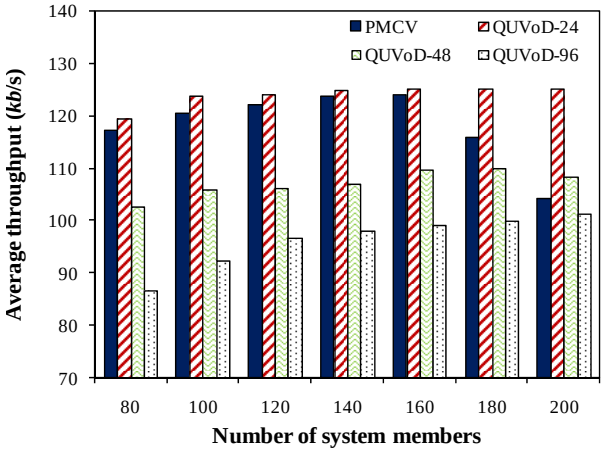


Fig. 11. Average throughput versus the number of system members

a stable period with slight fluctuations to the end. QUVoD-24 throughput reaches a peak value higher than those of PMCV, QUVoD-48 and QUVoD-96 due to its 4G network use, but PMCV outperforms both QUVoD-48 and QUVoD-96.

We compute the average throughput per video streaming session (each system member streams video content from its supplier) to show the video quality experienced by users. Fig. 11 presents the average throughput variation per video streaming session with the increasing number of system members. QUVoD-24's red throughput bars vary between 119 kb/s and 126 kb/s and are the highest among the four solutions, but PCMV's blue bars have very high levels only roughly 3% adrift for regular system load, too. The PMCV values are between 30% and 120% better than those of QUVoD-48 and QUVoD-96, respectively, clearly indicating PMCV's performance-related benefits. In highly loaded conditions, PMCV's throughput decreases, but still outperforms QUVoD-48 and QUVoD-96.

In these tests, the mobile nodes join the system following a Poisson distribution after 600 s. Transmission of numerous video data packets leads to data traffic increase with increasing number of system members. The use of the 4G network path by QUVoD ensures high transmission performance, including throughput, but also high monetary costs and higher energy consumption. In PMCV, the mobile communities group the mobile nodes with similar behavior of playback and movement to enhance the video content sharing efficiency. With increasing number of system members, the mobile communities are continually generated to enable fast search and video data transmission within close geographical distance. Therefore, the throughput of PMCV has relatively high increment and keeps the rise trend corresponding to decreasing packet loss rate with increasing number of system members from 80 to 160. However, due to the limited VANET bandwidth, when the number of nodes continues to increase from 180 to 200, the increase in the video data transmission results in the network congestion so that the throughput of PMCV decreases. This is as PMCV uses VANET only to transmit the video data. However the values of PMCV throughput are better than those of QUVoD-48 and QUVoD-96.

Average video quality: In order to assess the user perceived quality, we use Peak Signal-to-Noise Ratio (PSNR) metric, which measures video quality in decibels (dB), and estimate it according to eq. (18) [35].

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX\_Bitrate}{\sqrt{\left(EXP\_Thr - CRT\_Thr\right)^2}}\right) \quad (18)$$

where $MAX\_Bitrate$ is the average bitrate of the video stream as resulted from the encoding process, EXP_Thr is the average throughput expected from the delivery of the video stream over the network and CRT_Thr denotes the actual throughput measured during delivery. $MAX\_Bitrate$ and EXP_Thr are both 128 kb/s in terms of simulation settings.

We investigate the average PSNR of the video stream received by each node with increasing number of system members. As Fig. 12 shows, the QUVoD-24's results, illustrated by hashed red histograms, are higher than those of the other three solutions. However the PMCV's PSNR results, which correspond to the dark blue bars, have average values at good quality level for remotely delivered wireless mobile video [36] of roughly 16.4 dB for low number of system users (80) and 23.2 dB when the number of system members reaches 160. PMCV's PSNR values are 120% and 300% higher than those of QUVoD-48 and QUVoD-96, respectively. With increasing number of nodes, the system becomes increasingly congested and PMCV's PSNR decreases. However, the PMCV performance is still better than those QUVoD-48 and QUVoD-96.

QUVoD does not consider the mobility of nodes, so the variation of geographical distance between communication parties leads to a decrease in the performance of data transmission. The increasing number of requests for video content and the mobility of nodes negatively influences the performance of data transmission. However QUVoD employs 4G network communications (e.g. roughly 86% of QUVoD-24's nodes use the 4G path) to compensate for the reduced performance. The high dependency level on the 4G network increases the load on the 4G network and adds high monetary costs and increases

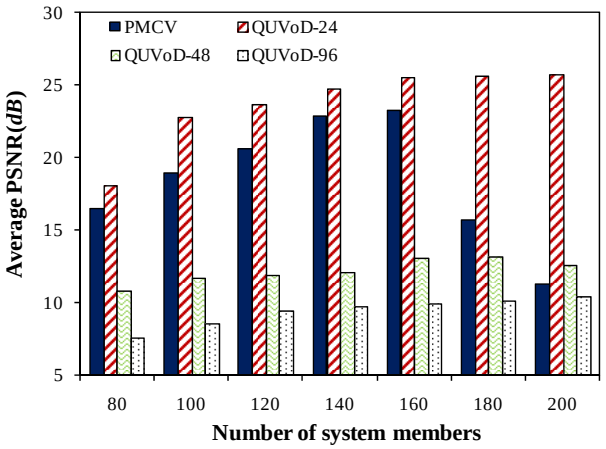


Fig. 12. Average PSNR versus the number of system members

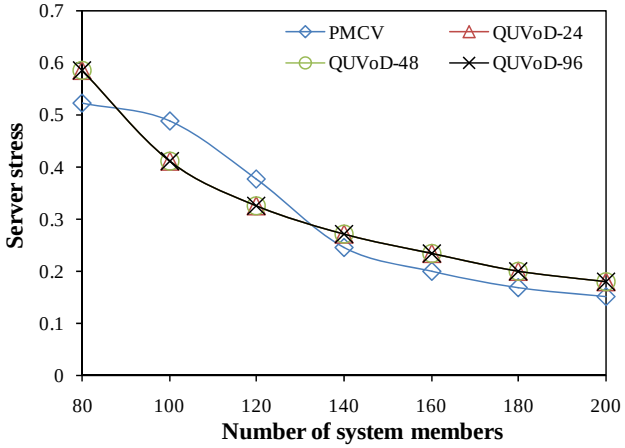


Fig. 13. Server stress versus the number of system members

the energy consumption in order to ensure high QoE.

In PMCV, the members in the mobile communities have close geographical distance and similar playback behavior. With the generation of mobile communities, the increasing efficiency of data transmission makes sure that the system members experience high video quality. However, the increase in the number of system members results in higher network traffic. The increasing number of requests for video resources results in higher PLR and larger delays due to limited network bandwidth. Therefore, when increasing the number of users from 180 to 200, PMCV's PSNR decreases. If the network bandwidth is sufficient, the PSNR of PMCV is good based on the very good performance of FAC and MSMM.

Media Server Stress: The ratio between the current number of streams serving mobile nodes at the media server side and the maximum number of streams supported by the server is defined as the server stress. The server stress is an important metric for the scalability of the system (low server stress indicates that the system has high scalability).

Fig. 13 illustrates the average server stress with increasing number of system members. Because QUVoD-24, QUVoD-48 and QUVoD-96 employ the same system architecture and the playback behaviors of mobile nodes are uniform, the curves of their media server stress are superimposed. We can see in the figure how the media server stress in QUVoD-24, QUVoD-48 and QUVoD-96 increases with the increase in the number of system nodes. However, the PMCV's server stress curve experiences a fast decrease with the generation of communities. It can be clearly seen how PMCV's server stress values are roughly 10% better than those of QUVoD with increasing number of system members from 130 to 200, clearly indicating its superior scalability.

The mobile nodes in QUVoD contribute with their static buffers to caching all video resources in terms of resource storage strategy and form a Chord structure where the nodes with similar resources are clustered into a group. Therefore, the server stress in QUVoD keeps at the high level before the number of nodes reaches 90 and fast decreases with increasing number of nodes. In PMCV, the members in the community can cache and regulate the chunks in the static buffer in terms of the interest level of members and chunk popularity (resource requirement of nodes). PMCV not only optimizes the distribution of resources, but also ensures the presence of video resources in the community, reducing significantly the number of streams served by the media server.

Overlay maintenance overhead: The average bandwidth used per second for exchanging overlay maintenance messages is considered as the overlay maintenance overhead.

Fig. 14 shows the overlay maintenance overhead with increasing number of system members. QUVoD-24, QUVoD-48 and QUVoD-96 have the same results for overlay maintenance overhead due to the same system architecture and uniform playback behaviors of the mobile nodes. The curve corresponding to these solutions maintains a fast increasing trend with increasing number of system members. The curve associated with the PMCV results shows a fast rise before the number of system members reaches at 120. With the increase in the number of system members from 140 to 200, the PMCV results are maintained at relatively low level, roughly 12% lower than those of QUVoD. Fig. 14 clearly shows how PMCV outperforms QUVoD in terms of overhead.

The nodes in PMCV are grouped into multiple communities in terms of playback behavior and mobility. The generation of mobile communities needs to perform message exchange between community members. PMCV's maintenance overhead fast increases before the number of system members reaches 120. With the slow increase in the number of mobile communities, the number of messages used to construct the mobile communities decreases. The members in these communities only periodically maintain the contact with other members in the same community. These members have similar playback behavior, so the mobile community structure is stable, reducing the number of messages used to maintain the mobile community. Therefore, PMCV's maintenance overhead increases slower than that of QUVoD. QUVoD needs to regularly maintain the Chord structure and the lists of peer pointers and therefore cannot reduce its maintenance overhead with increasing number of mobile nodes. Moreover, in QUVoD, the exchange of messages to balance the load between group members and the seeking messages to locate new supplier

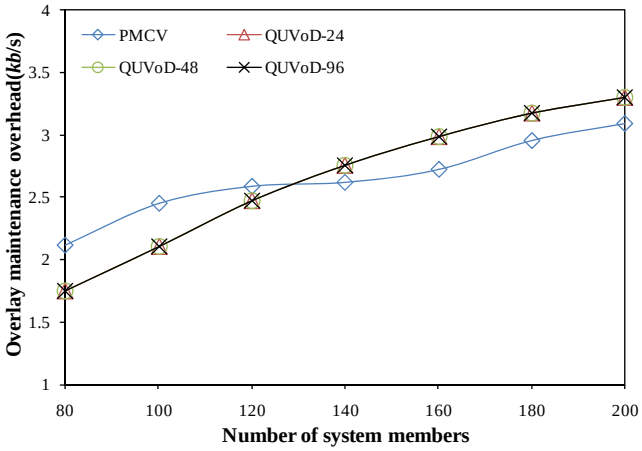


Fig. 14. Overlay maintenance overhead against the number of system members

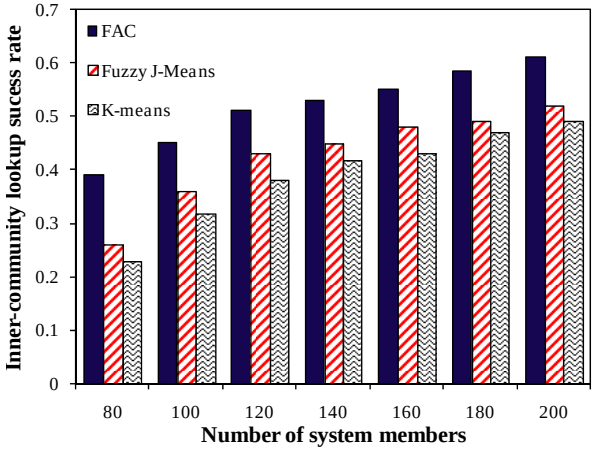


Fig. 15. In-group lookup success rate against the number of system members

when the sequential segments in one node can not supply the requested video chunks further increase the control overhead. Consequently, the overlay maintenance overhead of PMCV is lower than any QUVoD versions, with increasing number of system members.

Inner-community lookup success rate (ILSR): We compare the clustering accuracy of FAC in PMCV with two classical algorithms: Fuzzy J-means [34] and K-means [27] in term of inner-community lookup success rate. The ratio between the number of successful inner-community chunk lookups and the total number of inner-community chunk lookups is considered as the inner-community lookup success rate. We employ Fuzzy J-means and K-means to generate the playback patterns by mining the historical playback traces. In PMCV, the members preferentially search the inner-community resources. The higher the clustering accuracy is, the more similar the playback behavior between members is, namely the higher the similarity between sample (playback pattern) and members in the corresponding community is.

Fig. 15 shows the ILSR of PMCV, Fuzzy J-means and K-means with increasing the number of mobile nodes. The bars corresponding to the K-means results maintain a relatively slow increasing trend. The Fuzzy J-means bars have a fast rise from 80 to 120 and a stable increase from 140 to end. The bars corresponding to the PMCV results also show a fast increase from 80 to 120 and then a slight rise, with values roughly 15% and 25% higher than those of Fuzzy J-means and K-means, respectively.

The clustering accuracy determines the difference level of playback behavior between members. The members which have similar playback behavior request similar chunks. The members in the community collaboratively store the chunks in terms of chunk popularity and membership level to optimize the resource distribution, ensuring the resource security of the community and reducing the cross-community resource search. The higher the clustering accuracy for historical playback trace is, the lower the number of cross-community resource searches, namely the larger the inner-community lookup success rate is. The initial sample selection is very important for the clustering accuracy of K-means and Fuzzy J-means. However, K-means and Fuzzy J-means employ a random selection strategy to generate the initial sample, reducing the clustering accuracy. Moreover, the Fuzzy J-means makes use of a fuzzy membership parameter to iteratively regulate the cluster center, so Fuzzy J-means can obtain relatively accurate clustering results and its ILSR are greater than those of K-means. For PMCV, FAC makes use of the ant-colony clustering algorithm to determine the initial heap center (sample) in terms of viewing interest variation. FAC employs the fuzzy C-means to refine the heap centers to further enhance the clustering accuracy. Therefore, FAC obtains higher ILSR than both K-means and Fuzzy J-means.

VII. Conclusion and Future Work

This paper proposes a novel **Performance-Aware Mobile Community-based VoD streaming solution over vehicular ad hoc networks (PMCV)**. PMCV bases its efficiency on a newly designed mobile community detection scheme - which employs a novel *Fuzzy Ant Clustering algorithm* (FAC) to extract and refine playback patterns (samples) from historical playback traces for supporting high accuracy clustering of mobile nodes and makes use of a *mobility similarity measurement model* to estimate the similarity of mobility of mobile nodes for reducing the transmission cost of video data. Furthermore, PMCV design includes a novel *mobile community management mechanism* which performs the following steps: 1) assigns roles and tasks to members; 2) enables members joining the community; 3) supports members move between communities; 4) enables collaborative store and search for resources; 5) supports replacement of broker members. PMCV's performance was assessed in comparison with that of three versions of a state of the art alternative solution QUVoD via simulations. The results show that PMCV is an effective and low cost single network-based VoD streaming solution, which achieves low average seeking latency, low packet loss rate, high throughput, high video quality, low server stress, low overlay maintenance overhead and high average lookup success rate in pure vehicular ad hoc network environments. PMCV's system performance is very close to that of other state of the art solutions which avail from extensive use of

4G networks, while improving system scalability and reducing the overhead. Our future work will consider integrating novel intelligent strategies into PMCV in order to enable its deployment to both single and multi-network environments, while considering different network usage factors.

REFERENCES

[1] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato and Y. Nemoto, "A Stable Routing Protocol to Support ITS Services in VANET Networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337-3347, Nov. 2007.

[2] K. A. Hafeez, L. Zhao, J. W. Mark, X. Shen and Z. Niu, "Distributed Multichannel and Mobility-Aware Cluster-Based MAC Protocol for Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3886-3902, Oct. 2013.

[3] K. Pandit, D. Ghosal, H. M. Zhang and C.-N. Chuah, "Adaptive Traffic Signal Control With Vehicular Ad hoc Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459-1471, May 2013.

[4] C. Xu, F. Zhao, J. Guan, H. Zhang and G. Muntean, "QoE-driven User-centric VoD Services in Urban Multi-homed P2P-based Vehicular Networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273-2289, June 2013.

[5] 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.

[6] Y. Hsieh and K. Wang, "Dynamic overlay multicast for live multimedia streaming in urban VANETs," *Computer Networks*, vol. 56, no. 16, pp. 3609-3628, Nov. 2012.

[7] N. N. Qadri, M. Fleury, M. Altaf and M. Ghanbari, "Multi-source video streaming in a wireless vehicular ad hoc network," *IET Communications*, vol. 4, no. 11, pp. 1300-1311, July 2010.

[8] Z. Yang, M. Li and W. Lou, "CodePlay: Live multimedia streaming in VANETs using symbol-level network coding," in *Proc. IEEE ICNP*, vol, 29. no. 8, pp. 1698-1710, Oct. 2011.

[9] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. V. Kumar, D. D. Stancil and F. Bai, "Performance of the 802.11p Physical Layer in Vehicle-to-Vehicle Environments," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 3-14, Jan. 2012.

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

[12] D. Wang and C. K. Yeo, "Exploring Locality of Reference in P2P VoD Systems," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1309-1323, Aug. 2012.

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

[14] Y. Chen, B. Zhang, Y. Liu and W. Zhu, "Measurement and Modeling of Video Watching Time in a Large-Scale Internet Video-on-Demand System," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2087-2098, Dec. 2013.

[15] 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.

[16] S. Jia, C. Xu, J. Guan, H. Zhang and G.-M. Muntean, "A Novel Cooperative Content Fetching-Based Strategy to Increase the Quality of Video Delivery to Mobile Users in Wireless Networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 370-384, June 2014.

[17] H. M. N. D. Bandara and A. P. Jayasumana, "Community-Based Caching for Enhanced Lookup Performance in P2P Systems," *IEEE Transactions on Parallel and Distributed systems*, vol. 24, no. 9, pp. 1698-1710, Sep. 2013.

[18] H. Wang, F. Wang, J. Liu, C. Lin, K. Xu and C. Wang, "Accelerating Peer-to-Peer File Sharing with Social Relations," *IEEE Journal on Selected Areas in Communications/Supplement*, vol. 31, no. 9, pp. 66-74, Sep. 2013.

[19] A. Iamnitchi, M. Ripeanu, E. S. Neto and I. Foster, "The Small World of File Sharing," *IEEE Transactions on Parallel and Distributed systems*, vol. 22, no. 7, pp. 1120-1134, July 2011.

[20] H. Shen, Z. Li, Y. Lin and J. Li, "SocialTube: P2P-assisted Video Sharing in Online Social Networks," *IEEE Transactions on Parallel and Distributed systems*, vol. pp, no. 99, May 2013.

[21] C. Xu, S. Jia, L. Zhong, H. Zhang and G.-M. Muntean, "Ant-Inspired Mini-Community-Based Solution for Video-On-Demand Services in Wireless Mobile Networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322-335, June 2014.

[22] 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.

[23] C. P. Hong, E. H. Lee, C. C. Weems and S.-D. Kim, "A Profile-Based Multimedia Sharing Scheme With Virtual Community, Based on Personal Space in a Ubiquitous Computing Environment," *IEEE Transactions on Multimedia*, vol. 11. no. 9, pp. 1353-1361, Nov. 2009.

[24] 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, Sep. 2011.

[25] C. Doulkeridis, A. Vlachou, K. Norvag, Y. Kotidis and M. Vazirgiannis, "Efficient search based on content similarity over self-organizing P2P networks," *Peer-to-Peer Networking and Applications*, vol. 3, no. 1, pp. 67-79, Mar. 2010.

[26] S. Datta, C. R. Giannella and H. Kargupta, "Approximate Distributed K-Means Clustering over a Peer-to-Peer Network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1372-1388, Oct. 2009.

[27] K. M. Hammouda and M. S. Kamel, "Hierarchically Distributed Peer-to-Peer Document Clustering and Cluster Summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, May 2009.

[28] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75-174, Feb. 2010.

[29] R. Smarandache and P. O. Vontobel, "Quasi-Cyclic LDPC Codes: Influence of Proto- and Tanner-Graph Structure on Minimum Hamming Distance Upper Bounds," *IEEE Transactions on information theory*, vol. 58, no. 2, pp. 585-607, Feb. 2012.

[30] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," *Plenum Press*, New York, 1981.

[31] A. Papoulis, "Probability, Random Variables, and Stochastic Process," 1984.

[32] F. Bai, N. Sadagopan and A. Helmy, "The important framework for analyzing the impact of mobility on performance of routing protocols for ad hoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 383-403, Nov. 2003.

[33] 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*, Urbana-Champaign, IL, pp. 1-6, April 2007.

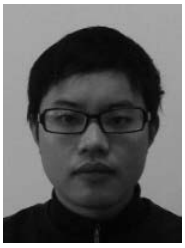
[34] N. Belacel, P. Hansen and N. Mladenovic, "Fuzzy J-Means: a new heuristic for fuzzy clustering," *Pattern Recognition*, vol. 35, no. 10, pp. 2193-2200, Oct. 2002.

[35] S.-B. Lee, G.-M. Muntean and A. F. Smeaton, "Performance-Aware Replication of Distributed Pre-Recorded IPTV Content," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516-526, Jun. 2009.

[36] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld and P. Tran-Gia, "Towards qoe management for scalable video streaming," *Proc. of the 21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE*, Miyazaki, Japan, March 2010.

Changqiao Xu received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences (ISCAS) in January 2009. He was an Assistant Research Fellow in ISCAS from 2002 to 2007, where he was a Research and Development Project Manager in the area of communication networks. During 2007 - 2009, he worked as a Researcher with the Software Research Institute at Athlone Institute of Technology, Athlone, Ireland. He joined Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in December 2009, and was an Assistant Professor from 2009 to 2011. Currently, he is 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 over 100 technical papers in prestigious international journals and conferences including IEEE transactions on mobile computing, IEEE transactions on vehicular technology, IEEE transactions on broadcasting, and Proceedings of ACM Multimedia. His research interests include wireless networking, multimedia communications, and next generation Internet technology. He serves as a Co-Chair and Technical Program Committee (TPC) member for a number of international conferences and workshops.

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.



Mu Wang received the BS degree in applied mathematics from the Xidian University, China, 2012. He is currently working toward the MS degree in the Institute of Network Technology at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. His research interests include P2P networks, wireless communications, and multimedia sharing over wireless 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, mobile internet technology, internet of things and vehicular networks.

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, Dublin, 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 (DCU), Dublin, Ireland. He is a 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, 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 personalized e-learning. He has published over 180 papers in prestigious international journals and conferences, has authored three books and 15 book chapters and has edited six other books. He is an 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.