QoE-Driven User-Centric VoD Services in Urban Multihomed P2P-Based Vehicular Networks

Changqiao Xu, *Member, IEEE*, Futao Zhao, Jianfeng Guan, Hongke Zhang, and Gabriel-Miro Muntean, *Member, IEEE*

***Abstract*—Recently, many cities around the world have witnessed large-scale deployment of terrestrial broadcasting mobile television (TV) to vehicles. This service is similar to the cable or satellite TV already in the home, and user-centric interactive mobile Video-on-Demand (VoD) over urban vehicular networks is in fact expected. However, providing this new service with focus on user Quality of Experience (QoE) constitutes a significant challenge. This paper introduces a QoE-driven User-centric solution for VoD services in urban vehicular network environments (QUVoD). QUVoD relies on a multihomed hierarchical peer-to-peer (P2P) and vehicular ad-hoc network (VANET) architecture. Vehicles construct a low-layer VANET via Wireless Access in the Vehicular Environment interfaces; they also form an upper layer P2P Chord overlay on top of a cellular network via Fourth-Generation (4G) interfaces. A novel grouping-based storage strategy that uniformly distributes the video segments along the Chord overlay is proposed, reducing segment seeking traffic while also enabling load balancing. A novel segment seeking and multipath delivery scheme that achieves high lookup success rate and very good video data delivery efficiency is also introduced, which achieves high lookup success rate and very good video data delivery efficiency. Furthermore, a new speculation-based prefetching strategy is proposed, which analyses users' interactive viewing behavior and, by estimating video segment playback order, employs prefetching of the expected segments, smoothening the video playback. Simulation results show how QUVoD is a highly efficient user-centric mobile VoD solution in urban vehicular networks in comparison with existing state-of-the-art solutions.**

***Index Terms*—Peer-to-peer (P2P), Quality of Experience (QoE), user-centric Video-on-Demand (VoD), vehicular network.**

Manuscript received January 18, 2012; revised October 3, 2012, November 6, 2012, and November 12, 2012; accepted November 12, 2012. Date of publication November 21, 2012; date of current version June 12, 2013. This work was supported in part by the National High-Tech Research and Development Program of China (863) under Grant 2011AA010701, by the National Basic Research Program of China (973 Program) under Grant 2013CB329100 and Grant 2013CB329102, by the National Natural Science Foundation of China (NSFC) under Grant 61001122, Grant 61003283, and Grant 61232017, by the Beijing Natural Science Foundation of China under Grant 4102064, by the Jiangsu Natural Science Foundation of China under Grant BK2011171, by the Jiangxi Natural Science Foundation of China under Grant 20122BAB201042, by the Fundamental Research Funds for the Central Universities under Grant 2012RC0603 and Grant 2011RC0507, and by Lero under Grant 10/CE/I1855. The review of this paper was coordinated by Dr. L. A. Grieco.

C. Xu is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Wuxi 214135, China (e-mail: cqxu@bupt.edu.cn).

F. Zhao and J. Guan are with the Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhaofutao@bupt.edu.cn; jfguan@bupt.edu.cn).

H. Zhang is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100044, China (e-mail: hkzhang@bupt.edu.cn).

G.-M. Muntean is with the Performance Engineering Laboratory, School of Electronic Engineering, Network Innovations Centre, Rince Institute, Dublin City University, Dublin 9, Ireland (e-mail: munteang@eeng.dcu.ie).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2012.2228682

I. Introduction

Lately, many cities around the world have witnessed large-scale deployment of terrestrial broadcasting mobile television (TV) services. For example, following the Beijing Olympics, almost all taxis (out of the over 700 thousand vehicles in the Chinese capital city) are equipped with onboard equipment that supports terrestrial TV broadcasting signal retrieval and multimedia playback. In terms of quality, this is like the passive cable or satellite TV programs that people watch at home. Additionally, we are witnessing extensive developments in wireless access technologies, including WiFi, Long-Term Evolution (LTE), LTE-A, WiMAX, etc. [1], and particularly in vehicular wireless technologies such as Wireless Access in the Vehicular Environment (WAVE) (IEEE 802.11p), enabling data delivery via vehicle-to-vehicle (V2V) [2], vehicle-to-infrastructure, and vehicle-to-road site unit (RSU) communications [3]. This paves the way toward vehicular Video-on-Demand (VoD) as mobile VoD is expected to become an attractive service over urban vehicular networks.

Peer-to-peer (P2P) multimedia content sharing from multiple sources to multiple destinations has been shown to be robust and highly scalable based on distributed solutions in the wired Internet [4]–[12]. Important research efforts are put to extend P2P content delivery and its performance benefits to the wireless domain. Lately, P2P-based multimedia services over vehicular ad-hoc networks (VANETs) have become a very interesting research topic [13]–[16]. However, although drawing increasing attention from the research community, studies on P2P media delivery over VANETs have been limited to simple sequential remote playback scenarios, which are similar to the broadcast mobile TV, where the passive user viewing pattern simplifies system design. However, user-centric mobile VoD services supporting interactivity can offer edited video clips to passengers on demand during their trips, extending the existing services offered by the likes of YouTube [4], for example, from the office or home to the road. This represents one of the most innovative trends and value-added services in urban vehicular networks.

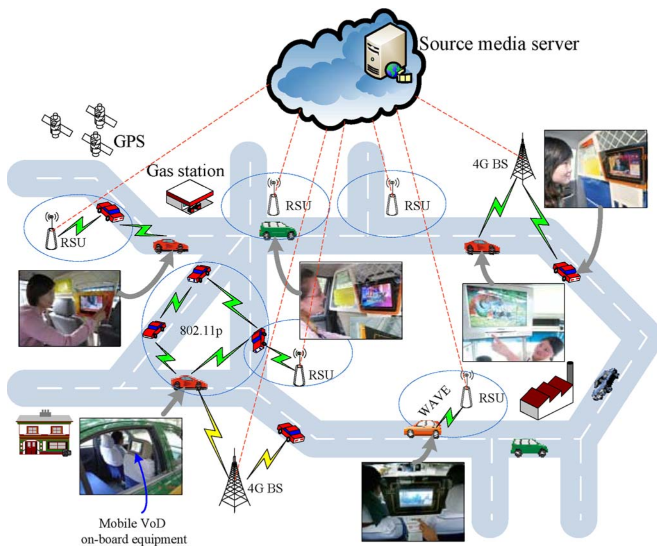


Fig. 1. QUVoD system for vehicular networks.

In this context, providing support for high Quality of Experience (QoE) for user-centric mobile VoD service in VANETs is a significant challenge. As we know, end-to-end video transmission quality plays an important role in maintaining high levels of user QoE in P2P-based VANETs. Most existing solutions disseminate video data via the underlying VANET in multihops [13]–[16]. However, VANETs easily become disconnected due to node mobility, leading to unreliable transmissions and negatively affecting user QoE levels. The latest technological advances enable vehicular devices to be equipped with multiple types of wireless interfaces to support access to different networks, forming a multihomed heterogeneous network environment. During communications, the source nodes can select and simultaneously use for data delivery one or several paths to the destination. The selection is performed dynamically following changes in the availability and/or characteristics of the access networks behind the nodes' interfaces. This significantly increases fault tolerance and supports provision of services with increased user QoE levels.

On the other hand, supporting user interactive viewing behavior in VoD services requires the proposal of a mechanism to enable smooth user video playback by fetching in advance the video segments in line to be played into the vehicle's buffer. The prefetching has become a default strategy and has been widely used in current Internet-based and wireless-based P2P multimedia delivery systems. However, most of them only prefetch the subsequent video segment for normal playback in strict sequential order. When video cassette recorder (VCR)-like operations occur, searching for the newly requested segment has to happen, and once found, it has to be downloaded, decoded, and then played back. This procedure is associated with long latency and seriously deteriorates the users' viewing QoE.

This paper introduces a novel efficient interactive Quality-oriented User-centric mobile VoD solution for vehicular networks (QUVoD), as illustrated in Fig. 1. QUVoD relies on a newly designed multihomed hierarchical P2P/VANET architecture and makes use of highly innovative algorithms for video segment storage, video resource search, multipath data transfer, and speculative segment prefetch to address the challenges of vehicle mobility and users' interactive viewing behavior. These novel algorithms will be described in detail in this paper alongside simulation-based performance evaluation. Testing results fully illustrate how QUVoD is an efficient user-centric mobile VoD solution in vehicular networks, which outperform existing state-of-the-art solutions.

II. Related Work

Recently, P2P content sharing for multimedia streaming services in vehicular networks has attracted increasing research interests from various scientists. Hsieh and Wang [13] proposed an effective dynamic overlay multicast strategy for live multimedia streaming in urban VANETs. Zhou *et al.* [14] developed a P2P media service scheme that jointly solves the content dissemination, cache update, and fairness problems for vehicular networks. Qadri *et al.* [15] presented a realistically modeled scheme for streaming video over VANETs by means of an overlay network with multiple sources. Using symbol-level network coding, Yang *et al.* [16] introduced a live multimedia streaming system CodePlay in VANETs. However, none of the foregoing works considered the vehicular interactive VoD service, in which users can jump to any watching point according to their interests. The interactive service significantly enhances user personal viewing experience but introduces important technical challenges, which need to be addressed.

Interactive P2P VoD over Internet is currently gaining momentum in the research community [5]–[12]. We previously proposed a balanced binary tree-based unstructured VoD distribution (BBTU) [5] and a distributed storage-assisted data-driven overlay network (SDNet) [6] to address VCR-related issues in wired nonmobile network environments. By adopting a mesh-based topology, Chang and Huang [7] introduced an interleaved video frame distribution scheme to support full VCR functionality by making use of gossip messages to search for resources. Zhou *et al.* [8] introduced a unifying request scheduling model and content replication strategy to minimize server load. Lee *et al.* [9] proposed a popularity-aware prefetching scheme to support interactive P2P Streaming. Wu and Lui [10] investigated how movie popularity can impact the server's workload and optimized the replication strategy of P2P-VoD systems.

Chord is a well-known P2P content distribution architecture. Nodes in Chord have embedded a "precursor-successor" relationship. By associating a key with each data item (i.e., video clip or video segment) and storing the key/data item pair at the node to which the key maps, efficient data storage and localization can be implemented [17]. For example, a Chord-based efficient interactive P2P VoD system named VMesh was introduced in [11] and [12]. All of the foregoing works address some important issues for P2P VoD, such as dissemination topology, video resources storage and search, support for VCR-like operations, video content replication, etc. However, so far, the research of interactive P2P VoD solutions lacks mobility support and, to the best of the authors' knowledge, does not focus at all on vehicular wireless networks.

P2P models can be generally classified in two types: unstructured and structured. Unstructured P2Ps are mainly based on

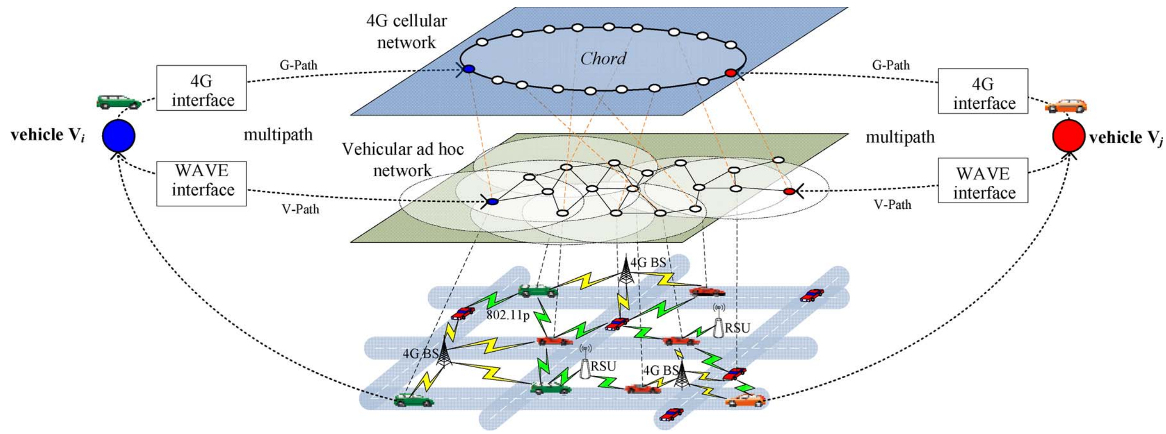


Fig. 2. QUVoD multihomed hierarchical architecture.

gossip and flood mechanisms, which produce large amounts of communication overhead. Most structured solutions use distributed hash tables (DHT), and as the typical lookup length of DHTs is $O(log(N))$, where N is the number of nodes in the system, the structured P2P solutions have high performance in terms of resource search. MeshChord [18] improves the basic Chord to efficiently utilize it in wireless mesh networks. Liu *et al.* [19], [20] proposed MChord to enhance the performance of Chord over VANETs. There are also other research works [21], [22] that improve the Chord algorithm trying to adapt it to the mobile environment. However, all the application-layer P2P Chord networks of the above solutions are built directly on top of the wireless ad-hoc network. It makes Chord seriously unstable as the ad-hoc network links are disconnection prone, and this happens frequently because of the nodes' mobility and the vehicles' different moving directions.

Recently, multihomed networks have attracted extensive academic research interests. Having multiple network interfaces, end hosts can establish multiple connections with other end hosts across different networks [23], a situation which protects the communications from the failure of any one of these networks. Multihomed mobile IP (M-MIP) [24] extended mobile IP to enable mobile hosts to use multiple care-of addresses simultaneously. Shim6 [25] inserts a sublayer into the network layer to support network layer multihoming and mobility. Okimoto *et al.* [26] described functional requirements and possible solutions for multihoming without the use of NAT in IPv6 for hosts. Multipath TCP (MPTCP) [27] modifies the Transmission Control Protocol and implements a multipath transport within a transport connection. In [28]–[30], we analyzed and proposed concurrent multipath video transfer solutions by making use of a multihoming stream control transmission protocol SCTP. It can be noted that multipath transport over multihomed networks increases the efficiency of the resource usage and thus increases the network capacity available to end hosts.

Based on the latest research progress in P2P multimedia content sharing in VANETs, Internet-based P2P VoD, and multihomed wireless networks, we introduce QUVoD, a novel mobile VoD solution in urban VANETs. To the best of the authors' knowledge, this is the first work to consider providing efficient P2P-based VoD services in urban multihomed vehicular networks.

III. Quality-Oriented User-Centric Mobile Video-on-Demand Solution for Vehicular Networks System Design

A. QUVoD Architecture

Fig. 2 shows the architecture of QUVoD. We assume that the vehicles in QUVoD are equipped with dual wireless interfaces: a fourth-generation (4G) cellular network interface (e.g., WiMAX, LTE-A) and a WAVE interface (supporting V2V communications). These vehicles construct a multihomed network, highly feasible to be built in reality. The cellular network is a standard for mobile telecommunications in wide-area mobile environments and supports real-time multimedia services [31]. A cellular network has relatively high stable connectivity merits. However, 4G is normally provided by internet service providers, which charge fees for data traffic. By making use of the WAVE interface in VANET, vehicles can forward data to their neighboring vehicles or RSUs, and data can transverse intermediate vehicles via ad-hoc communications with low network usage costs. However, VANETs easily become disconnected in situations with low vehicle density and high mobility, (i.e., high speed and/or different driving directions). Taking into account these factors, QUVoD constructs a multihomed hierarchical P2P/VANET architecture that employs both Chord and VANETs, as shown in Fig. 2. Vehicles construct low-layer VANETs via WAVE interfaces for V2V communications. The upper layer Chord overlay is formed on top of the cellular infrastructure via the 4G interfaces. Vehicle V_i can create two connections with vehicle V_j via the 4G path (G-Path) and VANET path (V-Path), respectively. The G-Path denotes packet transfer through the 4G infrastructure networks with high stability but also high cost. The V-Path indicates packet delivery over the VANET via V2V communications with increased instability but low cost. QUVoD transmits video data between vehicles V_i and V_j employing a multipath transfer mechanism. As 4G is stable, when vehicle V_i requests a video segment for its future playing, the request message is routed by the DHT for looking up the segment in the Chord network, which means in multihomed QUVoD, the G-Path is used for resource search. Both G-Path and V-Path are employed for video data multipath transmissions.

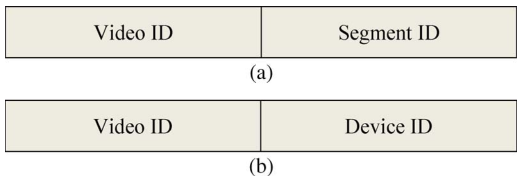


Fig. 3. Structure of resource ID and vehicle ID. (a) Structure of Resource ID. (b) Structure of Vehicle ID.

The multihomed hierarchical P2P/VANET architecture design of QUVoD is highly effective. In most current research, vehicles form an application-layer P2P network on top of the VANET directly, and P2P communication relies on the routing protocol of the underlying VANET [13]–[16]. With vehicle movement, the VANET topology changes quickly, which leads to P2P topology changing as well. Consequently, both resource lookup and data transfer fail when network disconnections between vehicles occur. This requires frequent redistribution and restoring of resources between different nodes as the nodes frequently join and leave the system due to their mobility. Unlike these solutions, QUVoD establishes Chord network communications over the 4G infrastructure network, which makes the P2P topology relatively stable. When a vehicle stores some video segments after it joins the system, it can carry the assigned segments all the time regardless of the driving requirements. For instances, in Beijing, each taxi normally has two drivers who work in shifts, which means the taxi can be 24 hours online. This enables for the target vehicle that stores the required video segments to always be located in Chord through the underlying 4G network instead of the failure prone VANET. In other cases, in urban environment, taxi density, traffic congestion, and light control in intersection provide video sharing opportunities via V2V VANET communications directly, which can make use of VANETs' own data distribution ability. The multipath transfer mechanism over G-Path and V-Path increases fault tolerance and improves data transfer capabilities. In this context, QUVoD overcomes the drawbacks of each of the 4G and VANET technologies, respectively, and makes full use of the advantages of a combined solution.

In addition to the multihomed structure that enables video segments sharing between vehicles, QUVoD has a media server. This server stores the original video resources and provides service when vehicles cannot fetch video segments from other vehicular nodes. The server also keeps all the video-related information, and users can search for the videos in which they are interested. Every video needs to be processed by the server before being distributed over the Chord network. The server assigns each video a unique Video ID according to the HASH algorithm [17]. In addition, for efficiency of management and delivery purposes, it divides each video into several equal size segments and assigns to each segment consecutive Segment IDs in the order of playback. Consequently, each video segment will have a unique Resource ID, which is composed of the Video ID and the Segment ID, as shown in Fig. 3(a).

Each vehicular node is assigned a unique Device ID by the server using the HASH algorithm when it joins the system. We define the Vehicle ID as being composed of the Video ID of the video playing and the Device ID of that vehicle as a unique identifier. Fig. 3(b) shows the composition of Vehicle ID. In QUVoD, all the vehicles playing the same video will have the same Video ID and make up a Chord subcircle according to the value of vehicular nodes' Device IDs. When a vehicle user is interested in a video, the vehicle node has to join the Chord subcircle corresponding to the video content sought. This is done by using the existing Chord algorithm [17]. Meanwhile, the vehicle node looks up for the initial video segments, downloads them via the multipath, and plays them when the requested data have arrived at its buffer. For video segment search and retrieval, the vehicular node employs the novel segment seeking and delivery scheme, which will be described in detail in the next section. After joining the system, the node then continues to request and download the next segments by making use of our newly proposed speculation-based prefetching strategy. When not all the bandwidth is in use, the vehicle will use its available bandwidth to download and store video segment(s) in its local storage using the QUVoD's novel distributed grouping video segments storage scheme. The vehicle leaving the system is treated based on the classic Chord leaving algorithm.

B. Distributed Grouping-Based Video Segments Storage

In the original Chord design, when the number of nodes is larger than that of video segments, there will be some nodes that do not store any resource as each video segment is stored in just one node. This creates load imbalance. Furthermore, if a node leaves or breaks down, although the video segment key is still stored in the system, the video segment resource is not accessible. Pitoura *et al.* [32] proposed a fair load distribution solution over DHTs using replication and multirotation hashing mechanism. A similar method is employed by VMesh [11], [12]. In VANET environments, due to the vehicular nodes' high dynamicity, it is difficult to guarantee a requester to properly obtain resources from only one supplier. Extending these works [32], [11], [12], QUVoD designs a distributed grouping-based storage scheme. It allows any video segment to be stored in multiple nodes and each node to store multiple video segments. This not only enhances the P2P network's stability but also balances the nodes' load.

As Fig. 4 shows, in QUVoD, the vehicular nodes in a Chord subcircle are divided into multiple groups, and the number of groups is equal to the number of video segments. The nodes in the same group will store several consecutive segments starting with the same segment. Denote the number of segments as $S_{\max}$, and the size of the Device ID field as L. Equation (1) gives the size of each group L_{group} as

$$L_{\text{group}} = 2^L / S_{\max} \tag{1}$$

where 2^L is the maximum value of the Device ID, and L_{group} is a constant value in a specific Chord subcircle network because the variables L and $S_{\max}$ are stationary. Assuming V_i and D_i are the latest vehicle node that has joined into the network and its ID, respectively, the ID to be assigned to the next node (i.e., V_{i+1}) is defined as

$$D_{i+1} = (D_i + L_{\text{group}} + 1) \bmod 2^L \tag{2}$$

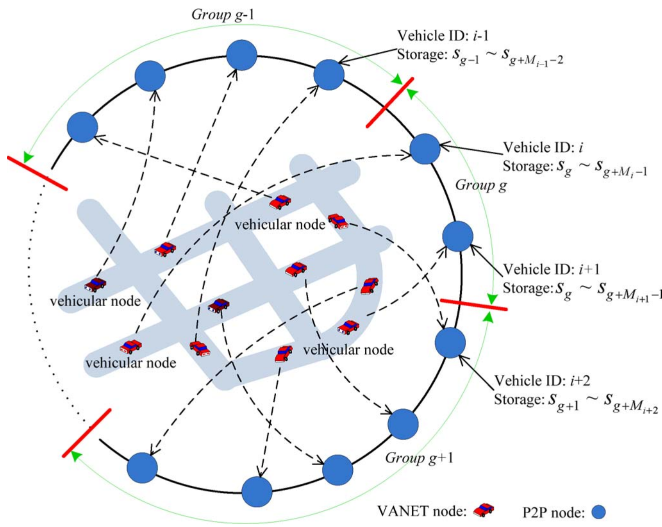


Fig. 4. Grouping-based video segments storage strategy.

which can have the nodes uniformly distributed over the Chord. We set F_g as the initial boundary Device ID for the gth group as in (3). For instance, the smallest Device ID of the nodes in the first group should be 0. g here also denotes the Segment ID of the starting segment that each node stores in the gth group

$$F_g = L_{\text{group}} \times (g - 1). \tag{3}$$

We set DS_g as the value set of all possible Device IDs of nodes in group g. DS_g can be defined by (4), where N is the set of natural numbers, and N^+ is the set of positive integers, i.e.,

$$\begin{aligned} DS_g &= \{id | id \in N, F_g \le id < F_{g+1}\} \\ &= \{id | id \in N, L_{\text{group}} \times (g-1) \le id < L_{\text{group}} \times g \\ &\quad 1 \le g \le S_{\max}, g \in N^+. \end{aligned} \tag{4}$$

As already mentioned, QUVoD divides each video in equal size segments, for example, $video \Leftrightarrow (s_1, s_2, \ldots, s_j, \ldots, s_n)$, where j is the Segment ID of segment s_j. Each node in QUVoD stores several successive video segments according to its available memory capacity. The nodes whose Device IDs are in the set DS_g will store the segment s_g and several of s_g's next segments. Assuming a V_i in group g has a storage memory space M_i, then it could store segment $s_g, s_{g+1}, \ldots, s_{g+M_i-1}$, which is indicated in

$$VS_i = \left\{s_j | j \in N^+, g \le j \le g + M_i - 1\right\} \tag{5}$$

where VS_i is the segment set composed of all the segments stored by V_i. Fig. 4 illustrates the distributed grouping-based video segments storage scheme.

When a node still has memory space available after having stored all the segments up to the last segment in the video sequence, it starts storing the first segments from the beginning of the sequence until the memory is used up. For example, a vehicle V_i in group $S_{\max}$ can store the segment $S_{\max}$ and then the segments 1, 2, 3 and so on. Assuming g is the group sequence V_i belongs to, s_j is a segment that V_i will store, and $device\_id_i$ is V_i's Device ID, Algorithm 1 illustrates the video segments' storage strategy after V_i joins the system.

Algorithm 1 Video segments storage for vehicle V_i

$g = device\_id_i / L_{\text{group}}$;
$j = g$; /* j denotes the segment ID*/
$idle\_memory = M_i$;
/*$idle\_memory$ denotes V_i's remaining memory size*/
while $(idle\_memory \ge \textbf{sizeof}(s_j))$;
  /*s_j denote a segment whose segment ID is j*/
  V_i search and download segment s_j using algorithm 2;
  $j++$;
  if $(j > S_{\max})$
    $j = 1$;
  end if
  $idle\_memory -= \textbf{sizeof}(s_j)$;
end while

C. Video Segments Seek

In QUVoD, a segment is stored in multiple nodes that are distributed in several consecutive groups. For example, the nodes in the gth group will certainly store the segment s_g, the nodes in the group $g-1$ will have a great probability of storing the segment s_g, the probability of the nodes in group $g-2$ storing segment s_g is smaller than that for the nodes in the group $g-1$, and so on.

In this context, we designed a novel segment seeking scheme to obtain segments. According to the QUVoD video segments storage scheme, to get a segment, we just need to locate a group to which the nodes storing the segment required belong to. We will choose the group in which the nodes have the highest probability of storing the required segment. That is, if we want to get the segment s_g, we will contact the gth group. We first find the first node in the group and then find the other nodes via this first node.

Since all the nodes in the gth group can serve the segment s_g, a data scheduling strategy is designed to balance the load between nodes when serving resources. The first node V_x in the gth group acts as a "scheduler". The "scheduler" maintains a group member table (GMT) to record all the vehicles' IDs and corresponding serving load information in the gth group. Any item in GMT is associated a 2-tuple $GMT = (NID, Load)$, where NID is the Vehicle ID of the member, and $Load$ is the number of nodes for which this member is the data supplier. The "scheduler" is responsible to assign the member with the lowest serving load as supplier to the requester. If V_z joins/leaves the gth group, V_z's precursor can be aware of V_z's variation and reports this change to V_x. V_x will add/remove V_z's information in GMT. The successor V_y of "scheduler" stores and periodically updates a duplicate of GMT to avoid the loss of GMT. If V_x leaves the group, V_y becomes the new "scheduler" and informs the members in the gth group about this change.

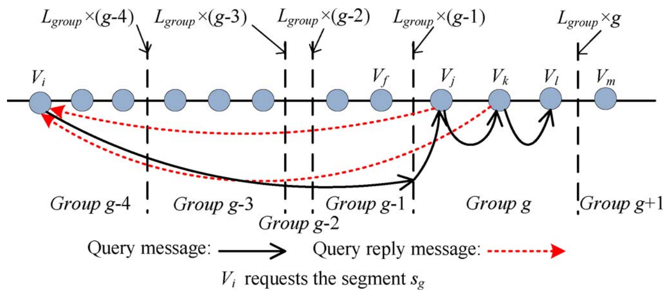


Fig. 5. Video segment seeking procedure.

Assuming vehicle V_i is playing a video and requests a segment s_g, it performs the following procedure.

1) V_i sends out a Query message by DHT for locating the node who possesses a group whose initial boundary is F_g by (3). According to the Chord algorithm, V_i could find the node whose Device ID equals or is closest to but higher than the group's initial boundary F_g along the Chord subcircle. As the Fig. 5 shows, we can find the first V_j in group g. Denote that, in Fig. 5, we have omitted the Video IDs for Vehicle ID, which are all the same in a Chord subcircle.
2) When V_j receives the Query message, by checking its GMT, it selects a member V_k with the lowest load as V_i's supplier. V_k will reply with a Query Reply message to V_i and serve s_g to V_i. V_j modifies V_k's load value in its GMT.
3) If V_i cannot locate the nodes that have the required s_g through the foregoing procedure, it will repeat the process to find nodes in groups $g-1, g-2, \ldots, g-M+1$ using the initial group boundary $F_{g-1}, F_{g-2}, \ldots, F_{g-M+1}$ whose values are $L_{\text{group}} \times (g-2)$, $L_{\text{group}} \times (g-3), \ldots, L_{\text{group}} \times (g-M)$, respectively. The parameter M is the maximum storage limit of all nodes in QUVoD.

Assuming V_k is the located vehicle node after the above search process, the vehicle V_i will download the required video segment s_g from vehicle V_k through G-path and V-Path by taking advantage of the multipath data delivery mechanism, which will be discussed in Section III-D. After downloading s_g, V_i also can continue to download $s_{g+1}, \ldots, s_{g+M_k-1}$ from V_k employing multipath directly, which avoids relocating $s_{g+1}, \ldots, s_{g+M_k-1}$ in Chord. Algorithm 2 describes the process of video segment seek and download.

Algorithm 2 Video segment seek and download algorithm for vehicle V_i

---

/* V_i requests segment s_g */
/* $|GMT|$ is the number of items in GMT */
for $(forth = 0; forth < M; forth++)$
  $fore\_limit = L_{\text{group}} \times (g - forth - 1)$;
  find $fore\_limit$'s Successor V_j by Chord algorithm;
  send a *Query* message to V_j;
  $min = \infty$; $vid = 0$;
  for $(i = 1; i < |GMT|$ of $V_j; i++)$
    if (s_g is local stored && $V_j.GMT[i].Load < min$)
      $min = V_j.GMT[i].Load$;
      $vid = V_j.GMT[i].NID$;
    end if;
  end for;
  if $(vid > 0)$ break; **end if**
end for;
if $(vid = 0)$
  V_i downloads s_g from media server employing multipath delivery mechanism; exit;
end if
V_{vid} sends a *Query Reply* message to V_i;
V_i downloads $s_g, s_{g+1}, \ldots, s_{g+M_i-1}$ from V_{vid} employing the multipath delivery mechanism;
V_j updates $V_j.GMT[i].Load$;
if (downloading $s_t (t \in [g, g+M_i-1])$ fails)
  V_j selects another V_l with low stress in V_j's GMT as V_i's supplier;
  V_i downloads segments between $[s_t, s_{t+M_k-1}]$ from V_l employing multipath delivery mechanism;
end if

To maintain GMT and respond to segment requests, the "scheduler" needs to exchange additional messages, which bring additional load. However, the group members storing multiple sequential segments can serve more than one segment in a row. This supports short-range sequential movement of the playback point, which reduces significantly the number of segment request messages sent to "schedulers." In addition, compared with the high-bandwidth streaming traffic, this overhead is light weighted. Furthermore, we can reduce the load on the "scheduler" by serving less video segments relative to other nodes to balance the load that the additional messages are responsible for. This will be considered by our future work.

D. Multipath Data Delivery

To improve the QUVoD's data delivery performance, we designed a multipath data transfer mechanism complying with the QUVoD multihomed structure. We implement the novel Next Generation Network-Oriented Mobile Equipment (NGN-Oriented ME) architecture [33] into our QUVoD system, which can enable one of the available interfaces according to the specified Quality of Service (QoS) of data.

The NGN-Oriented ME architecture can provide a dual mode such as WiMAX/WLAN interfaces within the Physical layer and an adaptive QoS module dubbed QoS-Cross-IP module (QXIP) within the IP layer. With QXIP, NGN-Oriented ME distributes the packet over either WLAN or WiMAX, depending on the packet type of service. This paper enhances the Dual-mode ME architecture [33] with an intelligent data distributor (IDD) integrated into QXIP. Fig. 6 shows the IDD-enhanced Dual-mode ME architecture.

Supposing a vehicle V_i wants to download video segments from a vehicle V_j, and there are two available paths through V-Path and G-Path, the proposed IDD aims to schedule data distribution over V-Path and G-Path according to their current estimated available bandwidth (AB). V_i launches IDD to

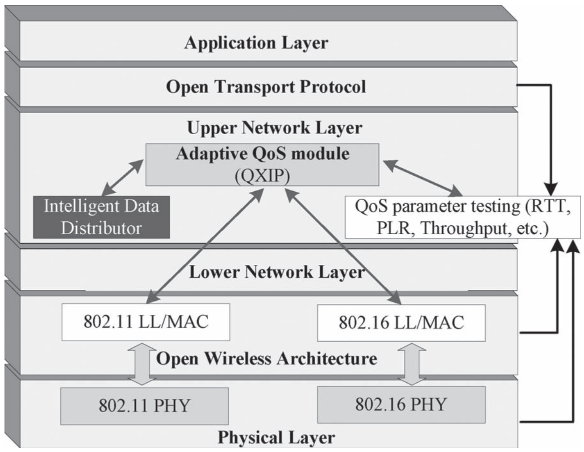


Fig. 6. Enhanced dual-mode ME architecture with IDD.

estimate AB of both V-Path and G-Path, following a well-known AB model for video transport [34], which is shown in

$$AB = \frac{const}{RTT \times \sqrt{PLR}}. \tag{6}$$

In (6), $const$ is a constant with value of 1.22 or 1.33, and PLR can be evaluated by the two-state discrete Markov Chain known as Gibert's model, which is detailed in [34]. RTT can be estimated by

$$RTT = \alpha \times \overline{RTT} + (1 - \alpha) \times (t - T_{\text{send}} - \Delta T). \tag{7}$$

In (7), $\overline{RTT}$ denotes the current round trip time of path, t is the timestamp at the time at which the packet ACK is received at the sender, α is a weighting parameter with a common value of 0.875, T_{send} is the timestamp at the packet sending time, and ΔT is the time interval of a packet handling time at the receiver (it can be omitted in general).

V_i probes for V-Path's AB (AB_v) and G-Path's AB (AB_g) periodically (e.g., per RTT). The V-Path is preferred to distribute packets in VANET. However, if AB_v is less than a threshold AB_{thres}, the gap $AB_{gap} = AB_v - AB_{\text{thres}}$ is greater than a specified value AB_{value}, and in that case, AB_g is higher than AB_{thres}, V_i will distribute packets over G-Path. Otherwise, V_i will enable V-Path to send packets to V_j.

E. Speculation-Based Prefetching

For a vehicle, to smoothen the passengers' video playback, the video segments need to be prefetched and downloaded in advance into the vehicle's buffer. Prefetching has widely been used in current P2P VoD systems [5], [6], [9], [11], [12]. However, most of them only prefetch the subsequent video segment for sequence playback. When a user employs a jump operation, the algorithm should locate the video segment corresponding to the new position, fetch it, and then play it back. The smooth playing will stop if long latency is encountered. Therefore, in interactivity cases, fast resource relocation and transmission is important but not sufficient.

Unlike most of the nodes in other wireless networks, in VANETs, vehicles can be equipped easily with large-capacity-storage and powerful-computing devices. For example, during the duration of a trip by taxi, passengers make up a captive audience, and large quantities of data can be stored in viewing logs. Each vehicle in QUVoD periodically performs statistical analysis of the characteristics of user viewing behavior. Based on this analysis, when a user is playing the current video segment, QUVoD proactively prefetches the segment that is likely to be requested by either sequential playback or VCR-like operations.

TABLE I
EXAMPLE OF VEHICLE VIEWING LOG FOR THE VIDEO IDENTIFIED BY RESOURCE ID

Log ID	Playback Records
1	1, 2, 3, 7, 8, 9, 11, 12, 13, 16, 17, 20
2	1, 4, 5, 6, 7, 10, 11, 12, 13, 16, 17, 18, 20
3	1, 2, 3, 4, 5, 9, 10, 12, 13, 16, 17, 6, 7
4	1, 2, 3, 8, 9, 12, 13, 14, 16, 17, 18, 9, 10
...	...
k	1, 2, 3, 4, 5, 9, 10, 11, 12, 13, 16, 17, 19, 20

A viewing log is actually a user's viewing trace in terms of these video segments and is recorded by the vehicle. Table I shows k user viewing logs, for the same video, identified by a unique Resource ID. The segments in each log are sequence sensitive. Existing studies employ statistical analysis of viewer playback records to calculate segment popularity to support interactivity-related business [9], [11], [12]. However, most of the studies neglect the association between segments. In this paper, we will consider segment-based and substring-based association to predict and prefetch the video segment that is most likely expected by the users. This approach is meant to significantly smoothen user video playing experience in VANETs. Furthermore, we also employ a feedback-based self-regulation mechanism to dynamically regulate the association probability.

Segment-Based Association: In terms of the sequence of segments, the likely playback association between segments can be obtained, for example, as follows: $1 \to 2$, $1 \to 4$, and $2 \to 3$. This is as user interactivity often breaks the sequential order of segments, namely, each segment s_i can be associated with $n - 1$ segments (if the jump target is within the current video segment, the jump behavior is called "inner-segment jump" and does not require special treatment). Our solution counts the frequency of association between each segment s_i and the other $n - 1$ video segments. For instance, in Table I, the frequency of the associations $1 \to 2$ and $1 \to 4$ is 3 and 1 in the first four logs, respectively. A matrix containing the frequencies of the associations for all segments in the k entry log can be defined as

$$\begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{pmatrix}.$$

In the matrix, the value of the main diagonal should be set to 0 due to the fact that the inner segment jump is in fact local seeking so it is not considered as an association. The total frequency of associations between s_i and the other $n - 1$

segments is in fact the sum of the frequency entries in the row corresponding to s_i and is expressed in

$$SUM(s_i) = \sum_{c=1}^{n} f_{ic},\ SUM(s_i) > 0. \tag{8}$$

If the users have accessed any segment s_i, the sum of the ith row in the matrix should be greater than 0. Considering the frequency of association between segment s_i and any segment s_j, denoted f_{ij}, and that $SUM(s_i) > 0$, the jump probability from s_i to segment s_j is computed according to

$$P_{ij}\{s_j|s_i\} = \frac{f_{ij}}{SUM(s_i)}. \tag{9}$$

Moreover, the frequency of s_i to s_j jumps in the k entry log can be considered as a weight factor. The higher the frequency of accessing s_j from s_i is, the more accurate the probability of the $s_i \to s_j$ association. The weight value of s_i can be obtained according to

$$w_{s_i} = \frac{r_{s_i}}{\sum\limits_{c=1}^{n} r_{s_c}} \tag{10}$$

where r_{s_i} is the overall frequency of s_i in k logs. $\sum_{c=1}^{n} r_{s_c}$ indicates the overall frequency of all segments in the k entry log. Next, we can obtain the weighted probability of the $s_i \to s_j$ association according to

$$WP_{ij}\{s_j|s_i\} = w_{s_i} \times w_{s_j} \times P_{ij}. \tag{11}$$

The weighted probability for the associations between each segment to other segments can be therefore obtained. Based on the matrix of association frequencies, we can build the matrix of weighted probability associations as follows:

$$\begin{pmatrix} WP_{11} & WP_{12} & \cdots & WP_{1n} \\ WP_{21} & WP_{22} & \cdots & WP_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ WP_{n1} & WP_{n2} & \cdots & WP_{nn} \end{pmatrix}.$$

In the matrix, WP_{ij} represents the association probability between two segments i and j only and cannot reflect the behavior of the continuous movement of the playback point, which most of the time includes a large series of segments. Therefore, we also use a substring-based weight factor to further influence the segment-based association probability.

Substring-Based Association: The continuous segments form a substring in their playback order. For example, there are five substrings: {1, 2, 3}, {7, 8, 9}, {11, 12, 13}, {16, 17}, {20} in the first entry in the k-entry log [1, 2, 3, 7, 8, 9, 11, 12, 13, 16, 17, 20] in Table I. Even if a single segment playback occurs in a log, it is also considered as a substring such as the case for {20}. Next we obtain a substring set by extracting the substring sample from the k-entry log, namely, $strSet \Leftrightarrow (str_1, str_2, \ldots, str_m)$. Similar to the segment-based matrix of weighted association probability, we can obtain the substring-based matrix of weighted probability associations as follows:

$$\begin{pmatrix} WP_{11}^{(s)} & WP_{12}^{(s)} & \cdots & WP_{1m}^{(s)} \\ WP_{21}^{(s)} & WP_{22}^{(s)} & \cdots & WP_{2m}^{(s)} \\ \vdots & \vdots & \ddots & \vdots \\ WP_{m1}^{(s)} & WP_{m2}^{(s)} & \cdots & WP_{mm}^{(s)} \end{pmatrix}.$$

Moreover, m, the length of $strSet$ ($m \leq \sum_{1\leq t\leq n}^{n}[(t^n/t!) \times \sum_{0\leq e\leq n-t}^{n-t}((-1)^e/e!)]$), is variable as new substrings may be generated at any time.

Assuming that the playback point has gone through h sequential segments before accessing s_i and is located into s_i, namely, there is a substring $str_{t(i)} = (s_{i-h}, s_{i-h+1}, \ldots, s_i)$, we use the accumulated value of substring-based probability from substring $str_{t(i)}$ to all substrings whose first segment is s_j to indicate the substring-based weight factor. The substring-based weighted combinatorial probability can be then calculated according to

$$D_{s_i \to s_j} = (WP_{ij})^{\alpha_{ij}} \times \sum WP_{t(i)c(j)}^{(s)} \tag{12}$$

where $\alpha_{ij} (0 < \alpha_{ij} \leq 1)$ is the impact factor of WP_{ij}, used to regulate WP_{ij}'s contribution according to the prediction results for prefetching. WP_{ij}'s value can be adjusted in real time based on the user playback interest variation instead of being updated after collecting large amounts of log data and recalculation, as in the classic approach. α_{ij} will be further discussed next. $WP_{t(i)c(j)}^{(s)}$ is the association probability between $str_{t(i)}$ and $str_{c(j)}$, whose first segment is s_j. $\sum WP_{t(i)c(j)}^{(s)}$ is the substring-based weight factor. Next, we obtain the combinatorial probability set from s_i to the other $n-1$ segments, namely, $CPS_i = (D_{s_i \to s_1}, D_{s_i \to s2}, \ldots, D_{s_i \to sn})$. Therefore, the target segment in terms of what the user wants to view next is s_j with probability $D_{s_i \to s_j} = MAX(CPS_i)$.

The prediction of playback behavior aims to ensure the users' quality of viewing experience.

1) As the target segment predicted according to the above-indicated algorithm is often the next or previous segment of the current segment, similar to VMesh, a peer in QUVoD keeps two lists of peer pointers, namely, next segment list (NSL) and previous segment list (PSL). By making use of NSL and PSL, the user can connect with the suppliers in overlay one hop and prefetch the video segment immediately.
2) If the target segment predicted is not the next or previous segment of the current segment, the user seeks the predicted segment and fetches it in advance. If the current available bandwidth permits, the next segment is also prefetched as well. Based on this, smooth playback supported whether sequential playout or jumps occur in practice.

With the increase in viewing log entries, the statistics of segment sequence and substrings continually updates so that the corresponding probability also continually changes, improving its accuracy. However, these frequencies are updated regularly every period of time UT set in such as manner not for the real-time updates to determine high computation costs. This efficiency design decision does not affect the computation accuracy of the probabilities too much, as few new viewing log entries do not influence too much the overall values of these probabilities. After every period UT, each node combines the incremental and history statistics to obtain the new probabilities. For instance, let INC_{ij} be the frequency increment for the association from s_i to s_j. We can obtain the new probability by modifying (9)–(13) [(13) is shown below]. This way, the computational complexity significantly reduces

$$P'_{ij}\{s_j|s_i\} = \frac{f_{ij} + INC_{ij}}{SUM(s_i) + INC_{ij}}. \quad (13)$$

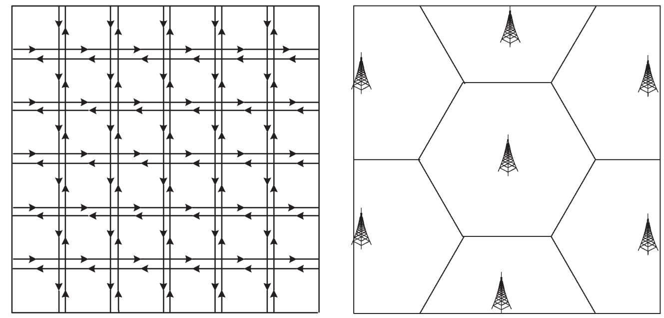
The statistics-based prediction scheme depends on the confidence level of the available heuristic information (viewing log). If the predicted s_j is the real segment the user is requesting after playing s_i, it means the prediction is accurate, and the weighted probability of the $s_i \rightarrow s_j$ association can be increased; otherwise, it should be reduced. We use the feedback information on the prediction accuracy to adjust the impact factors α_{ij} in (12). The new α'_{ij} can be updated according to

$$\alpha'_{ij} = \begin{cases} \alpha_{ij}^{(1+\rho)}, & I(+) \\ \alpha_{ij}^{(1-\rho)}, & I(-) \end{cases}, \quad 0 < \rho < 1 \quad (14)$$

where ρ is a global variable and considered as feedback factor. The $I(+)$ and $I(-)$ represent positive and negative feedback, respectively, namely, the success or failure of prediction. The above reinforce learning model of α_{ij} is to improve the accuracy of prediction by exploiting the user's feedback information. Algorithm 3 describes the speculation-based prefetching strategy.

Algorithm 3 Speculation-based prefetching strategy

```
//Let s_i be current playback segment;
//Let s_r and s_p be real and predicted target segment;
get probability set (WP_i1, WP_i2, ..., WP_in) by (11);
get weighted probability set CPS_i by (12);
select s_p with MAX(CPS_i) as the predicted segment;
if (s_p == s_{i+1} || s_p == s_{i-1})
    locate s_p by using NSL or PLS;
    prefetch s_p by employing multipath delivery
      mechanism;
else
    prefetch s_p by algorithm 2;
end if
if (s_r == s_p) regulate α_ij with I(+); end if //achieve
  smooth playback;
else if (s_r == s_{i+1} || s_r == s_{i-1})
    seek s_p by NSL or PSL;
    else seek and download s_p by algorithm 2;
    end if
    regulate α_ij with I(−); //prediction fails;
end if
```



(a) (b)

Fig. 7. VANET and 4G simulation scenarios. (a) Mobility model in VANET. (b) 4G WiMAX network architecture.

IV. Performance Evaluation

A. Simulation Methodology

QUVoD's performance is assessed in comparison with that of VMesh in VANETs by making use of the network simulator (NS2). The application-layer Chord network of both QUVoD and VMesh is implemented based on the MChord [19], [20], extended according to their different individual algorithms. For QUVoD, we consider two wireless communication interfaces for each of the vehicle nodes: the IEEE 802.11p WAVE network interface and the 4G WiMAX interface. The experiments are implemented using the multiinterface cross layer extension for NS2 (NS-MIRACLE) [35] as it enables the coexistence of multiple modules within each layer of the protocol stack, supporting the integration of IEEE 802.11p and WiMAX network interfaces. The IEEE 802.11p and WiMAX libraries of NS-MIRACLE are available at [36]. For IEEE 802.11p, two WAVE channels are used at the PHY/MAC layer. One is a control channel that is employed for delivering control information. The second one is a service channel, which is used for delivering data.

To reflect the realistic mobility characteristics of VANETs, we use simulation of urban mobility [37] to generate street scenarios. A street area of $2000 \times 2000 (m^2)$, which consists of five horizontal streets and five vertical streets, is used in the experiment. As Fig. 7(a) shows, every street has two lanes in each direction, and the Manhattan Mobility model [38] is employed for vehicular movement. The vehicular nodes are uniformly distributed over the streets and move along the directions shown by the arrows. Whenever a vehicle reaches an intersection, it is determined with some fixed probability whether it would keep moving in the same direction with 50% probability or turn left or right with 25% probability in each case. The 4G WiMAX layout [see Fig. 7(b)] is composed of seven hexagonal cells with 578 m cell radius, namely, seven numbers of base stations (BSs) with 578 m transmission range. We consider a scenario in which a maximum of 1000 vehicles request videos. Around 143 nodes connect and share the same downlink/uplink bandwidth available for each 4G base station. The BSs are connected through a 128-Mb/s wired link with a 2-ms delay. The horizontal handover of vehicle nodes between BSs adopts the hard handover mechanism in IEEE 802.16e [39]. This mechanism has been implemented in the NIST module [40] as extension of NS2 and also has been integrated in the NS-MIRACLE framework by the University of Karlstad [41]. Table II lists

TABLE II
SIMULATION PARAMETER SETTING FOR VANET AND 4G

Parameters		Values
VANET	Area	$2000 \times 2000(m^2)$
	Channel	Channel/WirelessChannel
	Network Interface	Phy/WirelessPhyExt
	MAC Interface	Mac/802_11Ext
	Peak Data Rate	11 Mbps
	Frequency	5.9 GHz
	Multiple Access	OFDM
	Channels Sharing	CSMA/CA
	Transmission Power	33 dBm
	Wireless Transmission Range	$250m$
	Interface Queue Type	Queue/DropTail/PriQueue
	Interface Queue Length	50 packets
	Antenna Type	Antenna/OmniAntenna
	Routing Protocol	AODV
	Peak Mobility Speed	30 *m/s*
	Mobility Model	Manhattan Mobility Model
	Transport-Layer Protocol	TCP
4G WiMAX	Operating Frequency	2.5 GHz
	Channel Bandwidth	10 MHz
	Channel Capacity	40 Mbps
	Radio Propagation Model	MPropagation/MFreeSpace
	Frame Duration	$5ms$
	Multiple Access	OFDM
	Duplexing Mode	TDD
	Interface Queue Length	50 packets
	Cell Layout (BS Number)	7 hexagonal cells
	Cell Radius (Transmission Range)	$578m$
	Transport-Layer Protocol	TCP

some important NS2 simulation parameters of the VANET and 4G WiMAX networks, respectively. Considering multipath data delivery, we set the value of AB_{thres} to 128 kb/s, which is equal to the streaming rate, and the AB_{value} is 8 kb/s in our experiments. Namely, if the value of V-Path's AB [in (6)] is less than 120 kb/s and G-Path's AB is greater than 128 kb/s, the packets will transfer over the G-Path. Otherwise, the V-Path is used to deliver the video segments.

As VMesh is a single-tier solution, we consider two P2P overlay construction ways for VMesh. One is VMesh(ah-hoc), in which vehicles form the Chord on top of VANET. The other case is VMesh(4G), in which the Chord is established over the 4G infrastructure. In VMesh(ad-hoc), packets transfer through V-Path via the WAVE interface. VMesh(4G) uses the G-Path via the WiMAX interface for data transfer, respectively. As in VANET the mobile nodes' position changes continually, VMesh's locality-aware segment management is unusable. Instead, the simulations use the more common case of VMesh, which applies random selection for segment storage.

Simulations consider a 5400-s-long video that is divided into 180 segments. The streaming rate is 128 kb/s. Each segment is 0.5 min long and about 0.47 MB in size. In QUVoD, the maximum storage capacity parameter M is set to 5. As discussed in Section III-E, by analyzing the user viewing logs data, QUVoD applies the speculation-based strategy to adapt to the users' interactive behavior. First, we created 5000 synthetic user viewing log entries based on the interactive actions according to the measurements and statistics from [42]. The video was one of the most popular matches in terms of the number of viewers. The impact factor α_{ij} in (12) was set to 1 to make the segment-based association and substring-based association have equal impact on the prediction. By using the same method, we then generate 1000 sole user viewing logs and assign each of them to each of the 1000 nodes to create the multiple requests scenario. The playback trace of a node can be determined by its assigned viewing record. For instance, a vehicle with playback record (1, 2, 3, 4, 7, 8, 9, 10, 14, 15, 16, 17, 18, 40, 28, 29, 30, 31, 44, 45, 46, 47, 52, 53, 60, 61) has a lifetime of 13 min and takes seven VCR operations in total.

We compare QUVoD with VMesh(ad-hoc) and VMesh(4G) in terms of the user QoE-related indicators, including average lookup success rate (ALSR), average segment seeking latency (ASSL), and hit ratio in supporting interactivity, and overhead-related metrics, including vehicles stress distribution, media server stress, and system control overhead.

B. Simulation Results

1) ALSR: The video segment ALSR describes the system serving capability. The ALSR is defined as the ratio between the number of times lookup success was obtained and the total number of lookup tries. Both VMesh and QUVoD are P2P-based VoD solutions, and to fairly compare the efficiency of their P2P-related algorithms, including video segments storage and seek strategy, etc., we neglect the requests from the server for computing ALSR. A high ALSR indicates it is easier for a vehicle node to obtain video resources from other suppliers instead of from the media server. This improves the system's resources sharing capability and indirectly reduces the media server stress. The experiments run involve variation in the mobility speed of VANET vehicles and different vehicular node numbers.

Fig. 8 shows the ALSR variation with the increase in the number of vehicles from 200 to 1000, where the mobility speed

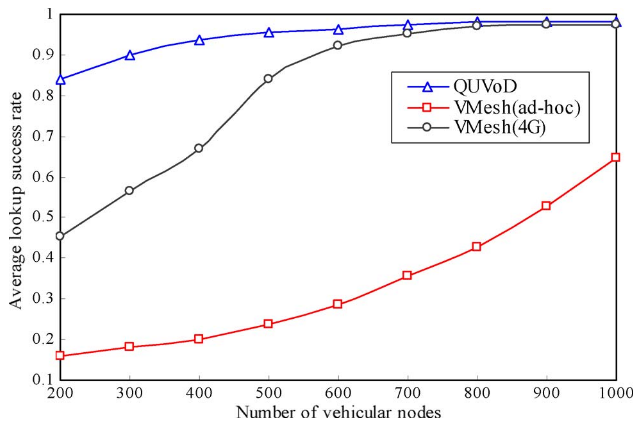


Fig. 8. ALSR against number of vehicular nodes.

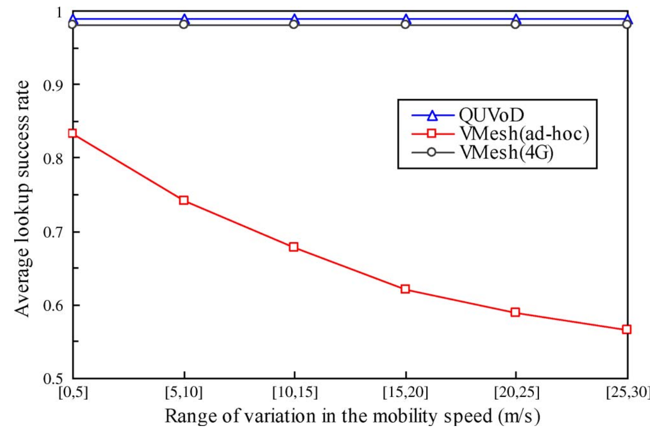


Fig. 9. ALSR against different vehicle mobility speed variation.

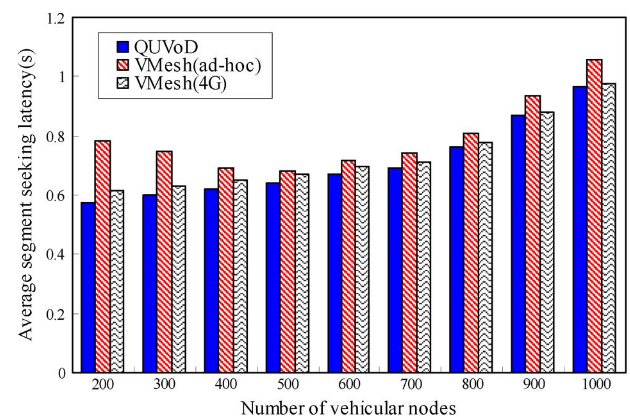


Fig. 10. ASSL against number of vehicular nodes.

varies in the [0, 30] m/s range. As the figure shows, with the system scale increase, the ALSRs of QUVoD and VMesh all increase. This is as the increasing number of vehicles determines higher segment sharing chances, and any vehicle can find an easier resource supplier. The ALSRs of QUVoD and VMesh increase quickly when the vehicle number increases from 200 to around 500. Above 600 nodes, the ALSRs of QUVoD and VMesh maintain a relatively stable value close to 1. However, when system scale is low (there are less than 700 nodes), the ALSR of QUVoD is higher than that of VMesh(4G). This is because QUVoD's grouping-based segment storage enhances the ALSR compared with the random segment storage used in VMesh. The ALSR of VMesh(ad-hoc) increases slowly and only arrives at about 0.645 when the number of vehicles increases to 1000. QUVoD and VMesh(4G) locate the video segments over Chord by using the underlying 4G network, which is stable and reliable. The vehicles' movement in a vehicular scenario does not influence the Chord circle. As long as there are enough vehicles, a supplier can always be found via 4G by DHT. However, as VMesh(ad-hoc) is formed on top of VANET, ALSR is easily affected by vehicle density, particularly in low-density situations. As the figure shows, when there are 1000 vehicles, the ALSRs of QUVoD and VMesh(4G) are about 35% higher than the ALSR of VMesh(ad-hoc). However, the difference between ALSRs can be higher than 70% when the number of nodes is 600.

Fig. 9 illustrates the ALSR comparison results with the variation in the mobility speed of VANET vehicles. To minimize the impact caused by the number of nodes, the number of vehicular nodes is set to 1000. We can see in the figure that the ALSRs of QUVoD and VMesh(4G) always maintain high levels. The ALSR of VMesh(ad-hoc) decreases with the increase of mobility speed variance. This is as in VMesh(ad-hoc), the increase in vehicle mobility determines the P2P overlay over the mobile ad hoc network to unstable, which results in some lookups not to reach the vehicles with requested segments. Unlike VMesh(ad-hoc), message communications in QUVoD and VMesh(4G) are through the more stable 4G network, and the increase in the movement of vehicles can hardly influence data communication.

2) ASSL: The ASSL is the average time from a vehicle requesting a video segment to having received the segment into its playback buffer. The ASSL mainly includes the average segment locating latency and the average segment downloading latency. When a vehicle joins the system, the ASSL determines how long a user needs to wait for the download of initial video segments to begin playing, so the ASSL has a significant impact on joining and startup latency. In addition, during the video playing process, a long ASSL can severely affect the user viewing QoE levels as late arriving data may cause buffer underflow. Consequently, there is a desire to have short ASSL. In simulations, we first compute the average segment locating latency in the Chord network and then compute the average end-to-end transfer delay for a segment between two vehicle nodes. The sum of these two values determines the ASSL. As vehicles may request segments many times, only the success in the seeking process will be considered in the calculation of the average ASSL.

Fig. 10 plots the comparison results when QUVoD and VMesh are used in turn for increasing the number of vehicular nodes, where the mobility speed varies in the [0, 30] m/s range. As the figure shows, the ASSL of VMesh(ad-hoc) is higher than that of QUVoD and VMesh(4G), particularly when the system scale is low (vehicle number is less than 400). This is because the low density of nodes in VANET reduces the connectivity between nodes. QUVoD performs better than the other two methods. For example, in the case of 200 vehicles, the QUVoD's ASSL is about 0.573 s, 7% lower than that of VMesh(4G) (0.613 s) and 36.8% lower than that of VMesh(ad-hoc) (around

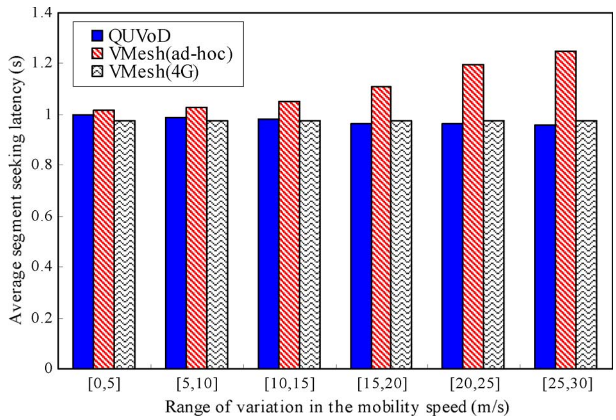


Fig. 11. ASSL against speed of vehicles.

0.784 s). Fig. 11 illustrates the ASSL comparison results for different ranges of mobility speeds, where the number of vehicular nodes is set to 1000. As the figure shows, when the nodes have low mobility speeds (less than 10 m/s), VMesh(ad-hoc) performs well. However, with increasing mobility speeds, VMesh(ad-hoc) performs the worst among the three methods compared. The QUVoD's ASSL is slightly higher than that of VMesh(4G) in the low-mobility speed case, and the reason is that QUVoD mainly uses V-Path for data delivery, which can be negatively impacted by V2V communications. However, the difference is very small, and in all other cases, the ASSL of QUVoD is maintained at a good level. This is as QUVoD stores several sequential segments in one node, the segments excluding the first segment do not need to be located again, which significantly reduces the average segment locating latency and mitigates the negative impact brought by the partial use of V-Path.

In VMesh(ad-hoc), both the search message and the data are transmitted through the VANET network, which is unreliable, so the ASSL and ALSR of VMesh(ad-hoc) are worse than those of QUVoD and VMesh(4G) in low-density and high-mobility speed situations. VMesh(4G) shows a relatively good performance for both ALSR and ASSL. However, the network usage cost of VMesh(4G) is the highest as all the search messages and data transfers are performed over the 4G network. For QUVoD, the search messages transfer in the 4G network, and the video data are based on the multipath transfer in the multihomed 4G/VANET network; therefore, QUVoD has both high seeking success rate and relatively low segment seeking delay.

3) G/V-Paths Evaluation: Fig. 12 illustrates the investigation on the average probability of using G-Path and V-Path to download video segments by applying the multipath data delivery mechanism in QUVoD. Fig. 12(a) investigates the probability against the increase in the number of vehicular nodes, where the mobility speed varies in the [0, 30] m/s range. As it is shown, when there are few vehicles in the system, for instance, 200 vehicles, the probability of employing G-Path is around 85.6% for segment download and 14.4% for using V-Path. However, as the system scale increases, the probability of using G-Path significantly decreases. For example, when the system scale reaches 1000 nodes, the probability of using G-Path is only 37.5%, and V-Path's usage probability reaches 62.5%. Fig. 12(b) evaluates the probability of using G/V-Paths in terms of range of variation in mobility speed. It illustrates how for slow speeds the probability of using G-Path is also small (e.g., when the speed is in the [0, 5] m/s range, the probability of using G-Path is 16.5% and that of employing V-Path is 83.5%). However, with speed increases belonging to the [25, 30] m/s range, the probability of using G-Path increases to 43.8% and that of V-Path decreases to 56.2%. Fig. 12 fully illustrates that QUVoD can combine the merits of VANET and 4G network and achieve a desirable tradeoff between quality and network usage cost.

4) Average Hit Ratio (AHR): We use the AHR metric to evaluate the performance in terms of supporting interactivity. After finish playing a segment, a vehicle then continues seeking a next segment for play-out. It will check first its playback buffer. If the buffer contains the required segment, it is considered as a hit event, and the vehicle can continue video playback smoothly without any suspension. Otherwise, it is seen as a miss event, and the vehicle needs to relocate and fetch the required segment, which involves long latency and seriously deteriorates the user QoE level. The AHR is the ratio between the total number of hit events and the total number of seeking requests. In this experiment, we investigate the AHR when the number of vehicles increases. The mobility speed is in the [0, 30] m/s range.

As Fig. 13 shows, if the system has a small scale, increasing the number of vehicles can improve the AHR. However, increasing the vehicle number beyond 600 sees almost no effect on the AHR values of QUVoD and VMesh. This is because the P2P sharing capability is no longer the influencing factor for AHR; the segment prefetching scheme has become the dominating factor for AHR. We can see that the AHR values of QUVoD are about 20% and 25% higher than those of VMesh(4G) and VMesh(ad-hoc), respectively. In VMesh, only the sequential playback behavior determines a hit and increases the AHR; any interactivity determines prefetching to fail, which decreases the AHR. However, in QUVoD, by employing speculation-based prefetching strategy, the segments that are not only for sequence playback but for jump as well can be prefetched with high possibility, which makes QUVoD achieve higher AHRs and better QoE than VMesh. Fig. 13 also shows how around 10% of prefetching fails in QUVoD, which means some prefetched segments cannot be used for playback. Although this wastes some additional bandwidth and storage for those failed prefetching, considering the increasing bandwidth on VANETs and storage capability on local vehicles, it actually offers a very desirable tradeoff between quality and cost.

Fig.14 (a)–(c) shows the average AHR distribution of QUVoD and VMesh related to the seeking event between segments, respectively. The average AHR of each seeking event $S_{s_i \to s_j}$ is defined as

$$HR_{ij} = \frac{f_{ij}^H}{\sum_{b=1}^{n}\sum_{c=1}^{n} f_{bc}}, \quad f_{ij}^H \leq f_{ij}, \quad HR_{ij} \in [0,1] \qquad (15)$$

where f_{ij} and f_{ij}^H are the occurrence frequency of $S_{s_i \to s_j}$ and its hit event, respectively. $\sum_{b=1}^{n}\sum_{c=1}^{n} f_{bc}$ is the occurrence

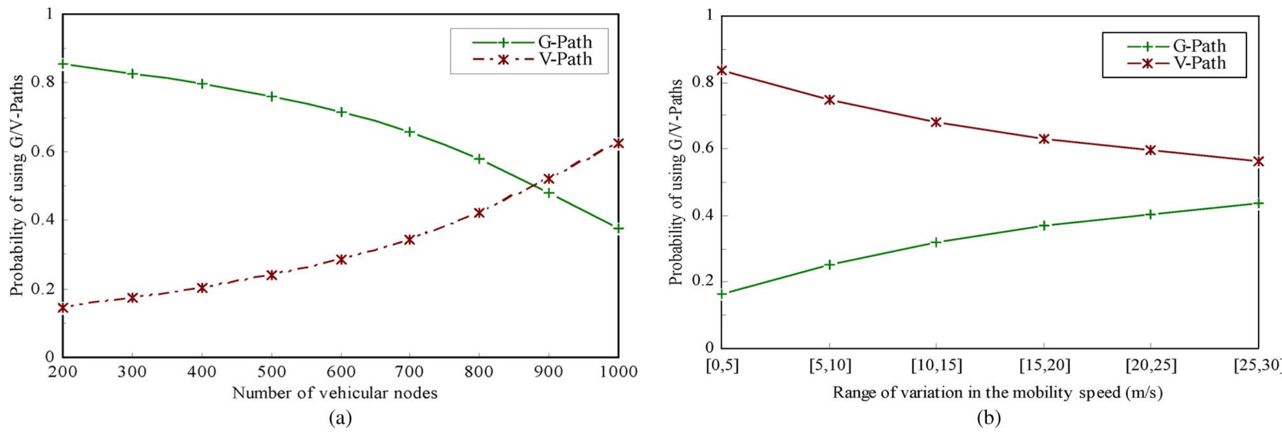


Fig. 12. Evaluation on probability of using G/V-Paths for downloading segments.

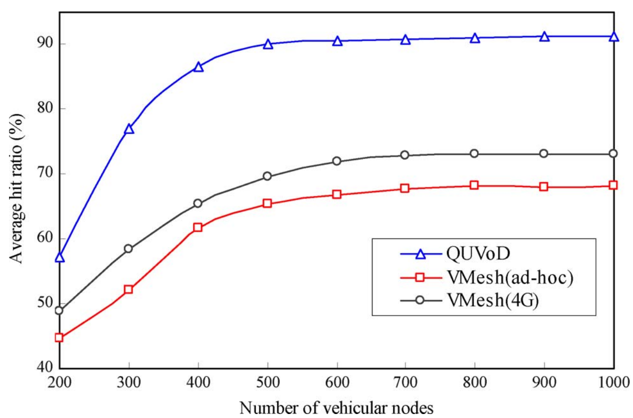


Fig. 13. AHR against number of vehicular nodes.

frequency of all seeking events. As Fig. 14 shows, the AHR values of VMesh are all distributed in a plane that is composed of coordinate axis Z and a straight line closed to the leading diagonal in the plane XOY. This is because the users in VMesh prefetch sequentially the next video segment only. The AHR values of VMesh(4G) are greater than those of VMesh(ad-hoc) due to the stability of the 4G network. In general, the AHR values of QUVoD are far greater than those obtained by VMesh. QUVoD can provide accurate prediction results for all seeking events to ensure high playback continuity with the help of the proposed speculation-based prefetching strategy.

5) Vehicles Stress: In QUVoD and VMesh, the vehicular nodes not only need to seek and download new video content from other vehicular nodes after consuming the current video segment but also respond to seek requests from other nodes and deliver the streaming data requested. We investigate the distribution of the number of vehicular nodes relative to the number of seeking events and corresponding cumulative distribution function (cdf) to support the analysis of vehicle stress, respectively. In Fig. 15(a), the curve corresponding to the distribution experiences first a rise and then a fall, with the peak value in the range [120, 140]. The number of seeking events is in the range [56, 189], and the average number of seeking events for each vehicular node is 128.97. Fig. 15(b) shows the cdf corresponding to the distribution in Fig. 15(a). As we know that the number of seeking events is greater than the total number of video segments (180), there is a possibility that some users watched the entire video. In Fig. 15(b), the curve in the range [180, 200] tends to a smooth horizontal line (cdf is close to 1), indicating that a small minority of users could have possibly viewed the whole video. The curve fast rises in the range [100, 160], which means that the number of seek events is in this range for most users.

The stress (load) distribution among the vehicular nodes in our simulation is a statistic about the number of vehicular nodes in different stress states. This is computed as the ratio between each node's stress value and the total stress of all vehicular nodes, where the number of served segments denotes the stress of each vehicular node. Fig. 16 illustrates the comparison statistics about the number of nodes in different ratio ranges between VMesh and QUVoD, where the total number of vehicular nodes is 1000. As the figure shows, more than 15% of nodes in VMesh do not bear any load from start to end. They just download segments from others but supply no resources. In QUVoD, each node bears a part of the system's load, and very few nodes are overloaded. It is important to note that the number of vehicles who bear load in the range (0, 2] exceeds 700 out of the total of 1000 nodes, which is a highly positive result for QUVoD.

In VMesh, the random storage scheme determines the uneven video segment distribution in the Chord network, which leads to the important load imbalance between nodes. However, in QUVoD, the grouping-based segment storage scheme enables uniform distribution of all the segments across the nodes. In general, the $S_{\max}$ numbers of nodes can store all the video segments of the requested video. Furthermore, the nodes in the same group can serve the same video segment request, and the nodes with low stress are preferred to provide service, which further helps achieve load balancing. This way, each vehicle node in QUVoD shares a part of the load of the system.

6) Media Server Stress: In addition, in the simulation, we also investigate the stress on the media server. The media server is a backup resource provider; the vehicular nodes will download segments directly from this server when they cannot fetch the required resource from other nodes successfully. Obviously, the fewer streams the media server delivers, the lower the stress on the server. Fig. 17 illustrates the server stress measured by the average ratio between the number of streams served from the server and the total number of streams in system during a

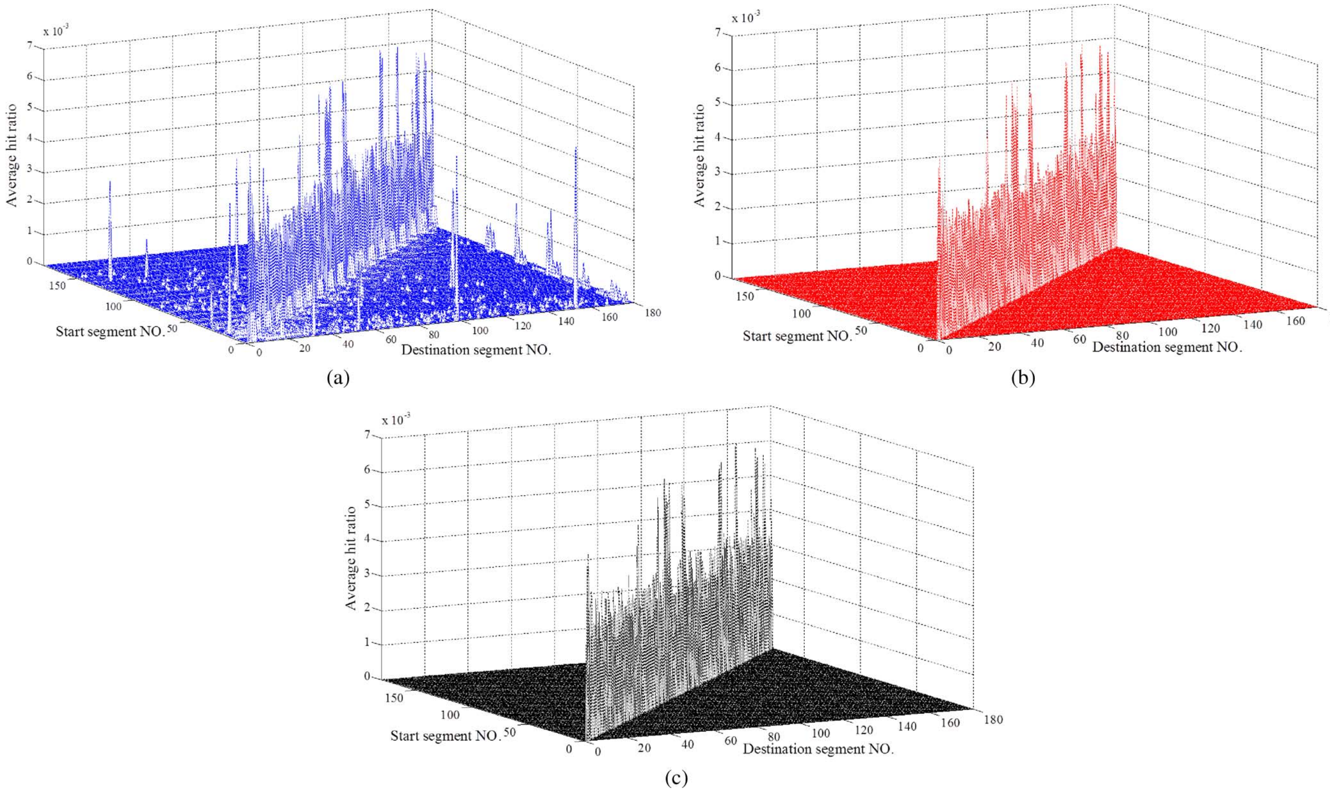


Fig. 14. AHR against seeking events. (a) AHR of seeking events for QUVoD. (b) VMesh(ad-hoc). (c) VMesh(4G).

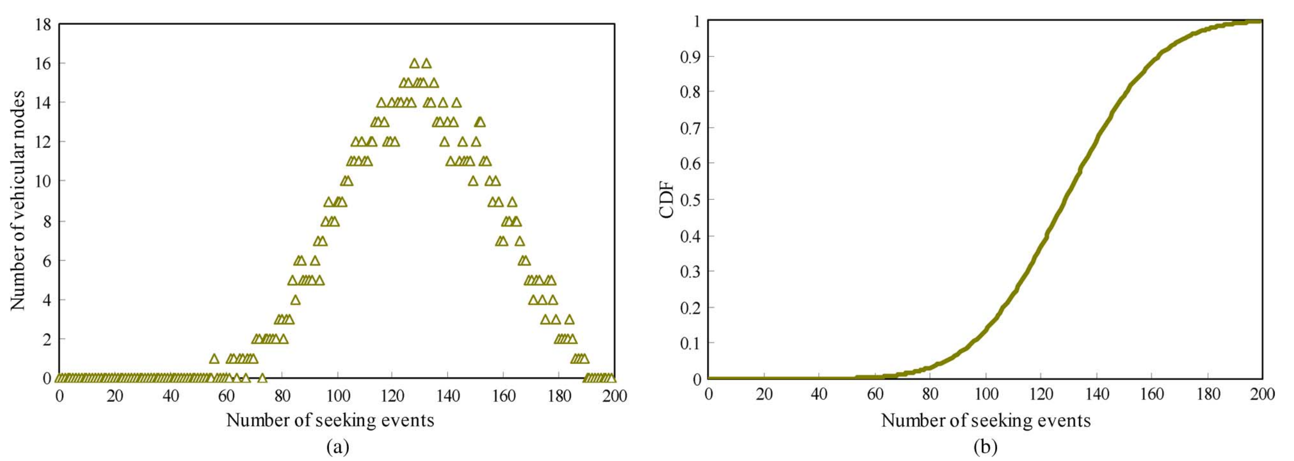


Fig. 15. Distribution and cdf of vehicular nodes against number of seeking events.

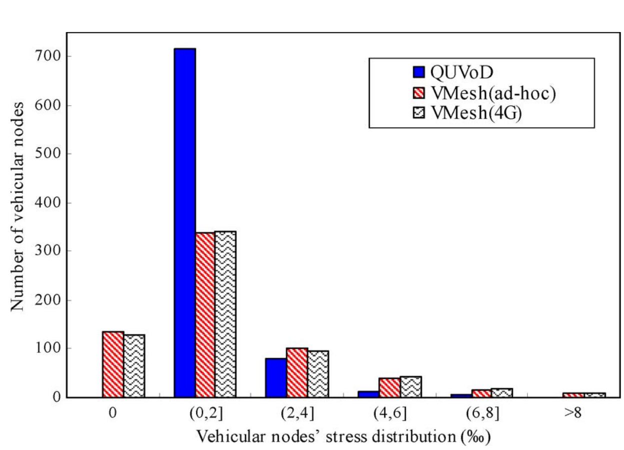


Fig. 16. Vehicular nodes' stress distribution.

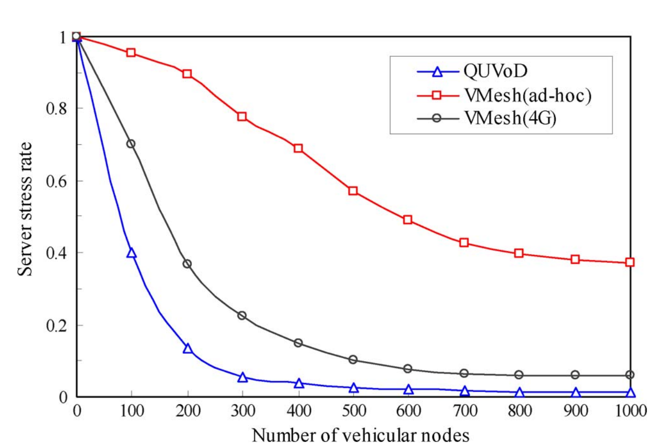


Fig. 17. Media server stress against the number of vehicular nodes.

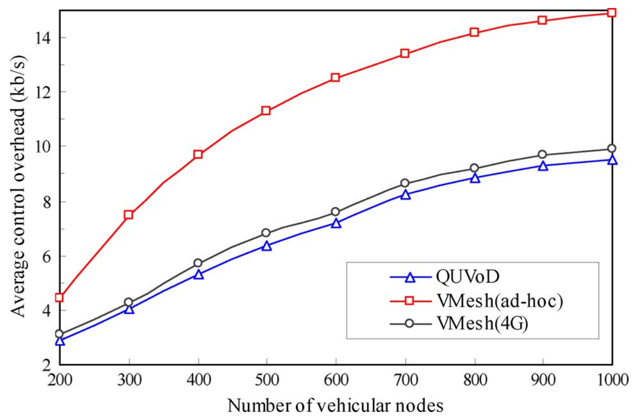


Fig. 18. Control traffic overhead.

period of simulation time. The vehicle number changes from 0 to 1000 with mobility speed in variation range of [0, 30] m/s. We can see in the figure how the media server stress rate in VMesh and QUVoD decreases with the increase in vehicle number as the sharing capability between vehicles increase with the system scale increase. However, compared with VMesh, the server stress of QUVoD remains at low levels and becomes close to zero after about 600 vehicles. This lightweight load is due to the grouping-based distributed segment storage strategy and utilizing the multipath transfer mechanism. Due to the random storage strategy and also VANET's instable characteristics, the server stress of VMesh(ad-hoc) is in a high state. As Fig. 17 shows, it can reach at about 0.4, even when the vehicle number increases to 1000. The server stress of VMesh(4G) performs better than VMesh(ad-hoc) due to the stability merit of the 4G network.

7) Control Overhead: In QUVoD and VMesh, nodes joining, building, and maintaining the Chord P2P network, and video segment seeking, etc., require nodes to exchange control messages. We count all control messages as control traffic overhead and use the control messages' average occupied bandwidth per second as the control overhead. As shown in Fig. 18, the control overhead of QUVoD and VMesh grows with the increase in the number of vehicles. This is mainly because the increase in system scale will determine an increase in the number of exchanged control messages between nodes and will result in an increase in the Chord overlay maintenance cost. VMesh(ad-hoc) performs the worst as the Chord is formed directly on top of VANET, the overlay topology is easily affected by the underlying vehicles' mobility, and each vehicle needs frequent exchange of messages to maintain in real time the connections for nodes frequently joining and leaving. However, as the Chord of QUVoD and VMesh(4G) is built over the 4G network, which is robust to variable network conditions, the topology maintenance messages can be significantly reduced. In QUVoD, because of the storage scheme by which we store several sequential segments in one node, more than one segment can be found at one time, which certainly greatly reduces the average seeking segment time; the control messages are also reduced correspondingly. Consequently, compared with VMesh(4G), QUVoD achieves lower control traffic overhead.

V. Conclusion and Future Work

The very attractive user-centric mobile VoD services can support different applications in VANETs. It is a great challenge to support high QoE for interactive mobile VoD services not only due to the high mobility of the devices but also due to the users' interactive viewing behavior. In this paper, we proposed an efficient user-centric mobile VoD solution called QUVoD in an urban vehicular network environment, which offers high QoE service level to vehicle passengers. Based on the proposed efficient designed multihomed hierarchical P2P/VANET structure, four novel mechanisms are introduced: 1) distributed grouping-based video segments storage scheme; 2) video segment seeking scheme; 3) multipath data delivery mechanism; and 4) speculation-based prefetching strategy. The storage scheme distributes the segments along the Chord network uniformly by groups, allowing each video segment to be stored in multiple nodes, and each node to store multiple video segments. This not only balances the nodes' load but also reduces the resource seeking times. By employing the video segment seeking scheme, the video segment lookup messages transverse the Chord network by the underlying 4G network. The stability of the 4G network ensures high lookup success rate. After lookup, the video segments are downloaded from the located node by the multipath data delivery mechanism over multihomed 4G/VANET networks, achieving high data rate and reliability. Furthermore, the presented prefetching strategy can sense the user's viewing behavior and make intelligent decisions for fetching expected segments in advance, which greatly smoothen the playback experience for interactive users. The simulation-based results show how QUVoD is an efficient interactive mobile VoD solution in urban vehicular networks.

Extending the sophisticated multipath data delivery mechanism presented in this paper, future work will consider employing a Concurrent Multipath Transfer (CMT) strategy [30] and other CMT-related technologies in QUVoD, and distribute data across over G-Path and V-Path concurrently to achieve bandwidth aggregation. It can accelerate video segment fetching speed to further improve QUVoD's performance. In addition, we will also explore multiple video streams delivered simultaneously. Those results will be very useful and are expected to provide support for the very attractive high-definition VoD services in future urban multihomed P2P-based vehicular networks.

References

[1] G. Piro, A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE cellular systems: An open source framework," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 498–513, Feb. 2011.

[2] 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 Trans. Veh. Technol.*, vol. 61, no. 1, pp. 3–14, Jan. 2012.

[3] Q. Wang, P. Fan, and K. B. Letaief, "On the joint V2I and V2V scheduling for cooperative VANETs with network coding," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 62–73, Jan. 2012.

[4] YouTube. [Online]. Available: http://www.youtube.com

[5] C. Xu, G.-M. Muntean, E. Fallon, and A. Hanley, "A balanced tree-based strategy for unstructured media distribution in P2P networks," in *Proc. IEEE ICC*, May 2008, pp. 1797–1801.

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

[7] C.-L. Chang and S.-P. Huang, "The interleaved video frame distribution for P2P-based VoD system with VCR functionality," *Comput. Netw.*, vol. 56, no. 5, pp. 1525–1537, Mar. 2012.

[8] Y. Zhou, T. Fu, and D. M. Chiu, "A unifying model and analysis of P2P VoD replication and scheduling," in *Proc. IEEE INFOCOM*, 2012, pp. 1530–1538.

[9] C. Lee, E. Hwang, and D. Pyeon, "A popularity-aware prefetching scheme to support interactive P2P streaming," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 382–388, May 2012.

[10] W. Wu and J. C. S. Lui, "Exploring the optimal replication strategy in P2P-VoD systems: Characterization and evaluation," in *Proc. IEEE INFOCOM*, 2011, pp. 1206–1214.

[11] W.-P. K. Yiu, X. Jin, and S.-H. G. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 1717–1731, Dec. 2007.

[12] S.-H. Gary Chan and W.-P. K. Yiu, "Distributed storage to support user interactivity in peer-to-peer video," U.S. Patent 7 925 781, Apr. 12, 2011.

[13] Y. Hsieh and K. Wang, "Dynamic overlay multicast for live multimedia streaming in urban VANETs," *Comput. Netw.*, vol. 56, no. 16, pp. 3609–3628, Nov. 2012.

[14] L. Zhou, Y. Zhang, K. Song, W. Jing, and A. V. Vasilakos, "Distributed media services in P2P-based vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, pp. 692–703, Feb. 2011.

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

[16] Z. Yang, M. Li, and W. Lou, "CodePlay: Live multimedia streaming in VANETs using symbol-level network coding," in *Proc. IEEE ICNP*, 2010, pp. 223–232.

[17] I. Stoica, R. Morris, D. L. Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for Internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[18] C. Canali, M. E. Renda, P. Santi, and S. Burresi, "Enabling efficient peer-to-peer resource sharing in wireless mesh networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 3, pp. 333–347, Mar. 2010.

[19] C.-L. Liu, C.-Y. Wang, and H.-Y. Wei, "Mobile chord: Enhancing P2P application performance over vehicular ad hoc network," in *Proc. IEEE Globecom*, 2008, pp. 1–8.

[20] C.-L. Liu, C.-Y. Wang, and H.-Y. Wei, "Cross-layer mobile chord P2P protocol design for VANET," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 6, no. 3, pp. 150–163, Aug. 2010.

[21] W. Ding and S. S. Iyengar, "Bootstrapping chord over MANETs—All roads lead to Rome," in *Proc. IEEE WCNC*, 2007, pp. 3501–3506.

[22] A. Forestiero, E. Leonardi, C. Mastroianni, and M. Meo, "Self-chord: A bio-inspired P2P framework for self-organizing distributed systems," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1651–1664, Oct. 2010.

[23] W. C. Ao, P.-Y. Chen, and K.-C. Chen, "Rate-reliability-delay tradeoff of multipath transmission using network coding," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2336–2342, Jun. 2012.

[24] C. Ahlund, R. Brannstrom, and A. Zaslavsky, "Traffic load metrics for multi-homed mobile IP and global connectivity," *Telecommun. Syst.*, vol. 33, no. 1–3, pp. 155–185, Dec. 2006.

[25] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming shim protocol for IPv6," IETF Trust, Fremont, CA, RFC 5533, Jun. 2009.

[26] T. Okimoto, D. Wing, D. Miles, and S. Matsushima, *IPv6 Multihoming Without Network Address Translation*. Fremont, CA: IETF, Feb 2012, Internet draft, work in progress.

[27] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development," IETF Trust, Fremont, CA, RFC 6182, Mar. 2011.

[28] C. Xu, E. Fallon, Y. Qiao, G.-M. Muntean, X. Li, and A. Hanley, "Performance evaluation of distributing real-time video over concurrent multipath," in *Proc. IEEE WCNC*, 2009, pp. 1–6.

[29] C. Xu, E. Fallon, Y. Qiao, L. Zhong, and G.-M. Muntean, "Performance evaluation of multimedia content distribution over multi-homed wireless networks," *IEEE Trans. Broadcast.*, vol. 57, no. 2, pp. 204–215, Jun. 2011.

[30] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: Quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Trans. Mobile Comput.*, Aug. 2012, to be published.

[31] G. Piro, L. A. Grieco, G. Boggia, R. Fortuna, and P. Camarda, "Two-level downlink scheduling for real-time multimedia services in LTE networks," *IEEE Trans. Multimedia*, vol. 13, no. 5, pp. 1052–1065, Oct. 2011.

[32] T. Pitoura, N. Ntarmos, and P. Triantafillou, "Replication, load balancing and efficient range query processing in DHTs," *Adv. Database Technol.*, vol. 3896, *LNCS*, pp. 131–148, 2006.

[33] T. Shuminoski and T. Janevski, "Cross-layer adaptive QoS provisioning for next generation wireless networks," *Int. J. Res. Rev. Next Gener. Netw.*, vol. 1, no. 1, pp. 7–13, Mar. 2011.

[34] Q. Zhang, W. Zhu, and Y. Q. Zhang, "Resource allocation for multimedia streaming over the Internet," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 339–355, Sep. 2001.

[35] N. Baldo, F. Maguolo, M. Miozzo, M. Rossi, and M. Zorzi, "Miracle: The multi-interface cross-layer extension of ns2," *EURASIP J. Wireless Commun. Netw.*, vol. 2010, p. 26, Apr. 2010.

[36] [Online]. Available: https://telecom.dei.unipd.it:/tlcrepos/nsmiracle-dev/trunk

[37] D. Kmjzewicz, G. Hertkorn, P. Wagner, and C. Rossel, "Simulation of urban mobility (SUMO): An open-source traffic simulation," in *Proc. 4th Middle East Symp. Simul. Model.*, 2002, pp. 1–25.

[38] 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 Netw.*, vol. 1, no. 4, pp. 383–403, Nov. 2003.

[39] *IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE Std. 802.16e-2005, 2005.

[40] The Network Simulator NS-2 NIST add-on—IEEE 802.16 model (MAC+PHY), Nat. Inst. Stand. Technol. Gaithersburg, MD, Tech. Rep., Jan. 2009.

[41] NS2MiracleWimax. [Online]. Available: http://sourceforge.net/projects/ns2miraclewimax/

[42] 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, Apr. 2007, pp. 1–6.

Changqiao Xu (M'02) received the Ph.D. degree from the Chinese Academy of Sciences (ISCAS), Beijing, China, in January 2009.

He was an Assistant Research Fellow with ISCAS from 2002 to 2007, where he was the Research and Development Project Manager in area communication networks. During 2007–2009, he was a Researcher with the Software Research Institute, Athlone Institute of Technology, Ireland. He joined the Beijing University of Posts and Telecommunications (BUPT), Beijing, in December 2009 and was an Assistant Professor from 2009 to 2011. He is currently an Associate Professor with the Institute of Network Technology and an Associate Director of the Next Generation Internet Technology Research Center, BUPT. His research interests include wireless networking, multimedia communications, and next-generation Internet technology.

Futao Zhao received the B.S. degree from Hangzhou Dianzi University, Hangzhou, China, in 2010. He is currently working toward the M.S. degree with the Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China.

His research interests include wireless sensor networks, vehicular networks, and multimedia streaming.

Jianfeng Guan received the Ph.D. degree in communications and information systems from the Beijing Jiaotong University, Beijing, China, in January 2010.

He is currently a Lecturer with the Institute of Network Technology, Beijing University of Posts and Telecommunications. His main research interests focus on mobile internet protocols, mobile multicast, and next-generation Internet technology.

Hongke Zhang received the Ph.D. degree in electrical and communication systems from the University of Electronic Science and Technology of China, Chengdu, China, in 1992.

From 1992 to 1994, he was a Postdoctoral Research Associate with the Beijing Jiaotong University, Beijing, China, and in July 1994, he became a Professor. He is also currently the Head of the Institute of Network Technology, Beijing University of Posts and Telecommunications, and the Chief Scientist of a National Basic Research Program of China. 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.

Gabriel-Miro Muntean (M'04) received the Ph.D. degree in quality-oriented adaptive multimedia streaming from Dublin City University (DCU), Dublin, Ireland, in 2003.

He is a Lecturer with the School of Electronic Engineering, Dublin City University. He is Co-Director of the DCU Performance Engineering Laboratory, Director of the Network Innovations Centre, part of the Rince Institute Ireland, and Consultant Professor with Beijing University of Posts and Telecommunications, Beijing, 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 120 papers in prestigious international journals and conferences, has authored two books and 12 book chapters, and has edited five other books.

Dr. Muntean is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, an Associate Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and a Reviewer for other important international journals, conferences, and funding agencies. He is a member of the Association for Computing Machinery and the IEEE Broadcast Technology Society.