IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received June 23, 2020, accepted July 13, 2020, date of publication July 22, 2020, date of current version August 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011111

Peer-to-Peer Data Dissemination for Deadline-Sensitive Streaming in VANETs

CHIA-CHENG HU, (Member, IEEE)
College of Artificial Intelligence, Yango University, Fuzhou 350015, China
e-mail: cchu.chiachenghu@gmail.com

ABSTRACT In this paper, a mechanism of Peer-to-Peer (P2P) data dissemination for distributing deadline-sensitive streaming files in Vehicular Ad-Hoc Networks (VANETs) is proposed. In a multi-media application sensitive to deadline, the data requested by the peer needs to be received before its playback deadline. For the purpose, we adopt different strategies from the previous works, as follows. First, a distinct lookup function is designed by just collecting the part information of data distribution to reduce the network overhead. It is executed simultaneously with the task of requesting data to take advantage of the timeliness of collected information. Second, in order to utilize the limited bandwidth of VANETs efficiently, the function in requesting data schedules only data sufficient to meet the playback rate requirement of the service. Third, a method is designed to provide deadline-sensitive segment dissemination and alleviate the problem of the highly mobile hosts in VANETs. It interactively uses the estimated and actual historical values of the transmission time between two neighboring hosts and the period of two neighboring hosts remaining connected. Fourth, to further alleviate the mobility problem, a route recovery function is designed to recover a disconnected route timely, and a system parameter can be set according to host speed. Fifth, the setting of the system parameter can be a trade-off between the playing quality of the service and the capacity of the network. Compared with a very recently related work, the effectiveness of this mechanism has been verified by a lot of simulations. When the hosts in the network are static, the admission peers of the mechanism increases by 18% compared with the recent work. On the other hand, when the hosts are dynamic, the mechanism improves 10-25% and 0-14% in admission peers and timely received data over the recent work, respectively.

INDEX TERMS Peer-to-peer (P2P), playback rate, deadline-sensitive streaming, segment scheduling, vehicular ad-hoc network (VANET).

I. INTRODUCTION

Motivated by the widespread deployment of inexpensive broadband wireless connections for users, many deadline-sensitive multimedia applications, such as Video-on-Demand (VoD) system, have become practical. Due to the huge scale of multimedia application files, the streaming technology divides the multimedia files into equal sized fragments, called segments, which are the smallest operating unit of transmission and cache.

In a deadline-sensitive multimedia application, the segments will be delivered to many asynchronous users with asynchronous VCR-like operations (e.g., pause, forward, and rewind). The Peer-to-Peer (P2P) networking technology has been used as a powerful approaching for solving the scalable and asynchronous problems. In a P2P network, the users (named as peers), can act as both clients and servers. A peer downloads some segments (part of the file) for playing by itself, and then caches them to serve the future requests of other peers [1]. The P2P approach can provide cost-effective large-scale streaming media [2].

Vehicular Ad-Hoc Network (VANET), which is a special kind of Mobile Ad-Hoc Network (MANET), is a new generation of wireless network technology to provide communication between vehicles [3]. Designing an efficient P2P data dissemination for distributing deadline-sensitive streaming files in VANETs is a necessary requirement for a large number of VANET users. For the purpose, several challenges and problems need to be properly addressed.

1) It is critical to provide deadline-sensitive data dissemination.

In a deadline-sensitive multimedia application, the requirement of segments with a definite deadline usually is success delivery before the deadline, i.e., the segments requested by

The associate editor coordinating the review of this manuscript and approving it for publication was Martin Reisslein.

peers need to be received before their playback deadlines [4]. In order to provide the feature, the transmission duration of requesting and receiving segments between peers should be known in advance, and its value depends on one-hop delay and link connected duration between two neighboring hosts. The one-hop delay is the transmission duration between two neighboring hosts, and the link connected duration is the period of two neighboring hosts remaining connected. It is still difficult to know or accurately estimate the above two types of values in advance since a host in MANETs/VANETs shares the radio channel with its neighboring hosts.

Recently, several works [5]–[31] have been proposed for P2P applications in MANETs/VANETs. Since the problem of estimating the values is not properly solved, they cannot provide deadline-sensitive data dissemination for deadline-sensitive streaming applications. However, good progress has been made in estimating the link connection duration from the works [32]–[42]. On the premise that the vehicle speed follows the specific distribution such as uniform distribution, Gaussian distribution and normal distribution, the estimated value is calculated.

2) According to the characteristics of VANETs, A lookup function of collecting necessary information for VANET peer should be designed carefully.

The necessary information includes the above two kinds of estimates and the segment distribution (i.e., what peers are the segments stored in). The information is necessary for the peers in disseminating the deadline-sensitive segments. Many P2P lookup functions proposed in [43]–[45] have been proposed for the collection on the Internet. However, they are not applicable for VANETs since the high mobility of VANET hosts quickly will quickly reduce the accuracy of the collected information.

3) A function of segment requesting designed for VANET peers should efficiently utilize the limited bandwidth of VANETs.

Most of the previous works in [5]–[31] use all available network bandwidth to maximize the number of received segments as soon as possible. However, this method will waste the limited bandwidth of VANETs when transmitting and storing the segments whose playback deadline is far from reaching.

In this paper, we adopt the distinct strategies of addressing the challenges and problems.

1) A distinct lookup function is designed for VANET peers in collecting the necessary information and requesting segments simultaneously.

The previous works [5]–[31] mainly use lookup functions by broadcasting control packets over the whole network to maintain and collect the information of segment distribution. There is no doubt that the lookup function will result in a huge overhead of exchanging messages to VANETs in collecting and maintaining the information. On the other hand, due to the high mobility of hosts in VANETs, the accuracy of the information decreases rapidly.

Additionally, if a requested peer cannot support deadline-sensitive segment request to a requesting peer, the requesting peer is not necessary to waste the overhead of maintaining and collecting the information of the requested peer. Based on the consideration, a distinct lookup function is designed by just maintaining and collecting the part information of segment distribution, where the requested peers can provide deadline-sensitive segment request to the requesting peers with high possibility. On the other hand, it is executed simultaneously with the function of segment requesting. This approach will greatly reduce network overhead and take advantage of the timeliness of the information collected.

2) In order to effectively utilize the limited bandwidth of VANETs, a distinct scheduling function is designed to schedule the segments sent to VANETs according to the playback rate of the service, and it is executed irregularly.

Different from the strategy of the previous works in [5]–[31], the designed function cannot only meet the needs of the playback service, but also save the bandwidth and storage in transmitting and caching the segments whose playback deadlines are far from being met.

3) In order to provide deadline-sensitive segment dissemination and reduce the inaccuracy of estimation of one-hop delay and link connected duration, the method of interaction between the estimated and actual historical values of one-hop delay and link connected duration is adopted.
4) A route recovery function is designed to recover a disconnected route timely while replying the requested segments to the requesting peer.

In this paper, the estimated values of the one-hop delay and the link connected duration are computed. Further, the actual historical values are obtained when the peers received the requested segments recently, and they will be used in requesting the subsequent segments. The estimated values will be used when the actual historical values are unknown.

1) The trade-off between the playing quality of the service and the capacity of the network can be made by setting a system parameter proposed. In addition, this setting also alleviates the problem of host mobility.

The remainder of the paper is organized as follows. In Section II, the related works are first reviewed. The methods of estimated one-hop delay and link connected duration are reviewed in Section III and Section IV, respectively. The detailed designed algorithms are given in Section V. In Section VI, the simulations are carried out. Finally, this paper concludes with some remarks in Section VII.

II. RELATED WORKS

A. DATA SHARING ALGORITHMS IN MANETs

In [5], a layered video monitoring coding method is proposed to improve the video quality in MANETs. In [6], [7], the mechanisms of broadcasting/multicasting for streaming media services to MANET users are proposed. However, the issue of asynchronous users in P2P applications is difficultly solved by repeated broadcasting/multicasting. In [8], the authors propose a middleware of supporting secure access for multimedia applications and cooperative services.

In [9], the best host is selected to store the shared information by studying the performance impact of different wireless MAC layers in information caching. In [10], a protocol for P2P resource sharing is proposed by using the location of peers in order to reduce message overhead in a wireless mesh network. In [11], a distributed algorithm is proposed to solve the scheduling problem of multiple senders in mobile P2P networks. Its purpose is to maximize the data rate of the service and minimize the power consumption of the network. In [12], a mechanism is proposed by selecting stable hosts as coordinators for file searching. In [13], a distributed protocol of data file replication is proposed by considering host storage.

In [14], a mechanism is designed to support a large number of parallel streaming phases in wireless access networks. It is used to stabilize the user's playback buffer near some target values under the dynamic network conditions due to user mobility and fading effects. In [15], a low complexity scalable video streaming system is proposed based on model-based rate adaptive algorithm in multihop wireless networks. Then, a quantifiable measure of end-to-end quality of service is proved as a function of link quality, and it can be converted into useful quality of experience metrics for video playback.

In [16], the paper introduces a new method of user cooperation in wireless video multicasting by using random distributed space-time code. After receiving all source packets, the sender and receiver use the code to generate and send parity check packets at the same time. As more parity packets are sent, more receivers can recover all source packets and add parity packets to the transmission. In [17], a cache placement strategy is proposed in a two-layer wireless content delivery network, which uses different channels for content dissemination and content service. In the system model, the authors take the delay cost caused by contention as the key measure of cache placement.

In [18], the authors study the multi hop data dissemination from one data source to multiple nodes in wireless networks to minimize network power consumption and social cost. Most of these works [5]–[18] aim to utilize the limited bandwidth more efficiently for the files shared among MANET peers. However, they cannot support timely P2P data dissemination. In [19], a delay-sensitive segment scheduling algorithm is proposed for P2P real-time streaming services in MANETs. Since link disconnected duration and route disconnected time are not considered, it is not applicable to VANETs.

B. DATA SHARING ALGORITHMS IN VANETs

In the literature [20]–[31], several algorithms have been proposed for data shared among peers in VANETs. In [20], this paper studies how to use information centric network to transmit multimedia streaming in VANETs, with emphasis on the trade-off between experience quality and energy efficiency. A framework of multimedia streaming media centered on green information is designed to make the system develop towards the best working point in practical application.

In [21], this paper proposes a framework of anti-interference multi-path video streaming based on the optimal path of link and node disjoint. A method of interference perception video stream considering the statistics of regular driving is developed. Based on the packet error rate and the shadow effect of non-circular transmission range, the quality of video link is measured. In [22], the authors propose a protocol for multi-hop peer-communication with fairness guarantee in VANETs. A vehicle can relay its data to other vehicles in a P2P manner.

In [23], the packet loss rate in VANETs is minimized by optimizing the allocation of video packets on multiple routes under the condition that the freeze delay and the number of transmitted video packets are met. In [24], this paper proposes a routing protocol for the transmission of vehicles from the same starting point, the same route and the same destination on the highway. The purpose of routing protocol is to provide cooperative video streaming services for members who belong to multiple sources to single target transmission.

In [25], a cache invalidation mechanism is proposed for the VANET vehicles to distinguish between location-dependent and location-independent data. In [26], a scheme of P2P cooperative caching is proposed for minimizing the system load of the infrastructure and traffic information among vehicles. In [27], this paper proposes a collaborative download scheme for popular content distribution in VANETs. A cell-based clustering scheme, which uses the strategy of inter-cluster relay selection, constructs a P2P network to accelerate the dissemination process.

In [28], a distributed routing protocol is proposed to make routing decisions by considering sparse and dense environments in VANETs. In [29], this paper studies the scheduling problem of collaborative data distribution in VANETs with a hybrid infrastructure-to-vehicle and vehicle-to-vehicle communication environment. Its goal is to maximize the number of vehicles to retrieve the data they request.

In [30], the rapid dissemination of content in device-to-device vehicle-to-vehicle Internet of vehicle networks is studied by combining the physical layer and the social layer information. An iterative matching algorithm is proposed to solve the problems of joint peer discovery, power control and channel selection under different QoS requirements. In [31], this paper introduces a popular content distribution scheme, which broadcasts content from roadside access point and further distributes content among vehicles, in VANETs.

III. ESTIMATION OF ONE-HOP DELAY

The proposed method of estimating the one-hop delay (named as *ohd*) is similar to the one proposed in [46] by monitoring the busy/idle ratio of the channel. The value of *ohd* is quantitatively estimated as follows:

$$ohd = (E[cw] \times (1+B(t)) + E[p]) \times E[ta] \times E[q] \quad (1)$$

where t denotes the period of time slots, $E[cw]$ denotes the mean back-off time slots during every period of time slots, $E[p]$ denotes the mean time slots during every period of time slots engaging in transmitting a single packet, $E[ta]$ denotes the mean number of transmission during every period of time slots, $E[q]$ denotes the mean number of packets waiting in the MAC queue, and $B(t)$ denotes the busy/idle ratio of the channel during t. Then, $E[p]$ is estimated as follows:

$$E(p) = \frac{(l + p\_overhead) \times 8}{r \times 20 \times 10^{-6}} \quad (2)$$

where l denotes the mean number of packet size in bytes, *p_overhead* denotes the PHY/MAC overhead, r denotes the transmission data rate, and 20×10^{-6} is a slot time in seconds. And, $B(t)$ is estimated as follows:

$$B(t) = \frac{b\_slot + \beta}{i\_slot - \beta}, \quad \text{with } \beta = \delta \times t \times (20 \times 10^{-6}) \times m \quad (3)$$

where β is the increment number of busy time slots induced by a new flow, m is the mean number of time slots occupied by a MAC packet, δ is the packet arrival rate of the new flow. The numbers of busy and idle time slots are denoted as *b_slot* and *i_slot*, respectively.

IV. ESTIMATION OF LINK CONNECTED DURATION

With the satellite communication technology, the Global Positioning System (GPS) has been widely adopted for offering geographic location information in intelligent network usages. In VANETs, every vehicle is equipped with a GPS receiver, and its location information (including position, speed, and direction) can be obtained by the GPS. Then, it maintains the location information of its neighboring vehicles within one hop by constantly exchanging HELLO messages.

Let v_i and v_j be two neighboring vehicles, h be the radius of the transmission range, $(x_i, y_i)((x_j, y_j))$ be the location of v_i (v_j), s_i (s_j) be the speed of v_i (v_j), and θ_i (θ_j) be the moving direction of v_i (v_j) (where $0 \le \theta_I \le 2\pi$ and $0 \le \theta_j \le 2\pi$). Further let $a = v_i \cos\theta_i - v_j \cos\theta_j$, $b = x_i - x_j$, $c = v_i \sin\theta_i - v_j \sin\theta_j$, and $d = y_i - y_j$.

Based on the similar link duration prediction methods proposed in [32-34], the link connected duration between v_i and v_j, denoted as $lcd_{i,j}$, i.e., the time period that v_i and v_j will be connected, is estimated as follows:

$$lcd_{i,j} = \frac{-(ab + cd) + \sqrt{(a^2 + c^2) - (ad - bc)^2}}{a^2 + c^2} \quad (4)$$

Then, the time v_i and v_j disconnected is denoted as $ldt_{i,j}$ and estimated as $ldt_{i,j} = ct + lcd_{i,j}$, where ct is current time.

V. PROPOSED ALGORITHM

Let v_c be a new coming peer intending to request a P2P deadline-sensitive streaming service in VANETs, v_p s be the peers keeping some segments of the service, and $v_f s$ be the forwarders. The service is achieved by $v_f s$ forwarding the requests (the replies) from v_c to v_ps (from v_ps to v_c). In this section, the designed ideas and details of the algorithms executed by v_c, v_ps, and $v_f s$ for the service are proposed in Section V.A and Section V.B, respectively. Further, the algorithm of recovering a disconnected route while replying the requested segments to v_c is proposed in Section V.C. In Section V.D, the algorithm of determining the time points of v_c executing the algorithm of requesting segments is proposed.

A. DESIGN IDEAS OF PROPOSED ALGORITHM

The new coming peer v_c schedules the requested segments based on the following methods. First, v_c arranges the segments according to their playback deadlines. Second, it only schedules enough fragments to meet the playback rate of the service, thus saving the limited bandwidth in VANETs. Third, in order to avoid scheduling the segments that are likely to not meet their playback deadlines, two values are used to determine the scheduled segments. One is the one-hop delays of v_c and v_c's neighboring hosts, the other is the link disconnected time between v_c and v_c's neighboring hosts. The time points of v_c scheduling the requested segments are irregular, their determination will be shown in Section V.D.

When a forwarder v_f receives a request (a reply) from v_c (a peer), it excludes the segments that have ever been forwarded before. In addition, it also excludes the segment whose playback deadline is likely to not be met. For the purpose, a segment will be excluded to be forwarded to v_f's neighboring hosts if its playback deadline requirement are very likely not to be satisfied by the route delay and route connection duration from one of v_f's neighboring hosts to v_c. After the exclusion, v_f forwards the remaining segments to its neighboring hosts.

When a peer v_p receives a request from a forwarder, it excludes the segments that have been ever replied before and those whose playback deadlines requirement are very likely not to be satisfied by the route delay and route connection duration from v_p to v_c. If some of the remaining requested segments (i.e., they are not excluded) are kept by v_p, v_p replies them to v_c. Otherwise, v_p acts as a forwarder by forwarding the rest remaining requested segments not kept by v_p to v_p's neighboring hosts. If the other peers reply them to v_c, v_p can also receive and keep them.

By the aid of the exclusions executed in forwarders and peers, the requested segment are only forwarded within a limited region, where the peers have high probabilities of replying them timely to the requested peers. Besides, the exclusion also avoids wasting the bandwidth of the hosts outside the limited region. Furthermore, a peer forwards the requested segments it does not keep, which will be helpful in disseminating the segments quickly.

SEG_REQ:

1. set $R_c = \{\}$ and U be the set consisting of the un-scheduled segments;
2. **while** U is not empty and $|R_c| \times prs < \alpha \times prf$ **do**
3. determine segment s_h so that $dls_h = \min\{dls_j | s_j \in U\}$;
4. **if** there exists a neighboring host v_m of v_c such that $ct + ohd_c + ohd_m < dls_h$ and $ct + ohd_c + ohd_m < ldt_{c,m}$
5. **then** add (s_h, dls_h) into R_c;
6. **endif**
7. delete s_h from U;
8. **endwhile**
9. **if** R_c is not empty
10. **then** broadcast R_c, $F = \{v_c\}$ to the neighboring hosts of v_c;
11. **endif**

B. DETAIL OF PROPOSED ALGORITHM

Let prf be the playback-rate of a P2P deadline-sensitive streaming service is, prs be the playback-rate of a segment for the streaming service, and dls_j be the playback deadline of the jth segment. For the flexibility of readers, Table 1 lists all symbols used in Section V.

TABLE 1. Symbols used in Section V.

Symbol	Meanings
v_c	new coming peer
v_f	forwarder
v_p	peer
ohd_i	one-hop delay of host v_i
$ldt_{i,j}$	link disconnected time between hosts v_i and v_j
prf	playback-rate in bytes per second of the requesting streaming service
α	A system parameter to control the amount of the scheduled segments
prs	playback-rate in bytes per second of a segment
s_j	jth segment
dls_j	playback deadline of the jth segment
R_i	a set consists of the pairs of elements (s_j, dls_j) in v_i
ct	current time
F	a collection of the hosts for a route
OHD	a collection of ohd_is for a route
$RDCT$	a collection of $rdct_{i,j}$s for a route
rd	route delay
$rdct$	route disconnected time
S_j	document of jth segment
$\widetilde{R_p}$	a set consists of the pairs of elements (S_j, dls_j)

1) NEW COMING PEER

The new coming peer v_c executes the following algorithm, denoted as SEG_REQ, to determine a set (denoted as R_c), that consists of the pairs (s_j, dls_j), where s_j and dls_j are the jth segment and its playback deadline, respectively. Then, it broadcasts R_c and F to its neighboring hosts, where F is a collection of the hosts for a route to v_c.

In the algorithm, the while-loop is executed at satisfying both of the following two conditions. One is when there exists an un-scheduled segment, i.e., the set U is not empty. The other is that the scheduled amount of segments is enough to satisfy the playback-rate by judging that the rate of transmitting the scheduled segments is smaller than the playback-rate, i.e., $|R_c| \times prs < \alpha \times prf$. The parameter α is used to control the amount of the scheduled segments and its value is larger than 1. Larger α can schedule more segments to meet the requirement of the playback-rate by pre-fetching more segments.

In each iteration, a segment s_h, that has the smallest playback deadline and is not scheduled before, is determined. It will be added to the schedule if there exists one neighboring host v_m of v_c and the following two conditions are met. First, v_c can receive s_h from v_m before the playback deadline of s_h if v_m is a peer and v_m keeps s_h, i.e., $ct + ohd_c + ohd_m < dls_h$. Second, the one-hop link duration between v_c and v_m should be long enough to finish the request and reply. In other words, the link disconnected time $ldt_{c,m}$ between v_c and v_m should be later than the time of finishing the request and reply, i.e., $ct + ohd_c + ohd_m < ldt_{c,m}$.

For example, referring to Fig. 1, the new coming peer v_c requests 5 segments s_is, where $1 \leq i \leq 5$. Let dls_i be the playback deadline of s_i. Refer to Fig. 1(a), after the algorithm execution, v_c broadcasts the request by carrying ((s_2, dls_2), (s_3, dls_3), (s_4, dls_4), and (s_5, dls_5)) to its neighboring hosts v_{f_1}, v_{f_2} and v_{p_1}, where v_{f_1} and v_{f_2} are two forwarders, and v_{p_1} is a peer. The segment s_1 is excluded from the schedule since at least one of the two conditions in step 4 of SEG_REQ cannot be met.

2) FORWARDER

When a forwarder receives a request or a reply, it not only considers the one-hop delays and the link disconnected time between it and its neighboring hosts, but also considers the duration from the time its neighboring hosts replying the requested segments to the time the requesting peer receiving the replied segments.

Suppose that a forwarder v_f receives a request from its neighboring host v_m and the request carries six parameters R_m, F, OHD, $RDCT$, rd, and $rdct$ (explained later). Then, v_f executes the following algorithm, denoted as SEG_FORWARD, to determine the forwarding segments.

In the algorithm, OHD and $RDCT$ are two collections of ohd_is and $ldt_{i,j}$ of the hosts along the route F, respectively. On the other hand, $rdct$ and rd are the disconnected time and the delay of the route to v_c, respectively. If the request is received from the new coming peer, i.e., $v_n = v_c$, we set $rdct = 0$, $rd = 0$, $OHD = \{\}$ and $RDCT = \{\}$. When v_f receives the request from v_n, the values of $rdct$ and rd are the disconnected time and the delay of the route from v_m to v_c. Then, v_f computes $rdct = \min(rdct, ldt_{f,m})$ and $rd = rd + ohm_f$. The new values of $rdct$ and rd are the disconnected time and the delay of the route from v_f to v_c, respectively.

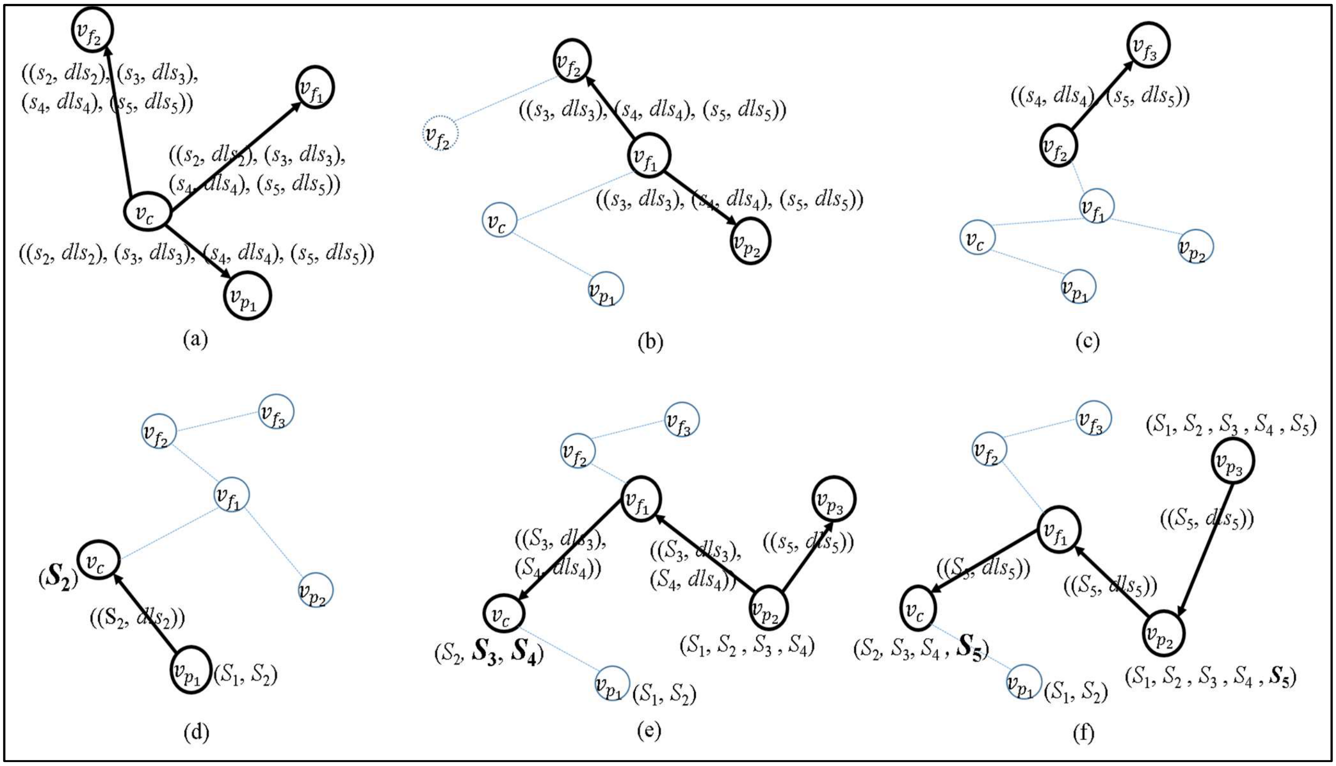


FIGURE 1. Segment requesting from v_c. (a) v_c broadcasts a request to v_{f_1}, v_{f_2} and v_{p_1}; (b) v_{f_1} forwards a request to v_{f_2} and v_{p_2}; (c) v_{f_2} forwards a request to v_{p_3}; (d) v_c receives S_2 from v_{p_1}; (e) v_{p_2} replies S_3, S_4 to v_c and forwards a request to v_{p_3}; (f) v_c and v_{p_2} receives S_5 from v_{p_3}.

SEG_FORWARD:

1. delete the segments that have been ever forwarded before from R_m;
2. set $R_f = \{\}$;
3. compute $rd = rd + ohm_f$;
4. compute $rdct = \min(rdct, ldt_{f,m})$;
5. **while** R_m is not empty **do**
6. select an element (s_h, dls_h) from R_m;
7. **if** there exists a neighboring host v_n of v_f $(v_m \neq v_n)$ such that $ct + ohd_f + ohd_n + rd < dls_h$,
$$ct + ohd_f + ohd_n + rd < ldt_{f,n},$$
$$ct + ohd_f + ohd_n + rd < rdct$$
8. **then** add (s_h, dls_h) into R_f;
9. **endif**
10. delete (s_h, dls_h) from R_m;
11. **endwhile**
12. **if** R_f is not empty
13. **then** add v_f into F;
14. add ohd_f into OHD;
15. add $ldt_{f,m}$ into $RDCT$;
16. broadcast R_f, F, OHD, $RDCT$, rd, $rdct$ to the neighboring hosts of v_f;
17. **endif**

In each iteration of the while-loop, a segment s_h selected from R_m will be added to R_f if there exists one neighboring host v_n of v_f and the following three conditions are met. First, v_f can receive s_h from v_n and transmits s_h to v_c before the playback deadline of s_h, i.e., $ct + ohd_f + ohd_n + rd < dls_h$. Second, the one-hop link duration between v_c and v_m should be long enough for finishing the request and reply, i.e., $ct + ohd_f + ohd_n + rd < ldt_{f,n}$. Third, there is sufficient time remaining for v_f to transmit s_h to v_c, i.e., $ct + ohd_f + ohd_n + rd < rdct$.

Refer to Fig. 1(b), v_{f_2} moves to a new position close to v_{f_1} and is a neighboring host of v_{f_1}. After the execution of SEG_FORWARD, v_{f_1} removes s_2 from the request received from v_c since s_2 violates at least one of the three conditions in step 7 of SEG_FORWARD, then v_{f_1} forwards $((s_3, dls_3), (s_4, dls_4), (s_5, dls_5))$ to its neighboring hosts v_{f_2} and thepeerv_{p_2}. On the other hand, v_{f_2} also receives the request from v_c as shown in Fig. 1(a). Since its high mobility to v_c, it removes all segments and stops forward the request. Refer to Fig. 1(c), v_{f_2} receives the forwarding request carrying $((s_3, dls_3), (s_4, dls_4), (s_5, dls_5))$ from v_{f_1} as shown in Fig. 1(b), but it only forwards $((s_4, dls_4)$ and $(s_5, dls_5))$ to theforwarderv_{f_3} since s_2 violates at least one of the three conditions in step 7 of SEG_FORWARD.

3) PEER

When a peer v_p receives a request from its neighboring host v_m, it executes the following algorithm, denoted as SEG_REPLY.

In the algorithm, a set $\tilde{R}_p$ that constains the pairs (S_j, dls_j) is used for v_p to reply the segments to the requesting peer, where S_j is the document of the jth segment. In each iteration of the

SEG_REPLY:

delete the segments that have been ever forwarded or replied before from R_m;
set $\tilde{R}_p = \{\}$ and $R_p = \{\}$;
compute $rd = rd + ohm_p$;
compute $rdct = \min(rdct, ldt_{p,m})$;
while R_m is not empty **do**
  select an element (s_h, dls_h) from R_m;
  if v_p keeps the hth segment S_h
    if $ct + rd < dls_h$ and $ct + rd < rdct$
    then add (S_h, dls_h) into $\tilde{R}_p$;
    endif
  else if there exists a neighboring host v_n of v_p $(v_m \neq v_n)$ such that ct+$ohd_p + ohd_n + rd < dls_h$, $ct + ohd_p + ohd_n < ldt_{p,n}$, $ct + ohd_p + ohd_n + rd < rdct$
    then add (s_h, dls_h) into R_p;
    endif
  endif
  delete (s_h, dls_h) from R_m;
endwhile
if $\tilde{R}_p$ is not empty
then delete v_m from F;
  transmit $\tilde{R}_p$, F to v_m;
endif
if R_p is not empty
then add v_p into F;
  add ohd_f into OHD;
  add $rdct_{f,m}$ into $RDCT$;
  broadcast R_p, F, OHD, $RDCT$, rd, $rdct$ to the neighboring hosts of v_p;
endif

while-loop, a segment s_h is selected from R_m. If v_p keeps s_h and the following two conditions are met, (S_h, dls_h) is added into $\tilde{R}_p$. First, v_p can transmit s_h to v_c by the route F before the playback deadline of s_h, i.e., $ct + rd < dls_h$. Second, the route disconnected time of F is after the time of finishing the transmission, i.e., $ct + rd < rdct$. On the other hand if v_p does not keep s_h, it acts as a forwarder by adding (S_h, dls_h) into R_p and then it forwards R_p to its neighboring hosts.

In Fig. 1(d), v_{p_1} receives the request carrying $((s_2, dls_2), (s_3, dls_3), (s_4, dls_4)$, and $(s_5, dls_5))$, from v_c as shown in Fig. 1(a). Since v_{p_1} keeps (S_1, S_2), it replies $((S_2, dls_2))$ to v_c. On the other hand in Fig. 1(e), v_{p_2} receives the request carrying $((s_3, dls_3), (s_4, dls_4)$, and $(s_5, dls_5))$ from v_{f_1} as shown in Fig. 1(b). Since v_{p_2} keeps (S_1, S_2, S_3, S_4), it replies $((S_3, dls_3), (S_4, dls_4))$ to v_c through v_{f_1} and forwards the request $((s_5, dls_5))$ to its neighboring host v_{p_3}. Refer to Fig. 1(f), the peer v_{p_3} replies $((S_5, dls_5))$ to v_c through v_{f_1} and v_{p_2}. Through the reply, v_{p_2} receives and keeps S_5.

C. RECOVERY OF THE ROUTES

Recall that the method mentioned in Section IV, each host v_i piggybacks its location and speed, provided by its GPS, onto a control packet. Then, v_i forwards the packet to its neighbors periodically. Upon receiving the message, a neighbor v_j of v_i can derive $ldt_{i,j}$ from the piggyback information. In order to recover the routes for transmitting the segments to the requesting peer, v_i further piggybacks its ohd_i, $ldt_{i,k}$s and ohd_ks onto the packet, where v_ks are the neighboring hosts of v_i.

On the other hand in the method of Section V.B, when each host v_i receives a request from another host v_j (where v_i and v_j are peers or forwarders), v_i piggybacks its ohd_i and $ldt_{i,j}$ onto the packets. By using the piggybacked information, it is helpful for replacing a disconnected link with a connected one.

Suppose that v_p - ... - v_{f_a} - v_{f_b} - v_{f_c} ... - v_{f_d} - v_c is a route from v_p to v_c. Once v_{f_a} receives a reply carrying S, F, OHD and $RDCT$, from its neighboring host and detects that the link to the next forwarder v_{f_b} is disconnected, v_{f_a} executes the following method to determine a new forwarder for replacing v_{f_b}.

Let D be the set of the hosts v_ms, where v_ms are the neighboring hosts of v_{f_a} and v_{f_c} is a neighboring host of v_ms. If D is not empty, a new forwarder $v_{f_m} \in D$, that can transmit the maximum number of the segments in S, is determined. If v_{f_m} can transmit a segment $s_h \in S$, the following two conditions should be met. First, v_c needs to receive s_h from v_{f_a} via v_{f_m} before the playback deadline of s_h, i.e., $ct + \check{rd} < dls_h$, where $\check{rd} = \sum_{v_i \in \check{F}} ohd_{v_i}$ is the route delay from v_{f_c} to v_c and $\check{F} = \{v_{f_c}, v_{f_m}, \ldots, v_{f_c}, v_{f_d}\}$. Second, there is sufficient time remaining for v_f to transmit s_h to v_c via v_{f_m}, i.e., ct+$\check{rd} < \overset{\frown}{rdct}$, where $\overset{\frown}{rdct} = \min_{v_i \in F} dct_{v_i}$.

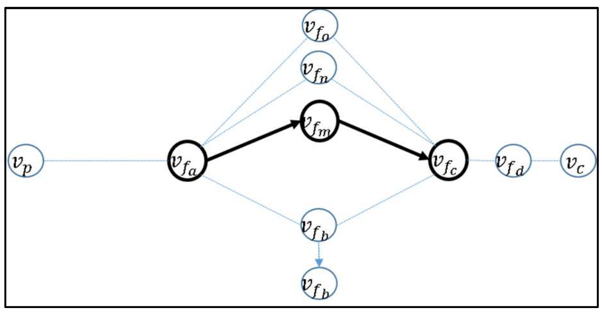


FIGURE 2. Route recovery by replacing$v_{f_a} - v_{f_b} - v_{f_c}$ with $v_{f_a} - v_{f_m} - v_{f_c}$.

An illustrative example is shown in Fig. 2. Suppose that v_{f_b} moves away and hence v_{f_a} becomes disconnected with v_{f_c}. Then, by the aid of the above method, v_{f_a} can determine a new forwarder v_{f_m}, which is selected from v_{f_m},v_{f_n} and $v_{f_{mo}}$. Thus, the disconnection is recovered by replacing $v_{f_a} - v_{f_b} - v_{f_c}$ with $v_{f_a} - v_{f_m} - v_{f_c}$.

D. TIME POINTS OF EXECUTING SEG_REQ

How to determine the next time point and frequencies of executing the algorithm SEG_REQ proposed in Section V.B for a new coming peer to request the subsequent segments is a compromise between the playing quality of the service and the utilization of the limited bandwidth in VANETs. A higher frequency induces an earlier next time point, and the new coming peer obtains a better quality. The result comes from that a higher frequency and an earlier next time point make the requested segments received early. However, it will waste the limited bandwidth of VANETs and the buffer of peers to transmit and store the segments that are far from the deadlines of playback.

In this section, a method of determining the next time point and frequencies of executing SEG_REQ is proposed for the compromise. In the proposed method, the history information of the actual route delays and route disconnected time obtained in the previous executing SEG_REQ is used to adjust the determined values according to the existing bandwidth conditions in VANETs. Further, since the replying peers can reply the requested segments in the previous executing SEG_REQ, there is a high possibility of obtaining the subsequent segments form them. Therefore, they are very suitable for candidates with a high reference value in the next executing SEG_REQ.

Let m segments be scheduled in the nth execution of SEG_REQ, and l segments be received while forwarding the l segments for other peers. Recall that the nth execution of SEG_REQ is terminated at meeting that all segments have been scheduled, or the playback-rate requirement of the service is satisfied by the amount of the scheduled segments. Therefore, the $(m+1)$th to $2m$th segments will be potential candidates for the $(n+1)$th scheduling to continuously meet the playback rate, and their playback deadline will be used to determine the time point of the $(n+1)$th execution of SEG_REQ. That is to say, the $(m+1)$th to $2m$th segments are regarded as the scheduling window for the $(n+1)$ execution of SEG_REQ. Refer to Fig. 3.

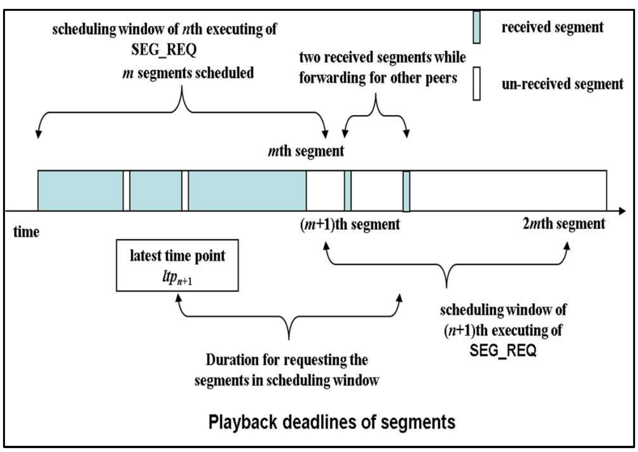


FIGURE 3. Latest time point of (n+1)*th* execution of SEG_REQ.

According to the example in Fig. 3, the latest time point for the $(n+1)$th execution of SEG_REQ depends on the duration for requesting the segments in the scheduling window, and it is denoted as ltp_{n+1}. Let W_{n+1} be the set consisting of the $(m+1)$th to $2m$th segments in the scheduling window of the $(n+1)$th execution of SEG_REQ, and Y be the set consisting of the l received segments. Then, ltp_{n+1} can be computed as follows.

$$ltp_{n+1} = \min_{s_j \in (w_{n+1} - Y)} ltps_j \tag{5}$$

where $ltps_j$ is the latest time point of requesting the segment s_j. The segment s_j should be requested before $ltps_j$, otherwise it will not be received before its playback deadline dls_j. Suppose that the peer v_{p_i} keeps s_j, its actual request delay of transmitting the request from the requesting peer to v_{p_i} is req_i, its actual reply delay of replying the request is rep_i, and the actual disconnected time of the route is $ardct_j$.

If v_{p_i} will be a candidate peer for requesting s_j in the $(n+1)$th execution of SEG_REQ, the requesting and replying for s_j should be finished before playback deadline dls_j and the actual disconnected time $ardct_i$. Thus, the latest time point (denoted as $ltpps_{i,j}$) of requesting the segment s_j from the peer v_{p_i} can be computed as follows.

$$ltpps_{i,j} = \min(dls_j, ardct_i) - (req_j + rep_j) \tag{6}$$

Since s_j may be kept by more than one peer, selecting a peer v_{p_i} with earlier or later $ltpps_{i,j}$ for determining $ltps_j$ is needed to be considered. Due to the high mobility of hosts in VANETs, the correctness of the history information will become lower and lower with time. Therefore, the history information can be fully utilized by selecting the peer v_{p_i} with the earliest $ltpps_{i,j}$, and the sooner s_j can be received. Thus, $ltps_j$ is computed as following, where K is a set consisting of the peers keeping s_j.

$$\text{ltps}_j = \min_{v_{p_i} \in K} ltpps_{i,j} \tag{7}$$

VI. SIMULATION

In this section, the performance of the proposed algorithm is evaluated by using the extended simulation implemented by the Network Simulator 2 package (NS-2, version 2.29) [47]. To the best of our knowledge, it presents the first work of P2P deadline-sensitive streaming for timely data dissemination in VANETs and therefore named as TDDV for convenience. In [19], a delay-sensitive segment scheduling algorithm (denoted as DSSSA) is proposed for P2P deadline-sensitive streaming services in MANETs, and we take as a baseline for performance comparison.

The differences of TDDV from DSSSA are stated as follows:

1) In TDDV, a distinct lookup function is designed to collect the necessary information and to request segments simultaneously within a limited region, whereas, DSSSA collects the necessary information over the entire network by using Chord [43] as the P2P lookup protocol and the delay-sensitive routing protocol proposed in [46] as the routing protocol.
2) TDDV estimates the delay and the connected duration of links and routes for achieving timely segment request.

It interactively uses the estimated values and the actual historical values to alleviate the problem of host mobility in VANETs. Additionally, a route recovery mechanism is designed to solve the problem.

In the simulation, 100 users are randomly distributed over a 1000 m × 1000 m area. The IEEE 802.11b is used as MAC/PHY protocol. The mobility model generator MOVE [48] is used to provide the mobility trace file containing information of realistic host movements. The maximum speed of hosts varies from 5 to 30 meters per second, and the realistic host speed is between 0 and the maximum value. In order to highlight the problem of host mobility in VANETs, the number of the hosts is increased to 100, and the speed of the hosts varies from 5 to 30 meters per second. Each host is equipped with a GPS receiver to provide its location and speed. The interval for each host to broadcast a control packet carrying its location and speed is 1.5 second.

In addition, 100 peers are generated with an arrival rate of 20 requests per minute, and they are randomly hosted on 100 hosts. All simulation runs use a special video stream whose streaming content is a 30 minute movie with an accurate rate control packet to generate a constant bit rate of 200 Kbps. The following three indices are measured: number of control bytes per second, successful receiving rate, and admission rate. The number of control bytes per second is used to reflect the overhead of requesting segments and maintaining/collecting segment distribution information. Successful receiving rate is the ratio of the number of requested segments received in time per second to the number of requested segments per second. Admission rate is the ratio of the number of admitted peers to the number of requesting peers. If the segments received per second can meet the playback rate, peers are admitted.

The simulations are performed in three aspects. First, simulations are executed under the static hosts. Second, the same simulations are executed for mobile hosts. In the aforementioned simulations, the value of the system parameter α is set 1.1. Third, the performance of TDDV is investigated by assigning different values of α. The thirty runs with different seed numbers are made for each scenario, and the data obtained by these runs are averaged.

A. STATIC HOSTS

In Fig. 4, the numbers of the control bytes per second generated by requesting peers varied from 25 to 100 peers. The simulation result of Fig. 4 shows TDDV generates fewer control bytes per second than DSSSA. The more control bytes in DSSSA are generated by executing the lookup protocol and the delay-sensitive routing protocol over the whole network, whereas the fewer control bytes in TDDV are caused by that the procedures of collecting the necessary information and requesting segments are executed simultaneously and are confined to a limited region.

In Fig. 5, the admission rate versus the number of requesting peers is demonstrated. Refer to Fig. 5. When the number of requesting peers is smaller than 36, DSSSA has higher admission rates than TDDV. It is a consequence that the lookup protocol and the delay-sensitive routing protocol are more helpful than the procedures proposed in TDDV for obtaining the information of the segment dissemination when the peers are rare.

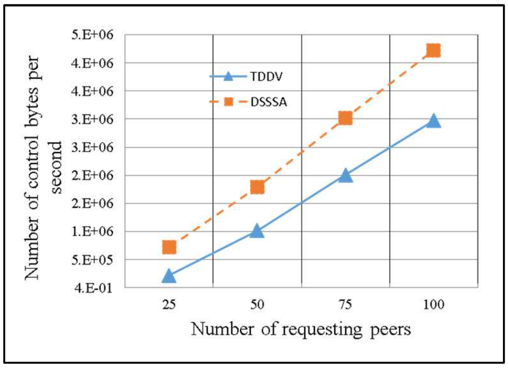


FIGURE 4. Number of control bytes per second versus a number of requesting peers.

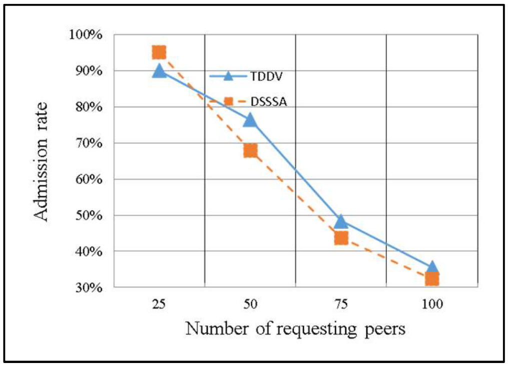


FIGURE 5. Admission rate versus a number of requesting peers.

On the other hand, when the number of requesting peers exceeds 36, DSSSA has lower admission rates than TDDV. The reason is that more peers increase the probabilities that TDDV finds and requests segments within the limited region, and the higher admission rates of TDDV is a consequence that few extra control bytes generated makes the network to admit more peers.

In Fig. 6, the average successful receiving rate versus the number of admitted peers is demonstrated. The values of the average successful receiving rate are averaged from the values of the successful receiving rates of the admitted peers. The average success rates (more than 95.8%) of TDDV and DSSSA are almost equal and high, which show that TDDV and DSSSA are effective in providing deadline sensitive segment requests. On the other hand, the differences (as mentioned above) of TDDV from DSSSA result in more admitted peers in TDDV, where TDDV and DSSSA admit 39 and 33 peers, respectively. In Fig. 7, an example of the successful receiving rate versus each individual admitted peer is demonstrated.

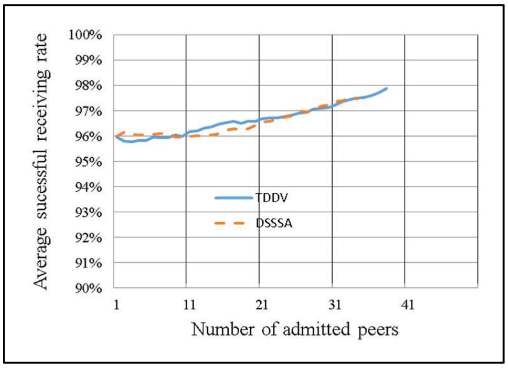
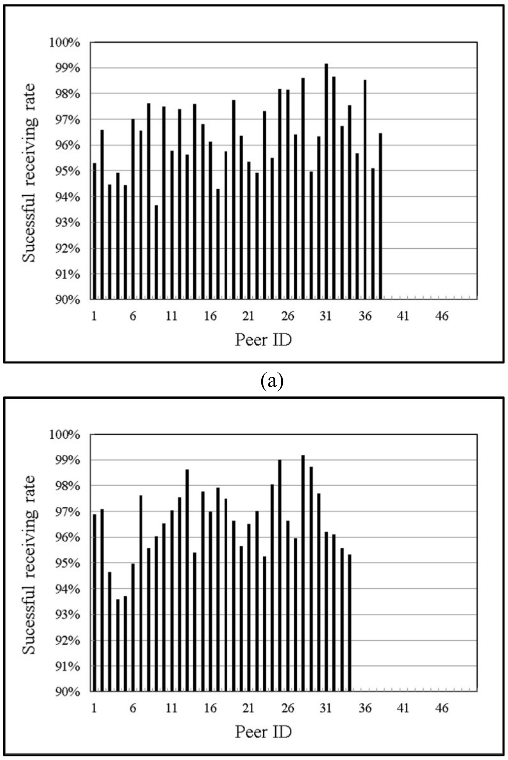


FIGURE 6. Average successful receiving rate versus a number of admitted peers.



(b)

FIGURE 7. Successful receiving rate versus peer ID. (a) TDDV; (b) DSSSA.

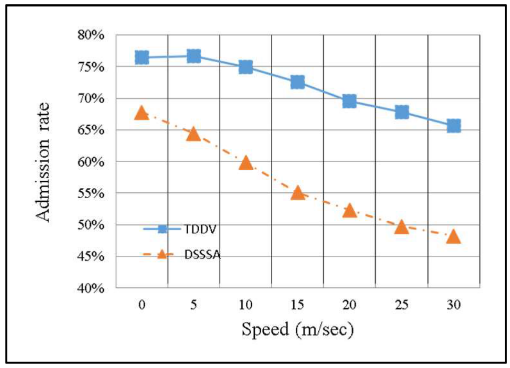


FIGURE 8. Admission rate versus host speed for 50 requesting peers.

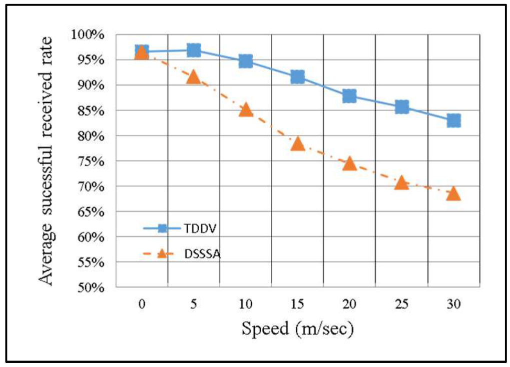


FIGURE 9. Average successful receiving rate versus number for 50 requesting peers.

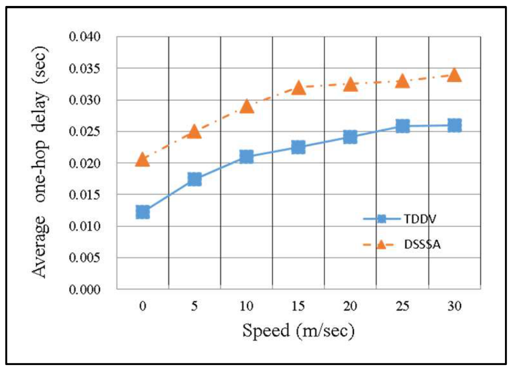


FIGURE 10. Average one-hop delay for 50 requesting peers.

B. MOBILE HOSTS

Fig. 8 and Fig. 9 demonstrate the impact of host mobility on admission rates and average successful receiving rate, respectively. There are 50 peers randomly hosted on the 100 hosts. When the host speed increases, the rates for TDDV and DSSSA decline, but TDDV is superior to DSSSA. The reason is explained in four aspects.

First, the correctness of information collected by the lookup protocol and the delay-sensitive routing protocol in DSSSA decreases rapidly with increasing speed of hosts. Whereas TDDV executes the procedures of collecting information and requesting segments at the same time, its information is more correct and timely. Second, the values of estimating link connected duration is used in TDDV to determine the routes with connection duration for meeting the timely requirements of requested segments.

Third, the quality of the route obtained in TDDV is better than that obtained in DSSSA. It is derived from that the limited region causes a short route with short route delay and long route connection duration. In order to verify the above statement, Fig. 10, Fig. 11, and Fig. 12, are used to show average one-hop delays, average route length, and average route delay. The simulation results of the figures show that TDDV is superior to DSSSA in the three indices.

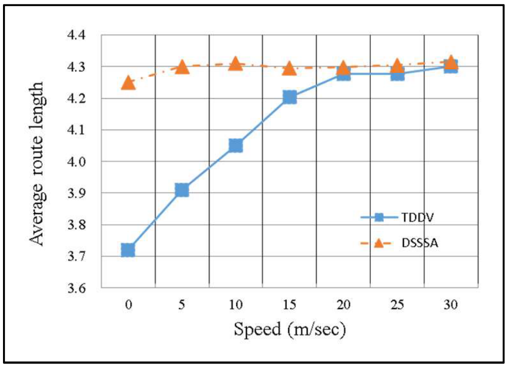


FIGURE 11. Average route length for 50 requesting peers.

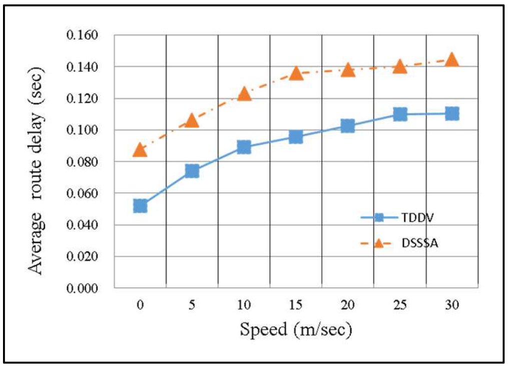


FIGURE 12. Average route delay for 50 requesting peers.

Recall that a forwarder does not forward the requested segments that have been ever forwarded before by the method of Section V.B. For the same requested segments received by the forwarder at a later time, it means that they have longer route delay from the forwarder to the requesting peer. If there are multiple routes from the forwarder to the requesting peer, the one with the least route delay will be selected in TDDV. A route with less route delay is an immediate consequence of less one-hop delay and shorter route length.

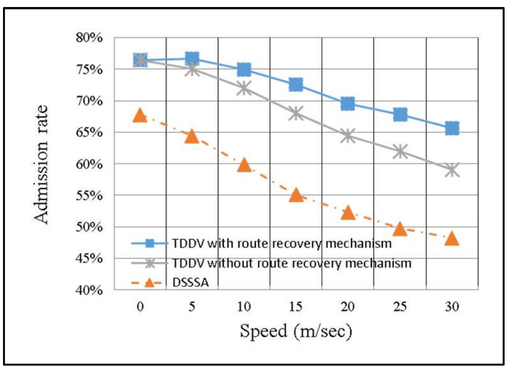


FIGURE 13. Admission rate versus host speed for 50 requesting peers.

Fourth, the route recovery mechanism further enhances TDDV to solve the problem of host mobility. In Fig. 13 and Fig. 14, the simulations are set with similar background patterns as those in Fig. 8 and Fig. 9, in which the difference is an added curve under TDDV without the route recovery mechanism. The simulation results of Fig. 13 and Fig. 14 validate the claim.

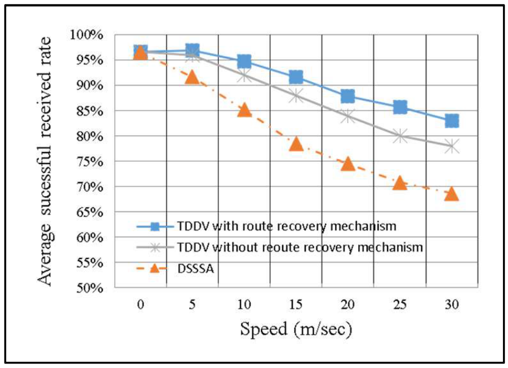


FIGURE 14. Average successful receiving rate versus number for 50 requesting peers.

C. PERFORMANCE COMPARISON BASED ON DIVERSE α

Recall that the method mentioned in Section V.B, the parameter α is a number larger than 1 and used to determine the number (i.e., $\alpha \times prf$) of requested segments. When the value of α gets greater, TDDV will select more requested segments transmitted over the network. That induces that the successful received rates of the requesting peers increase, but the admission rates of the network decrease since the more requested segments decrease the capacity of the network.

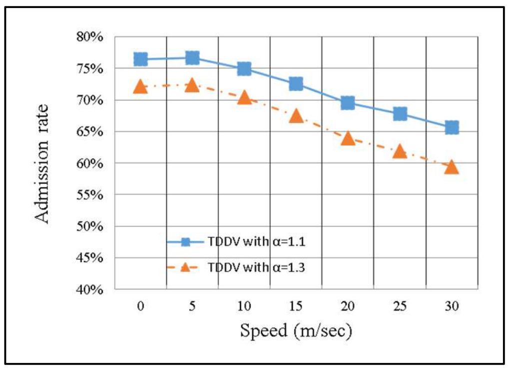


FIGURE 15. Admission rate versus host speed for 50 requesting peers under varied α.

On the other hand, the more requested segments caused by greater α helps to alleviate the problem of host mobility. Hence, as α increases, the quality of the service upgrades but the capacity of the network declines. The claim is validated by Fig. 15 and Fig. 16, where the simulations are set with similar background patterns as those in Fig. 8 and Fig. 9. The difference is an added curve with $\alpha = 1.3$ when TDDV is executed.

In summary, the determined value of α can be used as a decision parameter between the quality of the service and the capacity of the network. Its value also can be determined

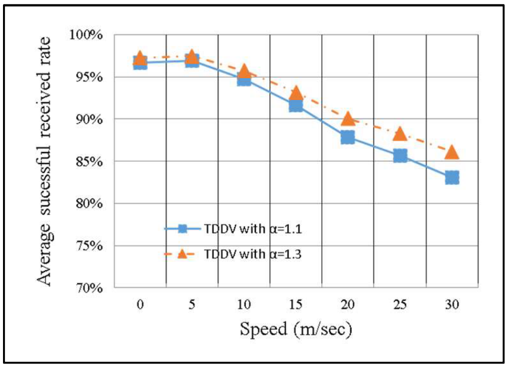


FIGURE 16. Average successful receiving rate versus number for 50 requesting peers under varied α.

based on the tolerable data loss rate of the service. If the tolerable data loss rate is high or the speed of the host is slow, the value of α can be lowered. Otherwise, a larger value is necessary.

VII. CONCLUSION

In this paper, a mechanism (named TDDV) of disseminating timely P2P segments is proposed for efficiently distributing deadline-sensitive streaming files in VANETs. TDDV addresses the issue by adopting different strategies from the previous works, as follows. First, it collects the part information of segment distribution and is executed simultaneously with the function of segment requesting. Second, it irregularly schedules the segments disseminated to peers according to the playback rate of the service. Third, it interactively uses the estimated and actual historical values of one-hop delay and the link connected duration. Fourth, it recovers the disconnected route timely when the requested segment is replied to the requesting peer. Fifth, it sets a system parameter to balance the playing quality of the service and the capacity of the network, and to alleviate the problem of host mobility.

Simulation results show that TDDV is superior to the other mechanism in terms of the number of control bytes per second, admission rate, successful receiving rate, one-hop delay, route length, and average route delay. Recall that the estimated values of link connected duration is calculated based on the assumptions that vehicle speeds follow some specific distributions. Thus, our one future work is to study the different effects of different distributions on TDDV, and refine TDDV accordingly.

REFERENCES

[1] E. Kurdoglu, Y. Liu, and Y. Wang, "Dealing with user heterogeneity in P2P multi-party video conferencing: Layered distribution versus partitioned simulcast," *IEEE Trans. Multimedia*, vol. 18, no. 1, pp. 90–101, Jan. 2016.
[2] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling xunlei kankan: Understanding hybrid CDN-P2P video-on-demand streaming," *IEEE Trans. Multimedia*, vol. 17, no. 2, pp. 229–242, Feb. 2015.
[3] K. Z. Ghafoor, "Video streaming over vehicular ad hoc networks: A comparative study and future perspectives," *ARO-Sci. J. Koya Univ.*, vol. 4, no. 2, pp. 25–36, 2016.
[4] J. Hu, C. Chen, T. Qiu, and Q. Pei, "Regional-centralized content dissemination for eV2X services in 5G mmWave-enabled IoVs," *IEEE Internet Things J.*, early access, Mar. 24, 2020.
[5] M. Qin and R. Zimmermann, "Improving mobile ad-hoc streaming performance through adaptive layer selection with scalable video coding," in *Proc. 15th ACM Int. Conf. Multimedia*, vol. 2007, pp. 717–726.
[6] H. Jenkac, T. Stockhammer, and W. Xu, "Asynchronous and reliable on-demand media broadcast," *IEEE Netw.*, vol. 20, no. 2, pp. 14–20, Mar. 2006.
[7] M.-F. Leung and S.-H.-G. Chan, "Broadcast-based Peer-to-Peer collaborative video streaming among mobiles," *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 350–361, Mar. 2007.
[8] D. Macedo, A. Dos Santos, J. S. Nogueira, and G. Pujolle, "A distributed information repository for autonomic context-aware MANETs," *IEEE Trans. Netw. Service Manage.*, vol. 6, no. 1, pp. 45–55, Mar. 2009.
[9] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless P2P networks: Design, implementation, and evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 2, pp. 229–241, Feb. 2010.
[10] 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.
[11] P. Si, F. Yu, H. Ji, and V. Leung, "Distributed sender scheduling for multimedia transmission in wireless mobile peer-to-peer networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4594–4603, Sep. 2009.
[12] K. Chen, H. Shen, and H. Zhang, "Leveraging social networks for P2P content-based file sharing in disconnected MANETs," *IEEE Trans. Mobile Comput.*, vol. 13, no. 2, pp. 235–249, Feb. 2014.
[13] K. Chen and H. Shen, "Maximizing P2P file access availability in mobile ad hoc networks though replication for efficient file sharing," *IEEE Trans. Comput.*, vol. 64, no. 4, pp. 1029–1042, Apr. 2015.
[14] K. Miller, D. Bethanabhotla, G. Caire, and A. Wolisz, "A control-theoretic approach to adaptive video streaming in dense wireless networks," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1309–1322, Aug. 2015.
[15] H. Al-Zubaidy, V. Fodor, G. Dan, and M. Flierl, "Reliable video streaming with strict playout deadline in multihop wireless networks," *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2238–2251, Oct. 2017.
[16] Z. Guo, Y. Wang, E. Erkip, and S. S. Panwar, "Wireless video multicast with cooperative and incremental transmission of parity packets," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1335–1346, Aug. 2015.
[17] J. Sung, M. Kim, K. Lim, and J.-K.-K. Rhee, "Efficient cache placement strategy in two-tier wireless content delivery network," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 1163–1174, Jun. 2016.
[18] M. Mousavi and A. Klein, "Energy and social cost minimization for data dissemination in wireless networks: Centralized and decentralized approaches," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5521–5534, May 2020.
[19] C. C. Hu, C. F. Lai, J. G. Hou, and Y. Min Huang, "Delay-sensitive segment scheduling algorithm for timely P2P streaming over MANETs," *Comput. Netw.*, vol. 127, pp. 56–67, 2017.
[20] C. Q. Xu, W. Quan, H. K. Zhang, and L. A. Grieco, "GrIMS: Green information-centric multimedia streaming framework in vehicular ad hoc networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 2, pp. 483–498, Feb. 2018.
[21] A. Aliyu, A. H. Abdullah, N. Aslam, A. Altameem, R. Z. Radzi, R. Kharel, M. Mahmud, S. Prakash, and U. M. Joda, "Interference-aware multipath video streaming in vehicular environments," *IEEE Access*, vol. 6, pp. 47610–47626, 2018.
[22] K. Yang, S. Ou, H.-H. Chen, and J. He, "A multihop peer-communication protocol with fairness guarantee for IEEE 802.16-based vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3358–3370, Nov. 2007.
[23] M. A. Salkuyeh and B. Abolhassani, "Optimal video packet distribution in multipath routing for urban VANETs," *J. Commun. Netw.*, vol. 20, no. 2, pp. 198–206, Apr. 2018.
[24] C.-M. Huang, T.-H. Lin, and K.-C. Tseng, "Data dissemination of application service by using member-centric routing protocol in a platoon of Internet of vehicle (IoV)," *IEEE Access*, vol. 7, pp. 127713–127727, 2019.
[25] A. Singh, N. Chand, and L. Awasthi, "Cache invalidation for location dependent and independent data in IVANETs," in *Proc. Int. Conf. Adv. Comput. Commun.*, 2011, pp. 609–614.
[26] N. Kumar and J.-H. Lee, "Peer-to-Peer cooperative caching for data dissemination in urban vehicular communications," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1136–1144, Dec. 2014.

[27] W. Huang and L. Wang, "ECDS: Efficient collaborative downloading scheme for popular content distribution in urban vehicular networks," *Comput. Netw.*, vol. 101, pp. 90–103, Jun. 2016.
[28] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, "Delay-aware grid-based geographic routing in urban VANETs: A backbone approach," *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2324–2337, Dec. 2019.
[29] K. Liu, J. K. Y. Ng, V. C. S. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a software defined network," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1759–1773, Jun. 2016.
[30] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social Big-Data-Based content dissemination in Internet of vehicles," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 768–777, Feb. 2018.
[31] M. Li, Z. Yang, and W. Lou, "CodeOn: Cooperative popular content distribution for vehicular networks using symbol level network coding," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 223–235, Jan. 2011.
[32] S.-S. Wang and Y.-S. Lin, "PassCAR: A passive clustering aided routing protocol for vehicular ad hoc networks," *Comput. Commun.*, vol. 36, no. 2, pp. 170–179, Jan. 2013.
[33] Y. Li, M. Zhao, and W. Wang, "Internode mobility correlation for group detection and analysis in VANETs," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4590–4601, Nov. 2013.
[34] E. Souza, I. Nikolaidis, and P. Gburzynski, "A new aggregate local mobility (ALM) clustering algorithm for VANETs," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–5.
[35] B. Hassanabadi, C. Shea, L. Zhang, and S. Valaee, "Clustering in vehicular ad hoc networks using affinity propagation," *Ad Hoc Netw.*, vol. 13, pp. 535–548, Feb. 2014.
[36] S. Shelly and A. V. Babu, "Probability distribution of link life time in vehicular ad hoc networks," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2013, pp. 1418–1423.
[37] X. Wang, C. Wang, G. Cui, and Q. Yang, "Practical link duration prediction model in vehicular ad hoc networks," *Int. J. Distrib. Sensor Netw.*, vol. 2015, no. 3, 2015, Art. no. 216934.
[38] X. Wang, C. Wang, G. Cui, Q. Yang, and X. Zhang, "ELDP: Extended link duration prediction model for vehicular networks," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 4, Apr. 2016, Art. no. 5767569.
[39] J. Zhang, M. Y. Ren, and H. Labiod, "Performance evaluation of link metrics in vehicle networks: A study from the cologne case," in *Proc. 6th ACM Symp. Develop. Anal. Intell. Veh. Netw. Appl.*, vol. 2016, pp. 123–130.
[40] S.-J. Lee, W. Su, and M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks," *Mobile Netw. Appl.*, vol. 7, no. 6, pp. 441–453, 2002.
[41] C. C. Hu, E. H. K. Wu, and G. H. Chen, "Stable backbone hosts and stable multicast routes in two-tier mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5020–5036, Nov. 2009.
[42] H. Yang, M. Yu, and X. Zeng, "Link available time prediction based GPSR for vehicular ad hoc networks," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 293–298.
[43] I. Stoic, R. Morris, D. R. Karger, M. F. Kaashoek, 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.
[44] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Int. Conf. Distrib. Syst. Platforms Open Distrib. Process.*, Berlin, Germany, 2001, pp. 329–350.
[45] Q. Luo, J. Cheng, Y. Guo, and C. Huang, "An optimization strategy of MANET based on chord algorithm," *Comput. Technol. Develop.*, vol. 21, no. 11, pp. 42–45, 2011.
[46] C.-C. Hu, "Delay-sensitive routing in multi-rate MANETs," *J. Parallel Distrib. Comput.*, vol. 71, no. 1, pp. 53–61, Jan. 2011.
[47] *Network Simulator (Version 2)*. Accessed: Jul. 2012. [Online]. Available: http://www-mash.cs.berkeley.edu/ns/
[48] F. K. Karnadi, Z. H. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for VANET," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 2506–2511.

CHIA-CHENG HU (Member, IEEE) received the M.S. degree in engineer science from National Cheng Kung University, in 1995, and the Ph.D. degree from the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, in 2005. He is currently a Professor with the College of Artificial Intelligence, Yango University.