

## Research Article

# A Novel Video Sharing Solution Based on Demand-Aware Resource Caching Optimization in Wireless Mobile Networks

Shijie Jia,<sup>1,2</sup> Ruiling Zhang,<sup>2</sup> Shengli Jiang,<sup>2</sup> and Mingchuan Zhang<sup>3</sup>

<sup>1</sup>Central Plains Economic Zone Wisdom Tourism Cooperative Innovation Center, Luoyang Normal University, Luoyang, Henan 471934, China

<sup>2</sup>Academy of Information Technology, Luoyang Normal University, Luoyang, Henan 471934, China

<sup>3</sup>Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Ruiling Zhang; [ruilingzhang@163.com](mailto:ruilingzhang@163.com)

Received 29 October 2016; Accepted 22 December 2016; Published 19 January 2017

Academic Editor: George Ghinea

Copyright © 2017 Shijie Jia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The video services in wireless mobile networks attract massive mobile video users, which triggers the demand for huge network traffic. In order to reduce network load and ensure user quality of experience (QoE), the MP2P-based video systems need to implement high-efficiency management and scheduling of video resources to optimize distribution and promote sharing with offloading traffic in underlying networks. The traditional solutions do not efficiently address the problems caused by demand variation and node mobility for distribution optimization and resource sharing, so that promotion of video sharing performance in wireless mobile networks for user QoE and reduction of network load still is a significant challenge. This paper proposes a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). VDRCO models the process of video dissemination in overlay networks in terms of the epidemic model and discusses the main influence factors for video dissemination. By estimation for user demand and video distribution variation, VDRCO makes use of clustering nodes with similar video demand and movement behaviors to achieve distribution optimization and efficient video sharing. Extensive tests show how VDRCO achieves much better performance results in comparison with a state-of-the-art solution.

## 1. Introduction

The deployment and popularity of video streaming services in wireless mobile networks are benefited from fast increase of wireless bandwidth and continuous improvement [1]. The promotion of wireless bandwidth capacity such as WiMAX, 4G, and 5G can accommodate massive network traffic brought by high-definition video application [2], which improves visual experience of users; on the other hand, the various technologies of network access such as MANETs, VANETs, and cellular network enable users to conveniently watch video content via smart terminal [3]. As Figure 1 shows, by leveraging the network virtualization methods, the video systems employ the mobile peer-to-peer (MP2P) technologies to achieve large-scale deployment of video applications and promote video sharing in wireless mobile networks [4–9]. The MP2P-based video systems make

use of management and scheduling for video resources in overlay networks to promote video sharing performance. The efficiency of video sharing is a key factor for system quality of service (QoS) and user quality of experience (QoE). The increase in the scale of video users brings huge traffic load for the core network. Offloading video traffic in underlying networks can relieve load of the core network and reduce delay of video lookup and transmission [10, 11]. However, this requires video requesters always can find and connect with geographic adjacent providers to fetch desired videos (called “near-end video fetching”) instead of geographic far providers (called “far-end video fetching”). The near-end video fetching in wireless mobile networks relies on the two perspectives: (1) the efficient resource management achieves optimal video distribution to ensure local balance between supply and demand of resources in geographic area; (2) the efficient resource scheduling supports fast discovery of

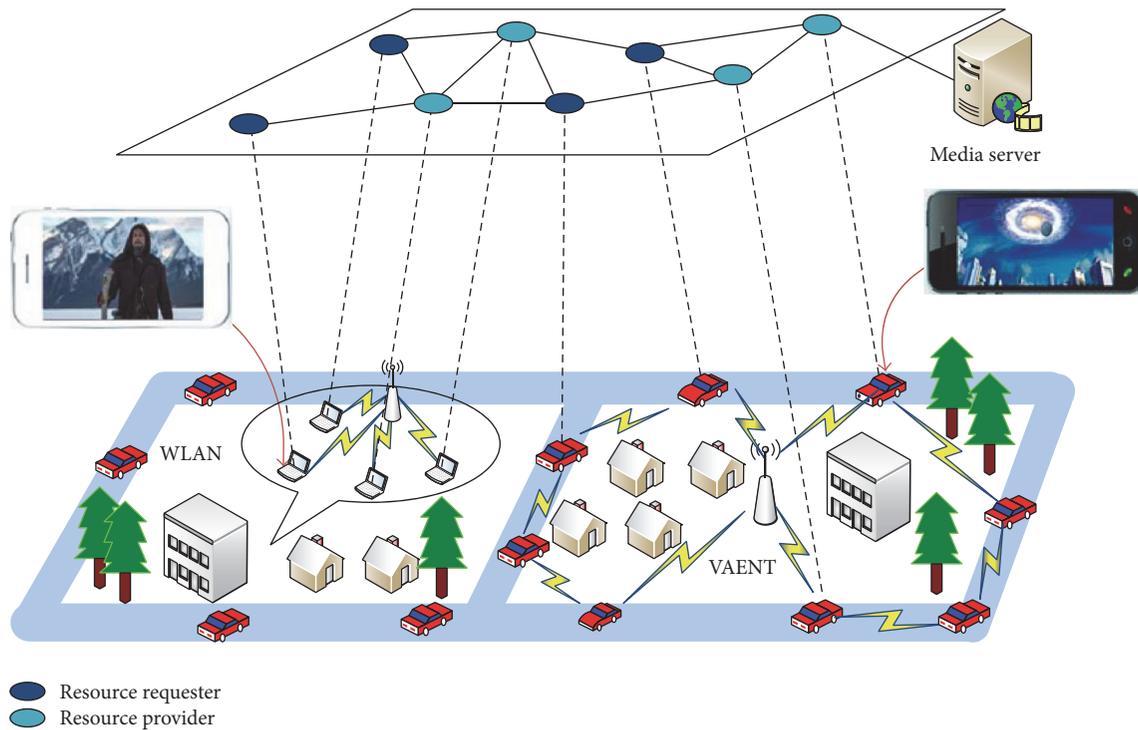


FIGURE 1: MP2P-based video streaming services in wireless mobile networks.

geographical adjacent video providers on the basis of the optimized distribution.

Recently, massive studies focus on making use of video caching and scheduling to promote sharing performance. For instance, the cooperative caching methods can make use of unoccupied storage space of nodes to cache popular videos in the future [12, 13]. The prefetching methods enable the nodes prestore the watched videos with high probability in the future, which reduces startup delay of request nodes and ensures smooth playback [14, 15]. The compelling caching assignment methods make use of the available storage resources to implement caching optimization in terms of variation of video distribution [16–21]. The social-based resource scheduling methods further promote resource sharing performance by cache integration in terms of social relationship [22–25]. However, the above methods do not efficiently address the negative influence caused by demand variation and node mobility for balance stability of video distribution, which results in low-efficiency scheduling and high-cost management [26]. In fact, user interest for video content is the determining factor for requesting or removing a video. The interest variation drives the nodes to request the desired videos regardless of whether near-end or far-end fetching. In other words, if the requested videos cannot be met by geographical adjacent video providers, the requesters also search geographical far providers, which increase delay of video lookup and transmission. Moreover, when the nodes lose interest for videos in local buffer, they remove the videos in order to cache interested videos. Obviously, variation of node interest is an important influence factor for

balance stability of video distribution. On the other hand, the mobile nodes carry the own video resources and move from a geographical area to another geographical area, which also causes variation of video distribution. A key issue for promotion of video sharing performance in wireless mobile networks is how to address the problems caused by interest variation and node mobility.

In this paper, we propose a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). By making use of the epidemic model, VDRCO models video dissemination process in overlay networks and analyzes the four influence factors for video dissemination and distribution: number and influence of initial providers and attraction and number of disseminated videos. The influence of initial providers and the attraction of disseminated videos bring severe influence for user demand. The number of initial providers and disseminated videos influences resource distribution in the process of video dissemination. Based on the above analysis, VDRCO designs the two measurement methods for user demand and video distribution variation. By clustering the nodes in terms of interaction frequency and similarity of interest and mobility, VDRCO further designs a video sharing strategy to promote efficiency of near-end video sharing based on the measurement results of user demand and video distribution variation. Simulation results show that VDRCO can obtain much better performance of video delivery in comparison with a state-of-the-art solution.

## 2. Related Work

Recently, many studies employ caching and scheduling methods to improve resource distribution and delivery. For instance, the asymmetric cooperative resource caching approach proposed in [13] enables the nodes to build cache layer, so that request and response of data are transmitted to the cache layer, which efficiently reduces overhead of data copying and decreases end-to-end transmission delay. In [14], Kozat et al. designed the three cache mechanisms in P2P-based systems: (1) cache prefetching: the peers prefetch all content even if the content is not desired; (2) opportunistic cache update: if a cached content is last in local buffer, the peers can replace the content; (3) hybrid scheme composed of the above two strategies, which can efficiently reduce the server load. In [16], Tan and Massoulié designed the caching strategies in P2P system based on the two scenarios: (1) pure P2P network and (2) distributed server networks. In pure P2P network, each peer assigns one cache slot for the most popular content and the remaining cache slots are assigned for the more popular content. In distributed server networks, the peers randomly cache content according to the overall popularity distribution and switch out content from cache space with uniformly probability. In [18], a commercial caching system is designed, which includes a video retailer (VR) and multiple network service providers (NSPs). By caching popular videos in the rented SBSs, the VR leases small-cell base stations (SBS) of NSPs to provide better local video services to the mobile users. In [19], Mokhtarian and Jacobsen designed the four algorithms: (1) a flexible ingress-efficient algorithm; (2) an offline caching algorithm based on awareness of future requests; (3) an optimal offline caching algorithm; (4) an adaptive ingress control algorithm, handling the demand for content caching on the servers of CDNs. In [20], a cache replacement method, Trend-Caching, is proposed, which optimizes caching performance according to the popularity trends of video content. Trend-Caching makes the decision of caching and removing videos in terms of popularity trend of video content, which achieves fast convergence and improves cache hit rate.

In order to further promote resource sharing performance, massive researchers have leveraged other technologies, such as the social network, to improve efficiency of resource caching and scheduling. For instance, in [22], SAVE considers channel switching as the interactions between channels and clusters channels with high-frequency interactions. SAVE builds contacts between channels with high-frequency and low-frequency interactions, which enables the nodes to stay in the same channel clusters or makes use of interchannel contacts to join a new channel instead of accessing server. The nodes also make use of the maintained friend lists to record common-interest channels and watching time shared by nodes, which achieves fast channel switching and reduces server load. Zhang et al. designed a resource prefetching strategy for video services based on online social network [24]. Because the popular videos are watched with high probability, the nodes can prefetch prefix chunks of popular videos, which reduce startup delay and achieves smooth playback. In [23], social-P2P groups nodes with

common interests and close social relationship into a cluster by investigation of social interests and relationships in social network. Furthermore, the relatively stable nodes in each cluster also are clustered in terms of the Distributed Hash Table (DHT). When any node requests video content, it sends request message to content providers by making use of the DHT structure, which improves lookup success rate and reduces lookup delay.

Moreover, some resource caching methods in new network architecture such as Content-Centric Networks (CCNs) and Information-Centric Networks (ICNs) are proposed. In [27], CodingCache makes use of network coding and random forwarding to support multipath routing in CCNs and improve diversity of cached content along different paths, which increases cache hit rate with reasonable cache privacy. Si et al. investigated the use of harvested bands for caching video contents closed to the interested nodes [28], in order to improve the performance of video distribution. Based on transforming the allocation of harvested bands (a Markov decision process) into a partially observable Markov decision process and a multiarmed bandit formulation, a spectrum management mechanism is designed, which improves efficiency of proactive video caching and spectrum utilization.

## 3. Model of Video Dissemination

The management for video resources cached by nodes in overlay networks focuses on achieving optimal video distribution to support high-efficiency video sharing. The local balance between supply and demand of videos is an important characteristic of optimal video distribution. For instance, when a node  $n_i$  always can find the video providers which cache the requested video from one-hop neighbor nodes of  $n_i$ , the one-hop neighbor relationship can minimize the number of forwarding request message and video data, which ensures low delay of video lookup and transmission and reduces risk of video data loss. Obviously, the local balance between supply and demand of videos can efficiently support near-end video fetching (e.g., the above one-hop video fetching) to promote video sharing performance.

The maintenance for stability of local balance of video resources in wireless mobile nodes is a significant challenge. This is as interest variation and node mobility bring severely negative influence for local balance of video resources. For instance, a node  $n_i$  has watched all videos cached in one-hop neighbor nodes of  $n_i$ , so that  $n_i$  loses interests for videos with high probability. In the other words,  $n_i$  does not request videos which have been watched and may fetch new videos from other nodes instead of one-hop neighbor nodes. Obviously, the interest variation results in imbalance of video distribution. On the other hand, because the mobile nodes are the carriers of video resources, the node mobility makes the video resources move in geographical area, which also results in variation of video distribution in geographical area. The maintenance for stability of local balance needs continuous adjustment (caching new videos and removing old videos) in terms of interest variation and node mobility. The variation of user interests can change the number of available resources in video systems, which determines the

performance of video lookup. The node mobility only leads to the variation of geographical distribution of video resources instead of changing total amount of available video resources, which determines the performance of video delivery.

**3.1. Construction of Video Dissemination Model.** For convenience, several notations used in sections are defined in Notations (notations used by models). The video distribution is continuous variation in the whole process of video dissemination. This is as the video dissemination process has obvious periodicity. When a new video  $v_i$  appears in overlay networks, the nodes have potential interest for  $v_i$ . Some nodes cache  $v_i$  and push  $v_i$  to other contacted nodes, which increases the range and speed of dissemination of  $v_i$ . At the moment, number of cached  $v_i$  in overlay networks fast increases. When the most of nodes have watched  $v_i$ , they lose the interests for  $v_i$  and remove  $v_i$  in local buffer in terms of their storage capacity. The number of cached  $v_i$  in overlay networks fast decreases. Therefore, the video dissemination process can be influenced by variation of user interests instead of the node mobility. The initial providers of  $v_i$  act as the infected nodes. The nodes interested in  $v_i$  are considered as the susceptible nodes. The nodes which have watched  $v_i$  are considered as the immune nodes. In the process of video dissemination, the susceptible nodes actively send request messages to search  $v_i$  (pull mode) and the initial providers send the advertisement messages to push  $v_i$  for other nodes (push mode). The two modes of pull and push enable  $v_i$  to be spread in overlay networks. Therefore, the dissemination process of  $v_i$  follows the epidemic model [29].  $N$  is the total number of nodes in overlay networks.  $N = N(S) + N(I) + N(R)$ , where  $N(S)$  is the number of nodes interested in  $v_i$ ;  $N(I)$  is the number of nodes which have cached  $v_i$ ;  $N(R)$  is the number of nodes which have been uninterested in  $v_i$  and have removed  $v_i$  in local buffer.  $S = N(S)/N$ ,  $I = N(I)/N$ , and  $R = N(R)/N$  denote the ratio between  $N(S)$ ,  $N(I)$ ,  $N(R)$ , and  $N$ , respectively. At the time  $t$ ,  $S$ ,  $I$ , and  $R$  meet the equation  $S(t) + R(t) + I(t) = 1$  and the dissemination model of  $v_i$  can be defined as

$$\begin{aligned} \frac{dI}{dt} &= \lambda SI - \mu I, \\ \frac{dS}{dt} &= -\lambda SI, \\ \frac{dR}{dt} &= \mu I, \end{aligned} \quad (1)$$

where  $\lambda$  and  $\mu$  are the increasing rate of nodes which request and remove  $v_i$ , respectively. If the content of  $v_i$  is wonderful, the probability of requesting  $v_i$  for nodes in overlay networks is high. If the nodes which have cached  $v_i$  have strong influence level for other nodes, the pushed  $v_i$  is accepted with high probability. Therefore, the wonderful content and strong influence level of push nodes make the value of  $\lambda$  keep high levels, so that  $v_i$  can be fast disseminated to the whole overlay networks (the value of  $I$  also keeps fast increase). If the nodes cached in  $v_i$  have watched  $v_i$ , they lose the interests for  $v_i$  and remove  $v_i$  in local buffer, so that the values of  $\mu$  also keep high levels (the value of  $R$  fast increases).

**3.2. Analysis for Video Distribution Variation.** We further describe variation process of  $\lambda$  and  $\mu$  to discuss the variation of distribution of  $v_i$ .

- (1) Initially, because the number of initial providers is less and there are some nodes which are uninterested in  $v_i$  in overlay networks, the dissemination rate of  $v_i$  is low and the value of  $\lambda$  slowly increases. At the moment, because the nodes which cache  $v_i$  do not finish the playback for  $v_i$ , the value of  $\mu$  is 0.
- (2) With the increase in the dissemination time, if the content of  $v_i$  is wonderful, the large amount of nodes make use of pull mode to request and cache  $v_i$ ; if the nodes which have cached  $v_i$  have strong influence level for other nodes, the pushed  $v_i$  easily is accepted. The value of  $\lambda$  fast increases and reaches the peak. At the moment, because some nodes have watched  $v_i$  and lose the interests for  $v_i$ , they remove  $v_i$  in local buffer. The value of  $\mu$  keeps the increase trend.
- (3) When the increment of number of nodes interested in  $v_i$  decreases, the value of  $\lambda$  decreases. At the moment, because the large amount of nodes have watched  $v_i$  and remove  $v_i$ , the value of  $\mu$  fast increases and reaches the peak.
- (4) When the number of nodes interested in  $v_i$  keeps relatively stable at the later stage of dissemination process, the value of  $\lambda$  keeps low level. Because the number of nodes which remove  $v_i$  decreases, the value of  $\mu$  fast decreases and also keeps low level.
- (5) Finally, when all nodes finish the playback and remove  $v_i$ , the values of  $I$  and  $S$  are 0, respectively (the values of  $\lambda$  and  $\mu$  are 0). In fact, the number of nodes in overlay networks always keeps dynamic variation in the real video services. The new mobile nodes join into the video systems, so the values of  $I$  and  $S$  always keep slight fluctuation in the reasonable range.

If the nodes finish the playback and immediately remove  $v_i$ , the values of  $\lambda$  and  $\mu$  have the same variation process. If the nodes finish the playback and do not immediately remove  $v_i$  in order to serve other request nodes, the values of  $\mu$  have longer and more moderate variation process than those of  $\lambda$ . In fact, we only consider the latter case in terms of the characteristic of resource sharing with each other in MP2P-based video systems.

On the other hand, the values of  $\lambda$  and  $\mu$  reflect the variation levels of distribution of  $v_i$ . The attraction of video content and the influence of providers are the two impact factors for the values of  $\lambda$  and  $\mu$  and variation of user interests. For instance, the two nodes are interested in the NBA and frequently share related videos with each other. If a node pushes videos of the CBA to another node, the latter may accept the pushed videos with high probability. However, if a node pushes videos of the World Cup to another node, the latter may reject the pushed videos with high probability. This is as the pushed videos go beyond the demand (interest) ranges of the latter (the latter is immune for the pushed videos). The wonderful video content can

attract large amount of nodes and the high-level influence of providers increases the probability of push success, which speeds up the video dissemination. As we know, the fast dissemination can increase the provision of available resources to reduce startup delay of request nodes caused by shortage of resources. In order to efficiently improve rate and range of video dissemination, a key issue is how to be aware of user demand and select appropriate push nodes.

Except for attraction of video content and influence of providers, number of initial providers and number of disseminated videos are the two influence factors for the dissemination process. The large amount of initial providers not only ensures enough provision of available resources but also increases the value of  $\lambda$  due to extension of push range. Moreover, the dissemination of multiple videos in overlay networks not only leads to the fast change of user interests but also causes the competition for bandwidth and storage resources of nodes. For instance, the small number of new videos lengthens the process of video dissemination. This is as the nodes do not fast lose the interests for the existing videos in overlay networks. The content providers do not eagerly remove  $v_i$  to fetch new videos due to limited storage space. Otherwise, if the massive new videos appear in overlay networks, the nodes continually request more interested videos, which reduce the stay time of  $v_i$  in local buffer of content providers. This speeds up the convergence of process of video dissemination, which results in the fast variation of distribution of  $v_i$ . Therefore, in order to improve video sharing performance, the management and scheduling of video resources need to consider awareness of user demand, selection of push nodes in terms of estimation of common interests between nodes, and monitoring of variation level of video distribution.

#### 4. VDRCO Detailed Design

As we know, the optimal video distribution can improve resources sharing performance of video systems and reduce delay of video lookup and delivery. The accurate awareness for user demand and distribution variation level is significant for optimization of video distribution. The awareness for user demand enables the video systems to be aware of caching what resources. The awareness for distribution variation level enables the video systems to be aware of period time of scheduling resources.

*4.1. Awareness of User Demand.* When the users are interested in videos, they send request messages to fetch and cache the requested videos. Obviously, the requested videos can embody the interests of users and the interest range of users is limited. For instance, a user has the highest interest level for the NBA-related videos among all videos. The user pays more time to search and watch the NBA-related videos. On the other hand, because the users may push some interested videos to other users, the pushed videos also reflect the interests levels of users. Therefore, the videos which are watched for a long time or are pushed by users can represent the interest of users. The interest range can be considered as demand range of users.

In order to accurately measure the interest range, we cluster the videos watched by users and define the interest range in terms of the interest levels of users for videos. The set of clustered videos is considered as the interest range of users. Let  $S_v(n_i)$  denote the set of watched videos of  $n_i$ . The similarity  $S_{jk}$  between any two videos  $v_j \in S_v(n_i)$  and  $v_k \in S_v(n_i)$  can be calculated according to our previous work [30]. If  $S_{jk}$  is the largest among the similarity values between  $v_k$  and other items in  $S_v(n_i)$ ,  $v_j$  and  $v_k$  form a new video subset  $S_{sub_j} = (v_j, v_k)$ , as Figure 2(a) shows. Similarly, if  $S_{hk}$  denotes the similarity between  $v_h \in S_v(n_i)$  and  $v_k \in S_v(n_i)$  and  $S_{hk}$  is larger than the similarity values between  $v_h$  and other items in  $S_v(n_i)$ ,  $v_h$  joins into  $S_{sub_j}$ , as Figure 2(b) shows. After the iteration of video clustering, the items in  $S_v(n_i)$  are grouped into several video subsets; namely,  $S_v(n_i) = (S_{sub_a}, S_{sub_b}, \dots, S_{sub_j})$ , as Figure 2(c) shows. Because any video in each subset  $S_{sub_j}$  only has a similarity value with another video in  $S_{sub_j}$ , all items in  $S_{sub_j}$  can be ranked in terms of the ascending similarity; namely,  $S_{sub_j} = (v_a, v_b, \dots, v_j)$ . In fact, the video subsets may include many noise items. For instance,  $v_s \in S_{sub_j}$  and the similarity value  $S_{sk}$  between  $v_s$  and  $v_k$  is larger than the values between  $v_s$  and other items in  $S_v(n_i)$ . If the value of  $S_{sk}$  is very small and the similarity values between  $v_s$  and other items in  $S_{sub_j}$  are also very small,  $v_s$  is unsuitable to become the member in  $S_{sub_j}$ ; namely,  $v_s$  is a noise item in  $S_{sub_j}$ . This is as the above video clustering only considers the similarity between  $v_s$  and  $v_k$  and neglects the similarity between  $v_s$  and other items in  $S_{sub_j}$ . In order to remove the noise items in video subsets, we firstly calculate the average value of similarity among all items in  $S_{sub_j}$  according to the following equation:

$$\bar{S}(S_{sub_j}) = \frac{\sum_{c=1}^{|S_{sub_j}|} \sum_{e=1}^{|S_{sub_j}|-1} S_{ce}}{|S_{sub_j}|(|S_{sub_j}|-1)}, \quad (2)$$

$$\bar{S}(S_{sub_j}) \in [0, 1],$$

where  $|S_{sub_j}|$  returns the number of items in  $S_{sub_j}$ . The initial average value of similarity among items in  $S_{sub_j}$  is defined as  $\bar{S}(S_{sub_j})^{(1)}$ . Because all items in  $S_{sub_j}$  have been ranked in terms of the ascending similarity and  $v_a$  is first item with minimum similarity,  $v_a$  is removed from  $S_{sub_j}$  and other items form a new set  $S_{sub_j}^{(2)}$ . The average value of similarity among all items in  $S_{sub_j}^{(2)}$  is recalculated according to (2); namely  $\bar{S}(S_{sub_j})^{(2)}$ . If  $\bar{S}(S_{sub_j})^{(2)} < \bar{S}(S_{sub_j})^{(1)}$ ,  $v_a$  is not a noise item and is readded to  $S_{sub_j}^{(2)}$ . Otherwise, if  $\bar{S}(S_{sub_j})^{(2)} > \bar{S}(S_{sub_j})^{(1)}$ ,  $v_a$  is considered as a noise item. Although the similarity value between  $v_a$  and another item in  $S_{sub_j}$  is high,  $v_a$  may have the low similarity with other items in  $S_{sub_j}$ . Therefore, when  $v_a$  is removed, the average value of similarity in  $S_{sub_j}^{(2)}$  increases. After making the decision of retaining or removing  $v_a$ ,  $S_{sub_j}^{(2)}$

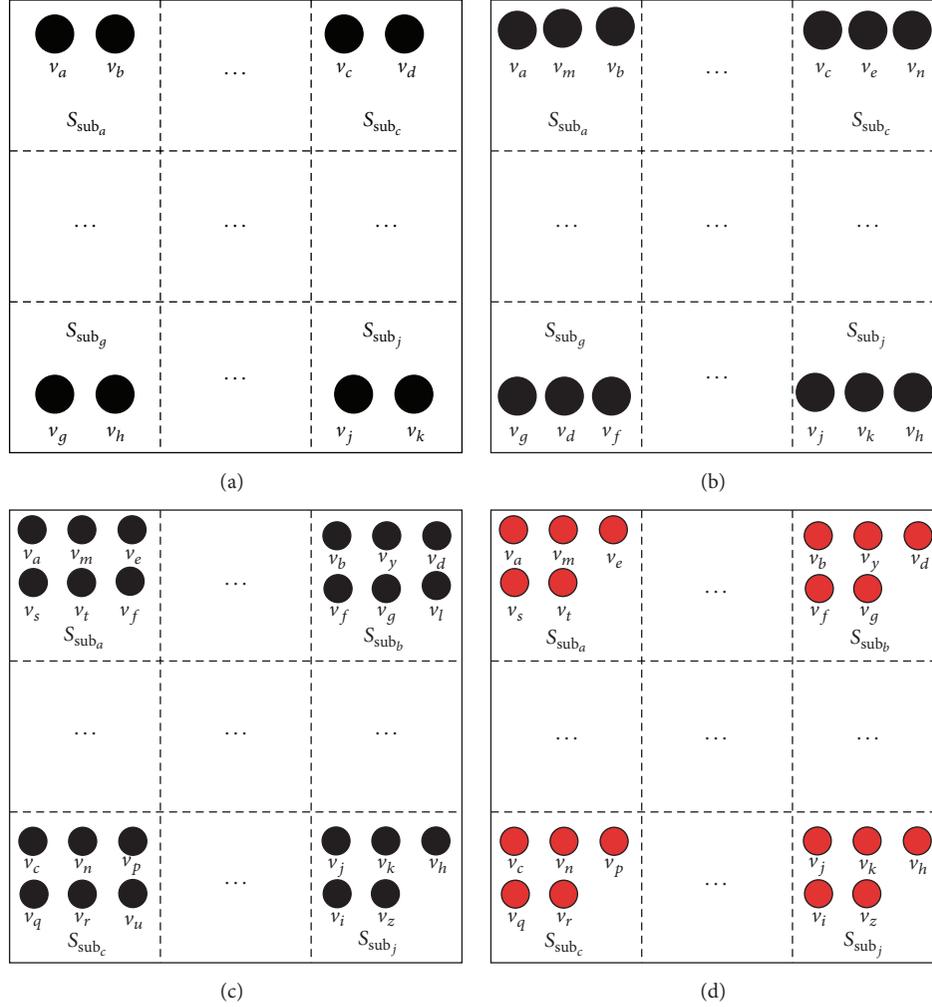


FIGURE 2: Estimation of interest range.

continues to remove  $v_b$ . Other items in  $S_{\text{sub}_j}^{(2)}$  form a new subset  $S_{\text{sub}_j}^{(3)}$ . If  $\bar{S}(S_{\text{sub}_j}^{(3)}) > \bar{S}(S_{\text{sub}_j}^{(2)})$ ,  $v_b$  is a noise item. Otherwise, if  $\bar{S}(S_{\text{sub}_j}^{(3)}) < \bar{S}(S_{\text{sub}_j}^{(2)})$ ,  $v_b$  is not a noise item and is readded to  $S_{\text{sub}_j}^{(2)}$ . By iteration of the above process, the noise videos in  $S_{\text{sub}_j}$  are removed, as Figure 2(d) shows. Similarly, the noise items in all video subsets can be removed and form a new video subset. We continue to cluster and refine items in subset composed of noise videos according to the above process. The condition of convergence is that each noise video forms a subset.

In order to extract the demand regions of  $n_i$  from the video subsets, we firstly make use of the watched time and pushed frequency to estimate the interest level for any video  $v_j$  according to the following equation:

$$I_i(v_j) = \alpha \frac{l_i(v_j)}{L_i} + \beta \frac{f_i(v_j)}{F_i}, \quad \alpha, \beta \in [0, 1], \quad (3)$$

where  $l_i(v_j)$  is the watched time of a node  $n_i$  for  $v_j$ ;  $L_i$  is the total playback time of  $n_i$  for all watched videos;  $f_i(v_j)$  is the

pushed frequency of  $n_i$  for  $v_j$ ;  $F_i$  is the total push frequency of  $n_i$  for all watched videos.  $\alpha$  and  $\beta$  are the weight values of watched time and push frequency, respectively, and  $\alpha + \beta = 1$ . Therefore, the value of  $I_i(v_j)$  is in the range  $[0, 1]$ . If a video  $v_j \in S_{\text{sub}_j}$  has the highest interest level among all items in  $S_{\text{sub}_j}$ ,  $v_j$  acts as the deputy video of  $S_{\text{sub}_j}$ . Each video subset  $S_{\text{sub}_j}$  has an average value of interest level according to the following equation:

$$\bar{I}(S_{\text{sub}_j}) = \frac{\sum_{c=1}^{|S_{\text{sub}_j}|} I_i(v_c)}{|S_{\text{sub}_j}|}. \quad (4)$$

Further, the average value of interest level of all items in  $S_v(n_i)$  can be defined as

$$\bar{I}(S_v(n_i)) = \frac{\sum_{c=1}^{|S_v(n_i)|} \bar{I}(S_{\text{sub}_j})}{|S_v(n_i)|}, \quad (5)$$

where  $|S_v(n_i)|$  returns the number of video subsets included in  $S_v(n_i)$ . If the value of  $\bar{I}(S_{\text{sub}_j})$  is greater than  $\bar{I}(S_v(n_i))$ ,  $S_{\text{sub}_j}$

is considered as a demand region of  $n_i$ . Therefore, the set of demand regions can be defined as  $D_i = (d_1, d_2, \dots, d_k)$ . When  $n_i$  requests and watches a new video  $v_d$ , the similarity between  $v_d$  and deputy videos in all demand regions is calculated. If the similarity between  $v_d$  and a deputy video  $v_j$  in  $S_{\text{sub}_j}$  is the largest similarity among all deputy videos,  $v_d$  is added to  $S_{\text{sub}_j}$ . By the purification for the subset, if  $v_d$  is a noise video, the similarity values between  $v_d$  and all noise videos are calculated.  $v_d$  is added to the video subset corresponding to the noise video which has the largest similarity; otherwise, if  $v_d$  is not a noise video, it stays in  $S_{\text{sub}_j}$ .

**4.2. Awareness of Variation Level of Video Distribution.** When the nodes want to watch new videos and the local buffer is full, they remove the videos which have been watched. The replacement of cached videos in local buffer of nodes directly results in the variation of video distribution in overlay networks. The stay time of cached videos in local buffer is an important factor for the measurement of replacement. For instance, when a request node  $n_j$  wants to watch a video  $v_k$ , it sends the request message in overlay networks and waits for the returned video data. If a video provider  $n_i$  stores  $v_k$  and receives the request message of  $n_j$ , it needs to transmit video data to  $n_j$  and cannot remove  $v_k$  in local buffer. In other words, the more the number of received request messages is, the longer the stay time of cached videos is. Obviously, the period time of video dissemination is the impact factor for stay time. The time of downloading  $v_k$  from  $n_i$  can be defined as

$$\text{TR}_i(v_k) = \frac{\text{size}_k}{\bar{B}_i(1 - \bar{P}_i)}, \quad (6)$$

where  $\text{size}_k$  is the size of  $v_k$ ;  $\bar{B}_i$  and  $\bar{P}_i$  are the average values of bandwidth and packet loss rate of  $n_i$  in the process of transmission. We assume that the video providers remove  $v_k$  in local buffer when they do not receive the request messages during a short period time  $\Delta t$ . If the number of request messages received by  $n_i$  is  $\text{NR}_i$  during a period time  $t_a < \text{NR}_i \times \text{TR}_i(v_k)$ , the stay time of  $v_k$  in local buffer of  $n_i$  is  $\text{NR}_i \times \text{TR}_i(v_k) + \Delta t$ . If  $\bar{B}_i$  is equal to the playback rate of requesters, the transmission time of  $v_k$  is equal to the playback time of requesters. If the number of requesters and providers is  $\text{NR}_k$  and  $\text{NP}_k$  for  $v_k$  during the period time  $t_b - t_a$ , respectively, and  $\text{NR}_k < \text{NP}_k$ , the  $\text{NR}_k$  video requesters cache and watch  $v_k$  and also become video providers of  $v_k$ . The number of cached  $v_k$  in networks is  $2 \times \text{NR}_k$  and the number of  $v_k$  removed by providers is  $\text{NP}_k - \text{NR}_k$ . This is as the  $\text{NP}_k - \text{NR}_k$  providers do not receive the request messages for  $v_k$ . The variation level of  $v_k$  distribution during  $t_b - t_a$  is defined as  $\text{VL}_k = \text{NR}_k / (\text{NP}_k - \text{NR}_k)$ .

On the other hand, the push between nodes also drives the video dissemination except for the video request. For instance, a provider  $n_i$  stores a video  $v_k$  and pushes  $v_k$  to another node  $n_j$ . If  $n_j$  accepts the pushed  $v_k$ , it returns an acknowledgement message to  $n_i$  and receives the video data from  $n_i$ . Otherwise, if  $n_j$  is uninterested in  $v_k$ , it neglects the pushed  $v_k$ . The push success rate also is an important

factor for period time of video dissemination. If the number of successful push for  $v_k$  during the period time  $t_b - t_a$  of video dissemination is  $\text{NU}_k$ , the variation level of  $v_k$  distribution during  $t_b - t_a$  is defined as  $\text{VL}_k = (\text{NR}_k + \text{NU}_k) / (\text{NP}_k - \text{NR}_k - \text{NU}_k)$ , where  $\text{NP}_k > \text{NR}_k + \text{NU}_k$ . Obviously,  $(\text{NR}_k + \text{NU}_k) / (t_b - t_a)$  and  $(\text{NP}_k - \text{NR}_k - \text{NU}_k) / (t_b - t_a)$  denote the increase and decrease rate of  $v_k$ , respectively ( $\lambda = (\text{NR}_k + \text{NU}_k) / (t_b - t_a)$  and  $\mu = (\text{NP}_k - \text{NR}_k - \text{NU}_k) / (t_b - t_a)$ ). The variation level of video distribution can be defined as  $\text{VL}_k = \lambda / \mu$ .

**4.3. Video Sharing Strategy.** Based on the awareness results for user demand and distribution variation level, the video systems make use of video caching and scheduling to optimize distribution and achieve high-efficiency sharing in the process of video dissemination. For instance, if the two nodes  $n_i$  and  $n_j$  frequently share cached videos with each other,  $n_i/n_j$  pushes videos which may be interested by  $n_j/n_i$  in terms of results of demand awareness, which increases probability of push success. When their video demand is out of original range and has high variation level, they can adjust cached videos in local buffer to maintain the stability of balance of video supply and demand between them. Moreover, if the nodes have similar demand and are clustered into a node set, the similar demand ensures the videos cached by them can meet the requirement with each other. The nodes in the same set share cached videos and adjust local cached videos according to variation of demand and cached resources in the set, which achieves balance between supply and demand in the whole overlay networks.

On the other hand, the construction of logical links between items in node sets relies on similarity level of demand and does not consider the factor of their geographical location. The adjacent geographical location between video providers and requesters can reduce number of relay nodes in the path of data transmission, which decreases delay of data transmission and probability of data loss and relieves traffic load of networks. The mobility results in dynamic geographical location of nodes, which brings severely negative influence for video delivery. For instance, the one-hop data transmission between requesters and providers may become multihop transmission with the continuous movement. The estimation of mobility similarity between nodes can measure the stability of relative distance between nodes. The construction of node sets (clustering nodes in overlay networks) needs to investigate similarity of demand and mobility of nodes, which achieves local balance between video supply and demand in geographical area.

We investigate the interaction (push and pull) between nodes to estimate similarity of demand. The high-frequency and successful interaction denotes that the nodes can meet the demand for the videos with each other, which increases lookup success rate and reduces lookup delay. The total number of interaction between any two nodes  $n_i$  and  $n_j$  is defined as

$$F_{ij} = f_i^{\text{push}} + f_j^{\text{push}} + f_i^{\text{pull}} + f_j^{\text{pull}}, \quad (7)$$

where  $f_i^{\text{push}}$  and  $f_i^{\text{pull}}$  are the number of videos pushed and requested by  $n_i$  for  $n_j$ ;  $f_j^{\text{push}}$  and  $f_j^{\text{pull}}$  are the number of

videos pushed and requested by  $n_j$  for  $n_i$ . The accepted video push and the successful response for the video request are considered the successful interaction.  $IR_{ij} = F_{ij}^{(s)}/F_{ij}$  denotes the interaction success rate where  $F_{ij}^{(s)}$  is the number of successful interaction. The higher the value of  $IR_{ij}$  is, the more efficient the video dissemination is. Moreover, the high-frequency interaction and high interaction success rate also reflect the closeness and stability of relationship between nodes. The close and stable relationship between nodes avoids the frequent message exchange to fetch state information with each other such as cached videos, which reduces the energy consumption of mobile nodes caused by huge message overhead.

Because the mobile nodes have low capacity in bandwidth, computation, storage, and energy, they difficultly bear the load for the estimation of mobility similarity with each other in large geographic areas. We investigate the stability of one-hop neighbor relationship between nodes to estimate their mobility similarity. The estimation of mobility similarity in small geographic areas can reduce the load of mobile nodes. If the two nodes have one-hop neighbor relationship at the time  $t_a$ , the two nodes are considered as encounter at  $t_a$ . The more the number of encounter is, the longer the period time of stable one-hop neighbor relationship is. The stability  $st$  of one-hop relationship between nodes can be estimated according to our previous work [31]. Therefore, the relationship level between any two nodes  $n_i$  and  $n_j$  can be defined as

$$GR_{ij} = IR_{ij} \times \left( 1 - \frac{\text{arccot}(st_{ij})}{\pi} \right). \quad (8)$$

Because the values of  $st_{ij}$  are in the range  $(-\infty, +\infty)$ , the values of  $1 - \text{arccot}(st_{ij})/\pi$  are in the range  $[0, 1]$  and  $GR_{ij} \in [0, 1]$ . Any node  $n_i$  records the information of nodes which encountered  $n_i$  and interacted with  $n_i$  and builds a node set  $GN_i$ .  $n_i$  estimates the relationship level with all items in  $GN_i$  according to (8). We cluster the nodes in overlay networks into multiple node subsets by employing the above demand region construction method in Section 4.1. If  $n_j$  has the highest relationship level with  $n_i$  among  $GN_i$ ,  $n_i$  and  $n_j$  form a node set  $GS_j$ . If another node  $n_k$  also has the highest relationship level with  $n_j$  among items in  $GN_k$ ,  $n_k$  joins into  $GS_j$ . After the iteration of the above clustering process, the built node subsets continue to remove the noise nodes according to the variation of average values of relationship between nodes in subsets. Finally, the nodes in overlay networks are grouped into multiple node subsets; namely,  $NS = (GS_1, GS_2, \dots, GS_n)$ . The nodes in the same subsets are aware of the deputy videos corresponding to demand regions with each other by periodical exchange of messages.

The nodes preferentially send request messages for the videos to members in the same subsets. Before  $n_i$  requests a video  $v_k$ , it firstly calculates the similarity between  $v_k$  and deputy videos of other nodes in current subset; namely,  $LS_i = (S_{ak}, S_{bk}, \dots, S_{nk})$ , where  $S_{ak}$  denotes the similarity value between  $v_a$  and  $v_k$ . If the similarity between the deputy

video  $v_h$  and  $v_k$  is the largest among all deputy videos and a node  $n_j$  in current subset stores  $v_h$ ,  $n_i$  sends the request message to  $n_j$ . If  $n_j$  also stores  $v_k$  in local buffer,  $n_j$  directly returns an acknowledge message and transmits data of  $v_k$  for  $n_i$ . Otherwise, if  $n_j$  does not store  $v_k$ , it returns a lookup failure message to  $n_i$ .  $n_i$  uses a threshold  $Thr_i$  to filter some items in  $LS_i$ . If the similarity values between  $v_k$  and items in  $LS_i$  is greater than  $Thr_i$ , these deputy videos are marked as the selected videos.  $n_i$  sends the request messages to the nodes which cache the selected videos except for  $n_j$ . If these nodes store  $v_k$ , they return the acknowledge messages to  $n_i$  and  $n_i$  selects a node as the provider. Otherwise, if these nodes do not store  $v_k$ ,  $n_i$  broadcasts the request message in the whole network after receiving the lookup failure messages.

On the other hand, because the nodes can be aware of number of request for  $v_k$  and number of nodes which cache  $v_k$  in subsets by message exchange, they estimate the variation level  $VL_k$  of distribution for  $v_k$  during a period time. If  $VL_k$  is the minimum value among the variation levels of all videos in current subset,  $v_k$  can be preferentially removed in local buffer of nodes after a period time  $\Delta t$ . Otherwise, if  $VL_k$  is not the minimum value among the variation levels of all videos in current subset and the nodes carrying  $v_k$  do not receive the request message during a period time  $(1 + VL_k)\Delta t$ , these carriers can remove  $v_k$  from their local buffer.

## 5. Testing and Test Results Analysis

**5.1. Testing Topology and Scenarios.** The performance of the proposed solution VDRCO compares with a state-of-the-art community-based video sharing solution AMCV [32]. We use in NS-2 to model and implement the two solutions VDRCO and AMCV. The 400 mobile nodes are deployed in the  $1000 \times 1000 \text{ m}^2$  area and have 200 m signal range. The mobile speed of nodes is in the range  $[1, 20] \text{ m/s}$ . The simulation time is set to 400 s. The data transmission protocol is UDP and the wireless routing protocol is DSR. In order to reduce the influence caused by network congestion for the simulation results, the server and the mobile nodes have 20 Mb/s and 10 Mb/s bandwidth, respectively. The transmission rate of video data (the playback rate of mobile nodes) is set to 128 kb/s.

The 200 mobile nodes request videos and join the video system, which are considered as the system members. The number of video files is 30 and the length of each file is 100 s. The 200 playback logs are generated for the system members, which are used to define the requested resources and playback time of system members. After simulation startup, the system members finish the first request for the videos in terms of the playback logs following the Poisson distribution from  $t = 0 \text{ s}$  to  $t = 100 \text{ s}$ . The requested resources and playback time of each system member in playback logs are randomly assigned. The number of videos requested by nodes in playback logs is 5 and the system members quit the system when they have watched all videos. The number of resources cached in each system member is 3. In VDRCO, the 200 historical playback logs are generated, which are used to estimate demand regions of the 200 system members. The historical logs of encounter and interaction of system

members are generated. The system members are clustered in terms of their historical logs of encounter and interaction before the simulation.  $\alpha$  and  $\beta$  are set to 0.5, respectively. The threshold Thr of all system members is set to 0.2.  $\Delta t$  is set to 20 s.

On the other hand, the 400 movement behavior logs are generated for the 400 mobile nodes. In the movement behavior logs, the location, speed, and moving target of mobile nodes are randomly assigned. The mobile nodes move from the initial location. When the mobile nodes arrive at the assigned target, they do not stay and continue to move in terms of the new speed and moving target defined in movement behavior logs. In VDRCO, the members in node subsets are reestimated the relationship between them in terms of the new movement traces and make the selection: leaving current subset and joining new subset or staying current subset.

AMCV is suitable for the performance comparison with VDRCO. AMCV has the flexible deployment strategy to implement video file sharing in wireless mobile networks. In AMCV, the nodes which request the same video chunk are grouped into a community. In the simulation, AMCV clusters the nodes which request the same video file into a community. The number of communities in AMCV is in the range (0, 30]. The maintenance strategy in AMCV also is suitable for the video file sharing. (2) The sharing strategy of video chunks in AMCV is suitable for the video file sharing. We assign the ID for the 30 video files, so the communities in AMCV can build the static connections with each other. We also generated the 5000 historical playback logs which include ID of videos requested by nodes. AMCV can construct the dynamic connections between communities according to the request behaviors of nodes.  $\alpha$  and  $\beta$  are also set to 0.5, respectively.  $PT_b$  and  $PT_r$  are set to 0.2 and 0.12, respectively. The value of UT is defined as 60 s. (3) AMCV and VDRCO make use of community-based MP2P network architecture and estimation for the video request behaviors of nodes to support high-efficiency video sharing. AMCV relies on the association relationship between videos requested by nodes to build the dynamic connections between communities and support the fast video lookup. VDRCO investigates the content similarity level between videos requested by nodes and node mobility to build the node subsets and support the video sharing between nodes in subsets. Before simulation startup, the system members also are clustered in terms of the community construction method.

**5.2. Performance Evaluation.** The performance comparison between VDRCO and AMCV includes the three aspects average startup delay (ASD), caching utilization rate, and packet loss rate (PLR).

(1) *ASD.* Let  $SD = t_r - t_v$  denote the definition of startup delay.  $t_r$  is the time of sending the packet containing request message at the application layer;  $t_v$  is the time of receiving first packet containing video data at the application layer.

Figure 3 shows the variation of ASD of the two solutions VDRCO and AMCV with increasing simulation time.  $ASD = \sum_{i=1}^m SD_i/m$  denotes the average startup delay where  $m$  is the

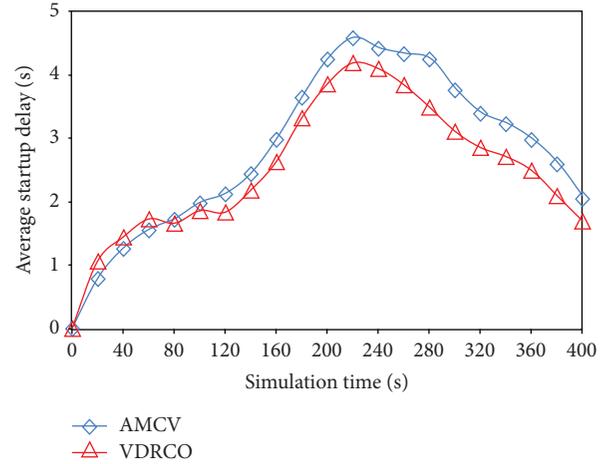


FIGURE 3: Average startup delay against simulation time.

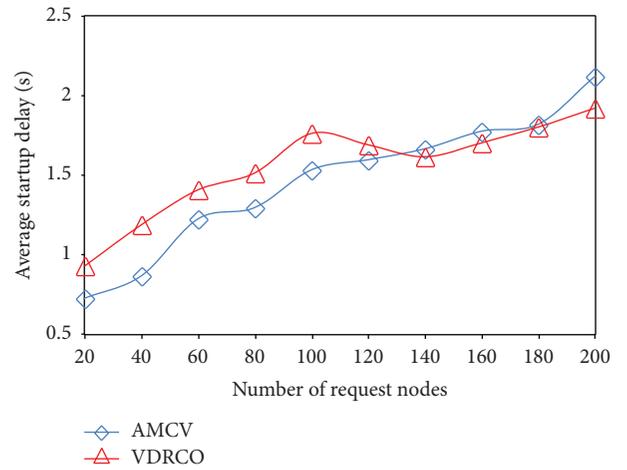


FIGURE 4: Average startup delay against number of request nodes.

number of nodes which have finished startup during a time interval  $T = 20$  s. The two curves corresponding to VDRCO and AMCV keep the fall trend from  $t = 220$  s to  $t = 400$  s after the fast rise from  $t = 0$  s to  $t = 200$  s. The VDRCO results fast increase from  $t = 0$  s to  $t = 60$  s, also keep slow rise from  $t = 80$  s to  $t = 120$  s, and reach the peak value at  $t = 220$  s. The AMCV results keep relatively stable rise from  $t = 0$  s to  $t = 120$  s and reach the peak values at  $t = 220$  s. Although the VDRCO results are higher than those of AMCV from  $t = 0$  s to  $t = 60$  s, the red curve corresponding to VDRCO is lower than that of AMCV from  $t = 80$  s to  $t = 400$  s and the peak value of VDRCO is less than that of AMCV.

Figure 4 shows the variation of ASD of VDRCO and AMCV with increasing number of request nodes.  $ASD = \sum_{i=1}^m SD_i/m$  also denotes the average startup delay where  $m$  is the number of nodes which have finished startup during a time interval of every 20 nodes requesting videos. The red curve corresponding to VDRCO keeps fast rise with increasing number of request nodes from 0 to 100 and also falls with increasing number of request nodes from 100 to 140. Finally, the red curve continues to keep the rise trend

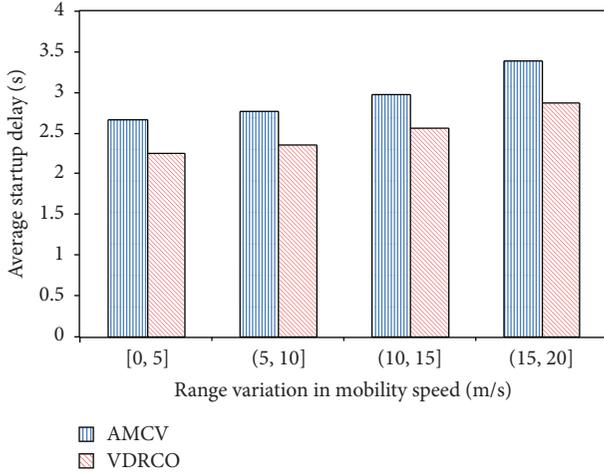


FIGURE 5: Average startup delay against range variation of mobile speed.

with increasing number of request nodes from 140 to 200. The blue curve corresponding to AMCv always fast increases in the process of the 200 request nodes joining. Although the red curve of VDRCO is higher than that of AMCv with increasing number of request nodes from 0 to 120, the VDRCO results are better than those of AMCv in the increase process of request nodes from 140 to 200.

Figure 5 shows the results of ASD of VDRCO and AMCv in the different range of mobile speed where the mean values of startup delay of all nodes during the whole simulation corresponding to the four range of mobile speed denote ASD. The bars corresponding to VDRCO and AMCv keep the rise trend with the increase in the mobile speed. The red bars of VDRCO are lower than those of AMCv in all range of mobile speed.

In AMCv, the system nodes which request the same video are grouped into a node community and these communities rely on the static and dynamic connections to support the node movement between communities (namely, video lookup). If there are the communities corresponding to all videos in overlay networks, the request nodes can fast find the video providers by making use of the connections between communities to forward the request messages. Therefore, the ASD results of AMCv in the initial simulation process are better than those of VDRCO. However, the system members continuously move between communities with the variation of requested videos, which leads to the dynamic and fragile community structure. In the other words, the interest variation for the video content results in the generation and disappearance of communities, which brings severely negative influence for the lookup success rate. If the request messages cannot be forwarded to the corresponding communities, they only are forwarded to the server. Therefore, the ASD values of AMCv fast increase. On the other hand, AMCv does not consider the mobility of mobile nodes in the process of video delivery, so that the far-end access for the video resources is not avoided. The fast increase in the number of requested videos brings large amount of data traffic, which

leads to the network congestion. The ASD values of AMCv keep high level in the process of network congestion from  $t = 140$  s to  $t = 340$  s, which means that the ASD values of AMCv are subjected to the severe influence of network congestion. When some system members have watched the 5 videos and quit the system, the decrease of traffic relieves the congestion level, so that the ASD values of AMCv can quickly decrease. Moreover, the mobility of mobile nodes may result in the increase of hop number between requesters and providers, so the increase of mobile speed also causes the rise of ASD results of AMCv. The ASD values of AMCv with low mobile speed are less than those with high mobile speed. In VDRCO, the system members are clustered in terms of the frequency of interaction and encounter between them and preferentially request and push the video content in current subsets. Because the system members in subsets continuously estimate the relationship between them according to the current frequency of interaction and encounter, the variation of subset structure brings severely negative influence for the sharing performance. Initially, the new movement traces lead to the change of subset structure; namely, the system members leave current subsets and join other subsets, which increase the ASD values of VDRCO. Once the subset structure keep relatively stable state, the intrasubset video sharing relies on the close geographical distance to obtain low transmission delay of request messages and video data. Further, the system members in the same subsets are aware of the demand regions with each other in the process of video sharing and dynamically regulate the local cached resources according to the demand variation and video dissemination. The adjustment of cached resources based on the awareness of demand variation ensures local balance between supply and demand to promote the intrasubset sharing and reduce lookup delay. Therefore, the ASD values of VDRCO can keep the level of decrease and stability from  $t = 60$  s to  $t = 120$  s. With the fast increase in the number of requested videos, VDRCO also is subjected to the network congestion from  $t = 140$  s to  $t = 340$  s, but the ASD values of VDRCO still are less than those of AMCv. The peak value of VDRCO is less than that of AMCv and the ASD decrease rate of VDRCO also is higher than that of AMCv. On the other hand, because the system members in the same subsets keep frequent encounter and have the stable one-hop neighbor relationship, the near-end video delivery reduces the influence caused by network congestion. The ASD values of VDRCO keep the rise trend with the increase in the mobile speed, but the increment of ASD values of VDRCO is less than that of AMCv.

(2) *Caching Utilization Rate.* Making use of video resources cached in local buffer of system members to serve the requesters denotes the caching utilization. Let  $N_{CU}$  and  $N_{RM}$  denote the number of caching utilization and all request messages, respectively. The caching utilization rate can be defined as  $R_{CU} = N_{CU}/N_{RM}$ .

Figure 6 shows the variation of caching utilization rate of VDRCO and AMCv every 40 s. The curves of VDRCO and AMCv have the rise trend during the whole simulation process. The blue curve corresponding to AMCv fast increases from  $t = 40$  s to  $t = 320$  s and keeps stable rise from  $t = 340$  s

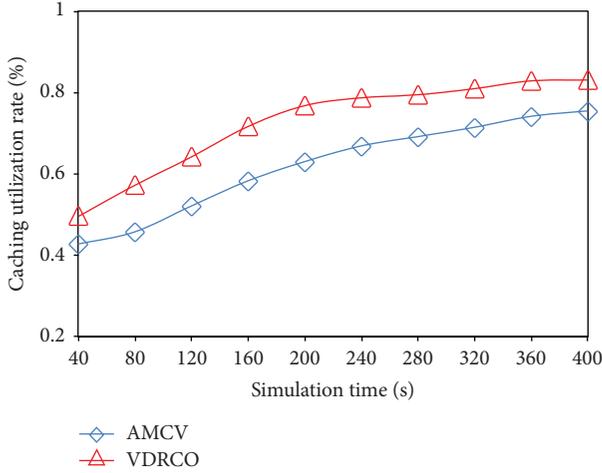


FIGURE 6: Caching utilization rate against simulation time.

to  $t = 400$  s. The red curve corresponding to VDRCO has the stable increase from  $t = 200$  s to  $t = 400$  s after a fast rise from  $t = 40$  s to  $t = 180$  s. The results of caching utilization rate of VDRCO are higher than those of AMCV.

In AMCV, the system members firstly send the request messages to the broker members in current communities. The broker members make use of the maintained connections with other communities to forward the request messages. If the requested videos are not played by the system members, the communities corresponding to the requested videos also are not built and the request nodes need to fetch the requested video resources from the server. Although the members in communities which have watched the requested videos and do not immediately remove the videos, they continue to request new videos, quit current communities, and join the new communities. The original videos cached by them are not used to serve other requesters. This is as these members have quit the original communities, which results in the link disconnection with the broker members of original communities. On the other hand, the interests of users have random and extensive characters; it is difficult to all videos being watched during the same period time. Therefore, the caching utilization rate of AMCV always keeps the lower level than those of VDRCO. In VDRCO, the system members in the same subsets have high-frequency interaction and similar interests, so they make use of local cached videos to serve with each other. Even if the system members do not fetch the desired video resources from the nodes in current subsets, they require other nodes in subsets forward the request messages or broadcast the request messages to obtain the requested videos from the nodes in overlay networks. Therefore, the results of caching utilization rate of VDRCO are higher than those of AMCV in the whole process of simulation. Moreover, the high caching utilization rate also means that VDRCO reduces the load of the server to obtain higher system scalability than AMCV.

(3) *Packet Loss Rate.* The ratio between number of lost data at the application layer in the transmission process and number

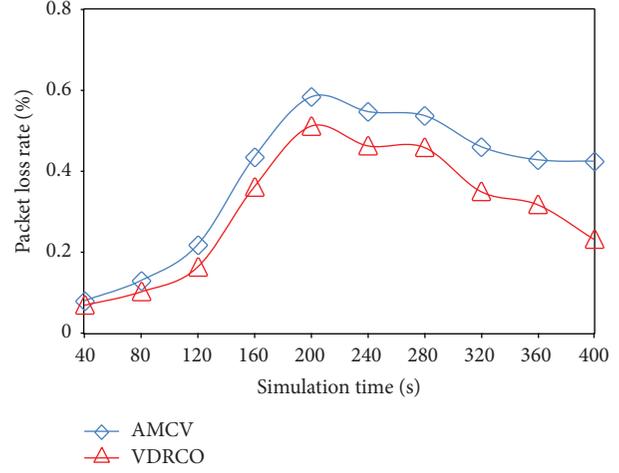


FIGURE 7: Average packet loss rate against simulation time.

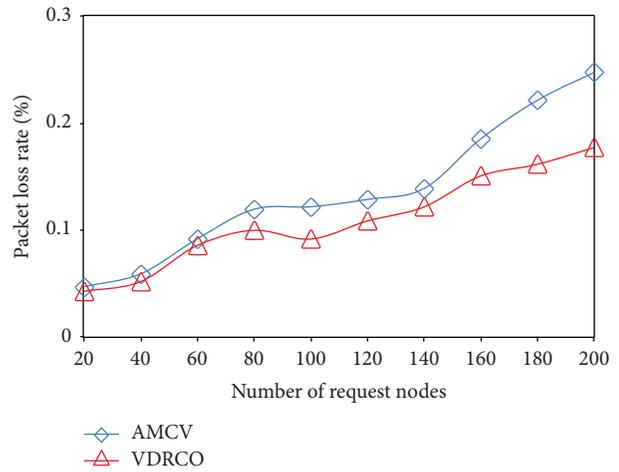


FIGURE 8: Average packet loss rate against number of request nodes.

of total number of data sent by video providers and the server denotes the packet loss rate.

Figure 7 shows the PLR variation of VDRCO and AMCV with increasing simulation time where the PLR values during a time interval  $T = 40$  s denote the PLR. The two curves corresponding to VDRCO and AMCV have similar variation process of PLR. The results of VDRCO and AMCV decrease from  $t = 200$  s to  $t = 400$  s after slow rise from  $t = 40$  s to  $t = 120$  s and fast increase from  $t = 120$  s to  $t = 200$  s. The blue curve of AMCV is higher than the red curve of VDRCO where AMCV's results have higher increment and peak value than those of VDRCO from  $t = 40$  s to  $t = 200$  s. The decrement of AMCV's results is lower than that of VDRCO from  $t = 240$  s to  $t = 400$  s.

Figure 8 shows the PLR variation of VDRCO and AMCV with increasing number of request nodes. Because the 200 system members request the videos following the Poisson distribution from  $t = 0$  s to  $t = 100$  s, the PLR results of VDRCO and AMCV with the increase in number of request nodes from  $t = 0$  s to  $t = 100$  s are shown in Figure 8. The two curves of VDRCO and AMCV have the similar variation

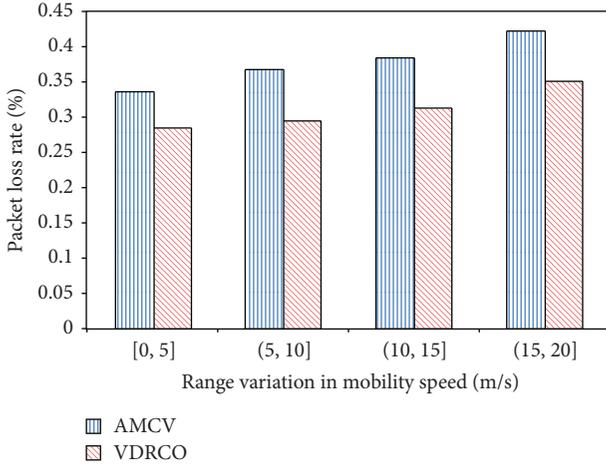


FIGURE 9: Average packet loss rate against range variation of mobile speed.

trend with variation of number of request nodes from 20 to 200. The red curve corresponding to VDRCO firstly keeps the fast rise with increasing number of request nodes from 20 to 80, experiences a transient fall from 80 to 100, and also continues to quickly increase from 120 to 200. The blue curve corresponding to AMC firstly has a fast rise with increasing number of request nodes from 20 to 80, experiences a stable process from 100 to 140, and continues to quickly increase from 160 to 200. The blue curve of AMC is higher than that of VDRCO during increase process in number of request nodes.

Figure 9 shows the PLR variation of VDRCO and AMC in different range of mobile speed. The bars of VDRCO and AMC keep the rise trend with increasing mobile speed. The red bars of VDRCO keep the low levels in the low range of mobile speed [0, 5] and (5, 10] and have the fast rise in the high range of mobile speed (10, 15] and (15, 20]. The blue bars of AMC also have low levels in the range of mobile speed [0, 5], keep the slow rise with the variation of mobile speed from (5, 10] and (10, 15], and have the distinct increase in (15, 20]. The PLR results of VDRCO are less than those of AMC in the four range of mobile speed.

AMC does not consider the negative influence caused by the node mobility for the video delivery; namely, AMC groups the system members into communities according to the videos requested by them. The forwarding process of request messages only reflects the logical distance between requesters and providers. The neglect for the geographical distance between requesters and providers leads to the increase in the probability of far-end video fetching, which brings the two types of negative influence for the video delivery. (1) The video delivery with geographical long distance increases the load core networks, which aggravates the level of network congestion. When the network congestion occurs, the PLR values of AMC keep high levels during a long period time from  $t = 200$  s to  $t = 400$  s, even if some system members continuously quit the video system. (2) There is high risk of packet loss in the process of data

transmission with geographical long distance. For instance, if the receiving buffer of a relay node is full, the relay node drops the subsequent data; if there are many relay nodes in the data transmission path, the fast movement of some relay nodes leads to the frequent change of data transmission path which increases the probability of packet loss. Therefore, AMC has the higher PLR values than VDRCO. VDRCO groups the system members into multiple subsets according to the frequency of interaction and encounter. The system members in the same subsets have stable one-hop neighbor relationship, so they achieve near-end video fetching with high probability. The near-end video fetching not only relieves the load of core network to reduce the level of network congestion but also avoids the risk of packet loss caused by node mobility in the process of data transmission. The PLR values of VDRCO keep relatively low level from  $t = 200$  s to  $t = 280$  s and fast decrease from  $t = 280$  s to  $t = 400$  s, which means that the negative influence caused by the network congestion for VDRCO is low relative to AMC. Moreover, because VDRCO considers the stability of mobility of system members in subsets (namely, stability of one-hop neighbor relationship), the PLR values of VDRCO keep low increase in the range of mobile speed [0, 5], (5, 10] and (10, 15]. Although the PLR values of VDRCO are subjected to the distinct negative influence in the high mobile speed (15, 20], VDRCO's PLR results are better than those of AMC.

## 6. Conclusion

In this paper, we propose a novel video sharing solution based on Demand-aware Resource Caching Optimization in wireless mobile networks (VDRCO). By construction of video dissemination process model in terms of the epidemic model, VDRCO discusses the main influence factors for the dissemination and distribution of videos. By estimation for user demand range and video distribution variation level, VDRCO groups the nodes into multiple subsets in terms of interaction frequency and similarity of interest and mobility, which optimizes video distribution. VDRCO further designs video sharing strategies to balance local supply and demand and make use of near-end video fetching to improve sharing performance. The simulation results also show how VDRCO has lower startup delay, higher caching utilization rate, and lower packet loss rate than AMC.

## Notations

$n_i$ :	A mobile node $i$
$v_i$ :	A video resource $i$
$N$ :	Total number of nodes in overlay networks
$N(S)$ :	Number of nodes interested in $v_i$
$N(I)$ :	Number of nodes which have cached $v_i$
$N(R)$ :	Number of nodes which lose interest for $v_i$ and have removed $v_i$
$\lambda$ :	Increasing rate of nodes which request $v_i$
$\mu$ :	Increasing rate of nodes which remove $v_i$
$S_v(n_i)$ :	Set of videos watched by $n_i$
$S_{sub_j}$ :	A video subset in $S_v(n_i)$

$S_{sk}$ :	Content similarity between $v_s$ and $v_k$
$\bar{S}(S_{\text{sub}_j})$ :	Average value of similarity among all items in $S_{\text{sub}_j}$
$l_i(v_j)$ :	Playback time of $n_i$ for $v_j$
$f_i(v_j)$ :	Frequency of $n_i$ pushing $v_j$
$\bar{I}(S_{\text{sub}_j})$ :	Average value of interest level of $n_i$ for all items in $S_{\text{sub}_j}$
$\bar{I}(S_v(n_i))$ :	Average value of interest level of $n_i$ for all items in $S_v(n_i)$
$D_i$ :	Set of demand regions of $n_i$
$TR_i(v_k)$ :	Time of downloading $v_k$ from $n_i$
$VL_k$ :	Variation level of distribution of $v_k$
$F_{ij}$ :	Total number of interaction between $n_i$ and $n_j$
$IR_{ij}$ :	Interaction success rate between $n_i$ and $n_j$
$GR_{ij}$ :	Relationship level between $n_i$ and $n_j$
$GN_i$ :	Set of nodes which encountered and interacted with $n_i$ .

## Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

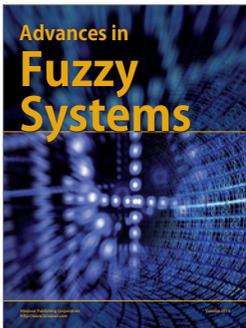
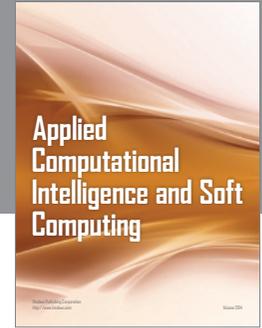
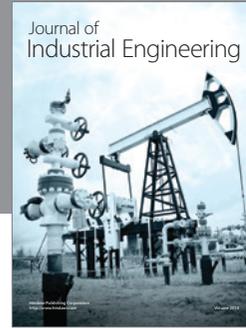
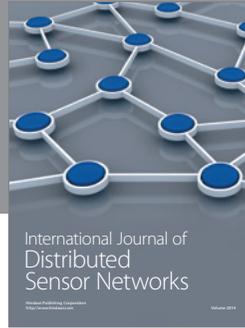
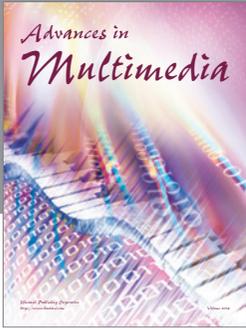
## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61501216, 61272015, 61050004, and U1404611, the Basic and Frontier Technology Research Project of Henan Province (142300410303), the Science and Technology Plan Projects (Openness & Cooperation) of Henan Province (152106000048), the Program for Science & Technology Innovation Talents in the University of Henan Province under Grant no. 16HASTIT035, and the Inter-Governmental Science and Technology Cooperation Project under Grant no. 2016YFE0104600.

## References

- [1] J. Yoon, H. Zhang, S. Banerjee, and S. Rangarajan, "Video multicast with joint resource allocation and adaptive modulation and coding in 4G networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1531–1544, 2014.
- [2] M. Hashem Eiza, Q. Ni, and Q. Shi, "Secure and privacy-aware cloud-assisted video reporting service in 5G-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7868–7881, 2016.
- [3] C. A. Chan, W. Li, S. Bian et al., "Assessing network energy consumption of mobile applications," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 182–191, 2015.
- [4] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.
- [5] L. Yang and W. Lou, "A contract-ruled economic model for QoS guarantee in mobile peer-to-peer streaming services," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1047–1061, 2016.
- [6] Y. Sun, Y. Guo, Z. Li et al., "The case for P2P mobile video system over wireless broadband networks: a practical study of challenges for a mobile video provider," *IEEE Network*, vol. 27, no. 2, pp. 22–27, 2013.
- [7] J.-L. Kuo, C.-H. Shih, C.-Y. Ho, and Y.-C. Chen, "A cross-layer approach for real-time multimedia streaming on wireless peer-to-peer ad hoc network," *Ad Hoc Networks*, vol. 11, no. 1, pp. 339–354, 2013.
- [8] S. Jia, C. Xu, A. V. Vasilakos, J. Guan, H. Zhang, and G.-M. Muntean, "Reliability-oriented ant colony optimization-based mobile peer-to-peer VoD solution in MANETs," *Wireless Networks*, vol. 20, no. 5, pp. 1185–1202, 2014.
- [9] A. Fiandrotti, V. Bioglio, M. Grangetto, R. Gaeta, and E. Magli, "Band codes for energy-efficient network coding with application to P2P mobile streaming," *IEEE Transactions on Multimedia*, vol. 16, no. 2, pp. 521–532, 2014.
- [10] C. Xu, J. Zhao, and G. Muntean, "Congestion control design for multipath transport protocols: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.
- [11] 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 Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.
- [12] O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Cooperative download in vehicular environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 663–678, 2012.
- [13] J. Zhao, P. Zhang, G. Cao, and C. R. Das, "Cooperative caching in wireless P2P networks: design, implementation, and evaluation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 229–241, 2010.
- [14] U. C. Kozat, Ö. Harmanci, S. Kanumuri, M. U. Demircin, and M. R. Civanlar, "Peer assisted video streaming with supply-demand-based cache optimization," *IEEE Transactions on Multimedia*, vol. 11, no. 3, pp. 494–508, 2009.
- [15] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.
- [16] B. Tan and L. Massoulié, "Optimal content placement for peer-to-peer video-on-demand systems," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 566–579, 2013.
- [17] L. Zhou, Z. Yang, Y. Wen, and J. J. P. C. Rodrigues, "Distributed wireless video scheduling with delayed control information," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 889–901, 2014.
- [18] J. Li, J. Sun, Y. Qian, F. Shu, M. Xiao, and W. Xiang, "A commercial video-caching system for small-cell cellular networks using game theory," *IEEE Access*, vol. 4, pp. 7519–7531, 2016.
- [19] K. Mokhtarian and H. Jacobsen, "Flexible caching algorithms for video content distribution networks," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2016.
- [20] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503–2516, 2016.
- [21] L. Zhou, Z. Yang, J. J. P. C. Rodrigues, and M. Guizani, "Exploring blind online scheduling for mobile cloud multimedia services," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 54–61, 2013.

- [22] H. Shen, Y. Lin, and J. Li, "A social-network-aided efficient peer-to-peer live streaming system," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 987–1000, 2015.
- [23] H. Shen, Z. Li, and K. Chen, "Social-P2P: an online social network based P2P file sharing system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2874–2889, 2015.
- [24] S. Zhang, Z. Shao, M. Chen, and L. Jiang, "Optimal distributed P2P streaming under node degree bounds," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 717–730, 2014.
- [25] H. Shen, Z. Li, Y. Lin, and J. Li, "Socialtube: P2P-assisted video sharing in online social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2428–2440, 2014.
- [26] C. Xu, S. Jia, L. Zhong, and G.-M. Muntean, "Socially aware mobile peer-to-peer communications for community multimedia streaming services," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 150–156, 2015.
- [27] Q. Wu, Z. Li, G. Tyson, S. Uhlig, M. A. Kaafar, and G. Xie, "Privacy-aware multipath video caching for content-centric networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2219–2230, 2016.
- [28] P. Si, H. Yue, Y. Zhang, and Y. Fang, "Spectrum management for proactive video caching in information-centric cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2247–2259, 2016.
- [29] A. Dadlani, M. S. Kumar, S. Murugan, and K. Kim, "System dynamics of a refined epidemic model for infection propagation over complex networks," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1316–1325, 2016.
- [30] R. Zhang and S. Xiong, "A novel mobile video community discovery scheme using ontology-based semantical interest capture," *Mobile Information Systems*, vol. 2016, Article ID 5279425, 12 pages, 2016.
- [31] S. Jia, C. Xu, J. Guan, H. Zhang, and G.-M. Muntean, "A novel cooperative content fetching-based strategy to increase the quality of video delivery to mobile users in wireless networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 370–384, 2014.
- [32] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.



**Hindawi**

Submit your manuscripts at  
<https://www.hindawi.com>

