

## Research Article

# A Novel Mobile Video Community Discovery Scheme Using Ontology-Based Semantical Interest Capture

Ruiling Zhang<sup>1,2</sup> and Shengwu Xiong<sup>1</sup>

<sup>1</sup>*School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei 430070, China*

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

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

Received 1 July 2016; Accepted 14 September 2016

Academic Editor: Gabriel-Miro Muntean

Copyright © 2016 R. Zhang and S. Xiong. 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.

Leveraging network virtualization technologies, the community-based video systems rely on the measurement of common interests to define and steady relationship between community members, which promotes video sharing performance and improves scalability community structure. In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI). An ontology-based semantical extension approach is proposed, which describes video content and measures video similarity according to video key word selection methods. In order to reduce the calculation load of video similarity, VCOSI designs a prefix-filtering-based estimation algorithm to decrease energy consumption of mobile nodes. VCOSI further proposes a member relationship estimate method to construct scalable and resilient node communities, which promotes video sharing capacity of video systems with the flexible and economic community maintenance. Extensive tests show how VCOSI obtains better performance results in comparison with other state-of-the-art solutions.

## 1. Introduction

The video streaming services are very popular multimedia applications in the Internet (e.g., the video traffic is now more than two-thirds of the network traffic) [1]. The convenient access of video content via mobile smart devices further promotes the development of video applications with the help of mobile networking technologies and increasing wireless bandwidth [2]. The huge traffic demand brings heavy load for the video systems, which requires scalable architecture and high-efficiency resource sharing to support large-scale deployment. Peer-to-Peer (P2P)/Mobile Peer-to-Peer (MP2P) technologies construct virtual networks in terms of resource demand of (mobile) users to manage and distribute resources [3–7]. The continuous increase in the number of nodes in video systems not only triggers huge traffic demand for video content but also increases the maintenance cost of overlay networks due to dynamic playback state of nodes. However, the traditional P2P-based video systems make use of the request-response relationship between nodes to construct overlay networks. The fragile relationship between

nodes is one of the main reasons for the frequent disconnection of logical links between them. Because the predefined relationship neglects the investigation for the resource supply and demand, it is difficult to achieve high-efficiency resource sharing and low-cost maintenance for overlay networks. This leads to the low quality of service (QoS) such as long startup delay and low system scalability.

Inspired by the investigation of relationship between nodes in social networks, the virtual community-based video systems rely on the measurement results of common interests for the video content to define relationship between community members [8–11]. As Figure 1 shows, the whole overlay network is divided into multiple communities composed of nodes with similar interests. The similar interests not only steady the logical link between community members to reduce maintenance cost for overlay networks, but also achieve efficient resource sharing to promote QoS of video systems. A key issue is how to accurately estimate the similarity of interests between nodes. The videos watched by users can reflect user interests; namely, the two users which watch the same or similar videos can be considered as having

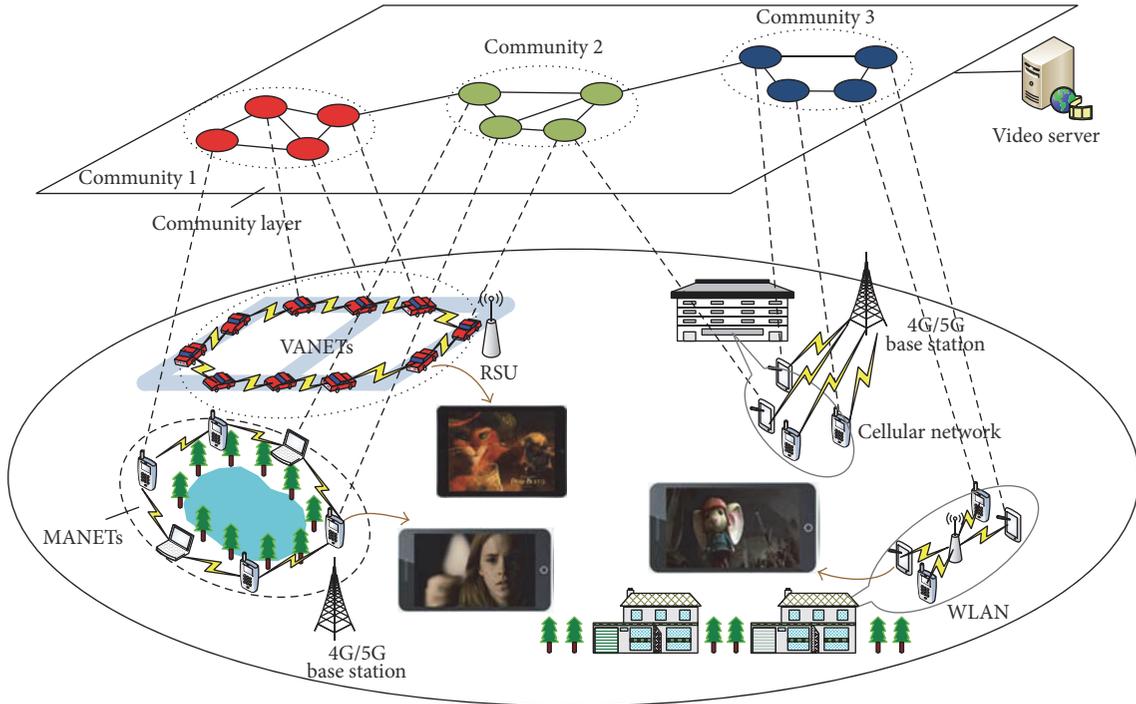


FIGURE 1: Multimedia streaming services in wireless mobile networks.

common interests. The video similarity estimation in the traditional methods only investigates file name of videos in historical playback traces of users. The file name difficultly reflects the whole video content, which cannot ensure estimation accuracy of interest similarity. The low accuracy of estimation results leads to fragile logical link between community members, which still triggers the frequent construction of community structure.

The description of video content is a key factor for the measurement accuracy of video similarity, which usually includes video name, title, type, actors, director, and abstract. The expanded video information is described by the short text. However, the noise words and valid data sparseness in the short text bring negative influence for the estimation accuracy, such as advertisement words in video abstract. A key issue is how to build a rich semantic-based description space for video content to efficiently promote estimation accuracy of video similarity. On the other hand, the expanded description brings more video information, which constructs the high-dimensional similarity measurement space composed of massive feature words. The video similarity calculation with high time complexity consumes large amount of energy of mobile nodes. Another key issue is how to design a light-duty estimation method to reduce calculation load of mobile nodes.

In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI). VCOSI employs an ontology-based semantical extension approach to describe video content and adopts three classification methods to select key words in the extended description of video content. VCOSI designs an

estimation method to calculate similarity of video content by making use of the selected key words. In order to reduce calculation complexity of similarity, VCOSI further proposes a prefix-filtering-based estimation algorithm to reduce the calculation load of mobile nodes. Based on the estimation results of video similarity, VCOSI measures closeness levels of common interests between nodes and builds scalable and resilient community structure, which promotes video sharing capacity of video systems. Simulation results show how VCOSI achieves much better performance results in comparison with other state-of-the-art solutions.

## 2. Related Work

The traditional P2P/MP2P-based video systems mainly employ structured and unstructured overlay networks to distribute video resources. The systems based on structured overlay networks, such as tree and Chord, can achieve fast resource location but need to consume large number of resources of bandwidth and computation to maintain overlay networks due to dynamic node state, which results in low scalability. The systems based on structured overlay networks do not handle real-time variation of node state, which obtains high scalability. However, the flooding-based resource lookup method wastes massive network bandwidth, which leads to network congestion and high startup delay.

The community-based video systems group the nodes into multiple communities in terms of closeness levels of relationship between them. Each community undertakes the tasks of state maintenance of intracommunity members; namely, the maintenance cost for the whole overlay networks

is distributed into multiple communities, which improves system scalability. Moreover, the tight relationship enables logical links between community members to become more stable, which further reduces maintenance cost of community structure. On the other hand, the community members store similar video resources and request similar video content with high probability in the future, so the video request of members is quickly responded by other intracommunity members, which reduces lookup delay.

SocialTube measures relationship between nodes by the investigation of number of watched videos [8]. The more the watched same videos are, the more similar the interests between nodes are. The nodes with similar interests form a community. The source nodes push the interested videos to the community members. However, it is difficult to obtain accurate measurement results of similarity by making use of number of watched videos to estimate interest similarity. This results in the low video push success rate and fragile logical link between nodes. SPOON also relies on investigating file name vector in historical playback traces to capture common interests between nodes [9]. The mobile nodes with similar interests are grouped into communities. The designed role assignment method distributes the maintenance cost for the community structure into multiple community members, which balances the load of mobile nodes and improves the community scalability. However, the interest similarity measurement based on file name difficultly ensures the estimation accuracy of common interests between nodes.

In order to address the problems of low estimation accuracy in traditional methods, some common interest capture methods expand the description of video content (e.g., video name, title, type, actors, director, and abstract). The expanded description enriches video information and is denoted by the short text, which increases the measurement accuracy of similarity between videos. However, the short text includes large number of noise words, which severely influences measurement accuracy of video similarity. The existing studies make use of semantic dictionary, feature extension, and topic model to address the problems of noise words.

(1) *Semantic Dictionaries*. The methods based on semantic dictionaries, such as WordNet and HowNet, can accurately estimate similarity of word pairs by the conversion from short text to word pairs. Mihalcea et al. [12] considered the weighted average values of similarities among word pairs as the similarities of short text. Li et al. [13] combine semantic and sequence similarity to calculate similarity of short text. Based on the work in [13], Islam and Inkpen [14] make use of the weighted sum of string similarity, semantic similarity, and sequence similarity to calculate similarity of short text. However, the semantic dictionary always is incomplete due to the continuous occurrence of new words, which cannot ensure high-accuracy similarity measurement results of short text.

(2) *Feature Extension*. Sahami and Heilman employed a similarity kernel function to estimate the short text similarity by making use of the search engine to extend features of short text [15]. Wang et al. [16] and Yuan [17] proposed a mining algorithm based on association rule to extract association

relationship between features included in training and testing sets, which further obtains the extended features corresponding to the words. Genc et al. built the mapping relationship between short text and Wikipedia pages to calculate similarities of short text [18]. Wang et al. [19] proposed a unified framework to expand short texts by making use of the convolutional neural network, which can address the problems of sparsity of short texts and semantic sensitivity. However, the above methods still do not avoid the negative influence caused by noisy words.

(3) *Topic Model*. Quan et al. make use of topic model based on the Latent Dirichlet Allocation (LDA) to calculate similarity of short text, which improves measurement accuracy based on the vector space model (VSM) model [20]. Phan et al. employed a similarity estimation method of short text based on the probability distribution of document topic [21]. Zhang and Zhong collected large-scale external data to build the topic model according to the LDA, which enables word topics to enrich feature representations of short text [22]. Vo and Ock employed a LDA-based method to discover hidden topic from universal datasets (e.g., Computer Science Bibliography), Lecture Notes in Computer Science book series (LNCS), and Wikipedia [23].

In addition to the above methods, there are many studies related to similarity measurement of short text, such as deep learning based methods [24–27], Earth Mover’s Distance- (EMD-) based method [28] and multilevel sentence similarity calculation [29].

### 3. VCOSI Detailed Design

3.1. *Video Similarity Measure Based on Semantic Feature Extension*. The noisy words in short text corresponding to video information do not reflect real video content, which results in severely negative influence for the estimation accuracy of video similarity. Therefore, it is essential to eliminate noisy words and select key words in short texts corresponding to video content (the key words can embody real video content). The video information mainly includes title, actor, director, and tag and is denoted by short text. For instance, if the two videos have many cooccurrence words in title, actor, and director of videos, they may have similar content (e.g., the TV series). The tags are the artificial sign to represent valuable information such as category and plot themes of videos. The above video information is used to recognize the key words; namely, *isTitleTerm*, *isActorTerm*, *isDirectorTerm*, and *isTag* are defined as the features of key words. Moreover, we also consider position and frequency of word as the features of key words. For instance, if a word has high frequency of occurrence in title and tag of video, it may carry important information for video content and is considered as the key words. Therefore, *termPOS*, *termFreq*, and *locToTitle* also are defined as the features of key words. Table 1 lists name and description of features of key words.

The key words disperse in short text, so the selection of key words is considered as the classification of words. In order to accurately identify the key words, we choose three kinds of classification methods including decision tree, SVM, and

TABLE 1: Features of key words of video content.

ID	Feature name	Note
<i>fea1</i>	<i>isTitleTerm</i>	Video title word
<i>fea2</i>	<i>isActorTerm</i>	Video actor word
<i>fea3</i>	<i>isDirectorTerm</i>	Video director word
<i>fea4</i>	<i>isTag</i>	Video tag word
<i>fea5</i>	<i>termPOS</i>	Part of speech of a word
<i>fea6</i>	<i>termFreq</i>	Word frequency in video description
<i>fea7</i>	<i>locToTitle</i>	Location in video description

TABLE 2: Accuracy rate of different classification methods.

Features	Classification methods		
	Decision tree	SVM	KNN
All features	45.16%	62.90%	67.74%
<i>fea1</i> – <i>fea6</i>	48.39%	66.13%	<b>82.26%</b>
<i>fea1</i> – <i>fea5</i>	48.39%	<b>74.19%</b>	58.06%
<i>fea1</i> , <i>fea5</i> , <i>fea6</i> , <i>fea7</i>	45.16%	62.90%	66.13%
<i>fea6</i>	46.77%	66.13%	53.23%

KNN. Table 2 lists the accuracy rate of decision tree, SVM, and KNN. Obviously, KNN and SVM have higher accuracy rate than that of decision tree. For instance, the maximum values of KNN and SVM are 88.26% and 74.19%, respectively. Because KNN has the best accuracy rate among decision tree, SVM, and KNN in the range from *fea1* to *fea6*, we use KNN based on the range from *fea1* to *fea6* to select the key words in the following experiments. Because KNN is a classical classification method, the classification process is no longer described in detail.

On the other hand, if the short text corresponding to video information only includes small number of valuable key words, the selected key words also cannot reflect the whole video content. We employ an ontology-based feature extension method to enrich key words because the vector space model does not deal with the synonym, hypernym, and hyponym in key words. The ontology is constructed by

domain experts, which includes rich semantic information based on the specific background. The text label, comments, instances, properties, and relations between other concepts are used to denote a concept in the ontology. We make use of the information of ontology to extend the key words of video content. Let  $R_k$  be the set of key words of a video  $v_k$ ; namely,  $R_k = \{r_1, r_2, \dots, r_i, \dots, r_m\}$ . Let  $F_e(r_i)$  denote the set of extended words of a key word  $r_i$  based on the built ontology information, which is defined as

$$F_e(r_i) = F_{\text{label}}(r_i) \cup F_{\text{structure}}(r_i) \cup F_{\text{property}}(r_i) \cup F_{\text{instance}}(r_i), \quad (1)$$

where  $F_{\text{label}}(r_i)$  is the set of literal features and includes key words of text label and comments of  $r_i$ ;  $F_{\text{structure}}(r_i)$  is the set of structure features and includes key words in ancestor of concept  $r_i$ ;  $F_{\text{property}}(r_i)$  is the set of property features and includes key words in the properties of concept  $r_i$ ;  $F_{\text{instance}}(r_i)$  is the set of instance features and includes key words in the instances of concept  $r_i$ . For example, a video, called “angry birds,” has the four feature sets corresponding to the key word “birds”; namely,  $F_{\text{label}}(\text{birds}) = \{\text{bird, warm-blooded, egg-laying, vertebrates, characterized, by, feathers, and, forelimbs, modified, as, wings}\}$ ,  $F_{\text{structure}}(\text{birds}) = \{\text{entity, physical, entity, object, whole, living, thing, organism, animal, chordate, vertebrate}\}$ ,  $F_{\text{property}}(\text{birds}) = \{\text{warm-blooded, egg-laying}\}$ , and  $F_{\text{instance}}(\text{birds}) = \{\text{Red, Chuck, Bomb, Matilda, Mighty Eagle}\}$ . Obviously, the ontology-based semantic description of video content efficiently enriches key words to reduce the probability of erroneous judgement.

The extended words of all items in  $R_k$  form a new set  $R_e = F_e(r_1) \cup F_e(r_2), \dots, F_e(r_{m-1}) \cup F_e(r_m)$ .  $R_e$  can be used to measure the similarity between videos. The traditional Tversky similarity model relies on common and distinctive features between two objects to estimate similarity. However, the users may focus more on common features in the process of similarity estimation. Therefore, we improve the Tversky similarity model and make use of the set  $R_e$  to estimate the content similarity between any two videos  $v_i$  and  $v_j$  according to the following equation.

$$\text{Sim}(v_i, v_j) = \text{Sim}(R_e(v_i), R_e(v_j)) = \frac{\alpha |R_e(v_i) \cap R_e(v_j)|}{\alpha |R_e(v_i) \cap R_e(v_j)| + \beta |R_e(v_i) - R_e(v_j)| + (1 - \beta) |R_e(v_j) - R_e(v_i)|}, \quad (2)$$

where  $\alpha$  and  $\beta$  are the weight factors added by us, which are used to adjust common and different features. For instance, we assume that  $R_e(v_i)$  is a subset of  $R_e(v_j)$  ( $R_e(v_i) \subset R_e(v_j)$ ) and the size of  $R_e(v_i)$  is much smaller than that of  $R_e(v_j)$  ( $|R_e(v_i)| \ll |R_e(v_j)|$ ). The similarity value of  $R_e(v_i)$  and  $R_e(v_j)$  based on the Jaccard similarity measure is very small; namely,  $R_e(v_i)$  and  $R_e(v_j)$  should be similar to some extent. Therefore, we use (2) to increase the similarity between  $R_e(v_i)$  and  $R_e(v_j)$  by the adjustment of  $\alpha$  and  $\beta$ . For

instance, we assume that  $R_e(v_i) = \{w_1, w_2, w_4\}$ ,  $R_e(v_j) = \{w_2, w_3, w_5, \dots, w_{m-1}, w_m\}$ , and  $|R_e(v_i) \cap R_e(v_j)| = |\{w_2\}| = 1$ ,  $|R_e(v_i) - R_e(v_j)| = |\{w_1, w_4\}| = 2$ , and  $|R_e(v_j) - R_e(v_i)| = |\{w_3, w_5, \dots, w_{m-1}, w_m\}| = 50$ . The Jaccard similarity of  $R_e(v_i)$  and  $R_e(v_j)$  is defined as  $\text{Jaccard}(R_e(v_i), R_e(v_j)) = 1/(1 + 2 + 50) = 0.018$ . According to (2), The similarity of  $R_e(v_i)$  and  $R_e(v_j)$  is defined as  $\text{Sim}(R_e(v_i), R_e(v_j)) = (\alpha * 1)/(\alpha * 1 + \beta * 2 + (1 - \beta) * 50) = 0.057$ , where  $\alpha$  is set to 1 and  $\beta$  is set to 0.7. Obviously,  $\text{Sim}(R_e(v_i), R_e(v_j)) > \text{Jaccard}(R_e(v_i), R_e(v_j))$ .

**3.2. Prefix-Filtering-Based Similarity Estimation Algorithm.** The extension of key words increases the available information for video content to improve measurement accuracy of video similarity. However, the increase in the extension scale of key words results in high computation complexity due to large number of duplicated comparisons in the process of similarity estimation. The high computation load consumes large amount of energy of mobile nodes. The prefix filtering technique based on the Jaccard similarity in [30] can efficiently reduce computation complexity. Because the video similarity  $\text{Sim}(v_i, v_j)$  in (2) is different with the Jaccard similarity, we derive the relationship of the Jaccard similarity and  $\text{Sim}(v_i, v_j)$  and convert  $\text{Sim}(v_i, v_j)$  to the Jaccard similarity. Further, we design a bloom-filter-based duplicated comparison free method, which reduces the computation load by prevention of duplicated comparisons.

### 3.2.1. Prefix Filtering Based on Derived Jaccard Similarity Threshold

**Lemma 1.** When  $\alpha + \beta \leq 1$  and  $\beta \leq 1/2$ , if  $J(v_i, v_j) < ((1 - \beta)/\alpha)\epsilon$ , then  $\text{Sim}(v_i, v_j) < \epsilon$ , where  $J(v_i, v_j)$  is the Jaccard similarity between  $v_i$  and  $v_j$ :

$$J(v_i, v_j) = \frac{|F(v_i) \cap F(v_j)|}{|F(v_i) \cap F(v_j)| + |F(v_i) - F(v_j)| + |F(v_j) - F(v_i)|}, \quad (3)$$

where  $\epsilon$  is the similarity threshold. If the similarity  $\text{Sim}(v_i, v_j)$  between  $v_i$  and  $v_j$  is smaller than  $\epsilon$ ,  $v_i$  and  $v_j$  are not similar.

*Proof.* For simplicity, let  $x$ ,  $y$ , and  $z$  represent  $|F(v_i) \cap F(v_j)|$ ,  $|F(v_i) - F(v_j)|$ , and  $|F(v_j) - F(v_i)|$ , respectively. The similarity between  $v_i$  and  $v_j$  can be redescribed as  $\text{Sim}(v_i, v_j) = \alpha x / (\alpha x + \beta y + (1 - \beta)z)$ .

$$\begin{aligned} \because \alpha + \beta &\leq 1 \\ \therefore \alpha &\leq 1 - \beta \\ \because \beta &\leq \frac{1}{2} \\ \therefore \beta &\leq 1 - \beta. \end{aligned} \quad (4)$$

So we can get

$$\begin{aligned} \text{Sim}(v_i, v_j) &= \frac{\alpha x}{\alpha x + \beta y + (1 - \beta)z} \\ &\leq \frac{\alpha x}{(1 - \beta)x + (1 - \beta)y + (1 - \beta)z} \\ &= \frac{\alpha}{1 - \beta} \frac{x}{x + y + z} = \frac{\alpha}{1 - \beta} J(v_i, v_j) \end{aligned} \quad (5)$$

$$\because J(v_i, v_j) < \frac{1 - \beta}{\alpha} \epsilon \implies$$

$$\text{Sim}(v_i, v_j) < \frac{\alpha}{1 - \beta} J(v_i, v_j) = \frac{\alpha}{1 - \beta} \frac{1 - \beta}{\alpha} \epsilon = \epsilon. \quad \square$$

**Lemma 2.** When  $\alpha + \beta > 1$  and  $\alpha < \beta$ , if  $J(v_i, v_j) < (\alpha/\beta)\epsilon$ , then  $\text{Sim}(v_i, v_j) < \epsilon$ .

*Proof.* For simplicity, we still use  $x$ ,  $y$ , and  $z$  to represent  $|F(v_i) \cap F(v_j)|$ ,  $|F(v_i) - F(v_j)|$ , and  $|F(v_j) - F(v_i)|$ , respectively. The similarity between  $v_i$  and  $v_j$  can be redescribed as  $\text{Sim}(v_i, v_j) = \alpha x / (\alpha x + \beta y + (1 - \beta)z)$ .

$$\begin{aligned} \because \alpha + \beta &> 1, \\ \alpha &< \beta \\ \therefore 1 &< 2\beta \implies \\ \beta &> \frac{1}{2} \implies \\ 1 - \beta &< \beta. \end{aligned} \quad (6)$$

So we can get

$$\begin{aligned} \text{Sim}(v_i, v_j) &= \frac{\alpha x}{\alpha x + \beta y + (1 - \beta)z} < \frac{\alpha x}{\beta x + \beta y + \beta z} \\ &= \frac{\alpha}{\beta} \frac{x}{x + y + z} = \frac{\alpha}{\beta} J(v_i, v_j) \end{aligned} \quad (7)$$

$$\because J(v_i, v_j) < \frac{\beta}{\alpha} \epsilon \implies$$

$$\text{Sim}(v_i, v_j) < \frac{\alpha}{\beta} J(v_i, v_j) < \frac{\alpha}{\beta} \frac{\beta}{\alpha} \epsilon = \epsilon. \quad \square$$

**3.2.2. Video Similarity Estimation.** Let  $\text{VT}(n_i)$  and  $\text{VT}(n_j)$  denote the set of videos watched by the two nodes  $n_i$  and  $n_j$ . The extended feature set of each video in  $\text{VT}(n_i)$  and  $\text{VT}(n_j)$  can be obtained according to (1), recorded as  $S$  and  $T$ , respectively (line 4). In order to improve the filtering effect, we construct a global ordering  $\mathcal{O}$  to investigate the feature frequency (line 5). The corresponding Jaccard similarity threshold  $\epsilon'$  can be calculated according to Lemmas 1 and 2. For instance, when  $\alpha + \beta \leq 1$  and  $\beta \leq 1/2$ ,  $\epsilon' = (1 - \beta)/\alpha$  (line 6). According to the principle of the prefix filtering scheme [30], we can obtain the prefix of each  $s_a \in S$ , recorded as  $\text{Pre}(s_a)$ .  $\text{Pre}(s_a)$  refers to the first  $|\text{Pre}(s_a)|$  items of  $s_a$ ,  $|\text{Pre}(s_a)| = |s_a| - \lceil \epsilon' |s_a| \rceil + 1$ . For instance, we assume that  $s_a = \{A, C, D, B, G, K\}$ ,  $\epsilon' = 0.45$ ; then  $|\text{Pre}(s_a)| = 3$ ,  $\text{Pre}(s_a) = \{A, C, D\}$ . The inverted index  $I_S$  based on all prefixes in  $S$  is constructed (line 7). The indexed terms in  $I_S$  are unique terms of prefixes in  $S$ , where each indexed term refers to a list composed of all videos whose prefixes contain the corresponding indexed term. In the following stage,  $s_p$  may be possible similar to  $t_b \in T$  by the inverted index  $I_S$  (line 8–19). For each item  $t_b \in T$ , we firstly figure out the prefix  $\text{Pre}(t_b)$ . For each term  $w_k \in \text{Pre}(t_b)$ , we obtain the corresponding inverted index list  $I_{w_k}$  in  $I_S$ . By traversing all items in  $I_{w_k}$  for each  $s_p \in I_{w_k}$ , the similarity of  $t_b$  and  $s_p$  can be calculated. If  $\text{Sim}(t_b, s_p) \geq \epsilon$ ,  $t_b$  and  $s_p$  are similar. We can add  $\langle t_b, s_p \rangle$  to the similar concept set  $\Theta$ . The pseudocode of the process of prefix-filtering-based video similarity estimation is detailed in Algorithm 1.

```

(1) Input: two video sets  $VT(n_i)$  and  $VT(n_j)$ ; similarity threshold  $\epsilon$ ;  $\alpha$  and  $\beta$ ;
(2) Output: similar video pairs set  $\Theta$ ;
(3)  $\Theta \leftarrow \emptyset$ ;
(4) gets extended feature set of each video from  $VT(n_i)$  and  $VT(n_j)$ , recorded as
     $S$  and  $T$ ,  $S = \{s_1, s_2, \dots, s_m\}$ ,  $T = \{t_1, t_2, \dots, t_n\}$ ;
(5) gets global ordering  $\mathcal{O}$  according to term frequency;
(6) gets Jaccard similarity threshold  $\epsilon'$  according to the parameters  $\alpha$ ,  $\beta$  and  $\epsilon$ .
    //refer to Lemmas 1 and 2
(7) constructs inverted index  $I_S$  based on the prefix of  $s_a$ ,  $s_a \in S$ ,  $a \in [1, m]$ ;
(8) for each  $t_b$  of  $T$ ,  $b \in [1, n]$ 
(9)     gets prefix of  $t_b$ , recorded as  $Pre(t_b)$ ;
(10)    for each term  $w_k \in Pre(t_b)$ 
(11)        gets according inverted index list  $I_{w_k}$ ;
(12)        for each  $s_p \in I_{w_k}$ 
(13)            calculates similarity of  $t_b$  and  $s_p$ ;
(14)            if  $(Sim(t_b, s_p) \geq \epsilon)$ 
(15)                 $\Theta \leftarrow \Theta \cup \langle t_b, s_p \rangle$ ;
(16)            end if
(17)        end for
(18)    end for
(19) end for

```

ALGORITHM 1: Prefix-filtering-based video similarity estimation.

### 3.3. Construction and Maintenance of Video Communities.

The mobile nodes joined into systems make use of the watched video similarity to calculate the similarity levels of interests. Let  $VT(n_i) = \{(v_a, t_a), \dots, (v_h, t_h)\}$  and  $VT(n_j) = \{(v_c, t_c), \dots, (v_m, t_m)\}$  denote historical playback traces of  $n_i$  and  $n_j$ , respectively.  $v_c$  and  $t_c$  are the video ID and time of playing video  $c$  for  $n_j$ .  $n_i$  and  $n_j$  exchange the information of  $VT(n_i)$  and  $VT(n_j)$  with each other and calculate the similarity of  $VT(n_i)$  and  $VT(n_j)$ . For instance,  $n_i$  calculates similarity value  $S(v_a, v_c)$  between  $v_a$  and  $v_c$  and mean value  $\bar{t}_{ac}$  of playback time ratio  $t_a = l_a/L_a$  and  $t_c = l_c/L_c$ , namely, two-tuple  $ST_{ac} = (S(v_a, v_c), \bar{t}_{ac})$ .  $l_a$  and  $L_a$  are watched time and length of  $v_a$ , respectively;  $l_c$  and  $L_c$  are watched time and length of  $v_c$ , respectively.  $n_i$  can obtain a two-tuple set  $ST$  of all videos in  $VT(n_i)$  and  $VT(n_j)$ . We use the Least Square Method (LSM) [31] to estimate the correlation coefficient  $R_{ij}$  of items in  $ST$  according to the following equation:

$$R_{ij} = \frac{\sum_{e=1}^k |S_e - \bar{S}_{ij}| \times |\bar{t}_e - \bar{t}_{ij}|}{\sqrt{\sum_{e=1}^k |S_e - \bar{S}_{ij}|^2} \times \sqrt{\sum_{e=1}^k |\bar{t}_e - \bar{t}_{ij}|^2}}, \quad (8)$$

where  $k$  is the number of items in  $ST$ ;  $S_e$  and  $\bar{t}_e$  are video similarity value and average playback time ratio of items in  $ST$ , respectively;  $\bar{S}_{ij}$  and  $\bar{t}_{ij}$  are the mean value of similarity and playback time ratio among all videos in  $VT(n_i)$  and  $VT(n_j)$ , respectively.  $R_{ij}$  is considered as the interest similarity value between  $n_i$  and  $n_j$ . If  $R_{ij} > \bar{S}_{ij}$ ,  $n_i$  and  $n_j$  have similar interest. We introduce the two discovery methods of community members, as follows.

(1) The nodes make use of the message exchange to find the nodes with similar interests in one-hop neighbor nodes.

For instance,  $n_i$  broadcasts the messages including the own playback trace to all one-hop neighbor nodes. If the neighbor nodes of  $n_i$  do not join any community, they estimate the interest similarity value with  $n_i$ . If there are similar interests between them, they return the acknowledgment messages to  $n_i$ .  $n_i$  and the neighbor nodes with similar interests form a community or the neighbor nodes join into the community corresponding to  $n_i$ .

(2) If  $n_i$  sends a request message to a supplier  $n_j$  of  $v_k$  or receives a request message from  $n_j$  where the message includes the playback trace of  $n_j$ ,  $n_i$  calculates the interest similarity value with  $n_j$ . If  $n_i$  and  $n_j$  have similar interest and  $n_j$  does not join any community,  $n_i$  invites  $n_j$  to join the community to  $n_i$  by the message exchange.

If the target node  $n_j$  invited by  $n_i$  has joined other communities,  $n_j$  decides whether to join the community to  $n_i$ . Let  $R_{uj}$  denote the interest similarity value between  $n_j$  and a corresponding node  $n_u$  in current community. If  $R_{uj} > R_{ij}$ ,  $n_j$  stays in current community; otherwise, if  $R_{uj} < R_{ij}$ ,  $n_j$  quits current community and joins new community to  $n_i$ . Obviously, the movement of nodes between communities leads to dynamic community structure and increases the load of community maintenance but also can be considered as a continuous optimization process of community structure. In other words, the nodes always seek the community which has the most similar interest value.

Except for the enter and departure of nodes, the community member role assignment and the community structure maintenance are very important for the video sharing performance. Due to the variation of community structure caused by the enter and departure of nodes, the community members need to maintain community structure and the maintenance load is assigned to the specific members. The real-time and

random variation of community structure bring huge maintenance load. In order to ensure community scalability, the maintenance load should be distributed to multiple members. Because the members join communities by the invitation with each other, there are the logical links among members. Therefore, the community members form an undirected and connected graph  $G = (E, V)$ . By the collection of video lookup-related communication paths between nodes in  $G$ , there is the member  $n_h$  which has the most occurrence frequency in all communication paths of a community. The high-frequency occurrence in communication paths means that  $n_h$  participates in most of message forwarding of video lookup.  $n_h$ 's neighbor nodes in  $G$  become the broker nodes of community and  $n_h$  becomes the head node of all broker nodes.

Because all paths include  $n_h$  in  $G$ , the paths are divided into multiple subpaths which do not have  $n_h$ . The neighbor nodes of  $n_h$  are responsible for maintaining the state of other members in the paths including them.  $n_h$  acts as the community interface to contact other communities. If a neighbor node  $n_s$  of  $n_h$  quits the system or current community,  $n_h$  reselects a neighbor node of  $n_s$  as the broker member. If  $n_h$  leaves current system or current community, the member which has the highest occurrence frequency among all neighbor nodes in  $n_h$  becomes new head node. All broker nodes exchange the information of community members during a period of time.

When a node  $n_i$  in a community requests a video  $v_k$ , it sends a request message to the contacted broker node  $n_p$ . If  $n_p$  is aware of the supplier cached  $v_k$ ,  $n_p$  directly forwards the request message to the supplier. Otherwise, if  $n_p$  does not store the information of suppliers cached  $v_k$ ,  $n_p$  forwards the request message to the head node  $n_h$ .  $n_h$  makes use of the maintained interface information between communities to help  $n_i$  search  $v_k$ . If  $n_i$  quits the system or joins other communities,  $n_i$  needs to send a message to inform  $n_p$ .  $n_p$  removes information of  $n_i$  from local member list. If a mobile node or a member in other communities joins current community, the inviter sends a message containing the information of new member to the corresponding broker node. The latter updates local member list.

## 4. Testing and Test Results Analysis

### 4.1. Testing Topology and Scenarios

**4.1.1. Testing Scenarios of Video Similarity.** The common interest of users is the key factor for the tight levels between community members, which determines the stability of community structure. We firstly compare the measurement performance of video similarity between the proposed ontology-based semantical estimation method, Semantic Feature Extension based Similarity Measure (SFEbSM) and Semantic Feature Extension based Similarity Measure using Key Word Selection (SFEbSMKS), and the two classical methods, Jaccard Similarity Measure using FileName only (JVSMfn) and Jaccard Similarity Measure using all features (JVSMaf).

Because there is no existing benchmark for video similarity estimation, we construct an annotated dataset composed

TABLE 3: 5000 video pairs with volunteer ratings.

Video A	Video B	Volunteer ratings
video 1	video 2	0.12
video 1	video 3	0.32
video 4	video 10	0.52
video 4	video 29	0.31
video 10	video 21	0.71
video 10	video 28	0.65
video 100	video 201	0.54
video 100	video 324	0.16
video 124	video 401	0.62
video 135	video 552	0.87
video 351	video 601	0.36
video 361	video 701	0.33
...	...	...

of 4000 videos selected from the internet. The attributes of videos include video title, video description, video type, tags, directors, and actors. We invite 100 student volunteers to estimate the similarity value for 5000 pairs of videos according to the video attributes where the similarity value range is between 0 and 1. The similarity value can be any real number with two decimal places between 0 and 1. Table 3 shows a part of evaluation results of volunteer for the 5000 video pairs.

**4.1.2. Testing Scenarios of Video Sharing.** We make use of the estimation results of video similarity to build the communities and further compare the video sharing performance of VCOSI with a state-of-the-art solution SPOON [9].

VCOSI and SPOON were modeled and implemented in NS-2. 500 mobile nodes are deployed in a wireless mobile network whose area is set to  $1000 \times 1000 \text{ m}^2$ . The mobile speed range of nodes is in the range  $[1, 20] \text{ m/s}$ . The simulation time is set to 500 s. The signal range of mobile nodes is set to 200 m. The transmission protocol of video data is TCP and the wireless routing protocol is DSR. The bandwidth of mobile nodes and server is set to 10 Mb/s and 20 Mb/s, respectively. The server acts as initial provider of video resources. The sufficient bandwidth of server not only avoids the overload caused by large-scale access due to low resource supply capacity in overlay network during initial simulation but also promotes the resource distribution. The mobile nodes request video and act as core network in the built wireless mobile environment. The settings of bandwidth of mobile nodes avoid severe congestion caused by large-scale request, which does not result in super-long congestion recovery time. The transmission rate of video data is set to 128 kb/s. The two solutions employ the same network environment. Moreover, the number of video files requested by all nodes is set to 20 and the length of each file is 180 s.

We created 300 historical playback traces for the 300 mobile nodes where the historical playback traces have video structure and different attribute values. Further, we also generated 300 playback logs for the 300 mobile nodes where the playback logs include the ID and time of watched videos. After the nodes join the system, they play videos in terms of

the playback logs. Once the nodes finish the playback for the videos in terms of the defined playback time, they continue to request new videos according to the playback logs. The surplus 200 mobile nodes do not join the system and are responsible for forwarding request messages and video data. We further describe the behaviours of 300 mobile nodes which join the system. Before the implementation of simulation, we assign the 100 historical playback traces for 100 mobile nodes. We make use of the proposed community construction method based on the measurement results of interest similarity to group the 100 mobile nodes into multiple communities. The members in the built communities request videos following Poisson distribution in terms of the playback logs and provide video resources for the request nodes. After the beginning of simulation, 100 mobile nodes join the system and request videos following Poisson distribution from the simulation time  $t = 200$  s to  $t = 500$  s. After the 100 nodes join the system, they join the corresponding communities in terms of the interest similarity values. In VCOSI, the surplus 100 mobile nodes do not actively request videos and join the system. Once they receive the invitation from the community members, they join the corresponding communities and request and play videos according to the assigned playback logs. In SPOON, the surplus 100 mobile nodes are randomly assigned the time of joining the system; namely, they randomly join the system from  $t = 0$  s to  $t = 500$  s according to the assigned time and play videos according to the playback logs.

We define a random mobility model for all mobile nodes. The mobile nodes move from current location to target location in terms of the specific speed. After the mobile nodes arrive at the target location, the mobile nodes move to new target location in terms of new speed. The speed and initial and target location of mobile nodes are randomly assigned.

#### 4.2. Performance Evaluation of Video Similarity Measurement

**4.2.1. Estimate Accuracy of Video Similarity.** We compare accuracy of similarity estimation of four methods: SFEbSM, SFEbSMKS, JVSMfn, and JVSMaf. Based on the similarity measurement results of video pairs by student volunteers, we firstly calculate the Pearson correlation values according to the following:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}. \quad (9)$$

We make use of the calculated results of the Pearson correlation to further evaluate the accuracy of the proposed similarity measures and compare with other approaches.

Figure 2 shows the correlation coefficient with different  $\alpha$  and  $\beta$ . The correlation coefficient is higher with  $\alpha \in [1.0, 1.8]$  and  $\beta \in [0.5, 0.8]$ , which means that the common features are more important than the different features for the similarity measure methods. Moreover, when  $\alpha = 1.5$  and  $\beta = 0.5$ , the correlation coefficient reaches the maximum value 0.919. Therefore, the values of  $\alpha = 1.5$  and  $\beta = 0.5$  are 1.5 and 0.5 in the following experiments, respectively.

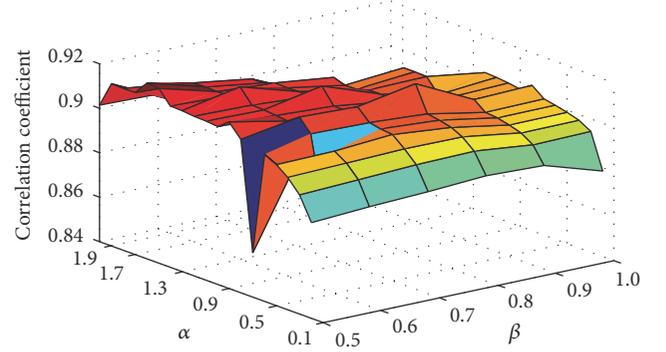


FIGURE 2: Correlation coefficient versus  $\alpha, \beta$ .

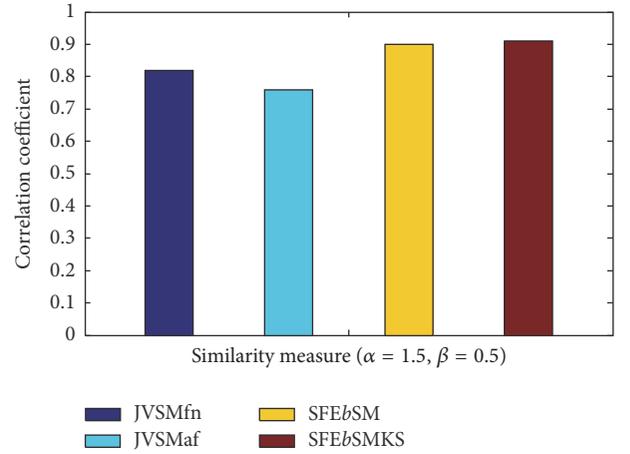


FIGURE 3: Correlation coefficient versus different similarity measure.

Figure 3 shows the measurement results of correlation coefficient of the four methods by using WordNet. Although JVSMfn only uses video name to estimate similarity between videos and JVSMaf investigates multiple features, JVSMaf cannot efficiently eliminate noisy words in the features. Therefore, the measure results of JVSMaf are lower than those of JVSMfn. SFEbSMKS and SFEbSM employ the semantic feature extension method to estimate the similarity of video content; they can obtain higher similarity accuracy than JVSMaf and JVSMfn by efficient elimination of noisy words and extraction of key words. Moreover, SFEbSMKS rely on the selection of key words based on the KNN classification to obtain better extraction effect of noisy words than SFEbSM, so that the estimation accuracy of SFEbSMKS is higher than that of SFEbSM.

**4.2.2. Run Time.** The execution time of similarity estimation is defined as the run time, in order to compare efficiency of the proposed prefix-filtering-based algorithm (Prefix) with the brute force method (Bruteforce).

As Figure 4 shows, the two curves corresponding to Prefix and Bruteforce have the same rise trend with the increase in the data size. The green curve of Prefix is lower than the blue curve of Bruteforce. The increment and peak value of Prefix are less than those of Bruteforce.

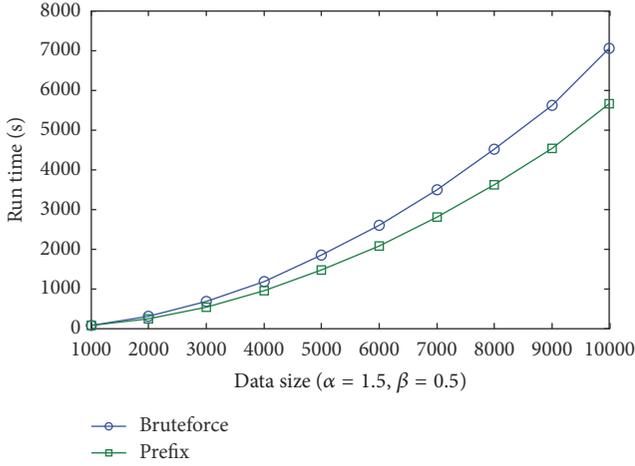


FIGURE 4: Run time with different data size.

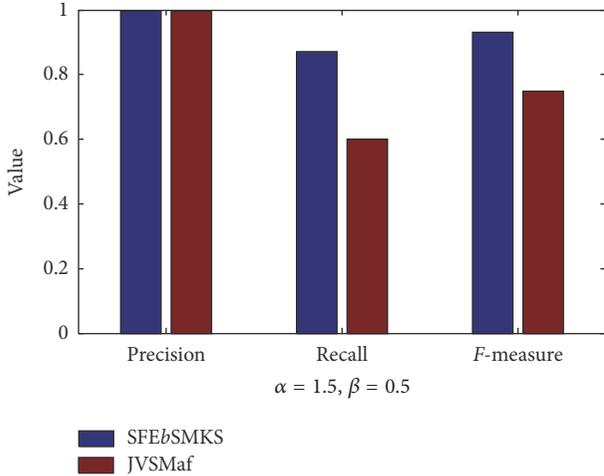


FIGURE 5: Precision, recall, and  $F$ -measure with different similarity measure.

Prefix makes use of the prefix filtering method to decrease the large number of repeated comparisons. In the other words, Prefix eliminates more repeated video pairs than those of Bruteforce. The computation complexity of Prefix is lower than that of Bruteforce, so that the run time of Prefix is less than that of Bruteforce.

**4.2.3. Precision, Recall, and  $F$ -Measure.** Figure 5 shows the precision, recall, and  $F$ -measure of SFEbSMKS and JVSMaf with  $\alpha = 1.5$  and  $\beta = 0.5$ . The precision results of SFEbSMKS and JVSMaf are 0.997 and 0.996, respectively. While the recall and  $F$ -measure of SFEbSMKS are higher than those of JVSMaf, obviously, the precision, recall, and  $F$ -measure of SFEbSMKS are better than those of JVSMaf. SFEbSMKS depends on the semantic feature extension and the key word selection to obtain better performance of similarity measure than JVSMaf.

**4.3. Performance Evaluation of Video Sharing.** The performance of VCOSI is compared with that of SPOON in terms

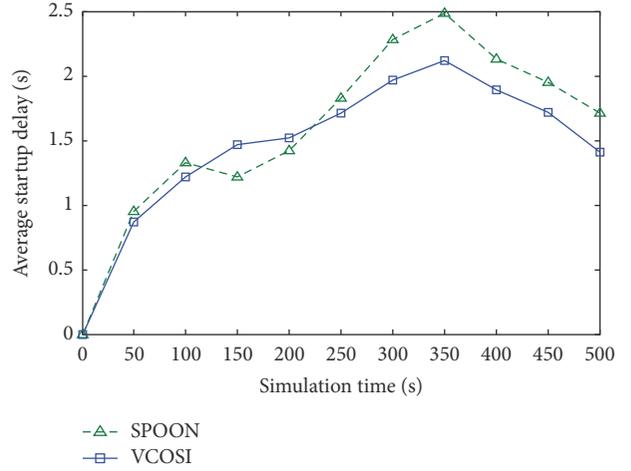


FIGURE 6: Average startup delay against simulation time.

of average startup delay, PSNR, and maintenance overhead, respectively.

**4.3.1. Average Startup Delay.** We use the time span between sending request message and receiving first video data to denote the startup delay.

Figure 6 shows the mean values of collected startup delay for the results of VCOSI and SPOON during a time interval  $T = 50$  s. As Figure 6 shows, SPOON's green curve has the two processes of rise and fall with the whole simulation time. The green curve firstly increases from  $t = 0$  s to  $t = 100$  s and slightly falls from  $t = 100$  s to  $t = 150$  s. It continues to quickly rise from  $t = 150$  s to  $t = 350$  s and quickly decreases from  $t = 350$  s to  $t = 500$  s. The blue curve of VCOSI keeps the rise trend with the fluctuation from  $t = 0$  s to  $t = 350$  s and quickly falls from  $t = 350$  s to  $t = 500$  s. VCOSI's blue curve is higher than that of SPOON from  $t = 0$  s to  $t = 200$  s, but VCOSI's results are less than those of SPOON from  $t = 250$  s to  $t = 500$  s.

Figure 7 includes the mean values of collected startup delay for the results of VCOSI and SPOON in the process of every 30 request nodes requesting videos. As Figure 7 shows, SPOON's green curve has fast increase with the fluctuation in the process of simulation. Although VCOSI's results also quickly rise, the increment of VCOSI's results is less than that of SPOON.

SPOON estimates the relationship between community members according to the similarity of file name of watched video (SPOON does not describe the measurement method of communication frequency between nodes). Although the file name reflects main content of videos, there is the information loss for the video content with rich-content. It is difficult to accurately denote the interest similarity relationship between members for the measurement method of similar interest in SPOON. This leads to fragile relationship between members and low sharing performance of system. For instance, once there are the different interests between members, the request messages for video content difficultly are quickly responded by other intracommunity members.

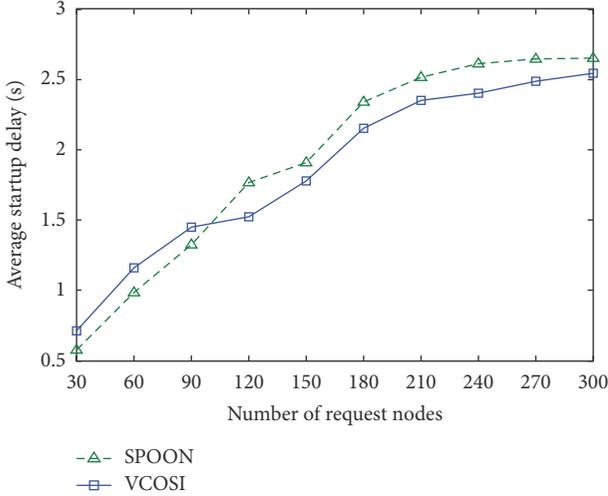


FIGURE 7: Average startup delay against number of request nodes.

The low video lookup success rate increases the lookup number, which increases the lookup delay. Because SPOON does not consider the mobility of members in the process of community construction, the transmission performance of video data is easily subjected by the influence of node mobility. The fast increase in the number of request nodes leads to quick rise of startup delay. VCOSI estimates the interest similarity between members by making use of ontology-based semantical interest capture to measure the extensional video information, which obtains the accurate measurement results of interest similarity. VCOSI further obtains high video lookup success rate; namely, the more video request can be responded by intracommunity members, which reduces the lookup delay. On the other hand, the community members make use of the message exchange to invite the one-hop neighbor nodes to join the communities, which reduces the geographical distance between video requesters and suppliers. The transmission delay of video data may be reduced. The two solutions VCOSI and SPOON experience slight network congestion (from  $t = 200$  s to  $t = 350$  s) with increasing number of nodes requesting videos. The community members in VCOSI need to invite the mobile nodes to join the communities, so VCOSI's results are higher than those of SPOON from  $t = 0$  s to  $t = 100$  s. However, after the mobile nodes join the communities, VCOSI's results are better than those of SPOON from  $t = 250$  s to  $t = 500$  s.

**4.3.2. Peak Signal to Noise Ratio (PSNR).** We use the PSNR of each video streaming to show the watched video quality according to the following equation which is defined in [32]:

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

where MAX\_Bitrate is the transmission rate of video data and is set to 128 k/s; EXP\_Thr and CRT\_Thr are expected and real throughput, respectively. The value of EXP\_Thr of each video streaming should be equal to the value of MAX\_Bitrate.

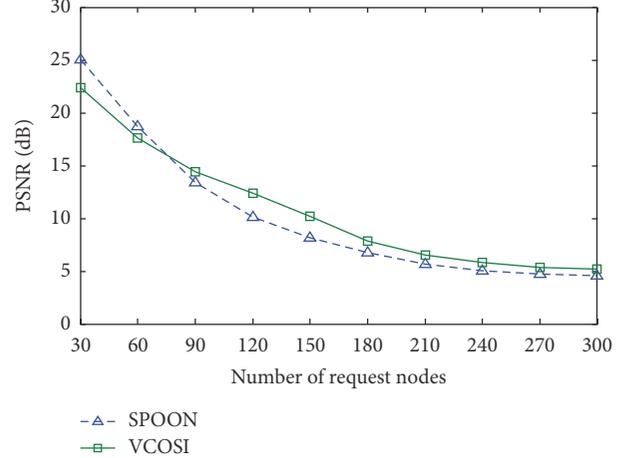


FIGURE 8: PSNR against number of request nodes.

The value of CRT\_Thr is real throughput of video streaming received by each node.

Figure 8 shows the mean values of PSNR for video streaming of each node corresponding to VCOSI and SPOON with increasing number of request nodes. As Figure 8 shows, the two curves corresponding to the results of VCOSI and SPOON have the same fall trend. The curve of VCOSI is lower than that of SPOON when the number of request nodes increases from 30 to 60. When the number of request nodes increases from 60 to 300, VCOSI's curve is higher than that of SPOON.

In SPOON, the measurement of relationship between community members mainly investigates the interest similarity based on the similarity calculation results of file name. The influence of node mobility cannot be avoided in the process of video data transmission due to the neglect for the investigation of mobility similarity. The increase in the number of request nodes brings the high demand for the network traffic, which results in the network congestion. The high packet loss rate caused by congestion severely influences the amount of video data received by nodes; namely, the decrease in the real average throughput of each node makes the values of SPOON's PSNR quickly fall. In VCOSI, the community members need to exchange the playback traces to find the potential community members with similar interests. The increasing number of request nodes, including nodes passively receiving invitation and nodes actively joining the system, brings high traffic demand. The high traffic of VCOSI also results in the increase in the number of lost packets, so that the values of PSNR of VCOSI are less than those of SPOON when the number of request nodes increases from 30 to 60. With the continuous increase in the number of request nodes, the network congestion also leads to the large number of packet losses and the fast decrease of VCOSI's PSNR. However, the negative influence levels of VCOSI caused by congestion are lower than those of SPOON (the curve of VCOSI's PSNR is higher than that of SPOON when the number of request nodes increases from 60 to 300). This is as the invited community members include the one-hop neighbor nodes of

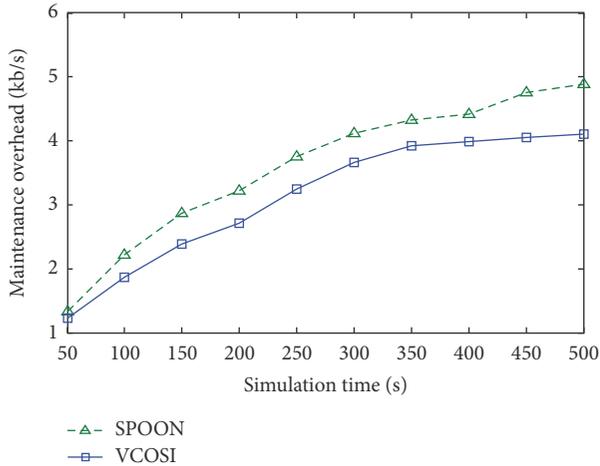


FIGURE 9: Maintenance overhead against simulation time.

inviters. The close geographical distance between inviters and invited nodes reduces the influence from the node mobility and the risk of packet loss.

**4.3.3. Maintenance Overhead.** The maintenance of member state, handling request message, and interaction between communities need to consume network bandwidth. The usage amount of bandwidth for maintaining the above information is considered as the maintenance overhead.

Figure 9 shows the maintenance overhead of VCOSI and SPOON during a time interval  $T = 50$  s. As Figure 9 shows, the two curves of VCOSI and SPOON experience the rise with the whole simulation time. VCOSI's blue curve is lower than that of SPOON and the increment of VCOSI's results gradually decreases. The increment of SPOON's results still keeps the increasing trend.

SPOON relies on the coordinators and ambassadors in communities to maintain the member state, handle request messages, and interact with other communities. The increase in the number of members results in the large amount of consumption of bandwidth in order to maintain the logical links between members. Moreover, the increase in the number of members also brings more request messages for video content, so that the members need to consume large amount of bandwidth to handle the massive request messages. On the other hand, the inaccurate interest similarity measurement leads to fragile relationship between members. The fragile relationship between members brings severely negative influence for stability of community structure; namely, the community members continuously leave current community and join into new community. The reconstruction of community structure results in huge consumption of bandwidth. The low lookup success rate also causes repetitive search, which increases the bandwidth consumption. Therefore, SPOON needs to consume massive bandwidth to maintain the overlay network. VCOSI has stable community structure relative to SPOON because of the accurate interest similarity measurement between members. The consumption of bandwidth from the maintenance cost of community structure keeps low

growth with increasing number of members. The high lookup success rate reduces the number of lookup messages, which reduces the bandwidth use. VCOSI has lower maintenance overhead than SPOON. Moreover, VCOSI distributes the maintenance overhead to multiple broker nodes, which does not cause the overload of broker nodes.

## 5. Conclusion

In this paper, we propose a novel mobile Video Community discovery scheme using ontology-based semantical interest capture (VCOSI) in order to enhance the stability and scalability of overlay networks in the process of network virtualization. VCOSI proposes an ontology-based semantical extension approach, which achieves distinct and precise description for video content and accurately measures the similarity between videos. In order to reduce the calculation load of mobile nodes, VCOSI employs a prefix-filtering-based estimation algorithm to reduce the comparison times between videos. VCOSI makes use of the similarity for watched videos to estimate the interest similarity level between nodes, construct node communities, and economically maintain community structure. The simulation results show how VCOSI has lower startup delay, higher video quality, and lower maintenance overhead than SPOON.

## Competing Interests

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

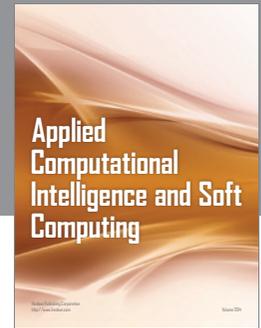
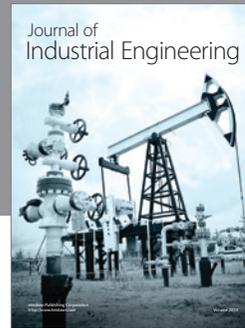
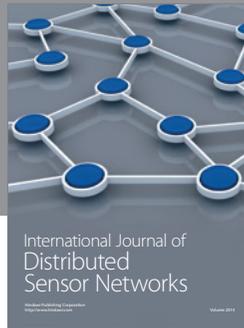
## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants nos. 61272015 and 61050004, the Basic and Frontier Technology Research Project of Henan province (142300410303), and the Science and Technology Plan Projects (Openness & Cooperation) of Henan province (152106000048).

## References

- [1] Y. Zhou, L. Chen, C. Yang, and D. M. Chiu, "Video popularity dynamics and its implication for replication," *IEEE Transactions on Multimedia*, vol. 17, no. 8, pp. 1273–1285, 2015.
- [2] J. Otmani, A. Agarwal, and A. K. Jagannatham, "Optimal scalable video scheduling policies for real-time single- and multiuser wireless video networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2424–2435, 2015.
- [3] C. Yang, Y. Zhou, L. Chen, T. Z. J. Fu, and D. M. Chiu, "Turbocharged video distribution via P2P," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 2, pp. 287–299, 2015.
- [4] H. Park and M. V. D. Schaar, "Coalition-based resource reciprocation strategies for P2P multimedia broadcasting," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 557–567, 2008.
- [5] 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.
- [6] 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.
  - [7] C.-L. Chang, W.-M. Chen, and C.-H. Hung, “Reliable consideration of P2P-based VoD system with interleaved video frame distribution,” *IEEE Systems Journal*, vol. 8, no. 1, pp. 304–312, 2014.
  - [8] 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.
  - [9] K. Chen, H. Shen, and H. Zhang, “Leveraging social networks for P2P content-based file sharing in disconnected MANETs,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 2, pp. 235–249, 2014.
  - [10] 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.
  - [11] 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.
  - [12] R. Mihalcea, C. Corley, and C. Strapparava, “Corpus-based and knowledge-based measures of text semantic similarity,” in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI ’06)*, pp. 775–780, Boston, Mass, USA, July 2006.
  - [13] Y. Li, D. McLean, Z. A. Bandar, J. D. O’Shea, and K. Crockett, “Sentence similarity based on semantic nets and corpus statistics,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 1138–1150, 2006.
  - [14] A. Islam and D. Inkpen, “Semantic text similarity using corpus-based word similarity and string similarity,” *ACM Transactions on Knowledge Discovery from Data*, vol. 2, no. 2, article 10, 2008.
  - [15] M. Sahami and T. D. Heilman, “A web-based kernel function for measuring the similarity of short text snippets,” in *Proceedings of the 15th International Conference on World Wide Web (WWW ’06)*, pp. 377–386, 2006.
  - [16] X. Wang, X. Fan, and J. Zhao, “Method of Chinese short text classification based on feature extension,” *Journal of Computer Applications*, vol. 29, no. 3, pp. 843–845, March 2009.
  - [17] M. Yuan, “Feature extension for short text categorization using frequent term sets,” *Procedia Computer Science*, vol. 31, pp. 663–670, 2014.
  - [18] Y. Genc, Y. Sakamoto, and J. Nickerson, “Discovering context: classifying tweets through a semantic transform based on wikipedia,” in *Foundations of Augmented Cognition. Directing the Future of Adaptive Systems: 6th International Conference, FAC 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9–14, 2011. Proceedings*, vol. 6780 of *Lecture Notes in Computer Science*, pp. 484–492, Springer, Berlin, Germany, 2011.
  - [19] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, “Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification,” *Neurocomputing*, vol. 174, pp. 806–814, 2016.
  - [20] X. Quan, G. Liu, Z. Lu, X. Ni, and L. Wenyin, “Short text similarity based on probabilistic topics,” *Knowledge and Information Systems*, vol. 25, no. 3, pp. 473–491, 2010.
  - [21] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, “Learning to classify short and sparse text & web with hidden topics from large-scale data collections,” in *Proceedings of the 17th International Conference on World Wide Web (WWW ’08)*, pp. 91–100, Beijing, China, April 2008.
  - [22] H. Zhang and G. Zhong, “Improving short text classification by learning vector representations of both words and hidden topics,” *Knowledge-Based Systems*, vol. 102, pp. 76–86, 2016.
  - [23] D.-T. Vo and C.-Y. Ock, “Learning to classify short text from scientific documents using topic models with various types of knowledge,” *Expert Systems with Applications*, vol. 42, no. 3, pp. 1684–1698, 2015.
  - [24] M. Wang, Z. Lu, H. Li, and Q. Liu, “Syntax-based deep matching of short texts,” <https://arxiv.org/abs/1503.02427>.
  - [25] A. Severyn and A. Moschitti, “Learning to rank short text pairs with convolutional deep neural networks,” in *Proceedings of the 38th International ACM SIGIR conference on research and development in Information Retrieval (SIGIR ’15)*, pp. 373–382, Santiago, Chile, 2015.
  - [26] Z. Lu and H. Li, “A deep architecture for matching short texts,” in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pp. 1367–1375, 2013.
  - [27] Z. Wang, H. Mi, and A. Ittycheriah, “Semi-supervised clustering for short text via deep representation learning,” 2016, <http://arxiv.org/abs/1602.06797>.
  - [28] X. Wan, “A novel document similarity measure based on earth mover’s distance,” *Information Sciences*, vol. 177, no. 18, pp. 3718–3730, 2007.
  - [29] H. Huang, Z. Chen, X. Zhang, and K. Zhang, “Approach of large-scale sentence similarity computation,” *Journal of Chinese Information Processing*, vol. 20, no. 1, pp. 47–52, 2006.
  - [30] R. Bayardo, Y. Ma, and R. Srikant, “Scaling up all pairs similarity search,” in *Proceedings of the 16th ACM International Conference on World Wide Web (WWW ’07)*, New York, NY, USA, 2007.
  - [31] B. Yang and C. A. Balanis, “Least square method to optimize the coefficients of complex finite-difference space stencils,” *IEEE Antennas and Wireless Propagation Letters*, vol. 5, no. 1, pp. 450–453, 2006.
  - [32] S.-B. Lee, G.-M. Muntean, and A. F. Smeaton, “Performance-aware replication of distributed pre-recorded IPTV content,” *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516–526, 2009.



**Hindawi**

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

