

Special Issue on Wireless Big Data

Embracing social big data in wireless system design

Yonggang Wen, Han Hu*, Fang Liu

Nanyang Technological University, Singapore 639798, Singapore

* Corresponding author, email: hhu@ntu.edu.sg

Abstract: The past decade has witnessed explosive growth in wireless big data, as well as in various big data analytics technologies. The intelligence mined from these massive datasets can be utilized to optimize wireless system design. Due to the open data policy of the mainstream OSN (Online Social Network) service providers and the pervasiveness of online social services, this paper studies how social big data can be embraced in wireless communication system design. We start with our first hand experience on crawling social big data and the principal of social-aware system design. Then we present five studies on utilizing social intelligence for system optimization, including community-aware social video distribution over cloud content delivery networks, public cloud assisted mobile social video sharing, data driven bitrate adjustment and spectrum allocation for mobile social video sharing, location-aware video streaming, and social video distribution over information-centric networking.

Keywords: wireless big data, social media, data analytics, video streaming, game theory

Citation: Y. G. Wen, H. Hu, F. Liu. Embracing social big data in wireless system design [J]. Journal of communications and information networks, 2017, 2(1): 81-96.

1 Introduction

Decades of exponential growth in Internet and data services have ushered in the big data era. Huge amount of multimedia data, deemed as a unique kind of big data, has been produced with the push of prevalence of online social network services that can be seamlessly integrated with the 3G/4G (third/fourth Generation) wireless networks as well as with smart mobile devices^[1,2]. In contrast to traditional passively received media content, online social networks prod users to be more active generating their own contents and sharing data widely among their “friends”. The existence of such highly diver-

sified content providers, along with the socialized sharing paradigm, further accelerates the increase in data. Take video content as an example, 300 hours of user generated videos were uploaded to YouTube every minute in 2016^[3].

Social media big data containing rich, valuable information on users’ behaviors and preferences has been deemed as the essential input of various multimedia applications and services, including multimedia search, recommendation, advertisement, smart city and so on^[4,5]. The richness and diversity of data allow us to build better computational models for discovering, mining, and analyzing the true value of the data. However, benefits are always accompanied

Manuscript received Dec. 31, 2016; accepted Feb. 8, 2017

This work is supported by the Ministry of Education Academic Research Fund Tier 1 (No. RG 17/14), SMART Innovation Grant (No. ING148077-ICT), Cisco Systems Inc. (No. M4061334.020), BCA Green Buildings Innovation Cluster R&D Grant (No. NRF2015ENC-GBICRD001-012).

with challenges. First, multimedia big data is usually unstructured and heterogeneous, which greatly increases the representation difficulty. The semantic character inherited by social media data also makes it hard to be understood. Second, powerful computation ability is necessary for processing social media big data. This incurs a bunch of resource optimization problems, such as optimizing storage, computation, networking, and bandwidth. At the same time, the real time requirements of online social services further encourage developers to adopt parallel and distributed processing to satisfy the end users' QoE (Quality of Experience)^[6]. Third, social media delivery becomes a tough task due to the large scales of the network traffic and user defined data sharing paradigm through social connections. Both the high popularity and wide propagation of dynamic content can also increase the difficulty of its delivery.

To address the aforementioned challenges, multiple revolutionary techniques and new network structures are proposed as key solutions. To manage and process the large amount of media data, the MapReduce^[7] computing framework, deemed as a general paradigm, is implemented in various applications, such as performance optimization with network traffic control. In addition, the performance of media big data computing has also been profoundly impacted by rapidly-changing networking technologies. For example, the SDN (Software-Defined Network) decouples the control plane from the data plane and places the control logic separately on logically centralized controllers. This design enables network operations that are more dynamic and flexible. Furthermore, many studies also focus on the way to infer valuable insights from data and further facilitate current systems and implementations. Here are some examples. Wang et al.^[8] utilized social information extracted from social videos to improve video distribution strategies. They proposed a hybrid CDN (Content Delivery Network) and a P2P architecture, based on which users within similar social groups can contribute to each other. Wang et al. also took the advantage of social propagation for media content replication^[9]. They associated a distinct

geographic influence index with each media content item, where a higher index value referred to a higher probability that the content would be propagated to more regions in the future.

In this paper, we conduct an overall study of social media big data for wireless communication systems. Basically, we summarize three general ways of using social media big data which are shown in Fig. 1: 1) optimizing media-based applications by mining meaningful information from big data, including user behavior, propagation feature, and content popularity; 2) enhancing both the scalability and efficiency of online social networks and their corresponding media processing with hybrid emerging networks; 3) using wireless resources, such as the bandwidth, appropriately and adaptively based on dynamic media information. To further illustrate these issues, we also introduce five specific cases, which represent some of our current efforts in utilizing social media big data.

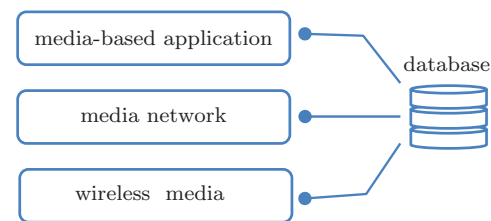


Figure 1 Issues related to social media big data

The rest of the paper is structured as follows. Section 2 investigates related studies. Section 3 describes how we crawl social big data as well as the principals of social aware system design. Section 4 presents five works on utilizing social media data for system optimization. Section 5 concludes the paper.

2 Related work

Online social networks have reshaped the way how information is generated and consumed. In particular, each social user is a content generator as well as a consumer. They are connected via social structure (e.g., friendship on Facebook). The contents generated by a specific user will be delivered to his/her friends, and this user will also enjoy contents from

them in return. Zhang et al.^[10] investigated a popular mobile social video sharing service via extensive measurement and illustrated its architecture as well as user viewing behaviors. Much effort has been expended in utilizing social structure to distribute contents over various novel infrastructures. In Ref. [11], the authors leveraged social information diffusion characteristics to adjust the impact on content distribution platforms. Since the OSN (Online Social Network) content is stored in key-value databases in private data centers, several studies^[12] aimed to minimize the number of content replicas required by connected users in order to reduce traffic across different servers while still maintaining load balance. Liu et al.^[13] considered the visit and update rates of different types of user data and determined which portions of data should be replicated. Wang et al.^[7] proposed a propagation-based replication strategy for social videos by utilizing edge clouds and P2P clients jointly. Jiao et al.^[14] studied social content placement over several clouds with the objectives of minimizing its carbon footprint, the inter-data center traffic, and its operational cost. Our previous studies^[15,16] presented the concept of utilizing social community for both content placement and request dispatching.

3 Social big data for wireless communication systems

In this section, we introduce a distributed crawler for gathering social big data, and discuss how social intelligence can be utilized in system optimization.

3.1 Social big data acquisition

The distributed data crawler aims to gather representative data with respect to entities (e.g., TV programs) through APIs provided by social media platforms. This is the basis for our social intelligence based system design. However, the data crawling strategy faces several fundamental challenges, including dealing with the huge volume of data, missing data, and API restrictions. To address these is-

ues, we design a novel strategy to rapidly obtain more social data, taking Sina Weibo as an example. There are two main parts to our strategy^[17,18]. First, to collect more microblog data, we design a program descriptor to generate keywords and key users dynamically. These keywords and key users are mapped to crawling tasks. Second, we implement a distributed crawling mechanism to accelerate the crawling rate.

A program descriptor contains a keyword and a user set and is used to issue search requests to the APIs provided by OSNs. The keywords and social users in this set have contextual relationships with the desired entity. For example, given a TV program, the title can be a keyword, and the official social media account can be the social user. In this study, we design four types of items to represent a TV program: fixed keywords, known accounts, dynamic keywords, and dynamic users. The generation procedure works as follows. 1) Given a TV program, we first manually select a few keywords that uniquely identify the TV program, such as the name of the TV program, character names, or famous lines of dialogue. Then, we identify a set of program-related accounts. These are typically the TV program's official accounts on the microblog platform that often post relevant microblogs about the TV program. We use both the fixed keywords and known accounts to issue search requests to crawl microblogs, which are viewed as the relevant data set. To elicit a more up-to-date and diverse set of relevant microblogs about the TV program, fixed keywords are not typically used, and content posted by unknown accounts is not collected. 2) Based on the amassed microblog repository, we extract a list of temporally emerging terms as dynamic keywords. In addition, we extract the names of social accounts that post relevant microblogs and rank them to identify the key dynamic users. These dynamic keywords and dynamic users are further utilized to crawl microblogs that are potentially relevant to the TV program. 3) The data collected in this way are a mix of microblogs both relevant and irrelevant to the TV program. In order to filter out noise data, we employ a standard

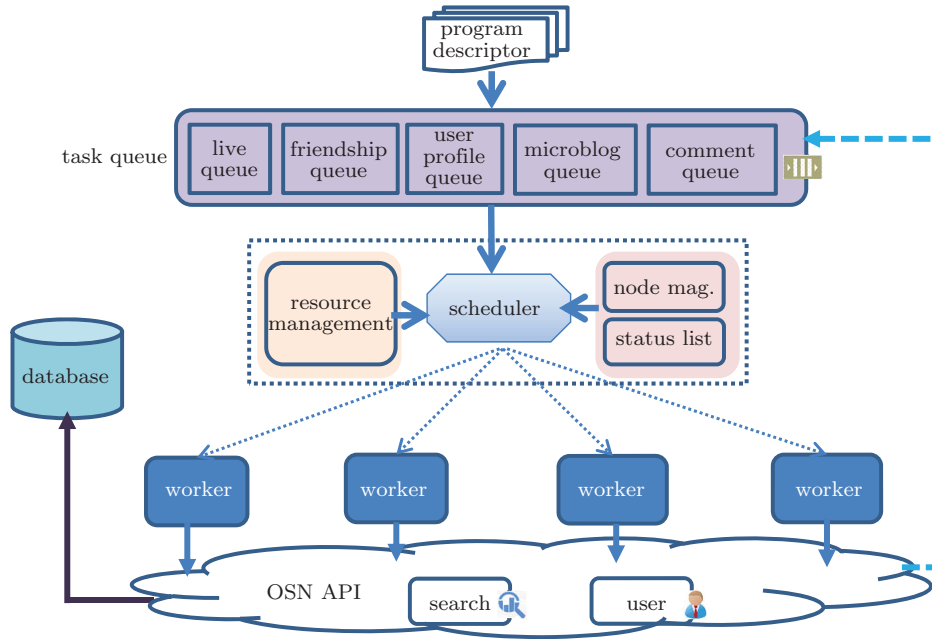


Figure 2 Architecture of the distributed data crawler comprised of four key components: a program descriptor, task queues, a request scheduling, and execution nodes. For each program of interest, we have a corresponding program descriptor, and the items in which will be mapped to multiple crawling tasks and then inserted into task queues. Request scheduling dispatches these tasks to distributed execution nodes while considering the request rate, resource utilization, and load balance.

two-class SVM classifier. For the training data, we regard the microblogs obtained from fixed keywords and known accounts as relevant.

Given the program descriptors, we can issue search items to the OSN APIs to gather microblogs. However, most live microblog services set limits on the amount of data that can be acquired and the frequency of acquisition. In general, there are three types of restriction. 1) Account restriction: each account can send only a limited number of search requests within a time period. 2) IP restriction: the request rate of each unique IP address is restricted. 3) Amount restriction: for an API keyword search, we can only obtain the most matched (or latest) 1 000 to 1 500 microblogs.

To break through the first and second restrictions, our system maintains a resource pool, consisting of OSN accounts and IP/proxy addresses, and exploits a distributed mechanism to utilize these resources to crawl microblogs. To address the amount restriction, our system automatically divides a search query into several sub-queries when the number of microblog search results to this query exceeds the threshold.

For example, we can constrain the query to a smaller time period. In this way, a regular query can be split into many sub-queries within a time slot or a region, such as “ $keyword = k, time = 0:00-1:00, region = r$ ”.

Fig. 2 presents the architecture of our proposed distributed data crawler. Each TV program is represented by a program descriptor, consisting of four types of items that can be expanded dynamically. Since a microblog is a mixture of different types of data, including user and associated user’s profiles, microblog contents, reposts, and comments, the particular data to be gathered depends on the purpose of analysis. For example, to study audience sentiment, we only need to crawl the microblog contents, whereas we need to crawl the users’ social relationships and their associated users’ profiles to analyze social structure of the audience. Therefore, we design different task queues for different types of data. For instance, the live queue is used to gather live microblogs related to a given keyword, and the friendship queue is used to gather the social relationships of a given social media account. For each item in

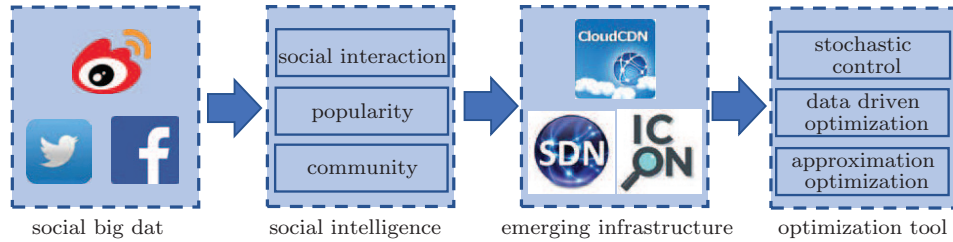


Figure 3 Framework of social intelligence based system design and optimization consisting of four components: social big data, mining social intelligence, emerging infrastructure, and optimization tool.

the program descriptor, our system will map it to a particular task queues according to the purpose of analysis. The scheduler fetches the crawling task from these queues to gather data. Considering the access constraints, we implement a resource pool to manage accounts and IP/proxy addresses. In addition, we exploit the Apache Zookeeper software to monitor the running statuses of all machines. According to the system status, the scheduler will dispatch tasks combined with account information to the execution nodes in a load-balanced way. When an allocated task is accomplished successfully, its execution node will notify the resource pool. In some cases, one successful execution will generate extra tasks in other task queues. For example, when we have crawled a microblog for a keyword and we want to know the contents of the comments, the execution node will insert this task into the comment queue. Once Zookeeper detects that one node is down, the allocated task will be re-scheduled to another active node. In the end, all of the data are stored to our storage system.

3.2 Mining social big data

Fig. 3 illustrates the general framework of social intelligence based system design and optimization^[19]. It consists of four components: social big data, mining social intelligence, emerging infrastructure, and optimization tool. First, we need to gather social big data to drive our research. Typical social media platforms for this include Sina Weibo, Twitter, and Facebook. After that, we can mine various types of social intelligence from social big data. For example, the information propagation pattern for online social

services is in-between friends, instead of from centralized servers to edge clients, and social messages are shared within small cliques. Using these intelligence, we can leverage the emerging networking infrastructures. For example, cloud CDN, built on the cloud computing infrastructure, is cost efficient due to its on-demand pricing strategy^[20,21]. Software defined networking^[22] separates the control plane and data plane from the traditional network design philosophy and adopts a centralized control mechanism to collect network status and make better decisions than the traditional distributed mechanisms do. Finally, we can utilize various optimization tools to find the optimal system deployment policy, such as data driven optimization, game theory, approximation optimization etc.

4 Cases in point of social media assisted system design

Using the intelligence gathered from social media data as described in the previous section, we discuss five cases that utilize the social characteristics for system design and optimization.

4.1 Community aware social video distribution

Recent years have witnessed a transformation that people tend to acquire and consume information from online social network services (e.g., Facebook, Twitter, and Instagram) instead of from television, newspapers and mainstream websites (e.g., CNN, Yahoo, etc.). In accordance with this change, videos are now integrated into social messages, in the forms

of urls or thumbnails embedded into updates or tweets and termed as social videos, and propagated among social users. On the largest microblog platform in China, Sina Weibo, 10% of short URLs in microblog messages are video links, and these links generate 23.6% of all network traffic. Foresee reported that 18% of social media users are influenced by social factors when accessing video contents. Social videos carry two types of characteristics, video-type and social-type, which bring significant challenges to social video distribution. In particular, the video-type characteristic refers to the vast volume of videos and close-to-uniform popularity profile. Service providers require a large amount of storage and network resources, and the number of requests made to a video is highly volatile. The social-type characteristic is the social propagation feature. For example, the social relationships and social activities that determine the propagation of videos among users.

Driven by the observation that information contained on OSNs is disseminated within small cliques, we cluster social users with social relationships, close geo-locations, and similar interests together into communities, and then develop a community based social video distribution framework under the cloud CDN architecture^[15,16].

Community classification: For a crowd of social users $v \in V$, $d_1(v, v')$ refers to a social relationship, $d_1(v, v') = 1$, if v and v' are friends. Otherwise $d_1(v, v') = 0$. The normalized geo-distance and interest similarity of video watching are represented by $d_2(v, v')$ and $d_3(v, v')$, respectively. We can employ a weighted graph $G(V, E)$ to describe the structure of these users, where V is the user set, and $E = \{e_{vv'}\}$ is the edge set. Each edge $e_{vv'}$ represents the connection between two users v and v' , which is expressed as

$$e_{vv'} = \omega_1 d_1(v, v') + \omega_2 d_2(v, v') + \omega_3 d_3(v, v').$$

There are three terms on the right hand side of the equation, which refer to the users' social relationship, geo-distance, and video viewing interest, respectively. ω_1 , ω_2 , and ω_3 are weight factors used to adjust the ratios of the three items in constructing this edge. There are many clustering and graph cut

algorithms for community detection. In this study, we utilize the affinity propagation algorithm to divide social users into a collection of communities to get rid of per-specifying the number of clusters.

Community-aware social video distribution: Our solution consists of two parts, including the cloud CDN architecture and a community based request scheduling.

Cloud CDN infrastructure: In each region, we rent resources in the forms of virtual machines from the local cloud service provider. These resources together, referred to a CDN node, are responsible for video storage and streaming and serve the nearby users. All of these CDN nodes constitute our cloud CDN infrastructure, as shown in Fig. 4. Each CDN node can store a small portion of the videos from the content source and stream them to social users. Depending on the content cache strategy, users can access videos from either the CDN node or the remote content source with higher latency.

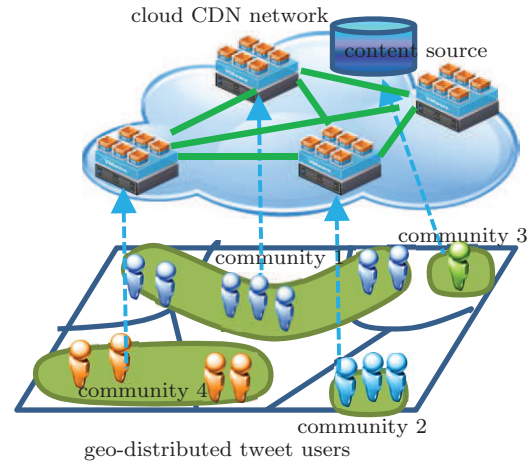


Figure 4 Cloud CDN infrastructure for community based social video replication. Social media users are grouped into various communities, and social users within a community can access videos from either a content source or the corresponding CDN nodes.

Request dispatching: Rather than adopting traditional DNS based request scheduling strategies in which each user is statically allocated to certain servers (usually three in total) according to geo-region, we adopt community based request scheduling. Using the aforementioned community classifica-

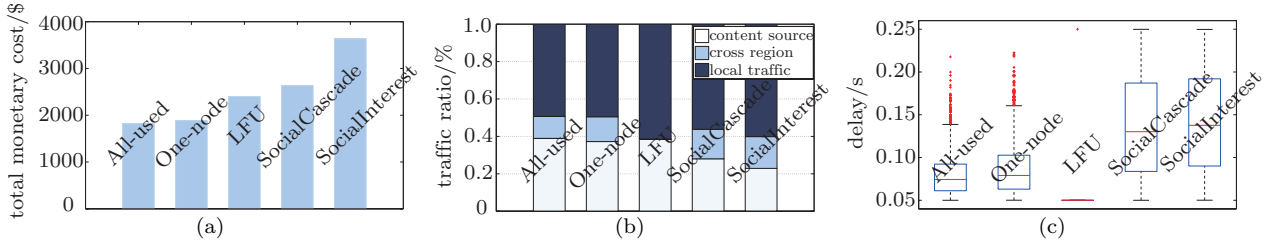


Figure 5 Performance comparison of our proposed strategies and alternative algorithms: (a) shows the total monetary cost for the five methods; (b) shows the data traffic from different sources for the five methods; (c) shows the time delay distribution for five the methods.

tion method, users within a community are scheduled to the corresponding servers. As a community may span several geo-regions, the main problem is how to determine the corresponding set of CDN nodes for each community. There are two straightforward strategies: 1) users from a community are scheduled to the CDN node corresponding to the region where most of them come from; 2) users from a community are scheduled to a set of CDN nodes corresponding to the regions covered by that community. Comparing these two strategies, the first is more practical because of its lower DNS query cost, and the second may achieve a better performance.

We utilize the microblog trace posted from January 1st to 15th, 2012 for community classification and user requests emulations. The provisioning cost is taken from the real pricing traces obtained from Amazon’s website. We compare our proposed methods with other legacy baseline algorithms, including an LFU algorithm, a SocialCascade algorithm, a SocialInterest algorithm, and our algorithms with two different request dispatching strategies, (i.e., All-used and One-node). The performance results, in terms of total monetary cost, traffic from different sources, and time delay, are shown in Fig. 5. The All-used algorithm can reduce the monetary cost by 5% compared to the One-node policy, 30% compared to the LFU algorithm, 43% compared to SocialCascade algorithm, and 89% compared to the SocialInterest algorithm. For the traffic saving, the All-used, One-node, and LFU algorithms save more traffic for the content source than the others do. For the time delay metric, our proposed All-used strategy achieves the lowest mean and variance in time delay.

4.2 Public cloud assisted mobile social video sharing

MSVS (Mobile Social Video Sharing) platforms, such as Vine, Miaopai, and Instagram, enable mobile users to create ultra-short videos (6 ~ 12 s long) and share them among their social media friends. As people are accustomed to watching videos via mobile devices, video traffic has grown to a tremendous scale. For instance, Vine has 200 million monthly active users who generate 1.5 billion video playbacks daily. It is a great challenge for both service providers and mobile users to distribute these massive videos to mobile users. In particular, for service providers, the huge volume of short videos demands vast bandwidth and storage resources, which are commonly relied on the local CDN (Content Delivery Network) and incur ever increasing operational expenditure. Mobile users may view multiple videos within a short time. However, the number of viewed videos is limited by both the device’s battery and the user’s wireless data plan. The legacy architectures for distributing video services rely on either the public CDN or private data centers, under which videos are dynamically replicated to servers at different geographical regions according to a traditional popularity profile, such as LFU (Least Frequently Used) or LRU (Least Recently Used). These schemes perform well for traditional Web service and video service, as the transmission mode is from the centralized server to end-users. However, social contents are being created and shared at the edge of the network, such as at the interest locality of a social community. Existing CDN-centric or data center-centric architectures are

ill-suited for these new patterns.

In this study, we propose a novel framework to tackle this problem based on public cloud storage^[23]. The idea is based on the following two observations.

- Social users may come from different georegions. Some regions may have local CDN nodes, while others may not have them. In the later case, when users access videos via CDN nodes, the service latency may vary. Fortunately, some regions without local CDN nodes may have local data centers. For instance, requests from social users in Singapore to video services provided by Facebook or Vine will be redirected to the CDN node in Malaysia or New Zealand. However, there are several data centers operated by Microsoft and Google in Singapore, and the access delay to these data centers is much smaller than that to the CDN node in Malaysia.

- Mainstream cloud storage service providers offer free storage service (e.g., 20 GB in Windows Azure) via data centers, and some users many have needless capacity. Thus, there is an opportunity to utilize the storage capacity of the local data centers to alleviate the traffic to remote CDN nodes.

Taking into account social users' selfishness, we design an incentive strategy for mobile users to support the cloud storage assisted MSVS architecture: users uploading videos to the local public cloud storage can access their friends' videos stored in the cloud with less delay, as shown in Fig. 6. In this example, users *b* and *c* upload the generated videos to the public cloud storage. User *c* can access the cloud storage for all five videos; user *b* needs to download 5 videos from the public cloud storage and 15 videos from the CDN node; user *a* has to download 5 videos from the CDN node due to no uploading. Mobile users can make decisions to balance the trade-off between the incurred uploading cost (e.g., energy consumption from a battery) and reduced access delay. We can utilize the game theory to model such a problem. In particular, the utility function of each user is the weighted sum of the individual's utility and friends' utilities, which are calculated by considering social users' willingness to help friends. We can prove that this problem is a supermodular game

and there exists at least a Nash Equilibrium.

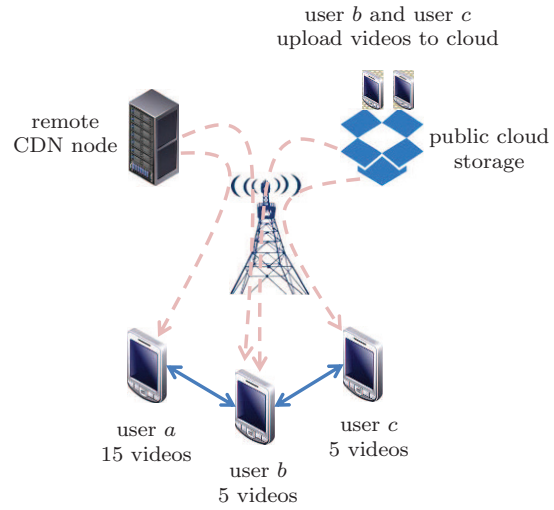


Figure 6 Public cloud storage assisted MSVS platform. Portions of videos are uploaded to the cloud storage, which can attain a balance between system performance and uploading cost.

We use two real social network topologies, including ego-Facebook and loc-Brightkite, to construct the mobile social graph. The individual utility function consists of both uploading cost and cloud access benefit. For the uploading cost, we consider the battery power consumption. For the cloud access benefit, we consider the reduction of access delay. We study the performance of the following algorithms: 1) G-SNE: it refers to the general Nash Equilibrium; 2) PO-SNE: it refers to the Pareto-optimal Nash Equilibrium. 3) G-OPT: this algorithm finds the social optimal solution to maximize social welfare. 4) I-OPT: this algorithm is a special case of our formulated problem in which each user aims to maximize the individual utility function.

From Fig. 7, we can observe that 1) with the growth of s_{nm} , the social welfare of algorithms G-SNE and PO-SNE will increase, because users care more about their friends and are willing to upload videos, and 2) our proposed PO-SNE algorithm can increase the social welfare by up to 27% in Facebook and 30% in Brightkite over the I-OPT algorithm, and the performance gaps are less than 8% in Facebook and 3% in Brightkite over the G-OPT algorithm.

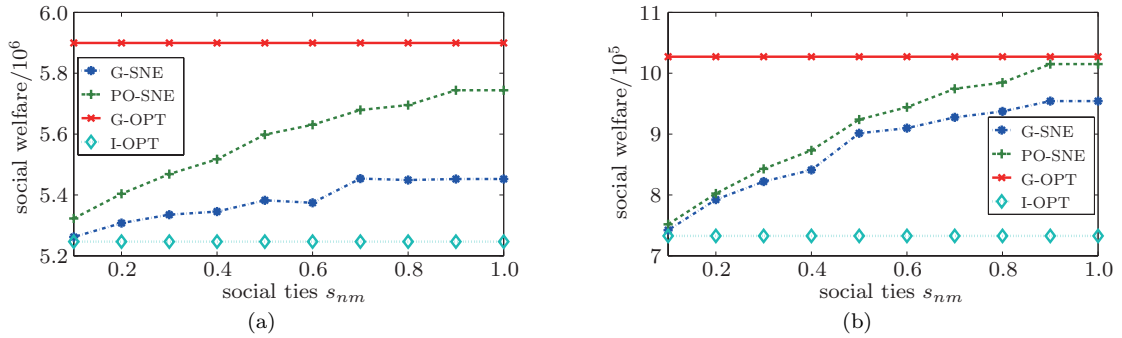


Figure 7 Social welfare comparison of four algorithms with different social ties s_{nm} (i.e., the willingness to help friends) on two social networks: (a) Facebook; (b) Brightkite

4.3 Data driven bitrate adjustment and channel assignment for MSVS

This section studies the MSVS service with regards to the legacy wireless infrastructure^[24]. For the sake of clarifying the problem, we present an illustrative example shown in Fig. 8. There are four mobile social users: users a , b , c , and d . Users a and b , b and c , and c and d are social media friends. Each user is connected to a base station. Within a specific duration of time, each user generates one video with the same time-length. Each video can be encoded into three resolution-level versions: high, median and low, with file size 1, 0.8, and 0.6 respectively. Without loss of generality, we assume that each user will enjoy all the videos generated by the friends. Thus, user a and user d need to download one video from user b and user c respectively. User b and user c need to download two videos. There are three channels: channel 1, channel 2, and channel 3. The network operator needs to assign four users to these channels for video downloading. We assume that each user has the same transmission power and channel gain to these channels. Furthermore, if the channel is occupied by one user, the transmission rate for this user is 2. If the channel is shared by two users connected to two base stations, the total transmission rate will be less than 2 due to inter-cell interference. To ease the following analysis, we assume that the transmission rate for each user is 0.7. We then study the spectrum assignment and bitrate adjustment separately as follows.

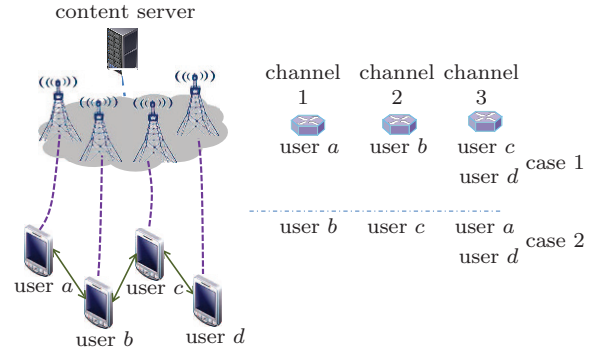


Figure 8 Spectrum assignment and bitrate adjustment for MSVS users. There are a collection of mobile users, each of whom is both a content generator and a consumer. Social media users will enjoy video clips generated by friends via base stations. The network operator needs to assign them to different channels for video downloading.

- **Spectrum assignment with a fixed bitrate:** the network operator chooses to provide the high-resolution version for mobile users, i.e., the file size for each video is 1. Therefore, the traffic volumes for users a and d are 1 unit and for users b and c are 2 units. To reduce the overall interference, one channel is shared by two users, and the rest of the channels are occupied by a single user. We present two cases in Fig. 8. In case 1, channels 1 and 2 are occupied by users a and b respectively. Channel 3 is shared by users c and d . As a result, the available traffic volumes for users a and b are both 2. They can download one video and two videos respectively. The traffic volumes for users c and d are 0.7, and they can download $0.7/2 = 35\%$ and $0.7/1 = 70\%$ of videos respectively. In case 2, channel 3 is shared

by users a and d . The available traffic volumes for users b and c are 2, and they can both download two videos. The available traffic volumes for users a and d are 0.7, and they can both download $0.7/1 = 70\%$ of videos. Comparing these two cases, the overall interference incurred by the two assignments is the same. However, both the overall traffic and fraction of downloaded videos are larger in case 1 than those in case 2, because channel 1 is not fully utilized in case 1.

- **Bitrate adjustment for a fixed spectrum assignment:** the network operator adopts case 2 for the spectrum assignment, and can choose among three different resolutions for video streaming. For high-resolution, users a and d can both download 70% of videos, and users b and c can download all of the required videos. For median-resolution, the traffic volumes for users a and d are both 0.8, while $2 \times 0.8 = 1.6$ for users b and c . According to the spectrum assignment, users a and d can enjoy $0.7/0.8 = 87.5\%$ of videos, and users b and c can enjoy all of the videos. For low-resolution, the traffic volumes for users a and d are both 0.6, while $2 \times 0.6 = 1.2$ for users b and c . Therefore, all of the users can enjoy the whole required videos. With the low-resolution setting, all of the users can enjoy the required videos, but the resolution is low, while with the high-resolution setting, some users can only enjoy a small portion of the required videos. In contrast, the median-resolution setting leads to a balance between smooth video playback and high video resolution.

From the above analysis, we can conclude that: 1) with the knowledge of the information propagation of social networks, we can achieve a better spectrum assignment and improve the system throughput, and 2) adjusting the bitrates of video streams can strike a balance between smooth video playback and high video resolution.

In this work, we adopt a hierarchical optimization structure for spectrum allocation and bitrate adjustment, as illustrated in Fig. 9. At the beginning of each time slot t , there is a set of mobile users in a region requesting spectrum channels. The net-

work operator of the base stations has no knowledge of the actual traffic demand of these mobile users, but it has the historical information of the mobile users' utilities. Using such information, the network operator can choose the bitrate version that should be requested by the mobile users and send the bitrate version to mobile users along with the vacant channels. Given the bitrate version and vacant channels, each mobile user, can get the information of the videos that he/she should download from the content server and make a channel selection.

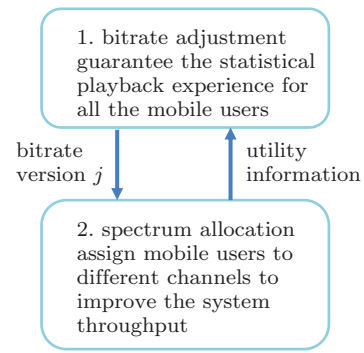


Figure 9 Hierarchical structure of spectrum access and bitrate adjustment. At the beginning of each time slot, the network operator needs to choose the bitrate version for all mobile users according to the historical statistical information. With a fixed bitrate version, mobile users make decisions on channel selection.

The dataset used for simulating a mobile social network is taken from the 9-day data usage records of mobile phone users in a city in China. A real video sequence (i.e., Sony Demo clip) is utilized to calculate the video traffic. For the communication model, we assume that mobile device users are randomly scattered over the coverage region. For the bitrate adjustment, we revise the AMBA (Autoregressive Model Based Algorithm)^[25] as the baseline algorithm to solve problem P1, termed as AR. For the spectrum allocation, we consider the generic case that the network operator aims to optimize the system throughput without the knowledge of the actual traffic demand of all the users, termed as SA. By combining different algorithms to solve these two sub-problems, we compare four algorithms: 1) OEBA+SSA, which adopts the proposed online QoS

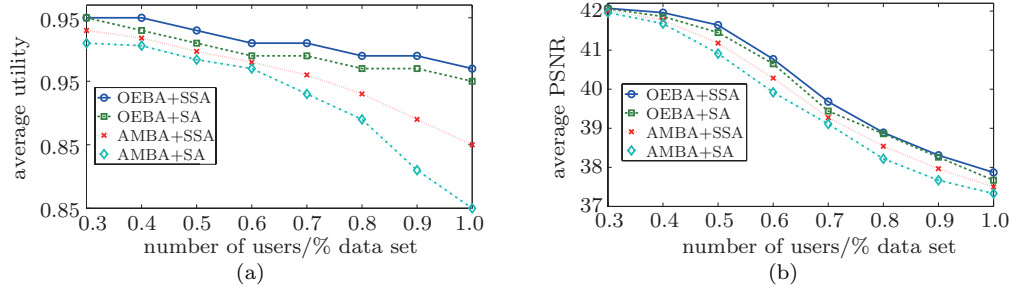


Figure 10 Performance comparison of four metrics under different numbers of mobile users: (a) average utility and (b) average PSNR

estimation strategy and spectrum allocation method; 2) OEBA+SA, which uses the proposed online QoS estimation strategy and the generic SA (Spectrum Allocation) method; 3) AMBA+SSA, which combines the AMBA algorithm for bitrate adjustment and the proposed spectrum allocation method; 4) AMBA+SA, which utilizes the AMBA algorithm and the generic spectrum allocation method.

Fig. 10(a) shows the comparison results on average utility. Both OEBA+SSA and OEBA+SA use our online QoS estimation method to adjust the bitrate version, which can guarantee the probability of the average utility is greater than the given threshold. Thus, the average utilities of these two algorithms are both above 0.9 (i.e., the threshold), and their performances are quite similar. Fig. 10(b) also presents the average PSNR for the four algorithms under different number of mobile users. It can be observed that our proposed algorithm OEBA+SSA achieves the best performance. When the sampling rate increase (i.e., there are more mobile users), the PSNR of all algorithms will decrease due to the limited available bandwidth.

4.4 Location aware video streaming

In recent years, video streaming has become one of the top mobile applications, following the exponential growth of the mobile devices market, including devices such as smartphones and tablets. However, video streaming is one of the most dominant factors in mobile battery exhaustion, which severely limits the user's experience. In considering how to pro-

long battery life, we discover that wireless stream data transmission, as a key factor of battery usage, is highly related to the condition of the bandwidth^[26]. Video streaming typically requires 18% more energy on wireless transmission with good bandwidth than it does for local video playback. If the bandwidth quality decreases from -50 dBm to -90 dBm, an incredible 8.1x more energy is spent on wireless transmission.

In this study, we hope to predict the future bandwidth to dynamically make video stream download decisions and minimize the overall cost. We address energy efficient video streaming for mobile devices by taking advantage of two observations. First, transmission bandwidth is strongly correlated with location. For example, bandwidth is often poor for underground tunnels and areas far from base stations, but it is better for open air freeways and near base stations. Second, user locations are predictable under many circumstances, and a substantial amount of location information can be extracted from online social networks. Based on such information, it is not hard to find that users often follow fixed movement patterns. For example when they go home, to work, or to other frequently visited destinations. In particular, once a user has started their journey, we can accurately predict their locations during the whole trip. Hence, using the correlation between location and bandwidth, we can predict the bandwidths the user will encounter at different times during the trip. The energy efficient mobile video streaming problem can be formulated as an optimization problem, where the cost is determined by a linear combination of the

amount of time that video playback freezes due to insufficient buffered data and the energy consumed for downloading data into the buffer. Based on the predicted bandwidth, we choose the optimal times to perform downloading in order to minimize the cost, providing both a satisfying user experience and energy savings.

Fig. 11 illustrates a system architecture for mobile video streaming. Data is downloaded from the streaming server via a wireless network and stored in the user device's buffer. Playback occurs only when the buffer is nonempty. To be more specific, each video segment needs to be first downloaded to the buffer, which consumes mobile energy by keeping the WNIC (Wireless Network Interface Card) active and transmitting data through wireless network. The mobile device can only playback a segment after it is fully downloaded to the buffer. When the buffer does not have sufficient space for the upcoming segment, no more segment can be downloaded until the buffer has enough space to accommodate the next segment. When the buffer is empty, the video freezes until the next segment is downloaded into the buffer.

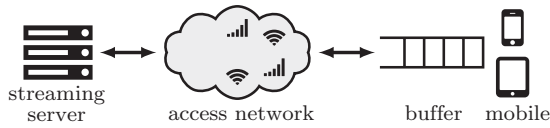


Figure 11 A system architecture for mobile video streaming. Data is downloaded from the streaming server via a wireless network and stored in the user device's buffer. Playback occurs only when the buffer is nonempty.

Basically, we want to minimize the overall cost of wireless video streaming along the road. The overall cost is given by a weighted combination of the amounts of video freezing and energy use, as well as a reward value assigned to the amount of time taken for playback. To this end, the energy efficient video streaming problem can be defined by Eq. (1).

$$\text{minimize } F(\mathbf{A}) - \beta_0(T - F(\mathbf{A})) + \beta_1 E(\mathbf{A}). \quad (1)$$

In this formula, T denotes the duration of the video being viewed and $F(\mathbf{A})$ is defined as the num-

ber of time slots in which the video is frozen. $E(\mathbf{A})$ represents the total energy including two major components: the WNIC (Wireless Network Interface Card) and wireless data transmission. $\beta_0, \beta_1 > 0$ are two weighting parameters. Larger values for β_0 represent more importance placed on smooth video playback, while larger values for β_1 represent more consideration on minimizing energy use. Obviously, there is a tradeoff between playback quality and energy use. Because minimizing video freezing usually requires more aggressive downloading which increases the corresponding energy usage.

To solve above minimization problem, we propose two algorithms. We first consider an offline setting for the problem, in which both the current and future bandwidths are known. An OPT (Optimal Algorithm) is proposed based on dynamic programming which can compute vector \mathbf{A} of download actions to minimize a combination of video freezing and energy use.

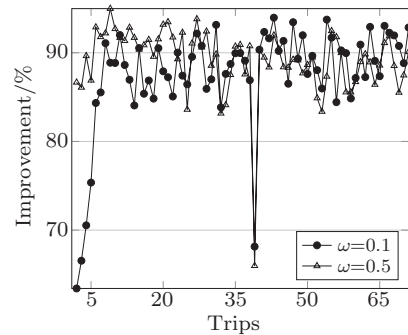


Figure 12 LAWS's performance compared to OPT for $\omega = 0.1$ and $\omega = 0.5$ using data from different numbers of trips. A point (k, y) indicates that LAWS's performance compared to OPT was y , when using data from the first k trips (excluding the first trip).

We also propose an online Location Aware Streaming algorithm named LAWS. LAWS does not assume that future bandwidths are known, but bases its decisions on historical location-aware bandwidth data. LAWS takes a user's planned route as input and then predicts bandwidths along the route to make online download decisions. We show that LAWS performs well in practice by simulation, and additionally requires only a small number of histori-

cal datasets for training, and tolerates high amounts of noise in the data. As shown by Fig. 12, LAWS never matches the performance of OPT. However, the performance gap is quite small. It is able to achieve 90.6% of OPT's performance on average when $\omega = 0.5$. Here, the parameter ω is the learning rate which controls the rate at which historical bandwidths are updated using new data. LAWS learns more quickly when the value of ω is higher. What is more, LAWS requires only around 5 trips to converge to a steady state.

4.5 Social video distribution over information centric networking

Recently, ICN (Information Centric Networking)^[27] has become a promising new paradigm for the future Internet architecture. One of the fundamental concepts of ICN is in-network caching, which enables storing multiple copies of data in routers, that are usually much closer to end-users. In this way, future requests can be served locally. This feature helps to reduce the load on servers and the congestion in the network, and it enhances end-users' delivery performances, offering a great opportunity for both the architectural design and optimization for social video distribution.

Current ICN solutions are designed for ordinary content, such as webpages and videos, and are not tuned to efficiently distribute social videos. The challenges lie in the following. 1) The legacy ICN solution only needs to consider the router topology to distribute contents from content source to different routers. In contrast, social connections determine the content exchange on OSNs. We need to jointly consider the structure of social network and the router topology. 2) Under the social structure and router topology, the cache capacity allocation among routers needs to be strategically designed, depending on the social user distribution, the social video generation/exchange quantity, the network topology, etc. 3) Even when receiving multiple interest packets for the same content in each ICN router, only one request will be sent out. This policy

may result in incorrect popularity estimates for different contents. Thus, the cache performance of the popularity-based cache strategy will be degraded.

Since social connections determine how the contents propagate among users in OSN, the access patterns of social contents are different from those of traditional webpage-based applications. In particular, in webpage-based applications, each content is passively requested by end-users and has a corresponding popularity profile. In contrast, the contents generated by a specific user will be proactively passed to his/her subscribers (e.g., followers on Twitter) and therefore have a similar popularity profile depending on the influence of the user (e.g., their number of friends). For this reason, we propose to transform the content-centric caching to user-centric caching. Our caching mechanism operates based on identifying popular social users and storing their contents. If we want to host a social user, it means the contents generated by this user within a particular time span will be cached. The benefit of this transformation is that it avoids individually handling each user-generated content, which have an extremely large number.

To adapt the ICN technology to the properties of social information diffusion, we revise the ICN naming structure by introducing the social information, including user ID and timestamp (i.e, when and by whom a specific content is generated), and modify the packet processing of the ICN router as described below. In contrast in other applications, information exchange of OSNs occurs among social friends. We devise a user-centric hierarchical naming scheme for the ICN router. Each social content is identified by a unique key in the form "APP_ID/User_ID/Timestamp/Content_ID", where "APP_ID" refers to the social service ID (e.g., Twitter, Vine); "User_ID" is the unique ID of the social user who posts this content and we can acquire this information via the collaboration with the OSN provider or by data crawling from the open OSN APIs, such as Twitter API; "Timestamp" represents the time when this content is posted; and "Content_ID" is the unique ID of the content. Using this

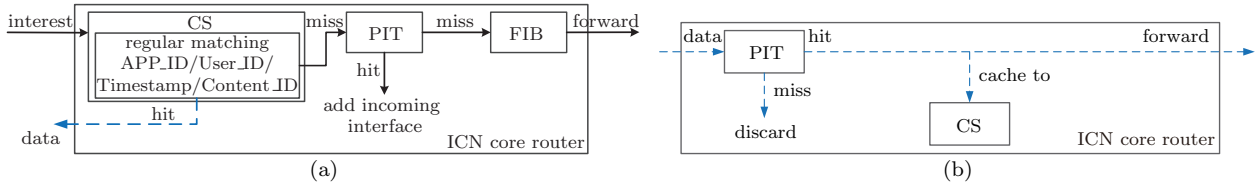


Figure 13 Packet processing engine in an ICN router. Based on the hierarchical naming scheme, we make the corresponding modification to process both interest and data packets. (a) Handling interest packet (control plane), regular matching is conducted in the Content Source to determine whether or not the request can be served locally; (b) handling data packet (data plane), social information is extracted from the hierarchical naming structure to determine whether or not the content will be cached.

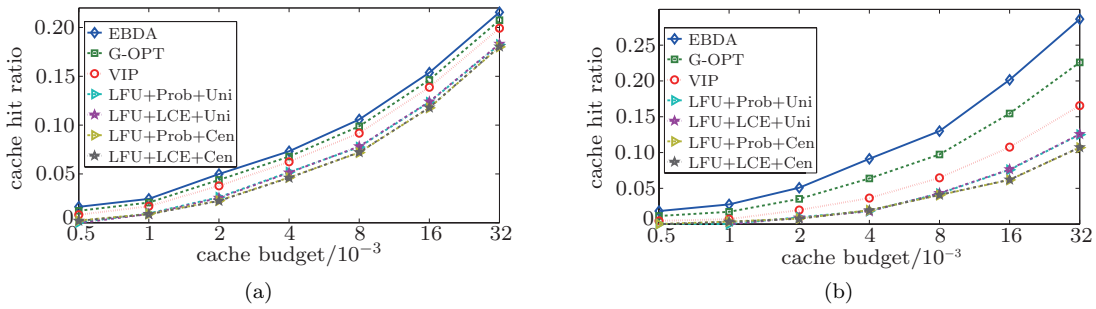


Figure 14 Cache hit ratio under different cache budgets. When α increases, the cache hit ratios of all of the algorithms increase gradually. EBDA algorithm achieves the best cache hit ratio in all cases. In comparison with the baseline algorithms, EBDA can increase the cache ratio by (a) up to 4% for Internet2 Layer3; (b) 23% for CERNET2 when $\alpha = 0.032$.

naming scheme, both the retrieval and storage for social content are transformed to become user-driven. In particular, when a social user (e.g., with an ID “David”) posts a message, it will be stored in the repository related to this user. The user’s followers will use the identifier “APP_ID/David/*/*” to send an Interest message to the ICN router, and all of the messages within the same time window related to “David” will be sent back.

Fig. 13 presents a packet processing engine in an ICN router. Specifically, Fig. 13(a) shows the workflows used to process interest packets (i.e., the request) in the control plane. And Fig. 13(b) shows the one for data packets (i.e., the response) in the data plane.

We use two real network topologies, including the real Internet2’s layer 3 network in the US and the CERNET2 (China Education and Research Network 2), to simulate the ICN network. For the social network model, we adopt an open Twitter dataset to simulate user interaction. We compare the performance of our proposed algorithm (EBDA) with

other baseline algorithms. Sun et al.^[28] utilized a large-scale real trace to evaluate 35 caching policies, of which the LFU+Prob and LFU+LCE achieved the best performances. In particular, LCE and Prob are content placement strategies, and stand for leaving a copy everywhere and the constant probability of caching on each router respectively. LFU, the abbreviation of least frequently used, is a content replacement strategy. For the cache capacity, we consider two allocation schemes: uniform allocation (Uni) and centrality-based allocation (Cen). In this way, we have four baseline algorithms using a combination of caching policy and cache capacity allocation, including LFU+Prob+Uni, LFU+Prob+Cen, LFU+LCE+Uni, and LFU+LCE+Cen. In addition, we compare our algorithms with the VIP algorithm^[29], as it partially solves the interest packet suppression issue. The G-OPT algorithm^[30] designs a optimal cache allocation algorithm based on the assumptions of an equal request rate distribution and a single routing path for each ICN router. We modify the request rate to the actual demand accord-

ing to the social relationships for each ICN router and take it as our baseline algorithm. As shown in Fig. 14, our proposed EBDA algorithm achieves the best cache hit ratio in all cases. In comparison with the G-OPT algorithm, EBDA can increase the cache ratio by up to 4% for Internet2 Layer3 and 23% for CERNET2 when $\alpha = 0.032$.

5 Conclusion

This paper discusses the possibility of utilizing social intelligence to optimize wireless system design. We first presented a distributed crawler system, which serves as the big data source to drive the following study, to gather online social network data, and the principle of mining social intelligence, including user behavior, community interaction, social interaction, etc. Then we presented five cases that utilized social intelligence for system design.

References

- [1] Y. G. Wen, X. Q. Zhu, J. Rodrigues, et al. Cloud mobile media: reflections and outlook [J]. *IEEE transactions on multimedia*, 2014, 16(4): 885-902.
- [2] J. J. Yu, J. W. Zhang. Recent progress on high-speed optical transmission [J]. *Digital communications and networks*, 2016, 2(2): 65-76.
- [3] Youtube Statistics-2016 [EB/OL]. <http://fortunelords.com/youtube-statistics/>.
- [4] W. W. Zhu, P. Cui, Z. Wang, et al. Multimedia big data computing [J]. *IEEE multimedia*, 2015, 22(3): 96-103.
- [5] A. Rodić, M. Jovanović, I. Stevanović, et al. Building technology platform aimed to develop service robot with embedded personality and enhanced communication with social environment [J]. *Digital communications and networks*, 2015, 1(2): 112-124.
- [6] W. W. Zhang, Y. G. Wen, Z. Z. Chen, et al. QoE-driven cache management for HTTP adaptive bit rate streaming over wireless networks [J]. *IEEE transactions on multimedia*, 2013, 15(6): 1431-1445.
- [7] J. Dean, S. Ghemawat. Mapreduce: simplified data processing on large clusters [J]. *Communications of the ACM*, 2008, 51(1): 107-113.
- [8] Z. Wang, L. F. Sun, X. W. Chen, et al. Propagation-based social-aware replication for social video contents [C]//The 20th ACM International Conference on Multimedia, Nara, Japan, 2012: 29-38.
- [9] Z. Wang, J. C. Liu, W. W. Zhu. Social-aware video delivery: challenges, approaches, and directions [J]. *IEEE network*, 2016, 30(5): 35-39.
- [10] L. Zhang, F. Wang, J. C. Liu. Understand instant video clip sharing on mobile platforms: Twitters Vine as a case study [C]//Network and Operating System Support on Digital Audio and Video Workshop, Singapore, Singapore, 2014: 85-90.
- [11] F. Malandrino, M. Kurant, A. Markopoulou, et al. Minimizing the peak load from information cascades: social networks meet cellular networks [J]. *IEEE transactions on mobile computing*, 2015, 22(3): 96-103.
- [12] J. Tang, X. Y. Tang, J. S. Yuan. Optimizing inter-server communication for online social networks [C]//IEEE 35th International Conference on Distributed Computing Systems (ICDCS), Columbus, USA, 2015: 215-224.
- [13] G. X. Liu, H. Y. Shen, H. Chandler. Selective data replication for online social networks with distributed data-centers [C]//The 21st IEEE International Conference on Network Protocols (ICNP), Göttingen, Germany, 2013: 1-10.
- [14] L. Jiao, J. Li, W. Du, et al. Multi-objective data placement for multi-cloud socially aware services [C]//IEEE Conference on Computer Communications (INFOCOM), Toronto, Canada, 2014: 28-36.
- [15] H. Hu, Y. G. Wen, T.-S. Chua, et al. Community based effective social video contents placement in cloud centric CDN network [C]//IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, 2014: 1-6.
- [16] H. Hu, Y. G. Wen, T.-S. Chua, et al. Joint content replication and request routing for social video distribution over cloud CDN: a community clustering method [J]. *IEEE transactions on circuits and systems for video technology*, 2016, 22(3): 96-103.
- [17] H. Hu, Y. G. Wen, Y. Gao, et al. Toward an SDN-enabled big data platform for social TV analytics [J]. *IEEE network*, 2015, 29(5): 43-49.
- [18] H. Hu, J. Huang, H. Zhao, et al. Social TV analytics: a novel paradigm to transform TV watching experience [C]//The 5th ACM Multimedia Systems Conference, Singapore, Singapore, 2014: 172-175.
- [19] H. Hu, Y. G. Wen, T.-S. Chua, et al. Toward scalable systems for big data analytics: a technology tutorial [J]. *IEEE access*, 2014, 2: 652-687.
- [20] Y. C. Jin, Y. G. Wen, H. Hu, et al. Reducing operational costs in cloud social TV: an opportunity for cloud cloning [J]. *IEEE transactions on multimedia*, 2014, 16(6): 1739-1751.
- [21] Y. C. Jin, Y. G. Wen, H. Hu. Minimizing monetary cost via cloud clone migration in multi-screen cloud social TV system [C]//Global Communications Conference (GLOBECOM), Atlanta, USA, 2013: 1747-1752.
- [22] Y. C. Jin, Y. G. Wen, C. Westphal. Towards joint resource allocation and routing to optimize video distri-

- bution over future Internet [C]//IFIP Networking Conference (IFIP Networking), Toulouse, France, 2015: 1-9.
- [23] H. Hu, Y. G. Wen, D. Niyato. Public cloud storage assisted mobile social video sharing: a supermodular game approach [J]. *IEEE journal on selected areas in communications*, 2017, doi: 10.1109/JSAC.2017.2659478.
- [24] H. Hu, Y. G. Wen, D. Niyato. Spectrum allocation and bitrate adjustment for mobile social video sharing: a potential game with online QoS learning approach [J]. *IEEE journal on selected areas in communications*, 2017, doi: 10.1109/JSAC.2017.2676598.
- [25] C. Chen, R. W. Heath, A. C. Bovik, et al. A Markov decision model for adaptive scheduling of stored scalable videos [J]. *IEEE transactions on circuits and systems for video technology*, 2013, 23(6): 1081-1095.
- [26] W. Zhang, R. Fan, Y. G. Wen, et al. Energy efficient mobile video streaming: a location-aware approach [J]. *ACM transactions on intelligent systems and technology*, 2017, accepted.
- [27] V. Jacobson, D. K. Smetters, J. D. Thornton, et al. Networking named content [C]//The 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, 2009: 1-12.
- [28] Y. Sun, S. K. Fayaz, Y. Guo, et al. Trace-driven analysis of ICN caching algorithms on video-on-demand workloads [C]//The 10th ACM International on Conference on Emerging Networking Experiments and Technologies, Sydney, Australia, 2014: 363-376.
- [29] E. Yeh, T. Ho, Y. Cui, et al. VIP: a framework for joint dynamic forwarding and caching in named data networks [C]//The 1st International Conference on Information-Centric Networking, Paris, France, 2014: 117-126.
- [30] Y. G. Wang, Z. Y. Li, G. Tyson, et al. Optimal cache allocation for content-centric networking [C]//The 21st IEEE International Conference on Network Protocols (ICNP), Göttingen, Germany, 2013: 1-10.

About the authors



Yonggang Wen (S'99-M'08-SM'14) is an associate professor with school of computer science and engineering at Nanyang Technological University, Singapore. He received his Ph.D. degree in electrical engineering and computer science (minor in Western Literature) from Massachusetts Institute of Technology (MIT), Cambridge, USA, in 2008. Previously he was with Cisco, USA, to lead product development in content delivery network, which had a revenue impact of 3 Billion US dollars globally. Dr. Wen has published over 140 papers in top journals and prestigious conferences. His work in Multi-Screen Cloud Social TV has been featured by global media in over 1600

news articles from over 29 countries and received ASEAN ICT Award 2013 (gold medal). His work on Cloud3DView, as the only academia entry, has won the Data Centre Dynamics Awards 2015 C APAC. He is a co-recipient of 2015 IEEE Multimedia Best Paper Award, and a co-recipient of Best Paper Awards at EAI/ICST Chinacom 2015, IEEE WCSP 2014, IEEE Globecom 2013 and IEEE EUC 2012. He serves on editorial boards for *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Wireless Communication Magazine*, *IEEE Communications Survey & Tutorials*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Signal and Information Processing over Networks*, *IEEE Access Journal* and *Elsevier Ad Hoc Networks*, and was elected as the chair for IEEE ComSoc Multimedia Communication Technical Committee (2014-2016). His research interests include cloud computing, green data center, big data analytics, multimedia network and mobile computing. (Email: ygwen@ntu.edu.sg)



Han Hu [corresponding author] received the B.S. degree and Ph.D. from University of Science and Technology of China (USTC) in 2007 and 2012 respectively. He is currently a research fellow with the School of Computer Engineering at Nanyang Technological University, Singapore. His research interests include social media analysis and distribution, big data analytics, multimedia communication, and green network. (Email: hhu@ntu.edu.sg)



Fang Liu received her B.E. and M.E. degree from Xidian University, China, in 2008 and 2011 respectively. She received her Ph.D. degree from Nanyang Technological University, Singapore, in 2016. She is currently a research fellow with the School of Computer Engineering at Nanyang Technological University, Singapore. Her research interests include cloud computing and data analytics. (Email: fliu@ieee.org)