Computer Communications 100 (2017) 104–117

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Crowd-Cache: Leveraging on spatio-temporal correlation in content popularity for mobile networking in proximity

Kanchana Thilakarathna[a,b], Fang-Zhou Jiang[a,b,*], Sirine Mrabet[a], Mohamed Ali Kaafar[a], Aruna Seneviratne[a,b], Gaogang Xie[c]

[a] Data61, CSIRO, Australia
[b] University of New South Wales, Australia
[c] ICT, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 19 July 2016
Revised 28 December 2016
Accepted 11 January 2017
Available online 19 January 2017

Keywords:
Mobile content distribution
Crowd-sourcing
Opportunistic content sharing
Mobile off-loading
Mobile networking in proximity

ABSTRACT

Mobile capped plans are being increasingly adopted by mobile operators due to an exponential data traffic growth. Users then often suffer high data consumption costs as well as poor quality of experience. In this paper, we introduce a novel content access scheme, *Crowd-Cache*, which enables mobile networking in proximity by exploiting the transient co-location of devices, the epidemic nature of content popularity, and the capabilities of smart mobile devices. *Crowd-Cache* provides mobile users access to popular content cheaply with low latency while improving the overall quality of experience. We model the *Crowd-Cache* system in a probabilistic framework using a real-life dataset of video content access. The simulation results show that, in a public transportation scenario, more than 80% of the passengers can save at least 40% on their cellular data usage during a typical average city bus commute of 10 minutes. Finally, we show the practical viability of the system by implementing and evaluating the system on Android devices.

1. Introduction

The pervasiveness of smart mobile devices and rich media services are giving rise to exponential growth in cellular traffic of which video is predicted to account for more than 75% of the total available bandwidth by 2020 [1]. To cope with this demand, network operators have introduced capped data plans. However, during peak times, the high demand introduces high latencies which yield to reduced user quality of experience (QoE) [2]. The predictions are that demand will outpace the increases in capacity that will be provided by new technologies such as LTE+ or 5G [3].

This produced significant research efforts to develop techniques for minimizing the cellular network data traffic and improving user *QoE*, broadly falling into three main areas: peer-to-peer systems [4,5], traffic offloading [6,7] and caching schemes [8–10]. Peer-to-peer systems, however, cannot guarantee timely content delivery, while traffic offloading schemes proposed so far assume the availability of a low cost network and deal with delays connecting to the lower cost networks. Traditional in-network caching systems, on the other hand, are not effective as the delay and cost bottlenecks quite often occur on the last hop wireless link in mobile networks [11,12]. Likewise, local caching schemes including content pre-fetching [9] rely on the ability to predict user interests, which is a challenging task, often inaccurate and prone to errors or can altogether lead to loss of privacy.

In this paper, we propose an alternative approach that exploits the transient co-location of mobile users and potential spatio-temporal correlation of content popularity. Along with the high capabilities of the modern mobile devices, we postulate and verify that it is possible to take advantage of the observed correlations of user behavioral patterns to enable mobile networking in proximity. We propose a novel content distribution system, the rationale of which is inspired by the delivery of information via the free newspapers in public transportation systems of major cities around the world such as Metro in NYC[1]. In the free newspaper distribution system, users consume content (reading the paper) whilst traveling but leave the content (the paper) when they leave. We propose *Crowd-Cache*, a system that provides storage facilities and a free local network in public places and transportation systems

* Corresponding author.
E-mail addresses: kanchana.thilakarathna@data61.csiro.au (K. Thilakarathna), fangzhou.jiang@data61.csiro.au (F.-Z. Jiang), sirine.mrabet@data61.csiro.au (S. Mrabet), dali.kaafar@data61.csiro.au (M. Ali Kaafar), aruna.seneviratne@data61.csiro.au (A. Seneviratne), xie@ict.ac.cn (G. Xie).

[1] http://www.readmetro.com/en/usa/new-york/

http://dx.doi.org/10.1016/j.comcom.2017.01.006
0140-3664/

where users can cache locally and share in vicinity the content they have downloaded via other networks. The users who contribute to the cache also gets access to the content in the cache. Thus, as the cache gets populated by content from contributors, users in the proximity can access it locally. *Crowd-Cache* creates a cyber-physical space where users can exploit and share content even when Internet may not be available. This is fundamentally different from an in-network cache system, as the cache is not in the data path between server and client, and content is pushed to the cache via a different network to which it was downloaded. Furthermore, although the concept is similar, *Crowd-Cache* differs from caching in small cells, i.e. femto-caching, that *Crowd-Cache* servers are mobile and provided by user devices for the benefit of other users in proximity.

There is no additional cost for the users of the *Crowd-Cache* system, except from a small energy penalty for pushing some content to the cache which we evaluate in this paper. There is no need to predict users' interests, as what is available in the cache will be what other users of the system consumed. The appropriateness, and the content integrity and authenticity can be perceived as an obstacle for *Crowd-Cache* deployment. These can be adequately addressed through well known techniques that are used in the user generated content hosting platforms as discussed later in this paper.

This paper makes the following contributions:

- We design a novel mobile system to reduce mobile data traffic exploiting the transient co-location nature of mobile users, the spatio-temporal correlation of content popularity and enabling mobile networking in proximity along with the capabilities of the modern smartphone devices.
- We model the video content access and the corresponding content consumption patterns and behavior of mobile users in public transportation systems using a unique large real-life dataset containing more than 500K users and 9.5M video requests from a popular video content provider.
- We study the transient aspects of content consumption patterns and show that even though number of requests change with the time of the day, the popularity of different video categories do not change with time. Furthermore, we show that the actual consumption is less than 30% of the total view time for more than 80% of the video requests and the view time exponentially decrease with regard to the length of the video.
- We simulate the proposed system and show that more than 80% of the passengers can save at least 40% on their cellular data usage during a typical average city bus commute of 10 min. In particular, the cache hit rate can vary from 25% to 65% depending on the shape of the content popularity distribution. In addition, the performances of various cache replacement policies are investigated and compared with real life content request patterns.
- We demonstrate the feasibility and practicality of the proposed system through the development of an Android application to evaluate the energy consumption and data transfer latency experimentally.

The remainder of the paper is organized as follows; Section 2 summarizes the related work. The *Crowd-Cache* system details are presented in Section 3. We model our proposed system using a real-world dataset in Section 4. In Section 5, the performance of the system is evaluated first through a simulation study and then through measurements of the Android-based real-life implementation in Section 6. Finally, Section 7 discusses future work and Section 8 concludes the paper.

2. Related Work

Of the three broad categories of peer-to-peer, traffic offloading and caching systems, the most relevant to *Crowd-Cache* is the work on caching systems. The work on caching again broadly falls into three areas: caching at the edge of the network, caching for personal use and opportunistic caching. Additionally, we also compare and contrast systems that provide similar functionality as *Crowd-Cache*.

2.1. Caching at the edge of the network

There are many proposals for proactive caching at the edge of of the network [13,14]. Mashhadi et al. [13] propose opportunistic download from dedicated Access Points (APs) that do proactive caching. This requires, however, the availability of APs with internet access. In addition, all nodes behave as peers, with all the trust, security and privacy issues associated with peer-to-peer systems. The cache in *Crowd-Cache* in contrast does not require internet access, and the nodes do not behave as peers. VideoFountain [14] deploys kiosks at popular venues to store and locally distribute content. The primary aim is to cache messages that users generate at different locations where the kiosk is located, which are then made available to users in the vicinity of a kiosk. Thus the system, similarly to opportunistic networks, relies on human mobility to carry content between kiosks. In contrast, *Crowd-Cache* does not rely on the users explicitly carrying information, but rather on the transient colocation of mobile users and the spatio-temporal correlation of content popularity. Erman et al. [15] propose a cost-benefit trade-off model to investigate the caching benefits at different levels of a cellular network. However as mentioned earlier, caching at the base stations is not effective as the bottlenecks quite often occur on the last hop wireless link in mobile networks.

2.2. Caching for personal use

Similar to in-network caching, caching at the end-user devices (e.g., smartphones) has also been studied previously [8,16]. Qian et al. [8] show that there are 17–20% redundant data transfers on cellular networks as a majority of current mobile web applications under-utilise the caching capabilities. In [16], the authors focus on the QoE improvement of web browsers and show that 60% of the requests can be served by a browser cache of only 6MB. Predicting user consumption of content and pre-fetching the content by caching it on the user's device to minimize the network congestion and cost, has been also evaluated in [9,17,18]. The effectiveness of content pre-fetching heavily relies on accurate prediction of future demand and the ability to find uncongested and lower cost network to pre-fetch the data. However, it has been shown that accurately predicting user behaviour and the network availability is a challenging task [19]. In addition, it also raises numerous privacy issues [20]. *Crowd-Cache* caches content and makes it available to other nearby users, which is inline with studies that argue in favour of exploiting redundant data transfers [8,16] and emanates the system from the need for user behavior and network availability predictions.

2.3. Opportunistic caching at mobile devices

Taghizadeh et al. [21] present a social community based cooperative caching system. It is aimed at minimising the cost of content distribution to users with common interests that are physically co-located. All users cache content via an ad-hoc network. Ioannidis et al. [4] and Han et al. [22] propose a distributed caching mechanism for the purpose of social welfare where users cache content

and opportunistically propagate it through a networking infrastructure. Similarly, Whitebeck et al. [5] propose a hybrid content delivery system in which content is distributed opportunistically, with acknowledgments to a central service provider. VIP delegation [23] replicates data using networking infrastructure on a few "socially important" (VIP) users in a mobile network. VIPs in turn distribute the content to other users opportunistically. In our proposed content sharing and collaboration scheme *uDrop* [24], user devices are leveraged to provide a distributed mobile cloud storage service. All these cooperative caching methods suffer from the variable delays, higher energy consumption and the trust, security and privacy issues of opportunistic networks. In contrast, *Crowd-Cache* users only need to trust the *Crowd-Cache* server, and does not introduce any significant energy costs or delays. In fact as we show in this paper *Crowd-Cache* improves the accessibility and hence the user *QoE* due to the higher data transferring rates of short-range networks.

2.4. Collaborative media sharing in public transport

There has been works proposing to install information hubs in public transport vehicles [25,26]. Bluespots [25] is a bluetooth based peer-to-peer distribution framework. While Bluespots is designed to function as an information waypoint in public transit, it does not act as a data transit system. Hence, streaming user QoE wont be improved as proposed in *Crowd-Cache*. DragonNet [26] leverages the spatial diversity of wireless signal quality across different parts or public transport vehicle to reduce the amount of Internet outage and improve throughput when traveling. The proposed protocol reduces average communication blackout from 6 s to 1.5 s, and at the same doubles the aggregate throughput. In addition, Tasiopoulos et al [27] propose a user QoE assessment framework for collaborative media streaming system in urban railway networks. Despite numerous studies in the area of content distribution in public transport, to the best of our knowledge, *Crowd-Cache* is the first to bridge theoretical work in video content distribution in public transport to a real-world system implementation in Android that could be deployed in large-scale.

3. Description of Crowd-Cache

Crowd-Cache leverages off the transient colocation of devices and the epidemic nature of content popularity, namely the observation that users in a particular location or users traveling to a same destination are likely to be interested in the same set of content with a high probability [28,29]. This is done by allowing the users to contribute the content that they have downloaded and consumed to a local store (*CC-server*) via a local network (*CC-LAN*), and making content on the *CC-server* available locally. If content is obtained from the *CC-server*, users will reduce their cellular data usage. Moreover, thanks to the higher capacity of the local networks, latency will be minimized and as a result, the user *QoE* will be improved.

3.1. Crowd-Cache: an overview

Users access the *Crowd-Cache* system via an application, *CC-app*, on their devices[2]. The *CC-app* enables users to access web content via the cellular network and via the *CC-LAN* when available. The *CC-server* is provided by either another mobile device or a dedicated device, acting as a local server. The *CC-server* also acts as an access point for the *CC-LAN*. Access to the *CC-LAN* is granted securely through the *CC-app* using standard WiFi authentication, i.e.

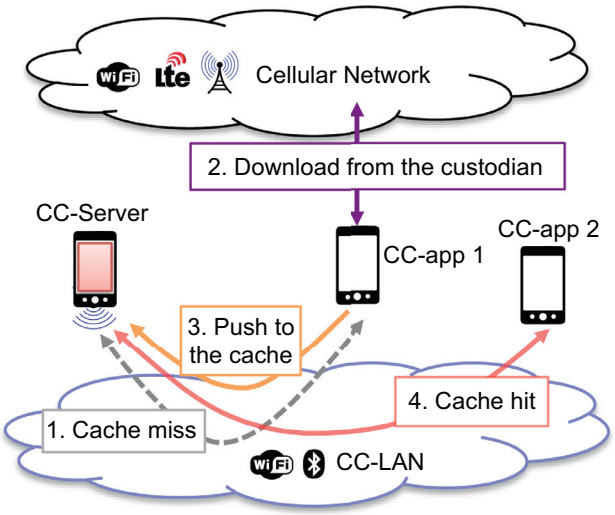


Fig. 1. Operations of the *Crowd-Cache* system.

WPA. When a user attempts to access content, the *CC-app* connects to the *CC-server* via the *CC-LAN*. If the content is available on the local *CC-server*, the content is served from the *CC-server*. If not, the *CC-app* connects to the cellular network and downloads the content. Once the content is downloaded and consumed, the *CC-app* pushes the newly downloaded content to the *CC-server* when the *CC-app* reconnects with the *CC-server*. The *CC-server* makes a local decision as to whether or not to update its cache with the newly pushed content using a content replacement strategy. The operations of the *Crowd-Cache* system are schematically shown in Fig. 1.

3.2. Application scenario and incentives

While we conceive several deployment scenarios of the *Crowd-Cache* system, we aim to focus on its deployment in public transport at this stage. It is reported that 83% of the US commuters use mobile phones during their daily commute[3]. Additionally, it has been observed that there is spatio-temporal correlation of content access patterns of commuters in public transport [29]. To this end, we believe that content distribution within city buses could be a potential application scenario for *Crowd-Cache*. In this scenario, a device under the control of the bus driver, powered via the power outlet of the bus, acts as the *CC-server* and the access point for the *CC-LAN*. Initially, the *CC server* would be either empty or only contains the data that was downloaded by the bus driver via a cellular network connection. The *CC-server* is then populated by commuters as they get on the bus and start using the *CC-app*. The users will not use the app and provider, e.g. transport companies, will not deploy the service, if they don't have tangible incentives from the service. The potential incentives and disincentives of the CC-service are described below:

For CC-app users: Incentives for users to subscribe to the *Crowd-Cache* system is naturally driven by the willingness to minimize their cellular data downloads. Furthermore, as will be shown in Section 6, when the content is delivered from the local *CC-server* via the *CC-LAN*, latency and the device energy consumption are reduced. User will be able to and prefer to use their own devices, which are already customized with their personal preferences, e.g., browser bookmarks, and user accounts, for in-bus/train entertainment compared to a fixed entertainment systems with screens attached to seats.

[2] The *CC-app* could also be an extension (plug-in) to mobile web-browsers.

[3] https://www.gfi.com/blog/survey-95-6-of-commuters-in-the-us-put-company-data-at-risk-over-free-public-wi-fi/

With respect to energy consumption, the extra energy consumed to push the downloaded content to the *CC-server* will be offset by the reduction in energy consumption when downloading data from the local *CC-server* since the energy consumed through a WiFi connection is provably lower than that of cellular networks [30]. The *CC-server*, on the other hand is, in our public transportation scenario, connected to a power source.

For CC-server providers: *CC-app* can either be supported via online advertisement or through subscriptions. The ad supported version of the app displays an advertisement whenever it receives data from the *CC-server*. However, as these ads are locally served ads, received data does not consume any of the (capped) cellular data plan of the users. Typically, local ads include company personalized promotional material to the travellers, including the purchase of tickets for future travels and trip information etc. The *CC-app* and *CC-server* update their advertisement impression counts whenever they connect to a low cost network, for example their home WiFi network, for accounting and detecting fraudulent activities. The revenue generated from the subscriptions or advertisements will be used to compensate the *CC-server* provider. However, *CC-app* users are not expected to spend more than what they would have saved for their cellular data expense.

In addition, there are multiple incentives for a transport company to become *CC-server* provider. As the services will be accessed through the travelers' own devices, the deployment costs are minimal for the transport company to providing in-bus/train entertainment to passengers. Since *CC-system* does not provide Internet access to passengers, it is cheaper to operate than traditional WiFi hotspots as there is no additional cost of broadband Internet connections. Furthermore, *CC-system* can be easily extended to provide relevant analytic and intelligence about customers to the transport company, such as identifying returning customers (possibility of reward program) and efficient customer feedback.

4. Crowd-Cache system model

To investigate the effectiveness of the cellular bandwidth saving for *CC-app* users and the benefits received by *CC-servers* through serving content to *CC-app* users, we first model the *Crowd-Cache* system using a real-world dataset (as described below), by deriving probabilistic models of user behavior for mobile content access. Then, the derived models are used to simulate the behavior of *Crowd-Cache* system under various workload/user conditions. The dataset provides insights of user consumption pattern, and we also vary the model parameters to gain extra insight under different scenarios.

4.1. Dataset in use

We use logs of video content access of mobile users of PPTV, one of the largest VoD service providers in China, for one week in December 2011. PPTV has 22 categories of videos [31]. The logs provide content requests from three major cities - Shanghai, Beijing and Tianjin, which correspond to 9.5 million requests, generated by more than 500K mobile users through mobile web-browsers or the client app on various mobile devices, e.g. iPhones, iPads, Android devices. Table 1 summarizes the statistics of the dataset.

There are several limitations of the dataset. First, due to the high percentage of movies and TV series episodes, the average length of a video is about 50 min, which is much larger than online video services such as YouTube. However, the average length of actual view time is about 18 min. Secondly, majority of requests are done using WiFi ($\sim$ 75%). Although user viewing behavior differs slightly when they are using cellular and WiFi as shown in [32], we believe that user of *Crowd-Cache* would behave similar to using WiFi due to free mobile data cost while accessing content. As a result, we do not treat WiFi requests and cellular requests separately in our system modeling. Additionally, the number of requests per user per day is low due to the factor that major of users are not heavy users, therefore, in the remainder of this section, we analyse the popularity and the size characteristics of videos, as well as the transient aspects of content access and consumption patterns of this data for the purpose of modelling the *Crowd-Cache* system.

Table 1
Summary of PPTV dataset.

Field	Statistics
Duration	7 days–Dec. 2011
No. users	516,149
No. content requests	9,579,576
Video categories	22 (news, movies, etc)
Per. of WiFi requests	76.15%
No. requests/user/day	2.65
Avg. length of a video	50 min
Avg. length of view time	18 min

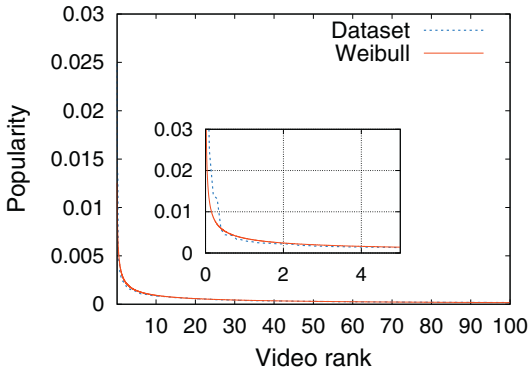


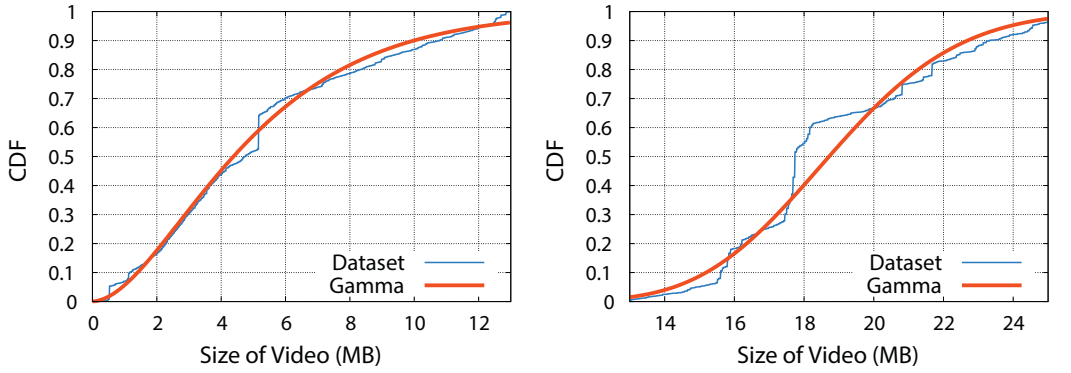
Fig. 2. Content popularity of PPTV dataset.

4.2. Popularity distribution of video content

It has been previously reported that the popularity of online video content is best modelled by a Zipf-like distribution, i.e. Weibull [32,33]. Fig. 2 shows the maximum likelihood estimation (MLE) fit of a Weibull distribution for the PPTV dataset using the SciPy Library[4]. The videos are ranked based on the number of requests. We obtain the MLE parameters for the Weibull distribution equal to a shape α of 0.48 and a scale λ of 1875. This suggests that a Weibull distribution represents a good approximate of the actual popularity distribution (with an R^2 goodness-of-fit value of 0.914).

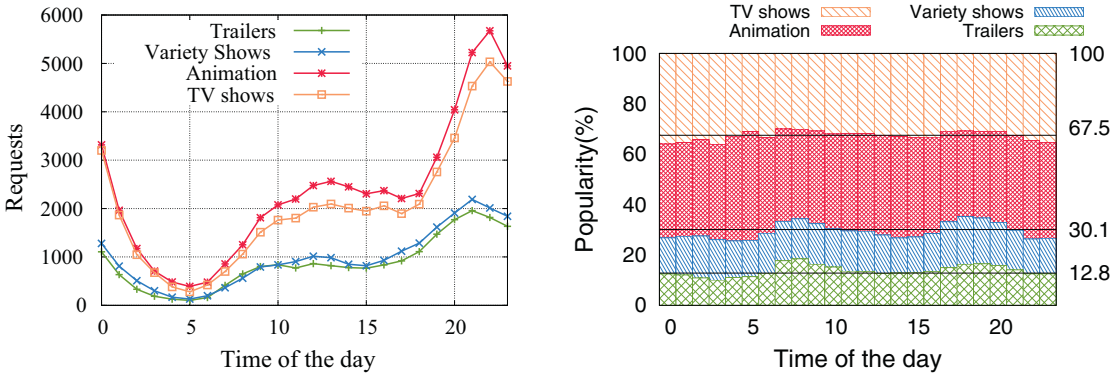
The popularity of content *i*, P_i is then considered as a Weibull probability function such that $P_i = \frac{\alpha}{\lambda}\left(\frac{i}{\lambda}\right)^{(\alpha-1)} e^{-(i/\lambda)^{\alpha}}$ where $i > 0$ and α, $\lambda > 0$. The popularity of a content is actually the probability of that content is being requested in the future. For contents that is already in the cache, all future requests will be a cache hit. Therefore, the sum of probabilities of requesting a content provides asymptotic expected cache hit rate for the cotnent that is in the cache. However, the popularity distribution in Fig. 2 has a considerable long tail due to the large variety of content access in a large geographical area. Since the *Crowd-Cache* system is mainly designed to operate in a confined geographical area for a limited period of time, the distribution of the content popularity should be narrower than the PPTV dataset. Hence, we further investigate the effects of content popularity in Section 4 by varying the shape

[4] SciPy Lib - http://docs.scipy.org/doc/



(a) Gamma distribution model $k_1 = 2.139, \theta_1 = 2.449$ (b) Gamma distribution model $k_2 = 52.388, \theta_2 = 0.343$

Fig. 3. Video size distribution of all content.



(a) Number of requests (b) Popularity of categories

Fig. 4. Popularity of video categories during a day.

and scale parameters of the Weibull distribution whilst considering different content popularity scenarios.

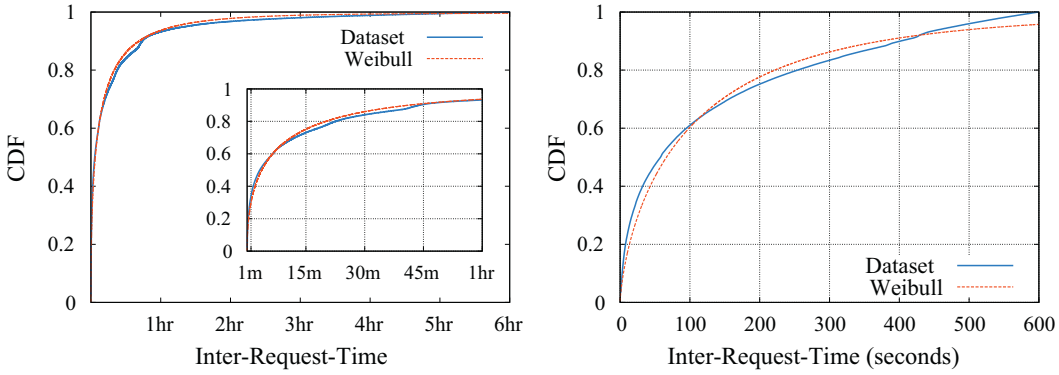
4.3. Size distribution and video categories

The video size (in bytes) directly affects the storage capacity of the *CC-server* devices. As our dataset contains the length of a video in seconds, we calculate the size of videos with a bit rate of 330Kbps [34]. For some videos, the bit rate could be larger. However, even with this smaller value, the dataset contains significantly large videos, i.e. movies, which does not match with our application scenario of city buses because the probability of someone watching a complete movie while commuting in a city bus would be negligible in practice. Therefore, we only considered videos of length less than 10 min which accounts for 990,219 unique set of videos. The size of videos has been reported Gamma distributed in various studies [33] such that the size of content i, corresponds to a probability function $S_i = \frac{i^{(k-1)}e^{-(i/\theta)}}{\theta^k \Gamma(k)}$ where i, k, $\theta > 0$ and $\Gamma(k)$ is the Gamma function evaluated at k (we denote k and θ the shape and scale parameters of the distribution respectively). However, the overall model does not fit well with the actual size distribution (R^2= 0.31), due to the high percentage of certain video sizes. Therefore, the size of videos are modeled in two categories: 0–13MB and 13MB–25MB as shown in Fig. 3. The probability of requesting content of the two categorizes is 1.8:1, based on the popularity of videos in the two size categories.

4.4. Transient aspects of content request and consumption

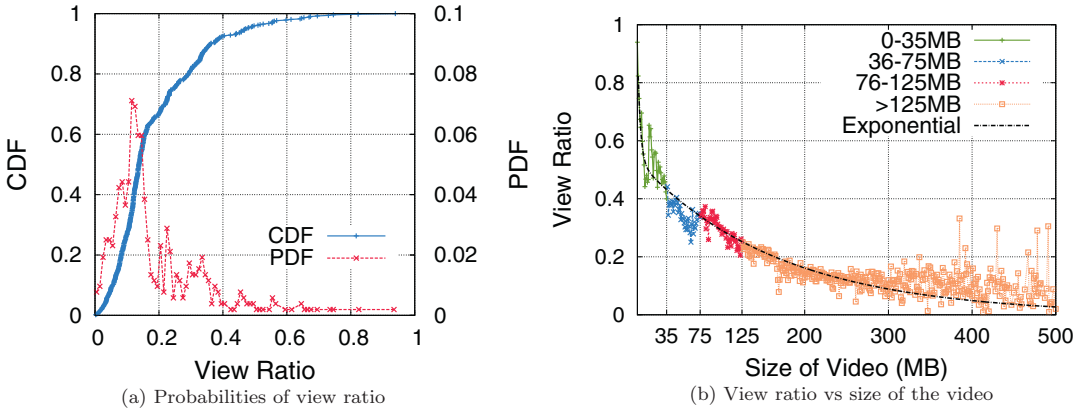
Next, we study the popularity of different content categories over the time of the day. Fig. 4a shows the average number of content requests of videos of top four main categories, namely movie trailers, variety shows, animation and TV shows, during each hour of the day. Only the top four categories are considered as it covers significant portion of the dataset and also clearly shows the dynamics of user interest over the time of the day. During the working hours ($\sim$ 9am to 6pm), the number of requests are relatively stable across all four categories. Since analyzing characteristics of different categories have been studies in prior works, i.e. [31], and is not the focus of this paper, we do not model user consumption pattern in a category basis. However, the model could be extended in our future work. The number of requests increases rapidly, peaking between 9pm–10pm. During this period, there is a greater number ($\sim$ 5500 requests per hour) of requests for the larger categories of animation and TV videos. However, if we normalize the popularity of each category from the total number of requests for the particular hour, all categories show approximately steady behavior throughout the day as illustrated in Fig. 4b. Therefore, we consider that the popularity of each video/size category does not change with the time of day.

The average number of videos per user per day is about 2.65, which is considerably low and also depicts the heavy-tailed distribution of inter-request-patterns. We extract inter-request-time for individual users from the dataset and then model the inter-request-time for content i, I_i as a power law distributed, Weibull, with variables γ and β as the shape and scale parameters respectively. Fig. 5a shows the CDF of inter-request-time of all users in PPTV and the Weibull distribution with MLE fit parameters of $\gamma = 0.5$ and $\beta = 456.14$. In particular, approximately 50% of users request a video at least every 5 min and more than 80% every 20 min. Since this is for bigger videos, the inter-request-time can be expected to be lower for other online video sharing services. In



(a) Global IRT (b) Distribution of IRT<10min

Fig. 5. Inter-Request-Time (IRT) time distribution of individual users.



(a) Probabilities of view ratio (b) View ratio vs size of the video

Fig. 6. Distributions of view ratio of videos.

addition, it is expected that in the application scenario of *Crowd-Cache* users would tend to request content more frequently as mobile devices are generally used for shorter periods by transit passengers [35]. Therefore, we focus on the distribution of inter-request-time of less than 10 min considering an average duration for a bus stop is approximately 10 min. Fig. 5b shows that again it follows the shape of a Weibull distribution with $\gamma = 0.7$ and $\beta = 110$ parameters and R^2 value of 0.9909, where 80% of users request new content in at least every 3–4 min.

Fig. 6a shows the probability distributions of the view ratio, defined as the view time normalized by the length of the video, for all videos. It shows that more than 80% of the videos have less than 0.3 view ratio. Even though the average length of a video is as large as 50 min, the users rarely watch a full video leading to an average actual view time of just 18 min (Table 1). Moreover, the median view time is as low as 1 min. This needs to be considered in designing systems such as *Crowd-Cache*, as it determines whether it is reasonable to cache the full content. However, the average view ratio is considerably higher (> 0.4) for the shorter videos (< 35MB) as shown in Fig. 6b, and as expected, the view ratio reduces as the size of video increase. We model this relationship of view ratios as a linear combination of exponential functions such that, the view ratio of a video of size $s > 0$ as $V_s = a * exp(\lambda_1) + b * exp(\lambda_2)$ where $a = 0.4$, $b = 0.53$, $\lambda_1 = -0.3$, $\lambda_2 = -0.006$ and $\exp(\lambda) = \lambda e^{-\lambda s}$. The exponential model provides considerably close representation of content view ratio of PPTV users with 0.995 R^2 goodness-of-fit test value.

4.5. Transient aspects of passengers on a bus

We consider a scenario of hosting the *CC-server* in a bus as described in Section 3, where users are within the communication range for the duration of the bus journey (T). The number of users, N, is dependent on the number of passengers which we consider equals a constant, say 50, for simplicity. Since *Crowd-Cache* is a passive content storage, the transient aspects of *CC-app* users such as the number of users in the bus at a given time, duration of the bus journey of each user and then, content access and consumption behaviors of users determine the effectiveness of the *Crowd-Cache* at a given time. We model the transient aspects of the bus journey as follows.

We assume there are bus stops every 10 minutes in a bus route. For simplicity, we also assume that all passengers on board use the *CC-app*. The time a user is connected to the *CC-server*, (τ), is considered to follow a Log-normal distribution which models well the bus journey travel time as suggested in [36]. That is, after $\tau = e^{\mu+\sigma^2/2}$ time on average *CC-app* users get off the bus and disconnect from the *CC-server*, where $\mu = 0.6197$ and $\sigma = 10.48$ are mean and standard deviation of τ.

An overview of the transient aspects for content requests are, therefore, modeled in Fig. 7. In a nutshell, during the period of interaction with *Crowd-Cache* system, passengers are to request content based on the modeled popularity and size distribution based on the extracted inter-request time distribution. The view ratio of each video content is also modeled depending on the size of the video as discussed. In addition, an overview of traffic model

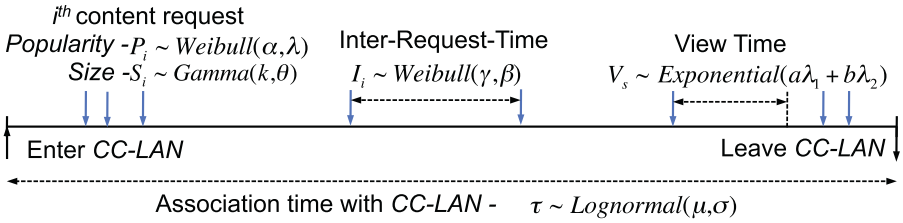


Fig. 7. Content request model diagram.

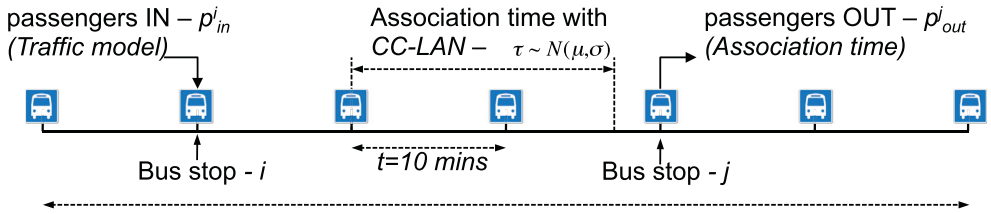


Fig. 8. Traffic model diagram.

Table 2
Summary of *Crowd-Cache* system model.

Transient nature of content access behavior	
Content popularity	Weibull - α=0.48, λ=1875
Size	Gamma - category1: k_1=2.139, k_2=52.388 category2: θ_1=2.4486, θ_2=0.3425
Category ratio	category1 : category2 = 1.8 : 1
Inter-Request-Time	Weibull - γ=0.7, β=110
View ratio	Exponential - a=0.4, λ_1=-0.3, b=0.3, λ_2=-0.006
Transients of bus scenario	
Duration of a ride	T=1 h
Bus stops	Every 10 mins
Capacity of a bus	N=50
No. of Passengers	Peak, Off-peak, Random
Association time	Log-normal (μ=0.6197, σ=10.48)

is shown in Fig. 8. We assume the following three traffic models, which broadly represent bus commuters.

- *Peak:* The bus is full all the time, i.e. $N = 50$, i.e. assuming the same number of passengers getting on and off the bus.
- *Off-peak:* 10 passengers get on board at every bus stop, if there is enough room. If not, the bus will be filled up to its maximum capacity.
- *Random:* At the bus stop, a random number of passengers gets on board. The number varies from 0 to the remaining capacity of the bus.

5. Performance evaluation

We developed an event-driven simulator based on Python to evaluate the performance of the *Crowd-Cache* system. Simulator uses system model parameters summarized in Table 2 as input to determine the cache hit rate and the bandwidth savings.

We assume that there are no packet losses and no significant transfer delays as the system only needs to support up to a few tens of users in a limited area (due to maximum number of concurrent WiFi connection). Higher number of users could be supported by multiple *Crowd-cache* at multiple positions. In the simulation, to reflect the variation of view ratio of the same content size, we randomly assign a view ratio for a particular content size from the actual dataset. This, we believe, will better reflect the actual user viewing behavior comparing to the simple view ratio model shown in Fig. 6b. For each evaluation metric, we run 20 simulations and show the average value. We first consider the case where there is an unlimited storage at the cache. Then, we also examine the system performance under practical resource constraints along with different cache replacement policies.

5.1. Performance with an unlimited cache – The cache hit rate and bandwidth saving

The expected cache hit rate, referred here as ($E[H_s]$) is the probability of finding a user requested item in the cache, thus, $E[H_s] = \sum_{\forall i} P_i$ where P_i is the popularity of content i. The actual cache hit rate is the ratio between the number of hits and the total number of requests for a considered time duration. If the content popularity distribution does not vary in time, the actual cache hit rate should converge to $E[H_s]$ asymptotically. Intuitively, regardless of content request pattern, the long term hit rate shall approach expected cache hit rate. However, real-world content access pattern is always highly dynamic and correlated. We present the comparison of expected hit rate with actual hit rate to evaluate the impact of request pattern correlation.

Fig. 9a shows the cache hit rate values after populating cache for 24 h with three traffic models under various content popularity distribution shapes by varying the parameters of the Weibull distribution. The scale of the content popularity distribution is considered equal to the empirical PPTV dataset ($\lambda = 1875$).

The peak traffic model gives the highest cache hit rates due to the larger number of contributors to the *Crowd-Cache*. In particular, the expected cache hit rate is higher than 55% in the peak traffic scenario, regardless of the type of popularity distribution. Notably, in the off-peak scenario, the observed minimum cache hit rate is still greater than 25% for the most distributed content popularity shape (α closer to one). The random traffic model yields to a 50% expected hit rate for the particular shape value ($\alpha = 0.48$) of the PPTV dataset, which suggests a considerable potential of high performance for PPTV customers and alike when using *Crowd-Cache*. Moreover, the random expected hit rate does not change significantly with the increase in the shape parameter validating the potential gain in local cache irrespective of the content popularity distribution. The difference in expected and actual value is significant, and actual value is always lower in comparison. The reason is mainly due to expected performance measures the long-term future performance of current Crowd-cache, while actual value measures the current traced hit rate considering prior measurements. In the case of unlimited cache, expected results would always be equal or higher than actual results. Similarly, Fig. 9b shows the impact of the scale parameter of the content popularity distribution

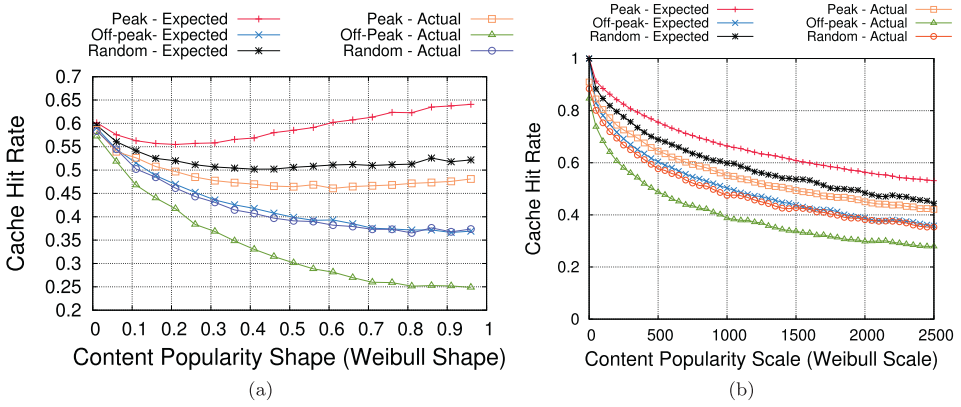


Fig. 9. The cache hit rate performance during a one hour bus ride. (a). as a function of the shape α of content popularity, and when $\lambda = 1875$, (b). as a function of the scale λ of content popularity, and when $\alpha = 0.48$.

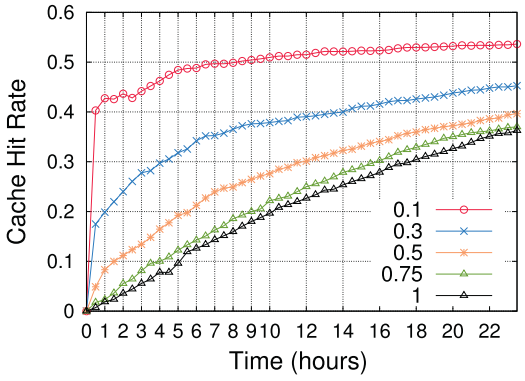


Fig. 10. Cache hit rate against time with varying α.

on the cache hit rate. As expected, hit rate drops as scale value of popularity distribution increases, because a larger scale represents a wider pool of content. Moreover, different traffic modes has a similar impact to the hit rate as in Fig. 9a.

Fig. 10 depicts the variation of the actual cache hit rate values during one day bus ride, starting from empty *Crowd-Cache* with random traffic model. It shows the results for five different content popularity distributions. $\alpha = 0.1$ represent the extreme case of very large long-tail popularity distribution. $\alpha = 0.5$ represents the popularity distribution for PPTV dataset. For $\alpha = 0.1$, the probability of requests for the most popular items is higher than other distributions. As a result, there is a rapid increase in the cache hit rate early during the bus journey and stabilizes after the peak observed at the first stop of the bus reaching a 50% hit rate, due to the disparity of the content requests. As expected, the actual cache hit rate monotonically increases with time, as the new users joining the system tend to request popular content that is already cached. The performance of *Crowd-Cache* reaches a hit rate of 40–50% after one day of operation.

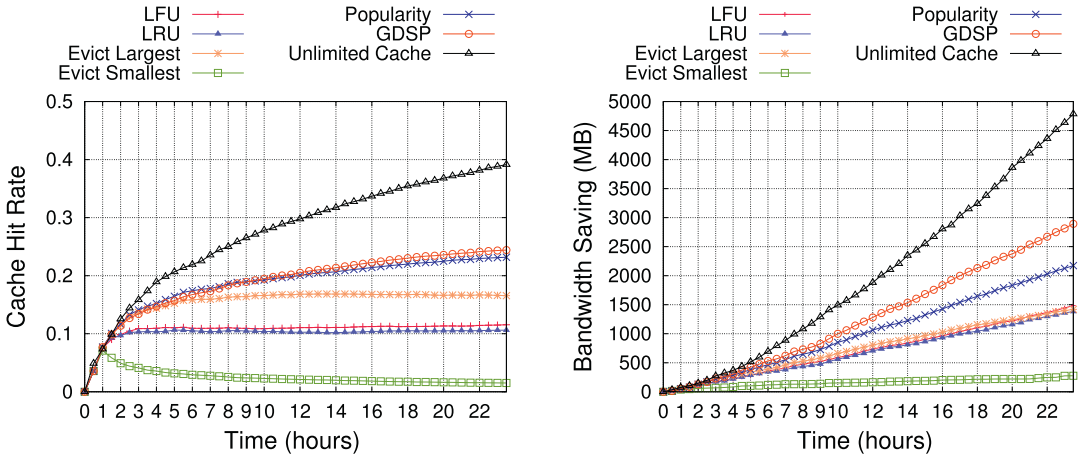
5.2. Performance with limited cache – impact of cache replacement policies

Isolated caching policy performance is a very well studied topic in the past decade. Cache replacement policy performances are highly dependent on two factors, the popularity dynamics of content and content request pattern. Intuitively, a highly dynamic content pool with large number of new arrivals might benefit more from recency based policies, while popularity based policies might perform better with periodic request patterns. However, the consensus has been that no single caching replacement policy would out perform all other policies in all scenarios. We picked some of the most representative policies to evaluate *Crowd-cache* performance under proposed public transit model.

To represent an extreme case, where the cache is provided by a low end smartphone, we restricted the cache size to 2GB. For this evaluation, the content popularity was considered to be similar to the PPTV dataset. We also assumed that the requested content's total size exceeds 2GB approximately in a 1 h period, after which the content was replaced according to the cache replacement policy. We considered a number of cache replacement policies. The recency based LRU (Least-Recently-Used) and frequency based LFU (Least-Frequently-Used), two intuitive schemes, namely Evict Smallest/Largest, and two function based schemes. The two function based schemes either evicted content that has the lowest popularity value (P_c) per unit size (S_c) (popularity/size) or content with lowest utility metric $U(c) = L + M_c P_c / S_c$, such that $L \leftarrow \min\{U(c)\colon \text{ every c in the cache}\}$ (Greedy-Dual-Size-Popularity (GDSP) [37]). The cache miss penalty M_c is considered to be one as the retrieval cost for cache misses are equal for all content items. GDSP was proposed to consider all three caching metrics, namely recency, size and popularity.

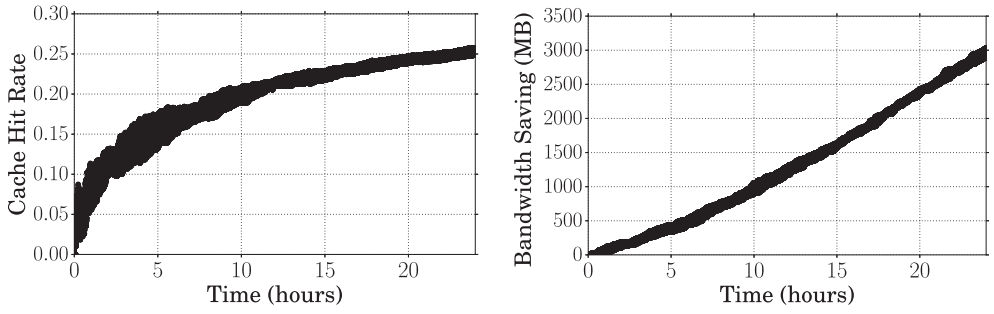
Fig. 11a shows the cache hit rates observed for different cache replacement strategies. All the strategies, except Evict-smallest, result in cache hit rates higher than 10% after a 1 h trip. As expected, the two content popularity-based strategies show relatively high cache hit rates. Although LRU and LFU are two of the most popular cache replacement strategies, their performance is not superior since content popularity is a better metrics than frequency in a independent reference model (IRM) of content popularity. GDSP and Popularity show similar and best results for the considered environments. However, GDSP is expected to perform better than Popularity when the content popularity change with time, since GDSP takes content access recency into account in its objective function.

Higher cache hit rate does not always result in the highest bandwidth saving as can be seen in Fig. 11b. Higher bandwidth saving also depends on the size of each hit content. For instance, even though cache hit rate for Evict-largest is higher than LRU and LFU by about 8%, bandwidth saving for both these strategies are almost similar. This is due to the fact that Evict-largest keeps more number of small videos in the cache. Furthermore, the difference between GDSP and Popularity are comparatively higher in bandwidth saving despite the fact that their hit rate performance is close. We do not focus on proposing a new caching replacement policy, while attempts to evaluate performance differences under our unique public transit scenario. Overall, if we employ a cache replacement policy which takes recency, popularity and the size of the content, the *Crowd-Cache* system performs reasonably well due



(a) Cache hit rate (b) Bandwidth saving of all users

Fig. 11. Performance with limited cache at *CC-server* device, cache size=2GB.



(a) Cache hit rate (b) Bandwidth saving of all users

Fig. 12. Performance variance of GDSP

to the observed spatio-temporal content popularity correlation in proximity even under extreme conditions with only a 2GB cache. Moreover, the results show the worst case performance, i.e. starting a day with an empty cache.

Lastly, we further show the performance variance of GDSP with regards to both hit rate and bandwidth saving in Fig. 12. Fig. 12a presents the variance of cache hit rate by plotting the traced hit rate at different time instance t of all requests in multiple repeated experiments. It could be seen that the performance varies greater during the first 10 h, and the fluctuation shrinks while Crowd-Cache is being filled up and performance stabilizes. Similarly, Fig. 12b shows the variance aspect of bandwidth saving, and the range of saving is considerably stable regardless of the status of cache.

5.3. Transient aspects of passengers

Fig. 13a illustrates the effects of travel time of bus commuters which is determined by the log-normally distributed association time with *CC-server* (τ) and the duration between two consecutive bus stops. Since the same user is not going to request the same content multiple times, the *Crowd-Cache* receives diverse set of content requests (from the long-tail part of the popularity distribution) when the same set of commuters travel for longer periods. As a result, *Crowd-Cache* achieves higher cache hit rate for short distance bus rides, where there is frequent arrival of new commuters as shown in Fig. 13a. In addition, mean association time is the deceive factor as the cache hit rate does not vary significantly with the bus stop length for a given association time. Therefore, *Crowd-Cache* performs better in metro type transport scenarios similar to our application scenario. Overall, the heat map depicts that for the majority of the cases cache hit rate after 24 h reaches more than 30% even for 2GB cache.

Additionally, we investigate system cache hit rate under GDSP with different cache sizes and IRT in Fig. 14. Both the shape γ and scale β parameters of IRT are varied to evaluate the system performance and cache size requirement under different workloads. In general, the γ value controls the distribution shape while the scale parameter β represents the skewness of the distribution. Fig. 14a shows that higher γ values translates to higher cache hit rate regardless of cache size due to the increase in total number of requests. In contrast, Fig. 14b illustrates that higher β values results in lower cache hit rate due to the large IRT value of the majority. In addition, cache size does not show significant effect after approximately 40GB which implicitly indicates the practical cache size requirement for *Crowd-Cache*.

5.4. Benefits for Crowd-Cache users

For a *CC-app* user, the amount of saving from the monthly data cap would be one of the primary objectives to use the *CC-app*. Therefore, we evaluate the cellular bandwidth saving for individual users during a bus ride. Each user associates with *CC-LAN* only for $\sim$ 12 min in average according to log-normally distributed association time. Since *CC-server* starts empty at time=0, the passengers during first several hours would be the users that receive lowest benefit.

In the first four hour, nearly 65% of the users does not save any bandwidth irrespective of the size of the cache as shown in Fig. 13b, while that reduces to $\sim$ 20% for the passengers that get

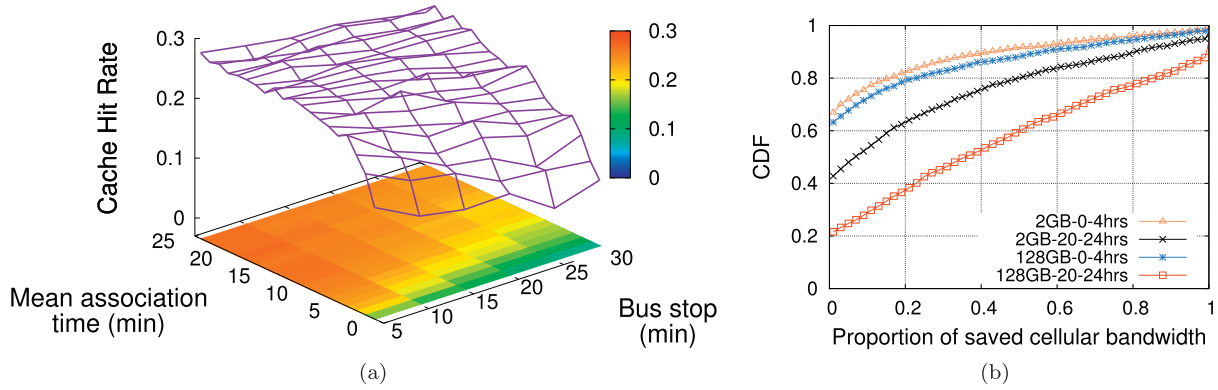


Fig. 13. (a) Cache hit rate against transient aspects of passengers, cache size=2GB and the random traffic model. (b) Individual bandwidth saving for $\alpha = 0.48$.

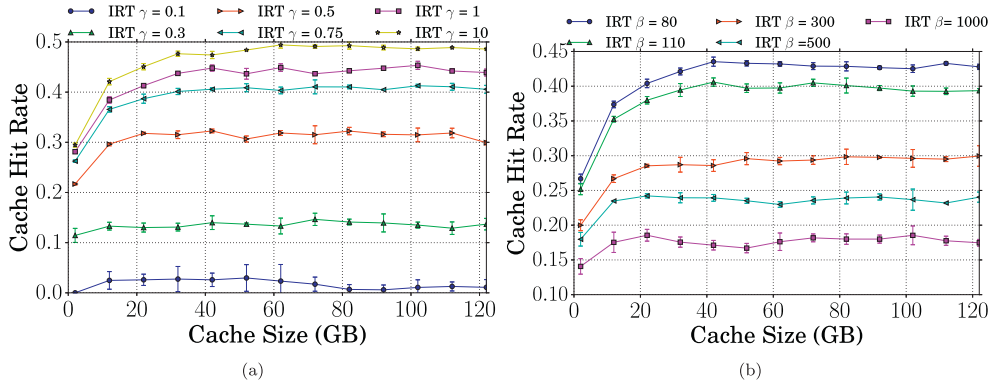


Fig. 14. (a) Cache hit rate against cache size and IRT shape γ (b) Cache hit rate against cache size and IRT shape β.

on-board at the end of the day. Since these results are for 2GB and 128GB caches, the savings are expected to be much higher in real-world systems as it is possible to upgrade the storage of the *CC-server* device with an external storage device. Fig. 13b illustrates that 80% of users save more than 40% of cellular bandwidth after 20 h of initial bootstrapping phase if the cache is at least 128GB in size.

5.5. Downloading and caching partial content

As we have shown in Section 4, users rarely watch a full video. In fact, longer the video is lower the proportion that the users consume. Therefore, it makes sense to download and cache only the parts of content that users actually consume. Although in real-world scenario applications will always download more than users would consume to reduce the time of buffing and improve user QoE, we consider the case which we are able to download exactly the length of video users are to actually consume.

Fractions of videos that are consumed by users are therefore cached in *Crowd-Cache*. Once requested, the existing part of the video will be served locally. If the request is longer than the cached version, the additional pieces of the content are to be fetched from the original content server. Thus, the cached video will be updated to the newer version with every request. As a result, in principle, enabling partial content downloading and caching will increase the efficiency of caches. Thus, resulting in a higher cache hit rate when the cache size is limited. In Fig. 15, the cache hit rate is compared whether partial caching is enabled. The cache size is limited to 2GB and cache hit rate is monitored for a period of 24 h. GDSP is used as the cache replacement policy for comparison. It can be seen that enabling partial caching could potentially increase the hit rate from 25% to over 35% after populating the cache for 24 hours.

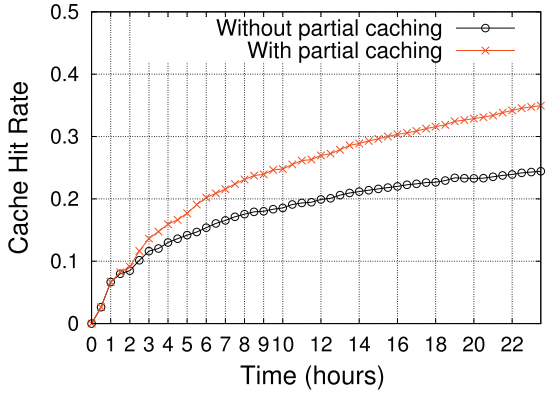


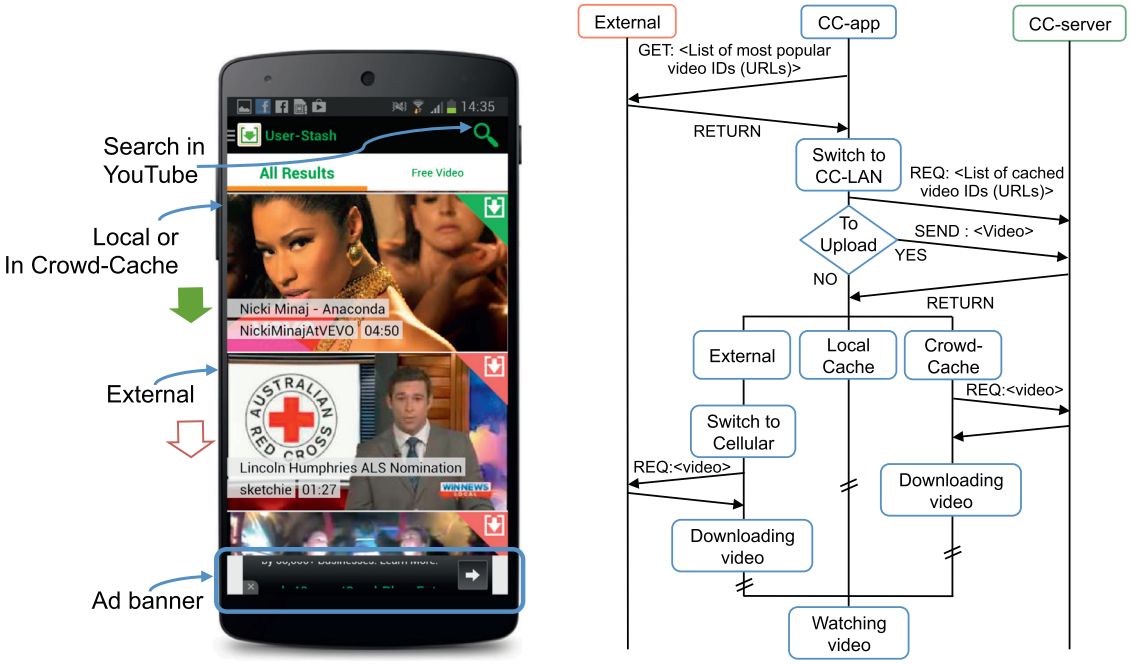
Fig. 15. Partial content caching comparison for 2GB cache size w.r.t. cache hit rate.

6. Experimental evaluation

We implemented *CC-app* as an Android app and *CC-server* as an Android app [38]. Although the system concepts are valid for any popular content type, in our implementation we focus on the distribution of video content.

The interface of the *CC-app* is shown in Fig. 16a. When the *CC-app* is launched, it requests the unique IDs (URLs) of the most popular videos from the video service providers. The current version includes both YouTube[5] and Dailymotion[6] content. The app also allows searching for a particular video or a set of related videos for a particular keyword. Once the relevant video IDs are received,

[5] https://developers.google.com/youtube/v3/
[6] http://www.dailymotion.com/doc/api/graph-api.html



(a) *CC-app*user interface (b) Communication protocol of *CC-app*

Fig. 16. Android implementation of *CC-app* and the communication protocol.

the networking interface switches to the available *CC-LAN* to obtain the list of videos that are cacheed in the *CC-server* as shown in Fig. 16b. If there is any new content in the local cache of the *CC-app*, the app pushes it to the *CC-server* in the background, while the user is scrolling through the search results.

Depending on the user request, the *CC-app* displays results, with an indication of whether the content can be obtained from the *CC-server* (green arrow) or needs to be downloaded via the cellular network (red arrow). If content can be obtained locally, the *CC-app* fetches the content from the *CC-server* via the *CC-LAN* at download rates between 2–6Mbps. If not, *CC-app* switches the network interface to the cellular network and downloads the video via the cellular network at a download rate of $\sim$ 400Kbps (the experiment was performed in 2014). The switching of networks is required because a majority of the current smartphones does not allow the simultaneous use of cellular and WiFi networking interfaces.

Measurements obtained using an implementation of the *CC-server* on a Samsung Galaxy S4 (i9306), and a *CC-app* on a variety of smartphones from different manufacturers and with various capabilities are shown in Fig. 17a.

6.1. Throughput and latency

Fig. 17a shows that the throughput obtained on different devices with different Android versions, when only one *CC-app* is in use. The maximum rate of $\sim$ 6Mbps was achieved by the Samsung i9306 and the lowest rate of $\sim$ 1Mbps was achieved by the Huawei U8950. As can be seen, the achievable throughput is dependent of the device type. Despite this, overall, the *Crowd-Cache* not surprisingly still achieves significantly faster data rates than practical cellular networks.

Fig. 17b illustrates switch time between the cellular and the WiFi networks. When a device attempts to connect with *CC-LAN* for the first time ("First connection"), it takes between 5 and 7 s to associate and connect. Subsequent connections only takes 2 to 3 s, since the authentication parameters for *CC-LAN* are stored under previously connected AP list. The switching time is barely noticeable as it occurs in the background whilst the users are scrolling through the search results. This is the worst case scenario, as these switching overhead are only temporary, since smartphones already start enabling the simultaneous use of multiple network interfaces[7].

6.2. CC-app device energy consumption

The device that hosts the *CC-server* is expected to be connected to a power source, and therefore energy usage will not be a concern. The *CC-app* however needs to be energy efficient. To investigate the energy usage of using *Crowd-Cache* system, the energy consumption of a smartphones when using the *CC-app* is measured by hijacking the battery which is connected to a shunt resistor ($R_s = 15m\Omega$). The voltage across the shunt resistor (V_s) is then measured using a National Instruments CCB-6008, a multifunction DAQ (NI-DAQ)[8]. We perform a measurement per every millisecond and export the results using NI-LabView. *CC-app* is also configured to log time-stamps for start and end of event categories shown in Fig. 17c. During the measurements, special caution has been taken not to introduce concurrent background activities.

We consider the cache hit and cache miss, assuming that the *CC-app* is in the foreground of the device, and that the device is first connected to the *CC-LAN*. We also measure the energy consumption when downloading the same content through a cellular network (3G case). For the 3G case, we eliminate any streaming protocol- and/or foreground user interface related power consumption discrepancies by using the *CC-app* to access videos in the two experiments. Fig. 17c shows the energy consumption normalized by the energy consumption of the 3G case.

As expected, cache misses results in more power consumption than 3G downloads due to the extra steps of checking the *CC-server* (Query time), and switching networks. It confirms that the larger

[7] http://galaxys5guide.com/samsung-galaxy-s5-features-explained/galaxy-s5-download-booster/

[8] http://sine.ni.com/nips/cds/view/p/lang/en/nid/201986.

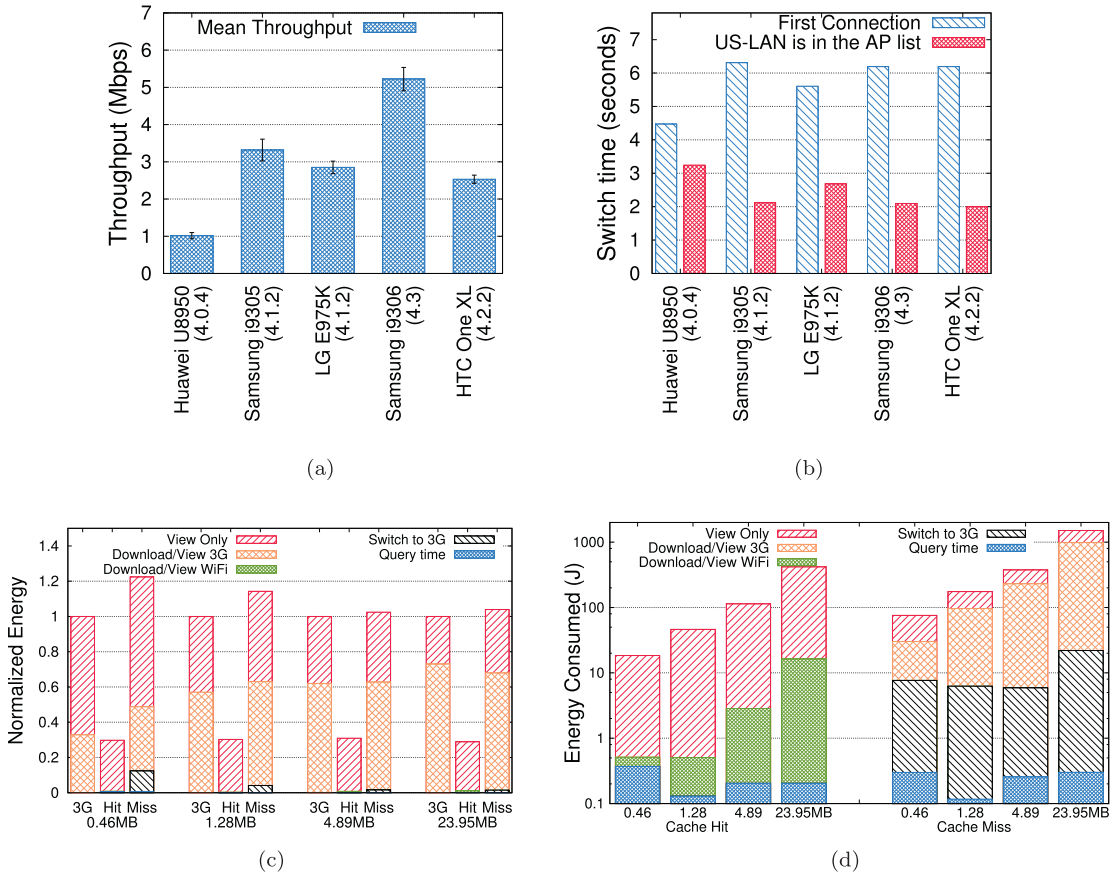


Fig. 17. Practical measurements of throughput and latency, (a). Throughput over *CC-LAN* for different device types, (b). Switch time to *CC-LAN* for different device types, (c). Client device energy consumption measurements for the two application scenarios of cache hit and cache miss compared to direct content access through cellular network, (d). Absolute energy value comparison (note the log scale).

the size of the requested file, the lower the relative energy consumption of the cache miss case. A cache hit results in significant energy savings compared to the 3G case regardless of the size of the video. There is a saving of $\sim$ 70%, primarily due to higher throughput achievable via the *CC-LAN*. The energy consumption for both download/view over *CC-LAN* and query time are small and is almost indistinguishable in Fig. 17c. Fig. 17d illustrates the absolute energy values of these small energy consumptions (note the log-scale). Query time proportion is larger for the smallest file as the total energy consumption is lower than other files. In the case of cache miss, switching to 3G consumes extra energy. However, it will not be a problem in future as the majority of smartphone will be able to simultaneously communicates over both WiFi and 3G interfaces. Overall, the amount of energy consumed in the cases of downloading via 3G networks is always larger than of local *Crowd-Cache* download, although the view only time is comparatively similar in all three cases. This is due to the fact that mobile device is still at the high power state even after downloading the content for the 3G case.

If we consider that there are four content size categories (denoted c), with a popularity likelihood of p_c and a normalized energy consumption of e_c^{hit} and e_c^{miss} for a cache hit and cache miss respectively, the expected normalized energy consumption E can be represented as a function of the cache hit ratio h as follows;

$$E(h) = h\sum_{\forall c} p_c e_c^{hit} + (1-h)\sum_{\forall c} p_c e_c^{miss}$$

We notice that $E(h)$ is a linear function of h with a gradient of $\sum_{\forall c} p_c(e_c^{hit} - e_c^{miss})$ and a y-intercept of $\sum_{\forall c} p_c e_c^{miss}$. If we assume that the content popularity of the four content sizes are similar to the PPTV dataset, $E(h)$ linearly decreases at a rate of $\sim$ 0.77 along with h. Moreover, for $h > 0.0968$, the normalized energy consumption $E(h)$ is less than one. Therefore, if the *Crowd-Cache* achieves at least a 10% hit rate, users are very likely to save on the device energy consumption.

7. Discussion and future work

The Android implementation of *CC-app* and *CC-server* presented here are expected to be developed further to enhance the practical feasibility of *Crowd-Cache* system in terms of user QoE and appropriateness, integrity, and authenticity of content. Standard practices such as hash filtering (registering with content identification databases[9]) will be carried out at the *CC-server* to identify and remove inappropriate content from the *Crowd-Cache* similar to any other user-generated-content distribution service. In addition, adequate take down policy, terms of service usage and privacy policy will be employed to address practical deployment issues of integrity and content authenticity.

We consider that the *CC-server* only caches content that are only cacheable as some of the video content providers may not allow users to cache content and as some content are exclusive to individual users. However, at the time when *Crowd-Cache* becomes a popular service, such an issue can be mitigated by negotiating with content providers since *Crowd-Cache* may enhance the distribution of their content. Moreover, we also intend to negotiate with local content providers such as PPTV. Since their customers

[9] e.g. https://www.audiblemagic.com/content-databases/

are monthly subscribed users, the service provider's requirement is to deliver the content to all users as cheapest as possible. The *CC-app* can be easily extended to support such a scenario, where there are separate channels for specific content providers, e.g. PPTV subscribed users to access PPTV content stored in a *Crowd-Cache*. We are in the process of conducting experimental evaluation of *Crowd-Cache* system in scale under real-world public transport networks in Thailand.

In this paper, we focused on the analysis of a single *Crowd-Cache* networks. However in practice, it might happen that multiple *Crowd-Caches* co-exist, and number of ways exist to take advantage of that. For instance, it is possible to synchronize the bus drivers' *Crowd-Caches* at least once a day at the bus depot. We aim to further investigate the effect of multiple *Crowd-Caches* in our future work. We also intend to evaluate the efficiency and the practical feasibility of the proposed advertisement distribution system as an incentive scheme integrating with the developed Android app.

8. Conclusion

Mobile video traffic has been driving an explosive growth in the mobile data traffic, with users of smartphone devices struggling to limit their usage to monthly capped data plans. We proposed a novel crowd-sourced mobile system - *Crowd-Cache*, that enables users to consume popular content for free in areas such as public transport through mobile networking in proximity. The *Crowd-Cache* provider selects a set of users to deploy crowd-sourced *Crowd-Cache* devices (*CC-server*) in public places and create a cyber-physical space where users can exploit content. The smartphone users can access the *CC-server*'s content storage via the (*CC-app*) mobile app. An advertisement based incentive scheme has been developed for users to become *CC-server* users. Using a real-world dataset and probabilistic modeling, we showed that more than 50% cache hit rate can be achieved during bus ride in peak hours regardless of the content popularity distribution. Moreover, with a 128GB of *Crowd-Cache* storage, more than 80% of the users reduce their cellular network usage at least by 40%. Results could be generalized to different scenarios by adjusting parameters in our proposed system model. Finally, we demonstrated the feasibility of the system by developing an Android application. Throughput, latency and device energy consumption of the *CC-app* was evaluated using the measurements from real devices. Results show that *CC-app* users lowers the device energy consumption compared to accessing the content through cellular networks, if the proposed system provides at least 10% of cache hit rate.

References

[1] Cisco, Visual networking index: Global mobile data traffic forecast update, 2015–2020 white paper.

[2] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, H. Yan, Modeling web quality-of-experience on cellular networks, MobiCom'14, 2014.

[3] J.A. Racoma, Attention smartphone users: your internet connection is just about to crawl to a halt, 2013. URL http://http://www.androidauthority.com/smartphone-data-use-outpacing-capacity-149489/.

[4] S. Ioannidis, L. Massoulie, A. Chaintreau, Distributed caching over heterogeneous mobile networks, SIGMETRICS Perform. Eval. Rev. 38 (1) (2010) 311–322.

[5] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, M.D. De Amorim, Push-and-track: Saving infrastructure bandwidth through opportunistic forwarding, Perv. Mob. Comput. 8 (2012) 682–697.

[6] A. Balasubramanian, R. Mahajan, A. Venkataramani, Augmenting mobile 3g using wifi, in: MobiSys '10, 2010, pp. 209–222.

[7] A.J. Nicholson, B.D. Noble, Breadcrumbs: forecasting mobile connectivity, in: ACM MobiCom'08, 2008, pp. 46–57. New York

[8] F. Qian, K.S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen, O. Spatscheck, Web caching on smartphones: Ideal vs. reality, in: ACM MobiSys'12, 2012, pp. 127–140.

[9] B.D. Higgins, J. Flinn, T. Giuli, B. Noble, C. Peplin, D. Watson, Informed mobile prefetching, in: ACM MobiSys'12, 2012, pp. 155–168.

[10] L. Robert, Data prefetching algorithm in mobile environments, Eur. J. Sci. Res. 28 (3) (2009) 478–491.

[11] F. Qian, Z. Wang, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, Characterizing radio resource allocation for 3g networks, in: ACM IMC'10, 2010, pp. 137–150.

[12] J. Huang, F. Qian, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, A close examination of performance and power characteristics of 4g lte networks, in: Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, ACM, 2012, pp. 225–238.

[13] A.J. Mashhadi, P. Hui, Proactive Caching for Hybrid Urban Mobile Networks, Tech. Rep, 2010, University College London, 2010.

[14] G.M. Lee, S. Rallapalli, W. Dong, Y.-C. Chen, L. Qiu, Y. Zhang, Mobile video delivery via human movement, in: IEEE SECON'13, IEEE, 2013, pp. 406–414.

[15] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, O. Spatscheck, To cache or not to cache: the 3g case, Internet Comput. IEEE 15 (2) (2011) 27–34.

[16] Z. Wang, F.X. Lin, L. Zhong, M. Chishtie, How far can client-only solutions go for mobile browser speed? in: ACM WWW'12, 2012, pp. 31–40.

[17] A. Finamore, M. Mellia, Z. Gilani, K. Papagiannaki, V. Erramilli, Y. Grunenberger, Is there a case for mobile phone content pre-staging? in: ACM CoNEXT'13, 2013, pp. 321–326.

[18] N. Gautam, H. Petander, J. Noel, A comparison of the cost and energy efficiency of prefetching and streaming of mobile video, in: Proc. of the 5th Workshop on Mobile Video, ACM, 2013, pp. 7–12.

[19] U. Rathnayake, M. Ott, A. Seneviratne, Network availability prediction with hidden context, Perform. Eval. 68 (9) (2011) 916–926. http://dx.doi.org/10.1016/j.peva.2011.03.003.

[20] J.B. Gomes, C. Phua, S. Krishnaswamy, Where will you go? mobile data mining for next place prediction, in: Data Warehousing and Knowledge Discovery, Springer, 2013, pp. 146–158.

[21] M. Taghizadeh, K. Micinski, S. Biswas, Distributed cooperative caching in social wireless networks, IEEE Trans. Mob. Comput. (2013), doi:10.1109/TMC.2012.66.

[22] B. Han, P. Hui, V. Kumar, M. Marathe, J. Shao, A. Srinivasan, Mobile data offloading through opportunistic communications and social participation, Mob. Comput. IEEE Trans. (99) (2011). 1–1

[23] M. Barbera, J. Stefa, A. Viana, M. de Amorim, M. Boc, Vip delegation: enabling vips to offload data in wireless social mobile networks, in: DCOSS'11, IEEE, 2011, pp. 1–8.

[24] F. Jiang, K. Thilakarathna, S. Mrabet, M.A. Kaafar, udrop: Pushing drop-box to the edge of mobile network, in: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE, 2016, pp. 1–3.

[25] J. LeBrun, C.-N. Chuah, Bluetooth content distribution stations on public transit, in: Proceedings of the 1st international workshop on Decentralized resource sharing in mobile computing and networking, ACM, 2006, pp. 63–65.

[26] F.P. Tso, L. Cui, L. Zhang, W. Jia, D. Yao, J. Teng, D. Xuan, Dragonnet: a robust mobile internet service system for long-distance trains, IEEE Trans. Mob. Comput. 12 (11) (2013) 2206–2218.

[27] A.G. Tasiopoulos, I. Psaras, V. Sourlas, G. Pavlou, Tube streaming: modelling collaborative media streaming in urban railway networks, IFIP Networking, 2016.

[28] A. Brodersen, S. Scellato, M. Wattenhofer, Youtube around the world: geographic popularity of videos, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 241–250.

[29] F. Jiang, K. Thilakarathna, M.A. Kaafar, F. Rosenbaum, A. Seneviratne, A spatio-temporal analysis of mobile internet traffic in public transportation systems: A view of web browsing from the bus, in: Proceedings of the 10th ACM MobiCom Workshop on Challenged Networks, in: CHANTS '15, ACM, New York, NY, USA, 2015, pp. 37–42.

[30] H. Petander, Energy-aware network selection using traffic estimation, in: Proc. of the 1st ACM MICNET '09, 2009, pp. 55–60.

[31] Z. Li, J. Lin, M.-I. Akodjenou, G. Xie, M.A. Kaafar, Y. Jin, G. Peng, Watching videos from everywhere: a study of the pptv mobile vod system, in: ACM IMC'12, 2012, pp. 185–198.

[32] F. Jiang, Z. Liu, K. Thilakarathna, Z. Li, Y. Ji, A. Seneviratne, TransFetch: a viewing behavior driven video distribution framework in public transport, in: 41st Annual IEEE Conference on Local Computer Networks (LCN 2016), 2016. Dubai, United Arab Emirates (UAE)

[33] A. Abhari, M. Soraya, Workload generation for youtube, Multimed. Tools Appl. 46 (1) (2010) 91–118.

[34] X. Cheng, C. Dale, J. Liu, Statistics and social network of youtube videos, in: Proc. of 16th International Workshop on Quality of Service, 2008, pp. 229–238. Enschede

[35] M. Berry, M. Hamilton, Changing urban spaces: Mobile phones on trains, Mobilities 5 (1) (2010) 111–129.

[36] D.A. Hensher, Measurement of the valuation of travel time savings, J. Transp. Econ. Pol. (JTEP) 35 (1) (2001) 71–98.

[37] S. Jin, A. Bestavros, Popularity-aware greedy dual-size web proxy caching algorithms, in: Distributed Computing Systems, 2000. Proceedings. 20th International Conference on, IEEE, 2000, pp. 254–261.

[38] K. Thilakarathna, F. Jiang, S. Mrabet, M.A. Kaafar, A. Seneviratne, P. Mohapatra, Demo: Crowd-cache – popular content for free, in: Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services, in: MobiSys '14, ACM, New York, NY, USA, 2014, pp. 358–359.

Kanchana Thilakarathna is a Researcher at the Mobile Systems Research Group at Data61. He received his Ph.D. in Electrical Engineering and Telecommunications from the University of New South Wales and a B.Sc. degree with First Class Honours specializing in Electronics and Telecommunications from the University of Moratuwa, Sri Lanka. He has more than three years of industry experience as a Mobile Radio Network Optimization Engineer, before pursuing postgraduate studies. Kanchana also holds Conjoint Lecturer appointment at the school of Computer Science and Engineering at UNSW. His current research interests are in developing novel mechanisms for efficiently delivering mobile data while preserving user privacy, and security when using online services.

Fang-Zhou Jiang is a Ph.D student in school of Electrical Engineering and Telecommunication from University of New South Wale, and an enhanced Ph.D student at Mobile System Research Group at Data61. He received the B.E. with First Class Honours in Telecommunications from the University of Sydney, Australia, in 2014. In 2013, he joined the Network Research Group in NICTA as a summer scholarship student, and later became a research intern. His current research interests include Mobile Content Distribution, Mobile Computing and Mobile System QoS/QoE optimization.

Sirine Mrabet is a Research Engineer at the Mobile Systems Research Group at Data61. She joined Data61 in February 2014 and is responsible for the design and the development of mobile networking products. Prior to joining Data61, Mrs Mrabet worked as a project manager for three years at KLS-Logistic Systems, a company dedicated to warehouse management, transport management and optimization. She was responsible of the design, development and delivery of final products to customers including Tag Heuer Swiss (Switzerland), ENTREMONT (France) and CHUs of Lyon, and Saint Etiennne.
Before that Sirine was a Research Engineer at the Research Unit in Networking (RUN) at the University of Liege, Belgium in 2007, and spend 2 years as a Software engineer at INRIA Sophia Antipolis in 2005. She worked in close collaboration with teams of researchers and engineers. She was responsible for the design and development of new software platforms. She was the main developer of OSA (Open Simulation Architecture). Mrs Mrabet obtained an Engineering degree and an M.S in Computer Science at the National School of Computer Sciences of Tunis Tunisia in 2004.

Dali Kaafar is a research leader at the Mobile Systems Research Group at Data61. He leads the research and development activities in Network measurement, modelling and performance evaluation as well as security, privacy and CyberCrime prevention with a focus on data-driven approaches. He holds the position of visiting professor of the Chinese Academy of Science (CAS). He was previously a researcher at the Privatics team at INRIA in France, and at the university of Liege. He is the main investigator and responsible of several European and Asia-Pacific research projects.
Dr. Kaafar obtained an Engineering degree, an M.S from University of Manouba and a Ph.D. in Computer Science from University of Nice Sophia Antipolis at INRIA. He published over 150 scientific peer-reviewed papers, three US patents and is the author of several publications in renowned conferences including ACM SIGCOMM and IEEE INFOCOM. Dali is also a member of the editorial board of the journal on Privacy Enhancing Technologies since 2013.

Aruna Seneviratne is the foundation Chair in Telecommunications and holds the Mahanakorn Chair of Telecommunications, and the leader of the Networks Research Group at NICTA. His current research interests are in mobile content distributions and preservation of privacy. He received his PhD in electrical engineering from the University of Bath, UK. He has held academic appointments at the University of Bradford, UK, Curtin University, and UTS.

Gaogang Xie received his B.S., M.S., and Ph.D. all from Hunan University and then joined Institute of Computing Technology, Chinese Academy of Sciences (ICT CAS) as assistant professor in 2002. Now he is a full Professor and the Director of Network Technology Research Center at ICT CAS. His research interests include network architecture, efficient data plane and network measurement and analysis. He has published over 100 scientific papers in refereed journals and conferences and held 21 patents granted. His research has been applied by many well-known companies in their products.