

TransFetch: A Viewing Behavior Driven Video Distribution Framework in Public Transport

Fangzhou Jiang*, Zhi Liu†, Kanchana Thilakarathna*, Zhenyu Li‡, Yusheng Ji§, Aruna Seneviratne*

*Data61, CSIRO& University of New South Wales, Australia, {firstname.lastname}@data61.csiro.au

†Waseda University, Japan, liuzhi@aoni.waseda.jp

§National Institute of Informatics, Japan, kei@nii.ac.jp

‡ICT, CAS, China, zyli@ict.ac.cn

Abstract—Mobile video traffic is exploding and it is particularly challenging to stream video when high density of users are “on the move”, e.g., in public transport systems. It becomes increasingly problematic as video traffic is predicted to account for more than 80% of Internet traffic by 2019. This will be exacerbated by factors such as cellular network coverage issues and unstable network throughput due to high speed mobility. By exploiting the predictable public transport mobility patterns, spatio-temporal correlation of user interests and users’ video viewing behaviors, we proposed *TransFetch* which uses intelligent caching on-board the public transport vehicles as well as a novel video chunk placement algorithm. We show through extensive simulations, that *TransFetch* reduces the system cellular data usage by up to 45% and improves the quality of video streaming by up to 35%. Finally, we demonstrate the practical feasibility of *TransFetch* by implementing caching units on a *Raspberry-Pi* and a mobile app on an Android device.

I. INTRODUCTION

Mobile data traffic has surpassed the total traffic consumed by desktops in 2012, and mobile video traffic in particular accounts for more than half of this mobile data traffic [1]. Smartphone is the primary device for video streaming for many users due to numerous benefits offered by mobile devices such as timely access and convenience. This is evidenced by a recent study which showed that 83% of commuters use their smartphones while on-board trains and buses in US, with a majority streaming videos [2].

However, cellular data is expensive, especially when users consume it outside of their mobile plan. User quality of experience (*QoE*), is also low as stalling and blurriness are often common in the case of mobile video streaming [3], [4]. It is even more challenging to distribute video in public transport as Internet access in public transport systems are provided via backhaul WiFi access through cellular dongles [5]. The high mobility of vehicles, unreliable cellular links, large numbers of users, often exacerbated the situation and lead to limited user *QoE* and network usability.

In public transport system, it has been shown that commuters in the same vehicle or on similar routes are more likely to be interested in similar contents or information known as the spatio-temporal correlation of user interest [6]. The observed correlation provides strong motivation to push the content caches across the last hop of wireless network to the vicinity of commuters as caching performance is shown to be greatly improved. Furthermore, trains and buses often operate in pre-determined routes and schedules. These predictable mobility

patterns offer great opportunity for scheduled offloading of video streams to alternative networks at the bus/train stations to reduce cost of cellular data of the vehicle and provide better *QoE*, although the stopping time and amount of data that can be transferred in each stop is usually limited.

In this paper, we propose a novel intelligent content caching and pre-fetching method, referred to as *TransFetch*, that exploits the fixed schedules of public transport systems, the spatio-temporal correlation of commuter interests as well as commuters’ video viewing behaviors. In *TransFetch*, video files are broken into “chunks”. All video chunks consumed by commuters on a vehicle, will be cached locally at the *On-board Unit (OBU)* caches inside the vehicles. Commuters with similar interests would be able to access the contents locally, without using cellular data. Moreover, this will minimize the access latency. In addition, a chunk placement algorithm is used to fetch the required video chunks from *Roadside Units (RSUs)* located in bus stops and train stations to *OBU* caches. The chunk placement algorithm uses the knowledge of commuter’s current video, commuter’s video consumption behavior, usage statistics of the current video and vehicle route information to determine the chunks to download at each stop. Although, the individual components of the *TransFetch* system has been addressed by numerous researchers, to the best of our knowledge, they have not been combined as proposed by the *TransFetch* system. Moreover, *TransFetch* system further optimizes transport and delivery of multimedia to public transport commuters by exploiting users’ viewing behavior. This paper makes the following contributions;

- We analyze a trace of nearly 570k sessions from mobile devices accessing *VoD* service in December 2014 to understand users’ mobile video viewing behaviors and show that these unique behavioral patterns could be exploited to further improve mobile multimedia delivery.
- We propose a novel chunk placement algorithm leveraging deterministic nature of the public transport mobility patterns and user information, and show that it is possible to minimize cellular data usage by approximately 45%, while improving streaming quality by up to 35%.
- We demonstrate the feasibility of *TransFetch* system by implementing *OBU* and *RSU* using *Raspberry-Pis* as well as designing an end-user application in *Android*.

The rest of the paper is organized as follows; We explore

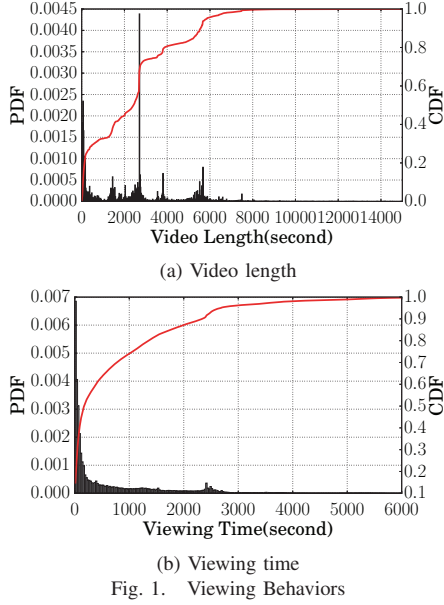


Fig. 1. Viewing Behaviors

mobile user viewing behavior in Section II. Section III describes the architecture and system model of *TransFetch*. We evaluate the system performance via simulation in Section V followed by describing our prototype implementation in Section VI. Section VII summarizes the related work. Finally, Section VIII provides the conclusion.

II. EXPLORING MOBILE VIDEO VIEWING BEHAVIORS

We first analyze characteristics of mobile video content and extract video viewing behavior of mobile users. Then, we model the characteristics and viewing behaviors that can be exploited for efficient video distribution in public transport.

A. Dataset Description

We use real-life mobile video dataset to analyze user behaviors and model video consumption patterns. In specific, we use mobile video access logs extracted from one of the largest online video on-demand service provider in the world¹. The logs were collected in December 2014, and contain 569,555 video requests generated by more than 420,000 mobile users. In this dataset, users request 70TB of mobile video content with a mean bit rate of 377Kbps. The dataset also contains implicit user activity, e.g., portion of the video watched by the user.

B. Video Viewing Behavior of Mobile Users

Mobile users often consume only small portions of requested video. Fig. 1 shows that the actual viewing time is densely concentrated for short periods of time, although video lengths are sparsely distributed from a few seconds up to several hours. In particular, the average viewing time remains under 10 minutes. View ratio is calculated by dividing view time of a particular video session over the video length.

Not only users' view ratio is affected by factors such as the video type, network condition and the length of the video, but it

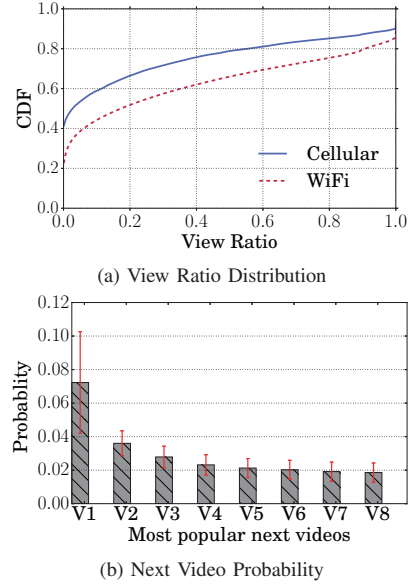


Fig. 2. Viewing Behaviors and Statistics

also depends on the access network used, i.e. WiFi or cellular. Fig. 2a illustrates the distribution of view ratio when the user is connected to the two different networks. In general, view ratio is lower in cellular network. This is mainly due to users being concerned about cellular data usage while streaming videos using their personal mobile data plans. The abandonment ratio (0% view ratio) is twice as much as in cellular networks (40%) than using WiFi (20%) due to longer startup time. Therefore, we believe users of *TransFetch* system would engage and behave more like using WiFi due to improved *QoE* and free access.

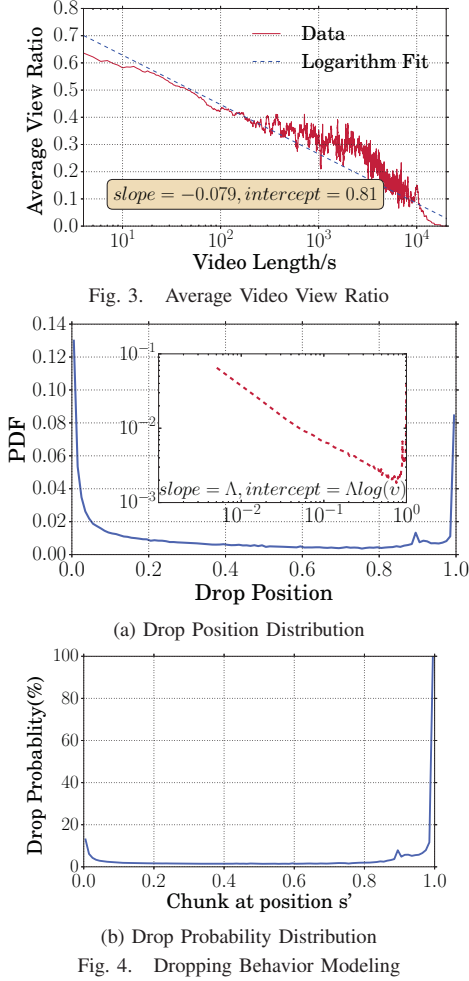
Fig. 2b illustrates the probability of requesting the same video out of top 100 popular videos knowing the video that the user is currently streaming. The results show that approximately 7% of users would watch the same top video knowing the video they are currently streaming. As a result, simply pre-fetching the top 5 candidates would result in around 20% accuracy on average when there is no bandwidth constraints. In Section IV, we propose a chunk placement algorithm taking into account of these observed user viewing behaviors.

C. Modeling Mobile Video Viewing Behaviors

In [7], authors model users' view ratio as linear functions of video length for different video types. The type-duration model essentially categorizes video into ranges of particular lengths, while our finding extends their model without losing generality. Fig. 3 displays the relationship of average view ratio to the length of video in seconds. As the mean view ratio drops linearly with regards to the video length in logarithm scale, we model the mean view ratio as a logarithm function of video length L_n : $\overline{\text{VR}}_n \sim \log(L_n)(\zeta, \eta)$, where $\overline{\text{VR}}_n$ stands for the mean view ratio of video n . The value of ζ and η are fitted using the dataset as slope and intercept respectively. Thus, the mean view ratio of video n is modeled as: $\overline{\text{VR}}_n = \zeta \log(L_n) + \eta$ ($\zeta = -0.079, \eta = 0.81$).

Clients request chunks of videos from server, and each chunk

¹<http://download.pptv.com/en/about.html>



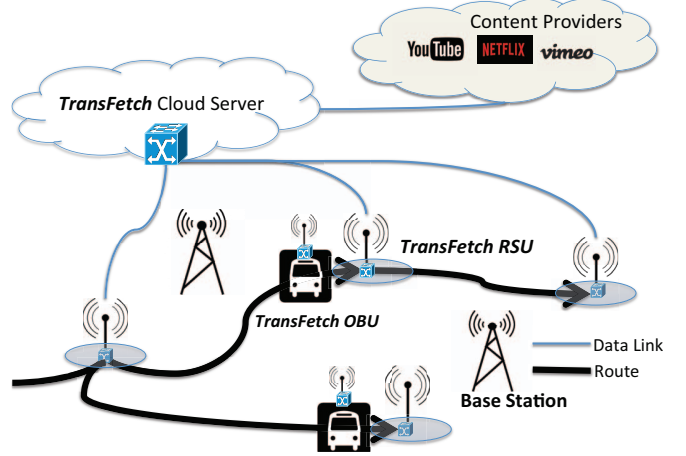
contains a fixed duration usually between 4s to 10s. Chunk position is used to represent the closest chunk. We further model view ratio distribution with regards to chunk position s . Fig. 4a shows the PDF distribution of drop positions DP. In log-log scale, $\log(\text{DP})$ against $\log(s)$ could be modelled by two linear equations with slope Λ and intercept $\Lambda \log(v)$ splitting at chunk position $s = 0.8$; $\log(\text{DP}) = \Lambda \log(s) + \Lambda \log(v)$. Therefore, DP could be modeled as two exponential functions with coefficients v and Λ ($v_1 = 0.0025, \Lambda_1 = -0.74, v_2 = 0.2077, \Lambda_2 = 179.86$);

$$\text{DP}(s) = \begin{cases} v_1 s^{\Lambda_1} & , s \leq 0.8 \\ v_2 s^{\Lambda_2} & , 0.8 < s \leq 1 \end{cases} \quad (1)$$

We derive the instantaneous drop probability of chunk at position s' , $D(s')$ via drop position distribution DP. Define event A: user drop at chunk of position s' , and event B: user has watched chunk at position s' . The probability of dropping the session at the end of chunk s' can be written as; $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)}{P(B)}$. $P(A \cap B) = P(A)$ because all users drop at chunk s' must have watched chunk s' .

$$D(s') = P(A|B) = \frac{\text{DP}(s')}{1 - \int_{s=0}^{s=s'} \text{DP} ds} = \frac{\text{DP}(s')}{\int_{s=s'}^{s=1} \text{DP} ds} \quad (2)$$

$\text{DP}(s')$ is the probability of a user drop at chunk position s' ,



and $P(B) = \int_{s=s'}^{s=1} \text{DP} ds$. The drop probability distribution is shown in Fig. 4b. Unlike the skew-normal PDF distribution of view ratio model in previous studies [7], user drop probability is high ($> 2\%$) for the first 10% of the video and the last 20% of video. Moreover, it is unlikely for users to drop a session once finishing watching the initial set of chunks of the video, while users are highly likely to drop a session approaching the end of video. This could be a results of user loosing the interest to continue watching the video as the user has already captured the main message of the video. This observed user drop behaviors and model is exploited in *TransFetch* system to build more effective mobile video distribution framework.

III. TRANSFETCH ARCHITECTURE

TransFetch is designed to reduce system cellular data usage while providing optimal user *QoE* within the constraints. Fig. 5 illustrates the system architecture of *TransFetch* system. We assume commuters on-board stream video using *TransFetch* instead of personal cellular connection because of cost incentives. We consider vehicles with deterministic routes and timetables are equipped with *TransFetch OBU*. One vehicle running through one whole route is regarded as a trip. *TransFetch OBU* provides Internet connections on-board, and is capable of connecting with *TransFetch RSUs* and cellular network. Furthermore, *TransFetch RSUs* are connected to *TransFetch* cloud servers which also have high capacity back-haul links to different content providers. Finally, we assume *TransFetch RSUs* are deployed in each transport stop, which is implemented similarly to *TransFetch OBU*.

Commuters stream videos using *TransFetch OBU*, and the actual video chunks are either stored in the *OBUs* cache or being streamed with the shared cellular link. Since *TransFetch RSUs* are normally connected to high speed wired networks, the bottleneck of the system lies in the limited amount of exchange time and bandwidth allowing vehicles to exchange data with *RSUs*. Hence, selecting and fetching video chunks that are most likely to be consumed is important for reducing cellular data usage and maintaining high stream quality. In *TransFetch*, a novel chunk placement algorithm is proposed to maximize the

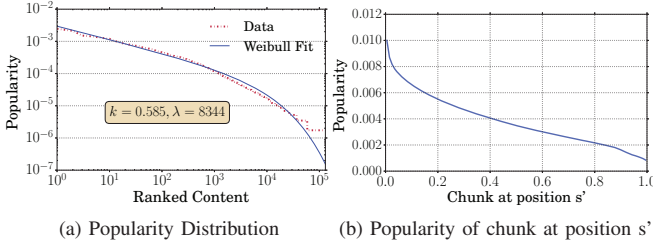


Fig. 6. Chunk Popularity Modeling

capability of *RSUs*. Chunks of video being streamed are marked as pre-staging chunks, and unmarked chunks are treated as pre-fetching chunks. We compute and rank pre-staging and pre-fetching chunks separately. The available bandwidth at each stop is used to fetch the highest ranked chunks with best effort, where pre-staging chunks are of higher priority. Cellular bandwidth is augmented to achieve optimal streaming quality that is reflected in the amount of stalling and freezing during streaming. *TransFetch* replaces chunks in *OBU* cache that are not popular with LRU in case *OBU* cache is full. Performances of cache replacement policies has been extensively studied, including in our prior work [8], and this paper will instead focus on chunk placement policy. Moreover, it is also shown that commuters in public transport at the same time are more likely to be interested in similar contents [6], thus videos cached and pre-fetched in *TransFetch OBU* are more likely to be requested again. Our system architecture design allow us to use minimal amount of cellular data to achieve optimal streaming quality. We model cellular bandwidth usage of *TransFetch* system and user streaming quality by considering expected missing chunk ratio $E(MCR)$. In addition, *OBU* hit rate is used to evaluate our pre-fetching performance.

A. System Model

1) *System Access Model*: We consider commuters start streaming video once on-board. At each stop, the number of boarding commuters varies from 0 to K_{cap} . For simplicity, we also assume that all passengers on-board use *TransFetch-app* to stream videos under natural leave pattern. In natural leave pattern, a user watches video continuously during a streaming session without interacting with the video. User might drop the session anytime and continue a new video. In addition, users are estimated to associate with the system for an average 35min, which is the average commuting time suggested in transport study [9]. The association time is, therefore, modeled as normal distribution: $\Delta t \sim \mathcal{N}(\mu, \sigma^2)$ ($\mu = 2100, \sigma = 600$).

2) *Video Popularity Distribution*: Recall the independent reference model (IRM), in which the popularity of n_{th} most popular video q_n follows Weibull probability function such that $q_n = \frac{\kappa}{\lambda} \left(\frac{n}{\lambda}\right)^{(\kappa-1)} e^{-(n/\lambda)^\kappa}$, where $n > 0$ and $\kappa, \lambda > 0$. Here q_n represents the popularity of the n_{th} most popular video, and κ, λ are shape and scale parameters of distribution. Similar to previous studies [10], video popularity distribution displays a heavy tail. Weibull distribution is fitted to our dataset in Fig. 6a ($\kappa = 0.585, \lambda = 8344$). There are also studies trying to take temporal-locality of content popularity into consideration, i.e. shot-noise model (SNM) [11]. However, the problem for

SNM lies in the difficulty of accurate classification and fitting. Since we evaluate our system performance over a short period of time, the dynamic popularity change is out of our scope. The IRM model still holds a balance between accuracy and simplicity.

3) *Popularity of Video Chunks*: The mean popularity of chunk at position s' , $Q(s')$, can also be written as the probability of consuming that chunk;

$$Q(s') = \frac{\int_{s=0}^{s=1} DPds}{\int_{s=0}^{s=1} DPds} = \frac{\int_{s=s'}^{s=1} DPds}{1}$$

$$= \begin{cases} \left[\frac{v_1}{\Lambda_1+1} s^{\Lambda_1+1} \right]_{s'}^{0.8} + \left[\frac{v_2}{\Lambda_2+1} s^{\Lambda_2+1} \right]_{0.8}^{1}, & s' \leq 0.8 \\ \left[\frac{v_2}{\Lambda_2+1} s^{\Lambda_2+1} \right]_{s'}^{1}, & 0.8 < s' \leq 1 \end{cases}$$

$$= \begin{cases} \frac{v_1}{\Lambda_1+1} 0.8^{\Lambda_1+1} - \frac{v_1}{\Lambda_1+1} s'^{\Lambda_1+1} + \frac{v_2}{\Lambda_2+1} - \frac{v_2}{\Lambda_2+1} 0.8^{\Lambda_2+1}, & s' \leq 0.8 \\ \frac{v_2}{\Lambda_2+1} - \frac{v_2}{\Lambda_2+1} s'^{\Lambda_2+1}, & 0.8 < s' \leq 1 \end{cases} \quad (3)$$

Hence, we plot $Q(s')$ distribution of our dataset in Fig. 6b. The popularity of chunk at position s' in video n written as $q_{n,s}$ can be represented by combining Weibull distribution of video popularity q_n and the chunk popularity distribution $Q(s')$; $q_{n,s} = q_n Q(s')$.

4) *Cellular Bandwidth Constraints*: The cellular data usage for downloading video n from i^{th} *RSU* to the $(i+1)^{th}$ *RSU* in a trip is denoted as $B_{n,i}$. Cellular data usage is limited by the cellular bandwidth, which is affected by a lot of factors, i.e. cellular coverage, mobility and network condition. The system cellular bandwidth constraint from i^{th} stop to $(i+1)^{th}$ stop can be written as; $\sum_{n \in streams} (B_{n,i}) \leq \bar{\omega}_i \cdot T_{ist,i}$, where $T_{ist,i}$ and $\bar{\omega}_i$ are inter-stop time and mean cellular bandwidth between the two stops. Inter-stop time is fitted using collected public transport dataset, $T_{ist,i} \sim \mathcal{N}'(\mu', \sigma'^2)$, ($\mu' = 82.3, \sigma' = 104.39$).

5) *Data Exchange with RSU*: Vehicle stop time, T_{stop} follows Gaussian distribution $\mathcal{N}''$ [12], where the mean bus stop time μ''_i is modeled as a function of total number of passengers aboard vehicle at stop i , $N_{on,i}$; $T_{stop,i} \sim \mathcal{N}''(\mu''_i, \sigma''^2)$, $\mu''_i = \chi \cdot N_{on,i}$, where χ is a stopping time coefficient. Amount of data exchange at i^{th} stop is denoted as BL_i . The second system constraint could be expressed as, $BL_i \leq \bar{\omega}'_i \cdot T_{contact,i} + \omega'_i|_{(d=0)} \cdot T_{stop,i}$. $T_{contact,i}$ is the amount of time vehicle in the range of i^{th} *RSU* apart from stopping time, which is dependent on the local WiFi range and vehicle speed. $\bar{\omega}'_i$ and $\omega'_i|_{(d=0)}$ stand for the mean local bandwidth and the bandwidth when vehicle is stopping at i^{th} stop. Our goal is to use BL as much and as effective as possible to satisfy on-going streaming users and future video requests.

B. System Objectives

In our bandwidth constraint system with multiple streams, not all video chunks would be downloaded in time before the deadline. We define missing chunk ratio as the total size of chunks that could not be downloaded before the deadline over the total size of chunks that is required to be download. Missing chunk ratio for stream session n from i^{th} to next *RSU* could be expressed as:

$$\text{MCR}_{n,i} \leq 1 - \left(\frac{\bar{\omega}_{n,i} T_{ist,i}}{\theta_{n,i}} + \frac{C_{n,i}}{\theta_{n,i}} \right) = \frac{\theta_{n,i} - \bar{\omega}_i T_{ist,i} - C_{n,i}}{\theta_{n,i}} \quad (4)$$

$\theta_{n,i}$ denotes the total size of chunks in stream n that is required to be downloaded from i^{th} RSU to the next one. $C_{n,i}$ stands for the part of $\theta_{n,i}$ that can be contributed from OBU cache. Any previous user consumed video chunk or placed video chunk would contribute to $C_{n,i}$. Note that the computed $\text{MCR}_{n,i}$ here is a lower bound comparing to real-world scenario, because we only treat a chunk missing deadline before arriving next RSU. In reality, the network dynamic is much higher, which could result in a larger measured MCR. The overall system expected MCR is;

$$E(\text{MCR}) = \frac{\sum_{u=1}^U \sum_{i=1}^{I_u} \sum_{n=1}^{N_i} \text{MCR}_{n,i}}{U \cdot I_u \cdot N_i}$$

U is the number of trips, I_u stands for the number of stops in each trip and N_i represents the number of streams from stop i to the next stop. We examine the optimal chunks of video to be placed to *TransFetch* OBU to satisfy the following two factors: (i) Minimal expected missing chunk ratio, $\text{Min}(E(\text{MCR}))$, (ii) Minimal cellular data usage. Our system objective is to minimize $E(\text{MCR})$, hence achieve the minimum cost of cellular data via more efficient usage of RSU and caching.

IV. TRANSFETCH CHUNK PLACEMENT ALGORITHM

In this section, we describe our proposed chunk placement algorithm to maximize the usability of chunks downloaded from RSU. Upon contacting the i^{th} RSU, future chunks in current live streams $v_{n,s'}$ would be marked as pre-staging chunks to distinguish with predicting user's future interests. Then all pre-staging chunks ($v_{n,s'}$) and pre-fetching chunks ($v_{f,s'}$) are ranked according to the computed score Γ respectively, where the priority is given to chunks higher in the list. Chunks that are not in OBU cache will be transmitted within the local bandwidth constraint BL_i . The detailed scheme is described in Algorithm 1. We will explain tagging and chunk ranking in following subsections.

A. Tagging Pre-staging Chunks

Assume user k is consuming chunk at position s of video n at the time of contact to i^{th} RSU. Chunks between position s and position $\min(s + \frac{T_{ist,i}}{L_n}, 1)$ would be marked as pre-staging chunks, meaning that we pre-stage chunks that are estimated to be streamed up until arriving at the next RSU.

B. Ranking Pre-staging and Pre-fetching Chunks

Pre-staging could be regarded as a special case of pre-fetching, where user interest is pre-determined. Pre-staging chunks are generally of higher importance than pre-fetching for the reason that the most popular chunk is in the magnitude of 10^{-5} . In other word, predicting user interest is much harder than maintaining a current stream because the drop probability is relatively low at each chunk. Therefore, we rank pre-staging and pre-fetching chunks in two lists separately. The probability of watching chunk position s' of video n , $v_{n,s'}$, could be calculated with knowledge of the drop probability distribution of user k 's current viewing chunk position s , and could be expressed as; $P(v_{n,s'}) = \frac{\int_{x=s}^{x=1} \text{DP} dx}{\int_{x=s}^{x=1} \text{DP} dx} = \frac{Q(s')}{Q(s)}$. We use the

Algorithm 1 TransFetch Chunk Placement Algorithm

Input: route info, user stream pattern, history, current streams

Output: chunks to place in OBU at i^{th} RSU

initialization: $s_{n,s'}, s_{f,s'}, \text{sizeUsed} \leftarrow 0$, new List L_1, L_2

forall the possible chunks $v_{n,s'}$ to exchange i^{th} RSU **do**

if n being streamed **and** $s \leq s' \leq \min(s + \frac{T_{ist,i}}{L_n}, 1)$ **then**

$\Gamma_{n,s'} \leftarrow \frac{Q(s')}{Q(s)}$
 $L_1.\text{add}(s_{n,s'})$

else

$\Gamma_{f,s'} \leftarrow \text{Eq. (6)}$
 $L_2.\text{add}(s_{f,s'})$

end

end

$L_1.\text{sort}(), L_2.\text{sort}(), L_3 = L_1 + L_2$

while $\text{sizeUsed} \leq BL_i$ **do**

for chunk v from head of List L_3 **do**

if not in *TransFetch* OBU **then**

 Place chunk v in OBU cache

$L_3.\text{remove}(v)$

$\text{sizeUsed} += \theta_{n,s'}$

end

end

end

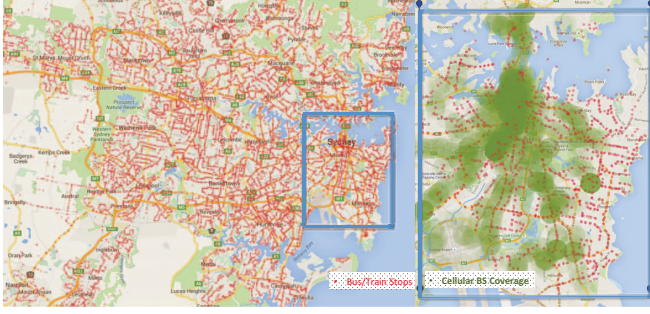
overall drop position distribution instead of the drop position of individual videos, because some unpopular videos with low view counts would give inaccurate drop prediction. The score for pre-staging chunk, $\Gamma_{n,s'}$ is computed simply as follows;

$$\Gamma_{n,s'} = P(v_{n,s'}) = \frac{Q(s')}{Q(s)} \quad (5)$$

We then take users' engagement into consideration while designing our pre-fetching ranking algorithm. In specific, apart from videos being streamed and chunks expected to be consumed, we predict users' interest leveraging viewing history, in particular video popularity and similar users' historical view ratios. The score of each pre-fetching chunk is composed of two parts;

$$\Gamma_{f,s'} = \begin{cases} \Omega_1 \cdot \frac{\sum_{k \in \text{users}} H(k) \mathbb{V}_{f,k}}{\sum_{k \in \text{users}} H(k)} + \Omega_2 \cdot q_{f,s'} & \sum_{k \in \text{users}} H(k) \neq 0 \\ \Omega_2 \cdot q_{f,s'} & \text{otherwise} \end{cases} \quad (6)$$

$q_{f,s'}$ stands for popularity rank for chunk position s' of video f . $H(k) = 1$ stands for video f has been watched by user k , and 0 otherwise. Ω_1, Ω_2 are a weighting parameter. When $\Omega_1 = 0$, we only use the popularity value. In addition, k belongs to a defined group of users with similar interest, e.g., users in the same bus/train as suggested in [6]. The first part stands for the normalized view ratio, where $\mathbb{V}_{f,k}$ is the view ratio history of user k on video f . The ratio of two parameters Ω_1/Ω_2 is to be adjusted according to different scenarios in order to reflect the optimal weighting between user engagement and chunk popularity. The pre-fetching ranking computation could be done in *TransFetch* cloud server once and update overtime. We later compare a group of randomly selected users with



(a) Sydney (b) Simulation Grid
Fig. 7. Public Transport Network

users of similar interest to show the performance impact of spatio-temporal correlation of user interest that is reflected in the algorithm.

C. Optimal Pre-fetching Coefficient Weighting

The *OBU* hit rate (HR) is calculated using the total number of *OBU* hits over total number of requests, while the first consumption of pre-staging chunks are not considered as hits. Additionally, the higher *OBU* hit rate also directly translate to a lower MCR under the same system bandwidth constraints. Thus *OBU* hit rate is used as a main metric to evaluate pre-fetching accuracy. We formulate a problem for assigning the optimal weighting for the two components in pre-fetching chunk ranking;

$$\text{Max } \text{HR}(\Omega_1, \Omega_2)$$

$$\text{s.t. } \sum_{n=1}^{N_i} (B_{n,i}) \leq \bar{\omega}_i \cdot T_{ist,i}, \forall i \in I_u \cdot U$$

$$BL_i \leq (\bar{\omega}_i \cdot T_{contact,i} + \omega'_i|_{(d=0)} \cdot T_{stop,i}), \forall i \in I_u \cdot U$$

$I_u \cdot U$ denotes all stops in all the trips. The optimum value of *OBU* is dependent on system input, i.e. request pattern, user interest and the popularity distribution. In evaluation, we vary the ratio of weighting and find the optimum ratio under real-life system constraints and inputs.

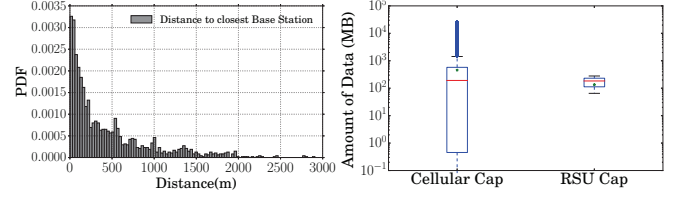
V. EVALUATION

Three real-world datasets that provide 1) mobility patterns of public transport vehicles, 2) mobile network performance and 3) video access logs are used for the evaluation of *TransFetch*. Table I summarizes the main statistics of the datasets that are succinctly described in the following.

DS1. The dataset contains operating schedule and stop, station information for train, bus and ferry services in both urban and rural areas in Sydney, Australia². There are 132,996 trips daily with over 5 million stop/station location information available in the dataset. The distribution of bus and train stops within the Sydney metropolitan area (38,388 stops) is illustrated in Fig. 7a. The inter-stop travel time is heavily concentrated under 8 minutes with mean value $T_{ist} = 82.3s$, while the inter-stop distance has a normal distribution with an average distance of 577m. We assume that the vehicle travels at a uniform speed between two stops which is determined by the scheduled departure and arrival time at the two stops.

TABLE I
SUMMARY OF STATISTICS

# Trips	132,996	# Stops	5,040,321
# RSU	38,388	D_{ist}	577m
T_{ist}	82.3s	$\bar{S}$	7.01m/s
# BS	2,394	Δt	35min [9]



(a) Distance to Closest BS (b) Bandwidth Constraints
Fig. 8. System Bandwidth Constraints

DS2. A cellular network trace was collected by traveling in Sydney metropolitan area with Android phones running a war-driving application Wiggle³. The application logs the location and signal strength of observed base-stations from one of the most popular mobile carrier in Australia. 2,394 unique base stations were located to derive a bandwidth map as shown in Fig. 7b, which is used in the simulation to describe cellular network bandwidth constraint.

DS3. The mobile video access logs dataset that is described in Section II is used as the workload generated by passengers in public transport vehicles. We use video access patterns, length of videos and view ratio of videos for this evaluation.

A. Simulation Description

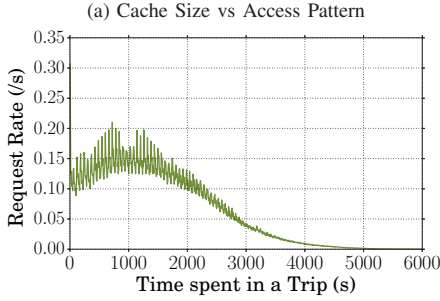
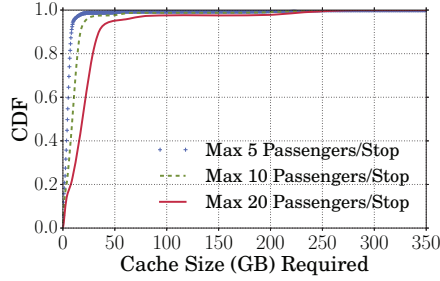
An event driven simulator based on Python was developed for evaluating the performance of *TransFetch*. We assume a *TransFetch* RSU is located at each bus/train stop in the simulation grid shown in Fig. 7b. Red points are locations of stops in transportation network (DS1) and green areas represents locations of collected cellular base stations (DS2). A vehicle with *TransFetch* *OBU* is deployed for each trip at the beginning of the day with empty *OBU* cache. We use a distance based formula to model the potential bandwidth of a location according to Shannon's theorem, $\omega(d) = B \log_2(1 + \frac{P}{N_0(\sqrt{d^2 + d_0^2})^\alpha})$. The same formula is used for both bandwidth map creation using DS2 and the bandwidth offered by the RSUs. B denotes the bandwidth used by wireless connection, and P is the transmission power. Noise is denoted as N_0 and the path loss exponent is α . For RSU bandwidth calculation, d is the distance between vehicle and stop located on road, while d_0 indicates the distance from RSU to the stop. Moreover, d is the distance between measured position to BS and d_0 is 0 for creating cellular bandwidth map. Additionally, some bandwidth related parameters used for our simulation are shown in Table II, and coefficients, i.e. Ω_1, Ω_2 are initialized to be 1:1 and later varied to investigate the optimal weighting. For instance, with the described model and parameters, 125.1MB data could be transmitted at each RSU with an average traveling speed of $\bar{S} = 7.01m/s$ and stop time of $T_{stop}=10s$. We investigate the system

²<https://tdx.transportnsw.info/>.

³<https://wigle.net>

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Cache Size C	1TB
Bandwidth B	10MHz
WiFi Transmission power P	10dBmW
Base Station Transmission power P	127.7W
Distance of RSU to road d_0	5m
Noise N_0	10^{-9} W
Path loss exponent α	5 [13]
Coefficients χ, Ω_1, Ω_2	1



(a) Cache Size vs Access Pattern
(b) Trip Request Rate
Fig. 9. System Cache Size and Request Rate

hit rate over 1 week in order to have a better understanding of the long term performance. For this simulation, we randomly sample users out of 17,384 unique users with more than three view requests from DS3 as passengers under system access model described in Section III-A1. We evaluate system performance mainly based on three metrics, cellular data usage, *OBU* hit rate and E(MCR). The simulation is repeated three times, and average value is taken in the following results.

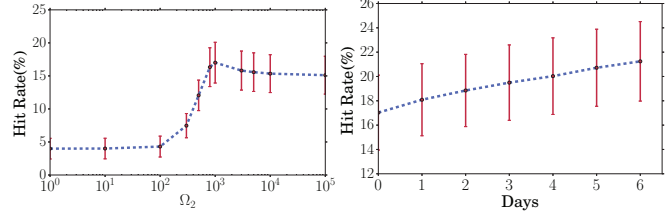
B. Dynamics of TransFetch Network and *OBU* size

The PDF distribution of the distance between stops to their nearest base station is displayed in Fig. 8a, where the majority of urban stops could be covered by base station within 500m range. The actual stop time would be dependent on the number $N_{on,i}$ and thus the *RSU* bandwidth constraint is also stop dependent. Moreover, Fig. 8b illustrates a box-plot of cellular bandwidth constraint in a stop to stop basis and *RSU* bandwidth constraint at each stop. The median cellular cap between two *RSUs* is 243.7MB, whereas the median *RSU* cap is approximately 184MB. The difference in bandwidth constraints indicates variance in cellular network is much larger than short-ranged *RSU*, causing higher unreliability.

OBU size and system access pattern would have a great

TABLE III
CELLULAR BANDWIDTH SAVING

Total Bandwidth Usage	180.09TB	100%
<i>OBU</i> Hit Saving	11.09TB	6.16%
Pre-staging Saving	45.64TB	25.34%
Cellular Data Usage	31.63TB	17.56%
Pre-fetching Potential	44.83TB	24.89%



(a) Optimum Ω_1/Ω_2 Ratio, $\Omega_1 = 1$ (b) Long Term Performance
Fig. 10. System Evaluation

impact on system performance. We first estimate the daily *OBU* size requirement under different system load K_{cap} . Fig. 9a illustrates the CDF of total cache size required for each *OBU* when K_{cap} is 5, 10 and 20 respectively. We are not showing K_{cap} for higher load due to the constraints of concurrent WiFi connections. Higher number of users could be supported by deploying one or multiple *TransFetch OBUs* in each car of the train. It shows that 80% of vehicles requires less than 30GB *OBU* size even for $K_{cap}=20$. For a more realistic scenario, we will simulate $K_{cap}=10$ in the following analysis unless otherwise specified. We believe the *OBU* size could be easily implemented to be a much larger size given the cost of SD cards and hard drives are substantially low. Therefore, we also set 1TB *OBU* size for our following evaluation. Moreover, the dynamics of *TransFetch* network is shown in Fig. 9b. The majority of the trips are under 1 hour, and on average request rate peaks approximately 0.2 req/s at around 800s under our system access pattern.

C. Performance of Chunk Placement Algorithm

Our proposed chunk placement algorithm is composed of both pre-staging and pre-fetching of chunks. We first evaluate the scenario only pre-staging is performed, and investigate the necessity of per-fetching component. In Table III, a breakdown of bandwidth usage in *TransFetch* is shown. *OBU* hit saving represents the bandwidth saving due to a local cached chunk to be requested again. On average, transport network consumes 180.09TB of video data traffic daily. *OBU* hits saves 11.09TB of cellular data usage, while pre-staging further reduces the number by additional 45.64TB accounting for approximately 25% of total video streaming traffic. Apart from *RSU* bandwidth used for pre-staging, an extra 44.83TB of data that could potentially be further utilized. Therefore, predicting user interest with the residual *RSU* bandwidth would further improve system performance.

We then evaluate the optimal pre-fetching weighting Ω_1/Ω_2 to achieve optimal system performance with regards to *OBU*

hit rate. In Fig. 10a, we vary the pre-fetching coefficient weighting for optimal pre-fetching accuracy. *OBU* hit rate (HR) peaks at $\Omega_1/\Omega_2 = 1/1000$ to approximately 17%, which is the optimal weighting to consider both user engagement and chunk popularity under our current system input and constraints. We will use the optimal weighting in the rest of the analysis.

Overall, in *TransFetch*, a minimum $E(MCR) = 0.4424$ ($\Omega_1/\Omega_2 = 1/1000$) could be achieved by consuming 31.8TB of cellular data without considering the spatio-temporal correlation of user interest. We compare *TransFetch* with the most related practical implementation cacheable WiFi access point, where the AP attempts to cache all user consumed contents. As a baseline, in cacheable AP, users would consume a total of 57.3TB cellular data with $E(MCR) = 0.685$. A comparison of *TransFetch* with cacheable WiFi and pre-staging only is shown in the bar chart in Fig. 11a. *TransFetch* system could improve streaming quality by as much as 35.43%, and reduce cellular data usage by 44.5%. Here streaming quality is measured by missing chunk ratio, and a lowered MCR leads to improvements of two most important *QoE* metrics – startup delay and rebuffering time. In practice, we believe the system performance could be much higher due to, a) the performance shown here is for 1 day, while the long term performance could be further improved, and b) the underlining spatio-temporal correlation of user interest in public transport system is not considered.

As a result, we further evaluate the system performance considering long term performance and the correlation of user interest. Firstly, same vehicles are used for each trip for a week to investigate the long term hit rate evolution. Figure 10b shows the hit rate trace over 7 days, and the average *OBU* hit rate increases to over 21% at the end of week. Note that *OBU* hit rate keeps improving and tends to saturate because dynamic popularity evolution is not considered in our model. Secondly, we investigate the impact of spatio-temporal correlation of user interest to the pre-fetching performance. We regard users who watched any of the top 50 most popular videos as cohort of users that have similar interest and compare the performance with randomly selected users in Fig. 11b. The *OBU* hit rate for users of similar interest within 24 hrs increases to approximately 35% even at the edge of mobile network, and total cellular data saving is improved by 12 times comparing to randomly select uses. The strong performance of user with similar interest in *TransFetch* indicates that in practice the spatio-temporal correlation observed would result in a much improved pre-fetching performance than simulation results we already showed for randomly selected users.

VI. PROTOTYPE IMPLEMENTATION AND FUTURE WORK

We implement *TransFetch-app* as an application for Android users and *TransFetch RSU/OBU* in a *Raspberry-Pi*. The user interface is shown in Fig. 12. *TransFetch-app* displays the most popular videos from popular content providers, i.e. YouTube, DailyMotion with provided *API* upon launching, and searching could be done across these different video providers. *TransFetch* also provides a framework for any multimedia content provider to offer better experience for users in public transport, while

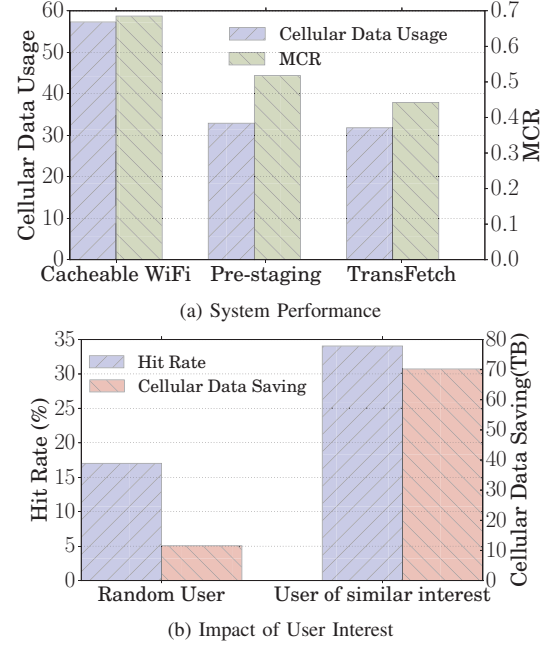


Fig. 11. System Evaluation

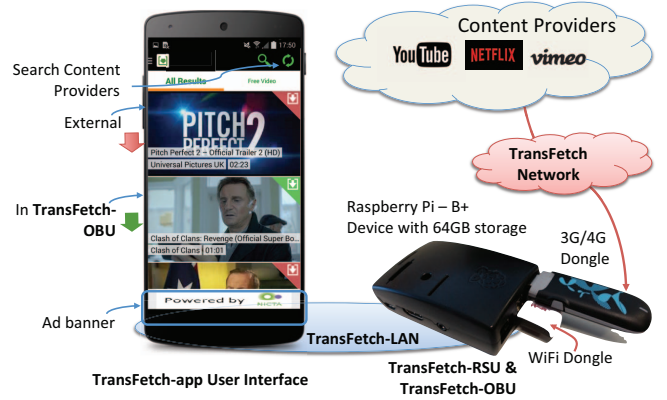


Fig. 12. User interface of TransFetch app

it could also be implemented as an extension for existing video content providers. Videos cached in *TransFetch OBU* are displayed with a green arrow indicating better *QoE*, while red arrow meaning the video needs to be streamed via cellular networks. *TransFetch* is optimized for public transport network as an extension to our previous implementation crowd-cache [14]. We are also planning to deploy *TransFetch* system in pilot projects with transport companies and on-board Internet providers, hence experimentally collect real-world user statistics to validate the system design and performance.

VII. RELATED WORK

Mobile content distribution, including the distribution of multimedia contents, has been well studied in the research community. The works that are most related to *TransFetch* lies in caching [15], [16] and pre-fetching [17], [18], [19], [20]. Qian et al. showed in [15] that 18% to 20% of total HTTP traffic volumes are redundant and could be further optimized by caching on the device. Finamore et al. [16] further

presented a general idea of pushing caching paradigm into the vicinity of mobile users in order to address the bottleneck of wireless access links. Cellular data usage could be reduced by 20% based on their proposed proactive content caching scheme. *TransFetch* shares the same idea of further taking the network caches closer to end users in cellular network, while at the same time optimizing streaming quality via proposed chunk placement schemes using road-side cache units. In [17], authors analysed the feasibility of WiFi offloading from moving vehicles. Although they show that the availability of WiFi is as low as 11%, their proposed offloading scheme *Wiffler* could reduce cellular data usage by 45%. *TransFetch* is optimized for real-time video streaming to achieve minimum missing chunk ratio, while *Wiffler* assumes a delay tolerance of 60s.

In the area of video pre-fetching, Liang et al. [20] presented an integrated pre-fetching algorithm for HTTP-based adaptive video streaming services. Their objective is to maximize byte-hit ratio, and is shown to have the best lower bound prediction performance comparing to different proposed schemes. Koch et al. [19] proposed to leverage user context and video meta information from social network to improve the potential of video pre-fetching. Similar to the goal of minimizing cellular data usage in *TransFetch*, Higgins et al. [18] proposed informed mobile prefetching that uses a cost-benefit analysis to decide the time of pre-fetching. On the contrast, *TransFetch* exploits the predictable mobility pattern in public transport, and user information to minimize cellular data usage. In addition, *TransFetch* further optimize multimedia distribution framework by taking into consideration the difference lies in different video chunks. Similar to predicting user interest, there has also been great research effect in the field of personalized video recommendation (top-N recommendation task) [21], [22], [23]. The goal of these systems is to find a few specific contents that are most appealing to the user, where click through rate (CTR) or RMSE are often metrics of system performance. Rather than helping users explore contents, *TransFetch* objectively differs from these systems as it provides lower cellular data usage and better video streaming quality.

VIII. CONCLUDING REMARKS

In this paper, we propose and investigate the feasibility of a video streaming framework *TransFetch* in the challenging scenario of public transport based on the video viewing behaviors of commuters. The framework consists of on-board caches and road-side caches located at public transport stations that intelligently pre-fetch chunks of videos to increase on-board cache hit rates. In specific, a novel chunk placement algorithm is proposed to determine the optimal video chunks to fetch in advance at each road-side unit. We model *TransFetch* system and evaluate its performance with large scale real-life datasets under public transport system. We show that on average the user quality of experience can be increased by approximately 35%, and at the same time cellular data usage can be reduced by as much as 45%. In the future work, we aim to experimentally validate *TransFetch* using the developed prototype implementation on *Raspberry-Pi* and *Android* devices.

REFERENCES

- [1] Cisco, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2010 - 2015," Tech. Rep., 2016.
- [2] D. Kelleher, "95.6% of Commuters in the US put Company Data at Risk over Free Public Wi-Fi," Tech. Rep., 2013.
- [3] O. Software, "Mobile operators and their customers still struggling with video stalling," Tech. Rep., 2014. [Online]. Available: <http://www.operasoftware.com/press/releases/mobile/2014-02-24-2>
- [4] D. Ghadiyaram, A. C. Bovik, H. Yeganeh, R. Kordasiewicz, and M. Galant, "Study of the effects of stalling events on the quality of experience of mobile streaming videos," in *GlobalSIP'14*. IEEE, 2014, pp. 989–993.
- [5] Oxyfi, "Oxyfi supplies on-board Internet to Norrtag," Tech. Rep., 2013. [Online]. Available: <http://iis.oxyfi.se/En/NewsDisplay/4/35/24>
- [6] F. Jiang, K. Thilakarathna, M. A. Kaafar, F. Rosenbaum, and 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 ACM MobiCom Chants'15*. ACM, 2015, pp. 37–42.
- [7] Y. Chen, B. Zhang, Y. Liu, and W. Zhu, "Measurement and modeling of video watching time in a large-scale internet video-on-demand system," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2087–2098, 2013.
- [8] F. Jiang, "Factors influencing performance of smart mobile user stash," *NICTA, Tech. Rep*, 2014.
- [9] IBM, "Australian Commuter Pain Study White Paper," Tech. Rep., 2010.
- [10] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 5, pp. 1357–1370, 2009.
- [11] S. Traverso, P. Torino, M. Garetto, U. Torino, and P. Giaccone, "Temporal Locality in Today's Content Caching : Why it Matters and How to Model it," vol. 43, no. 5, pp. 6–12, 2013.
- [12] S. Bae, D. Ban, D. Han, J. Kim, K.-h. Lee, S. Lim, W. Park, and C.-k. Kim, "StreetSense: Effect of Bus Wi-Fi APs on Pedestrian Smartphone," in *Proceedings of the ACM IMC'15*. ACM, 2015, pp. 347–353.
- [13] E. M. Van Eenennaam, "A survey of propagation models used in vehicular ad hoc network (VANET) Research," *Paper written for course Mobile Radio Communication, University of Twente*, 2008.
- [14] K. Thilakarathna, F. Jiang, S. Mrabet, M. A. Kaafar, A. Seneviratne, and P. Mohapatra, "Demo: Crowd-cache-popular content for free," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 2014, pp. 358–359.
- [15] F. Qian, K. S. Quah, J. Huang, J. Erman, A. Gerber, Z. Mao, S. Sen, and O. Spatscheck, "Web caching on smartphones: Ideal vs. reality," in *Proceedings of Mobisys'12*. ACM, 2012, pp. 127–140.
- [16] A. Finamore, M. Mellia, Z. Gilani, K. Papagiannaki, V. Erramilli, and Y. Gruenberger, "Is there a case for mobile phone content pre-staging?" in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. ACM, 2013, pp. 321–326.
- [17] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," in *Proceedings of the Mobisys'10*. ACM, 2010, pp. 209–222.
- [18] B. D. Higgins, J. Flinn, T. J. Giuli, B. Noble, C. Peplin, and D. Watson, "Informed mobile prefetching," in *Proceedings of the MobiSys'12*. ACM, 2012, pp. 155–168.
- [19] C. Koch and D. Hausheer, "Optimizing mobile prefetching by leveraging usage patterns and social information," in *Network Protocols (ICNP'14)*. IEEE, 2014, pp. 293–295.
- [20] K. Liang, J. Hao, R. Zimmermann, and D. K. Y. Yau, "Integrated prefetching and caching for adaptive video streaming over HTTP: an online approach," in *Proceedings of the 6th ACM MM*. ACM, 2015, pp. 142–152.
- [21] J. Davidson, B. Liebal, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston et al., "The youtube video recommendation system," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 293–296.
- [22] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 39–46.
- [23] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun, "Temporal recommendation on graphs via long-and short-term preference fusion," in *Proceedings of the 16th ACM SIGKDD*. ACM, 2010, pp. 723–732.