APRANK: JOINT MOBILITY AND PREFERENCE-BASED MOBILE VIDEO PREFETCHING

Ge Ma[1]*, Zhi Wang*[2]*, Minghua Chen*[4]*, Wenwu Zhu*[1,3]

[1]Tsinghua-Berkeley Shenzhen Institue, Tsinghua University, Shenzhen, China
[2]Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[4]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China
mg15@mails.tsinghua.edu.cn, wangzhi@sz.tsinghua.edu.cn, minghua@ie.cuhk.edu.hk, wwzhu@tsinghua.edu.cn

ABSTRACT

Today's internet has witnessed a fast growth of mobile video streaming. Different from traditional PC/laptop-based video streaming, mobile video streaming relies on the usage of mobile devices and wireless networks, allowing people to receive video content on the *move*. The change has challenged traditional video content delivery, which uses centralized infrastructure (e.g., CDN) inside the network for content distribution, in a sense that mobile users (connected to Wi-Fi or cellular networks) encounter large delay and small download speed. One promising solution is to prefetch content in the edge of the network, e.g., on access points (APs). However, it faces the great challenges: 1) It is difficult to prefetch content in such edge APs with limited storage capacity; 2) Users' mobility cross APs affects the content delivery; 3) Popularity of content may change significantly across APs. Previous approaches make mobile video content delivery inefficient without jointly considering these problems. In this paper, we propose an AP-assisted mobile video delivery framework to solve these problems. First, using large-scale measurement studies of users' trajectories and preferences of videos, we reveal that both users' mobility patterns and their intrinsic preferences are important for AP-assisted content delivery. Second, we formulate the AP content prefetching as an optimization problem, and develop an online solution, *APRank*, to solve it. Third, we evaluate the effectiveness of our design, compared with four baselines, random-based, popularity-based, preference-based and offline prefetching.

1. INTRODUCTION

Today's Internet has witnessed an unprecedented global growth of mobile data traffic that is expected to reach 11 EB/month at the end of 2017, where mobile video traffic accounts for 60%. According to Cisco Forecast, mobile video traffic will occupy over 78% of the world mobile data traffic by 2021 [1]. Different from traditional PC/laptop-based video streaming, mobile video streaming relies on the usage of mobile devices and wireless networks, allowing people to receive video content on the *move*.

The explosive increase of mobile video streaming is changing the video delivery landscape, and the change has challenged traditional video content delivery, which uses centralized infrastructure (e.g., content delivery network) inside the Internet backbone for content distribution, in a sense that mobile users (connected to Wi-Fi or cellular networks) encounter large delay and small download speed. Many researchers realize that the abundant resources, such as set-top, small cell base station (SBS) [2], and smartrouter [3], can be utilized to assist the video content delivery and offload the traffic from the backbone.

One promising solution in recent years for mobile video streaming has been to prefetch content in the edge of the network (e.g., on APs) [4, 5]. According to [1], a large number of such edge smart APs (over 6.5M) have been deployed in China to perform content delivery. Youku, one of the largest online video providers in China, has deployed over 300K smartrouters in their users' homes, expecting to turn a large fraction of its users (250M) to such content delivery AP nodes [6].

Though the idea seems simple, it faces the great challenges: 1) It is difficult to prefetch content in such edge APs with limited storage capacity (usually several GBs). Given the large amount of video content uploaded to today's mobile video services, it is impossible for any individual AP to cache a full set of content as conventional CDN servers do [4, 5]. 2) Users' mobility cross APs affects the content delivery. It has become common for users to receive video content in different locations and on the move. Thus, a user may have to download content from different APs when s/he is in different locations. The mobility patterns also affect the content prefetch for these APs, while previous studies have usually focused on social and usage information [7]. 3) Popularity of content may change significantly across APs. According to previous measurement studies [8], the popularity distributions for even adjacent APs are different; and previous content

This work is supported in part by the National Natural Science Foundation of China under Grant No. U1611461, 61402247 and 61531006, in part by the National Basic Research Program of China (973) under Grant No. 2015CB352300 and 2013CB336700, in part by the University Grants Committee of the Hong Kong Special Administrative Region, China, under Grant C7036-15G, and in part by the research fund of Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

placement strategies do not consider such fine-grained popularity difference.

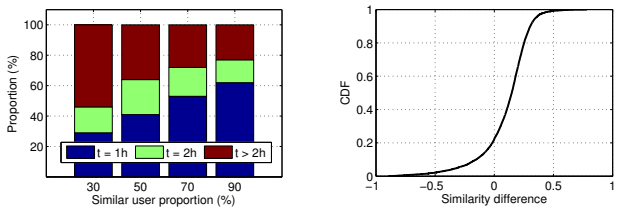
To tackle these challenges, we propose a joint mobility- and preference-based mobile video delivery framework. Different from previous studies, we use a data-driven approach to guide our design by carrying out measurement studies on real-world mobile video streaming traces, and propose a prefetching approach using both user mobility and preference prediction to maximize the hit rate with the limited edge AP cache capacity. Our contributions can be summarized as follows.

▷ First, using large-scale measurement studies of users' trajectories and preferences of videos, we reveal that both users' mobility patterns and their intrinsic preferences are important for AP-assisted content delivery. We perform large-scale measurement studies on over 1 million APs and 30 million records covering 2 million users watching more than 0.3 million videos in 2 weeks. Our key observations include: 1) More than 70% users are likely to watch videos in the same category (e.g., TV series, Animation and variety show); 2) The user preference is stable over time, and few users' similar preferences (based on the history of requested videos) are time-varying; 3) Mobile users have similar request patterns at the regularly visited APs.

▷ Second, we formulate the AP content prefetching as an optimization problem, and develop an online solution, *APRank*, to solve it, which employs the Markov random fields (MRFs) theory. Driven by the measurement studies, we design and build user mobility and preference predictive models to capture the popularity distribution of content in different edge APs. Instead of identifying and modeling personal mobility pattern, we propose to capture crowd mobility patterns eliminating the randomness of personal mobility to predict the propagation of content. We employ the theory of MRFs to infer the user preference, and design an algorithm to predict the content propagation due to the user mobility. Using the predictive models, we then formulate the content prefetching as a knapsack problem. The objective is to maximize the hit ratio for all edge APs using limited cache capacity. We then propose a greedy strategy to practically solve this problem in an online manner.

▷ Third, we evaluate the effectiveness of our design, compared with four baselines, random-based, popularity-based, preference-based and offline prefetching. Based on real-world trace-driven experiments, the results show that our design achieves 20% (resp. 30%) hit rate improvement and 20% (resp. 30%) service rate improvement, compared with the popularity-based (resp. random-based) prefetching.

The rest of the paper is organized as follows. We present the measurement studies that motivate our design in Section 2. We present the system architecture and formulate the problem in Section 3. In Section 4, we present the details of our prefetching approach based on mobility and preference predictive models. We evaluate our design in Section 5. Finally, we conclude the paper in Section 6.



(a) The distribution of request time against different proportions of similar users.

(b) CDF of similarity difference between two consecutive periods.

Fig. 1: User preference analysis.

2. MEASUREMENT STUDIES

In this section, we present the measurement results that motivate our study.

2.1. Datasets

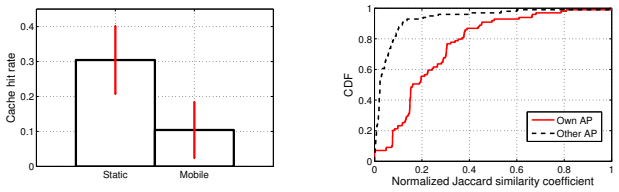
Traces of AP information. The dataset is provided by a mobile App, which asks users to respond to questions on how they use wireless networks. In particular, we collected over 1 million APs in Beijing city, including the Basic Service Set Identifier (BSSID), timestamp and location of each AP. Using these traces, we are able to infer the users' connections to edge APs.

Traces of mobile video sessions. The mobile video dataset is collected by one of the most popular video providers in China. How users watch videos in the mobile video streaming App has been recorded. The dataset spanning 2 weeks covers 2 million users watching 0.3 million unique videos in Beijing city. In each trace item, the following information is recorded: the user ID, the timestamp and location when and where the user watches the video, the title and duration of the video. Based on these traces, we can study the user preference and mobility.

2.2. User Preference Patterns

First of all, we classify the videos into several categories according to the video duration. There are noticeable peaks around 2 minutes, 11 minutes, 24 minutes, 45 minutes in the video duration distribution. So we classify them into the following types: trailer, short variety show, animation, TV series and others (e.g., movie and reality show). We then explore the users' preferences to each video category. We observe that more than 70% users are likely to watch videos in the same category, which is similar to the result in [8].

Next, we investigate the characteristics among different users. In the analysis, we focus on the users who watch videos at least once a day. For each user, we calculate the Jaccard similarity coefficients of the video sets that the user and others have watched. We select the top 10% ones as his/her similar users. As illustrated in Figure 1(a), we explore whether the time when a user will watch a video is related to the proportion of his/her similar users who have watched the same video. It shows that about 41% (resp. 64%) of the users will watch the same videos within an hour (resp. two hours), when



(a) Impact of user mobility.

(b) Similarity between mobile users and APs.

Fig. 2: User mobility analysis.

the proportions of their similar users who have watched the videos reach 50%. Furthermore, we study whether the similarity among users is varying greatly over time. For each user, we calculate the average similarity coefficient of his/her similar users, in the first and second week respectively. Figure 1(b) plots the CDF of similarity coefficient differences between the first and second week. The value greater (resp. less) than 0 indicates that the similarity decreases (resp. increases) gradually over time. The result shows that about 20% users' similarities increase over time, and 40% (resp. 30%) users' similarities decrease less than 0.2 (resp. 0.3). Given these results, we claim that each user has category preference and the preference shows obvious stability.

2.3. User Mobility Patterns

We study the impact of user mobility on the edge network caching performance. In particular, we measure the cache hit rates (the fraction of the number of requests served by cache over the total number of requests) for mobile and static users. Figure 2(a) presents that in each AP the cache hit rate of mobile users is much lower than that of static users under Least Recently Used (LRU) caching strategy. So the user mobility is a significant factor in caching and prefetching strategies.

Then we investigate whether user mobility results in content propagation. We define that a user's *dominant AP* is the AP where the user issues his/her most requests. Figure 2(b) shows the Jacarrd similarity between mobile users and their dominant APs. The curve of own dominant AP is the CDF of normalized similarity coefficient between the videos that a mobile user will watch during the period $[t, t+\Delta t]$ and all the videos watched by the users at his/her dominant AP during the period $[t-\Delta t, t]$. The curve of other dominant AP is the CDF of similarity coefficient between the videos that the same user will watch and the videos watched by the users at a dominant AP, randomly chosen from another mobile users. We observe that mobile users are more likely to have similar request patterns with their dominant APs. It is hard that mobile users have good quality of experience, except for in their dominant APs. The above results give us the basic characteristics of mobile users.

3. ARCHITECTURE AND FORMULATION

In this section, we present the system architecture and formulate the content prefetching problem as a classic 0/1 Knapsack problem.

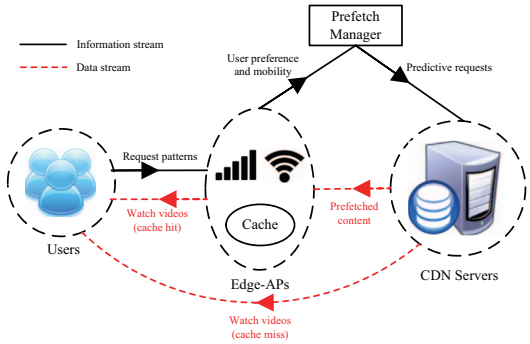


Fig. 3: System architecture.

3.1. System Architecture

We present the proposed system architecture in Figure 3. The proposed prefetching approach is applied in edge APs between users and the CDN servers which store all the videos. Since the edge APs are equipped with storage space and computing capacity, they are potential to assist the content delivery. In this system, the edge APs and CDN servers are operated by the content service providers. The edge APs handle all the users' requests and the videos received from the CDN servers. Meanwhile, the Prefetch Manager in the edge APs learns the request pattern of each user based on his/her mobility and preference, then predicts future requests and prefetches corresponding videos from the CDN servers to their caches. When new requests arrive, they will be served directly by the edge APs if the requested videos have been prefetched. Otherwise, the requests will be served by the remote CDN servers.

3.2. Problem Formulation

In our AP-assisted content prefetching problem, our goal is to devise the *prefetching strategy* $\mathbf{x}$ that maximizes the total cache hit rate of all edge APs. Since we assume that prefetching strategy in each AP does not affect the rest APs, maximizing total cache hit rate of all edge APs is equivalent to maximizing the hit rate of each AP. Thus, the problem can be formulated as a classic 0/1 Knapsack problem:

$$\begin{aligned} \max_{\{\mathbf{x}^t\}} \quad & \mathrm{J}(\mathbf{x}^t) = \sum_{v=1}^{V} \tfrac{1}{w_v} r_v^t x_v^t \\ \text{s.t.} \quad & \sum_{v=1}^{V} w_v x_v^t \le C, \\ & x_v^t = \{0,1\}, \forall v \in \mathcal{V}. \end{aligned} \tag{1}$$

where $\mathcal{V} = \{1, \ldots, V\}$ is the set of V video files, C is the cache capacity, w_v and r_v^t denote the weight and request demand of video v, w_v is defined as the size of the corresponding video. The request demand r_v^t is the number of users that may watch v within the time $[t, t+\Delta t]$. The optimization variable x_v^t indicates whether v will be prefetched ($x_v^t = 1$) at time t or not ($x_v^t = 0$). Then, the prefetching strategy at time t is given by vector $\mathbf{x}^t = (x_1^t, \ldots, x_V^t)$. Since w_v is fixed, the key to determine the prefetching strategy $\mathbf{x}^t$ is to predict the

value of r_v^t. We will present the details of r_v^t prediction based on user mobility and preference in the following section.

4. PREFETCHING POLICIES FOR MOBILE VIDEO STREAMING

Motivated by the measurement studies, we design a joint user mobility- and preference-based prefetching approach for mobile video delivery. We build user mobility and preference predictive models to capture the popularity distribution of content in different edge APs. Based on these models, we predict whether a video will be highly requested due to the users' mobility and preferences in a particular edge AP.

4.1. User Preference-based Popularity Prediction

Based on the previous observation that similar users are more likely to watch the same video during a period of time and the assumption that the user state y at time $t + \Delta t$ can be calculated from his previous state at time t, we employ the theory of Markov random fields (MRFs). We will first discuss the basic ideas of our approach. Using the history of users' request traces, we calculate the Jaccard similarity coefficients among users to build a user-user similarity network. If the request patterns of two users are similar (coefficient is greater than 0.3), they are neighbors of each other. For a given video, if most neighbors of a user have watch the video, we are more likely to believe that the user will watch the video, illustrated in Figure 4. We want to associate each user with a confidence probability that the user watches the video. In the MRFs, the problems are how to assign different weights to the parameters and how to estimate the probabilities based on the network. To solve the problems, there are two assumptions: 1) If a video is more popular than other videos, the probability that a user will watch the video should be higher than other videos; 2) Weights on far away neighbors are less than close neighbors. Let $y_{iv}^t = 1$ if the i-th user has watched the video v at time t and 0 otherwise. For each user, we define his neighbors, $S(i)$, as the set of users similar with i-th user. Using theory of MRFs, the probability of video labelling is proportional to $\exp(-U(y_{iv}^{t+\Delta t}))$, where

$$\begin{aligned} U(y_{iv}^{t+\Delta t}) &= -\alpha N_1 - \beta N_{10} - \gamma N_{11} - N_{00} \\ &= -\alpha y_{iv}^{t+\Delta t} - \beta \sum_{j \in S(i)} (1 - y_{iv}^{t+\Delta t}) y_{jv}^t \\ &\quad + (1 - y_{jv}^t) y_{iv}^{t+\Delta t} - \gamma \sum_{j \in S(i)} y_{iv}^{t+\Delta t} y_{jv}^t \\ &\quad - \sum_{j \in S(i)} (1 - y_{iv}^{t+1})(1 - y_{jv}^t). \end{aligned} \tag{2}$$

In the terminology of MRF, $U(y)$ is referred as the *potential function*. This function defines a gloabl Gibbs distribution of the entire network, the probability $P(y_{iv}^{t+\Delta t}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp(-U(y_{iv}^{t+\Delta t}))$, where $\boldsymbol{\theta} = (\alpha, \beta, \gamma)$ are parameters and $Z(\boldsymbol{\theta})$ is a normalized constant as the partition function in the theory of MRF. To calculate the probability, we use a Gibbs sampler:

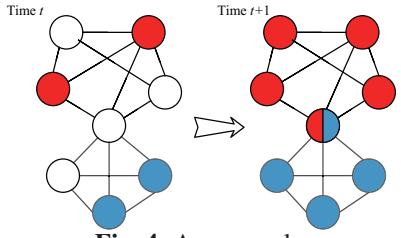


Fig. 4: An example.

$$\begin{aligned} P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]v}^t, \boldsymbol{\theta}) &= \frac{P(y_{iv}^{t+\Delta t} = 1, \mathbf{y}_{[-i]v}^t | \boldsymbol{\theta})}{\sum_{k=0}^{1} P(y_{iv}^{t+\Delta t} = k, \mathbf{y}_{[-i]v}^t | \boldsymbol{\theta})} \\ &= \frac{e^{\alpha + (\beta-1) M_{iv}^0 + (\gamma - \beta) M_{iv}^1}}{1 + e^{\alpha + (\beta-1) M_{iv}^0 + (\gamma - \beta) M_{iv}^1}} \end{aligned} \tag{3}$$

where $\mathbf{y}_{[-i]v}^t = (y_{1v}^t, \cdots, y_{i-1,v}^t, y_{i+1,v}^t, ..., y_{Nv}^t)$, N is the number of users, M_{iv}^0 and M_{iv}^1 are the numbers of neighbors of i-th user, labelled with 0 and 1, respectively.

It is difficult to use the maximum likelihood estimation method directly to estimate the parameters $\boldsymbol{\theta} = (\alpha, \beta, \gamma)$, because the partition function $Z(\boldsymbol{\theta})$ is also a function of the parameters. Thus, we use the logistic regression approach,

$$\begin{aligned} &\log \frac{P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]t}^t, \boldsymbol{\theta})}{1 - P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]t}^t, \boldsymbol{\theta})} \\ =&\alpha + (\beta - 1) M_{iv}^0 + (\gamma - \beta) M_{iv}^1. \end{aligned} \tag{4}$$

The details of this method are presented in Algorithm 1. In particular, we first initialize the parameters $\boldsymbol{\theta}$, e.g. $\mathbf{0}$ (line 1), and estimate them using a quasi-likelihood estimation method based on a linear logistic model (line 2). Then, in each time slot, we utilize the Gibbs sampling to iteratively obtain the stabilized posterior probabilities $P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]v}^t, \boldsymbol{\theta})$ (line 3–line 10).

Algorithm 1: Preference-based Popularity Prediction

Input: user-user similarity network $\mathcal{G}$ and video labels $\mathbf{y}_v^t$
Output: $P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]v}^t, \boldsymbol{\theta})$

1 initialize the parameters $\boldsymbol{\theta}$ (e.g., $\mathbf{0}$)
2 estimate $\boldsymbol{\theta}$ with Eq. (4)
3 **for** $k = 1, \cdots, K$ **do**
4 update the value of y_{iv}^t with Eq. (3)
5 $p = rand(0, 1)$
6 **if** $y_{iv}^t > p$ **then**
7 $n_1 = n_1 + 1$
8 **end**
9 **end**
10 $P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]v}^t, \boldsymbol{\theta}) = \frac{n_1}{K}$
11 **return** $P(y_{iv}^{t+\Delta t} = 1 | \mathbf{y}_{[-i]v}^t, \boldsymbol{\theta})$

4.2. Crowd Mobility-based Content Propagation

In order to eliminate the randomness and uncertainty of personal mobility, we develop an algorithm to model corwd mobility to predict the propagation of content. Based on our previous measurement studies, we observe that the edge APs not only have different popularities, but also different correlation levels between each other – users tend to have similar request patterns with their dominant APs more than that of others. In our design, we focus on the proportion of mobile users between different edge APs instead of each mobile user.

We consider a general network architecture where a set $\mathcal{L}$ of L edge APs provide video content access to their users. Let $\mathbf{u}_l^t = (u_{1l}^t, \cdots, u_{Ll}^t)$ denotes the number of users from other APs to l-th AP at time t. The real demand distribution of each AP is given, $\mathbf{d}_l^t = (d_{l1}^t, \cdots, d_{lv}^t, \cdots, d_{lV}^t)$, where d_{lv}^t is the request proportion of video v in l-th AP at time t. Our goal is to predict the change of popularity in each AP due to the content propagation. Without loss of generality, given a specific l-th AP, we want to obtain the future demand distribution $\mathbf{d}_l^{t+\Delta t}$. The problem $\mathbf{d}_l^{t+\Delta t}$ can be solved using a crowd mobility-based content propagation (CMCP) solution derived from PageRank [9],

$$\mathbf{d}_l^{t+\Delta t} = \mathbf{d}_{l,k}^t = \mathrm{CMCP}(\mathbf{u}_l^t, \mathbf{d}_{l,k-1}^t),$$
$$\mathrm{CMCP}(\mathbf{u}_l^t, \mathbf{d}_{l,k-1}^t) = \frac{\sum_{i=1}^{L}\sum_{j=1}^{V} u_{il}^t d_{ij,k-1}^t}{\sum_{i=1}^{L} u_{il}^t}. \quad (5)$$

The details are presented in Algorithm 2. It adopts the value iteration technique, which extends the PageRank and uses L_2 norm to estimate errors. At every iteration, the future demand distribution $\mathbf{d}_{\mathcal{L}}^{t+\Delta t}$ is updated based on the user migration and previous iteration result (line 5). This process continues until $\mathbf{d}_{\mathcal{L}}^{t+\Delta t}$ begins to converge.

Merging the results of Algorithm 1 and Algorithm 2, we can predict the request demand r_v^t in Eq. (1) based on user mobility and preference. In order to maximize the hit rate for each edge AP, we can adopt a greedy strategy that prefetches the greatest demand video units ($\frac{1}{w_v} r_v^t$ in Eq. (1)) no more than the cache capacity.

Prefetching Timing. Note that the prefetching will download uncached videos before users watch them. As a result, a user has better quality of experience if s/he watches a video that has been prefetched. However, prefetching can be a waste of network resource if the prefetched videos are not watched by users. Also, prefetching too frequently degrades the user perceived quality because it incurs additional bandwidth consumption in both the edge APs and content servers. In real-world systems, the prefetching timing should be scheduled according to not only the gain of user experience, but also the bandwidth cost of the dedicated servers.

Algorithm 2: Mobility-based Content Propagation

Input: user migration of all APs $\mathbf{u}_{\mathcal{L}}^t$ and
real demand distribution of all APs $\mathbf{d}_{\mathcal{L}}^t$
the convergence threshold ε
Output: $\mathbf{d}_{\mathcal{L}}^{t+\Delta t}$

1 initialize $\mathbf{d}_{\mathcal{L},0}^t = \mathbf{d}_{\mathcal{L}}^t$
2 **while** $E \leq \varepsilon$ **do**
3 $E = 0$
4 **for all** $l \in \mathcal{L}$ **do**
5 update the value of $\mathbf{d}_{l,k}^t$ with Eq. (5)
6 $\Delta = \left\| \mathbf{d}_{l,k}^t - \mathbf{d}_{l,k-1}^t \right\|_2$
7 $E = E + \Delta$
8 **end**
9 **end**
10 $\mathbf{d}_{\mathcal{L}}^{t+\Delta t} = \mathbf{d}_{\mathcal{L},n}^t$
11 **return** $\mathbf{d}_{\mathcal{L}}^{t+\Delta t}$

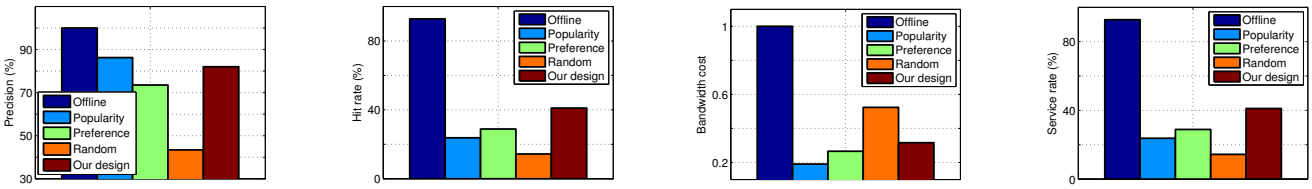
5. PERFORMANCE EVALUATION

In this section, we use trace-driven experiments to verify the effectiveness of our design.

5.1. Experiment Setup

In our experiments, we use the users in our traces who watch at least one video per day, and we use the top 10% edge APs with the most requests, occupying more than 30% of total requests. We use the traces of the first week to train the user similarity network and the traces of the second week to evaluate the performance. Considering the constraints in Eq. 1, we set the cache capacity $C = \lambda \times unique\ \ video\ \ number \times average\ \ video\ \ size$, where $\lambda = 0.1‰$, the number of unique video is 100K and the average size of video is 180 MB; thus the cache capacity $C \approx$ 2GB. Intuitively, with a larger value of λ, the larger cache capacity is, the better performance prefetching has. However, more contents to prefetch, in turn, brings more bandwidth cost, which degrades the users' quality of experience. In our experiments, the running time ($< 1\ min$) of our design can be ignored compared with the video transmission time ($< 1\ h$). So we set the time interval Δt between two consecutive prefetching processes at the same edge APs $1\ h$.

In our study, we compare our design with baselines as follows: 1) Popularity-based prefetching. It prefetches the most popular video first until the cache is full. 2) Preference-based prefetching. It prefetches the highest probability video first according to Algorithm 1 until the cache is full. 3) Random-based prefetching. Given a specific AP, there may exist mobile users between the AP and other APs. The random prefetching selects an AP based on the distribution of mobile users and then randomly selects a candidate video from the selected AP. 4) Offline prefetching. It assumes that not only the history information, but also the future infor-



(a) Precision (b) Hit rate (c) Bandwidth cost (d) Service rate

Fig. 5: Perfromance comparison.

mation are known, e.g., where, when and what the user will watch are all available for scheduling. It prefetches the videos to achieve the optimal performance.

5.2. Experiment Results

We use the following metrics to verify the performance of our design.

Precision. It is the fraction of prefetched videos that will be downloaded by users. Figure 5(a) plots the precision results of different prefetching algorithms. It shows that the precision values of popularity-based prefetching and our design are greater than 80%. The results indicate that our design can achieve a relatively high precision by only prefetching a small fraction of the most popular content.

Hit rate. It is the fraction of the number of requests served by prefetched content over the number of total requests. As illustrated in Figure 5(b): 1) The optimal hit rate (offline prefetching) is smaller than 100% due to the limited cache capacity. 2) The hit rate of preference-based prefetching is higher than that of popularity-based. It indicates that preference-based approach improves the hit rate with prefetching frequently, which in turn decreases the precision. 3) Our design can achieve a higher hit rate with about 20% (resp. 30%) improvement compared with the popularity-based (resp. random-based) prefetching.

Bandwidth cost. It is the additional bandwidth cost of prefetching, which is proportional to the number of prefetching. Since the offline algorithm prefetches as much as possible, its bandwidth cost is highest. We select offline prefetching as the baseline and set its cost to 1. As illustrated in Figure 5(c), the bandwidth cost of popularity-based prefetching is smallest because the most popular videos are only a small fraction of total videos. Our design can bring about 15% additional bandwidth cost of CDN server compared with the popularity-based prefetching, while it has much higher hit rate. Compared with bandwidth cost with no prefetching, though our design uses about 10% additional bandwidth to prefetch content, it significantly increases the hit rate which saves about 20% bandwidth cost of CDN server.

Service rate. It is the fraction of the number of AP-served users over the number of total users. As illustrated in Figure 5(d), our design outperforms the popularity-based and random-based prefetching with respect to service rate, with improvements up to 20% and 27% respectively. It indicates that our design is sensitive to the users' mobility behaviors.

6. CONCLUSIONS

New video encoding/transcoding technology, smart devices and ubiquitous wireless networks jointly enable the fast growth of mobile video services. Allowing people to receive video content on the move, mobile video service has challenged the centralized conventional content delivery paradigms. To improve the quality of user experience for mobile video services, edge-assisted content delivery paradigms have been proposed and utilized by real-world systems. In this paper, we study prefetching strategies in AP-assisted content delivery, and propose a joint mobile- and preference-aware prefetching framework, which is different from conventional strategies based purely on content popularity. We design *APRank*, an online algorithm that provides predictions of both people's mobility and preference, and schedules prefetching for each individual AP. Compared with traditional schemes, our design can significantly improve hit rate for edge APs, especially when people watch videos from multiple locations regularly.

References

[1] Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2016-2021," *San Jose, USA: Cisco White paper*, 2017.

[2] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *Communications, IEEE Transactions on*, vol. 62, no. 10, pp. 3665–3677, 2014.

[3] L. Chen, Y. Zhou, M. Jing, and R. TB Ma, "Thunder crystal: a novel crowdsourcing-based content distribution platform," in *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2015, pp. 43–48.

[4] D. K. Krishnappa, S. Khemmarat, L. Gao, and M. Zink, "On the feasibility of prefetching and caching for online TV services: a measurement study on hulu," in *International Conference on Passive and Active Network Measurement*. Springer, 2011, pp. 72–80.

[5] Z. Wang, L. Sun, S. Yang, and W. Zhu, "Prefetching strategy in peer-assisted social video streaming," in *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 2011, pp. 1233–1236.

[6] M. Ma, Z. Wang, K. Su, and L. Sun, "Understanding content placement strategies in smartrouter-based peer video CDN," in *Proceedings of the 26th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2016, p. 7.

[7] C. Koch and D. Hausheer, "Optimizing mobile prefetching by leveraging usage patterns and social information," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 293–295.

[8] W. Hu, Z. Wang, M. Ma, and L. Sun, "Edge video CDN: A Wi-Fi content hotspot solution," *Journal of Computer Science and Technology*, vol. 31, no. 6, pp. 1072–1086, 2016.

[9] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web.," *Stanford InfoLab*, 1999.