EDGE-BOOST: ENHANCING MULTIMEDIA DELIVERY WITH MOBILE EDGE CACHING IN 5G-D2D NETWORKS

Venkatraman Balasubramanian, Mu Wang, Martin Reisslein, Changqiao Xu.

{vbalas11, mwang184, reisslein}@asu.edu, cqxu@bupt.edu.cn

ABSTRACT

By moving computation and caching to the network edge, Mobile Edge Computing (MEC) offloads core networks and shortens data access latencies, which is important for large scale mobile multimedia services. Increasing the density of edge data centers to service these multimedia requests is uneconomical. Recent research has proven the benefits of involving devices in the delivery of multimedia services. This is done by exploiting the idle computation and storage resources via device-to-device (D2D) communication, i.e., by forming a so-called Mobile Device Cloud (MDC). Despite the flexibility and cost efficiency of this MDC paradigm, the timely allocation of caching resources to satisfy the dynamic user demands is challenging. This is mainly due to the uncertainty in resource availability of mobile devices. To this end, we propose Edge-Boost, a novel MDC caching architecture for low-latency multimedia streaming services. We develop a novel fluid-based model to capture the dynamically changing network status. Additionally, we propose a dynamic caching allocation to jointly minimize caching cost and service latency. Edge-Boost achieves over 20% higher average cache utilization and 15% shorter average access latency than the state-of-the-art MDC approach.

Index Terms— Mobile Edge Computing, Multimedia Processing, Low Latency

1. INTRODUCTION

Recent estimates predict a more than 20-fold increase in virtual reality traffic and over 25% increase in internet video traffic over the next five years. In this context, Mobile Edge Computing (MEC), which deploys computation and storage resource at the network edge, has been showcased as a promising solution for large scale multimedia services [1], [2]. However, MEC deployments are very costly. Generally, it is becoming more and more difficult to keep up with the increasing demands for multimedia services in upcoming 5G networks.

Recently, leveraging Device-to-Device (D2D) communications to involve the mobile equipment nodes (MEs) in the loop of multimedia delivery has gained attention in both academia and industry. In this device-involved paradigm, MEs contribute their own storage resources [3], [4] to service each other via D2D links, hence flexibly enhancing the edge cloud capability (scalability). Such a resource-rich environment of mobile devices that are substituted for the expensive MEC is called a Mobile Device Cloud (MDC). Such an MDC can be managed through extending MEC management frameworks, e.g., [5]. However, despite the MDC promises and general management frameworks, several operational challenges need to be investigated before bringing this paradigm into reality: First, unlike edge data centers, the availability of caching resources at MEs is highly dynamic due to the random ME behaviors. On the other hand, MEs contribute their own bandwidth, storage, and energy resources to provide services; these resources may get exhausted when they are aggressively utilized. In addition, the continuous ME movements and the limited D2D communication range may result in intermittent content delivery. All these factors make it challenging to reliably deliver low-latency multimedia services in MDCs.

This paper makes the following contributions:

- We design an Edge-Boost framework for edge caching in MDCs. To capture the variations of content replicas based on video demands, we develop a novel algorithm relying on a fluid model.
- We formulate the caching configuration in MDCs as an online optimization problem and propose an Edge-Boost caching algorithm. Contrary to the conventional caching methods that statically allocate the caching resources, Edge-Boost first estimates the population of different states in each time slot and then jointly minimizes the peak population of clients waiting for content and the number of replicas; thus, Edge-Boost optimizes the video access delay and the number of content copies.
- We conduct simulations to validate the performance of the proposed Edge-Boost framework. By measuring the average access latency and cache hit ratio of Edge-Boost under different scenarios, we show that Edge-Boost achieves better delay reduction and cache utilization than the state-of-the-art caching strategy.

The rest of the paper is organized as follows. Section 2 delineates the novelty of the proposed solution with a brief dis-

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00290

cussion on the most recent related work. Sections 3 and 4 present the system design and the associated problem formulation. Section 5 presents the proposed algorithm. Section 6 presents the performance evaluation and Section 7 provides concluding remarks.

2. RELATED WORK

Extensive studies have been conducted recently for replacing the edge caching capacity by device storage resources via D2D links. For instance, Zhang *et al.* in [6] utilized mobile vehicles as smart caching agents to offload the caching tasks from the Base Station (BS) using a vehicular edge structure. However, the random vehicular movements change vehicular caching, which had not been considered. Neglecting this characteristic of the vehicular environment significantly impairs the caching demand estimation, which in turn negatively affects the caching performance.

Wu *et al.* built a content sharing framework relying on the D2D assisted caching paradigm [7]. A collaborative cache management scheme is proposed that includes a distributed caching decision and updating policy. Li *et al.* proposed a delay-aware caching algorithm over D2D links [8]. By locating the best carrier, the proposed caching policy aims to minimize the transmission delay and to improve the throughput. Although these solutions positively regulate the caching capacity, they ignore the demand variations.

In [9], a Chord-based overlay structure is employed for effectively searching content providers in D2D networks. A two stage PID-based LTE traffic controller is then proposed to determine the offloading scale. However, the instability of the overlay structure incurs high maintenance overhead. Despite the caching benefits, all of the above solutions rely on the probability-based content popularity estimation. These models consider the request dynamics in mobile environments which result in inaccurate estimation of content popularity. Thus, to provide high efficiency video caching by estimating the video content requests, it is critical to observe the high variations of user demands.

In [10], 5G D2D caching for information centric networking (ICN) is proposed by formulating a fluid-based model that considers the 5G ICN caching dynamics. However, the formulated model ignores caching demand variations in highly mobile scenarios, such as in edge device clouds. In contrast, we address the challenge of satisfying demand variations that [10] did not consider. Since, the states of the mobile clients should be redefined according to the content distribution process of an MDC, our design involves five novel state definitions that encompass all possible state transitions in a D2D system and are easy to implement.

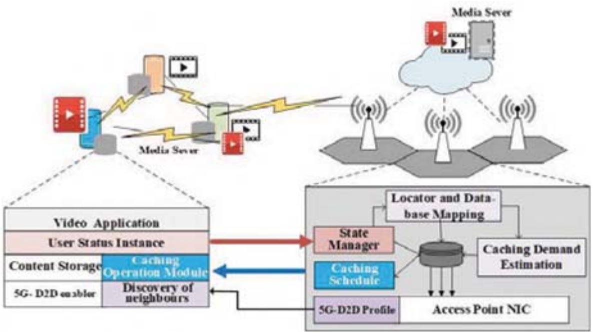

Fig. 1. Edge-Boost resource management framework

3. SYSTEM ARCHITECTURE

Fig. 1 illustrates the proposed video distribution framework over an MDC-enabled mobile network. The proposed edge-boost framework focuses on the caching time scale; we assume that the video streaming at the time scale of individual video frames employs some conventional high-performance video streaming scheme, e.g., [11, 12]. In the edge-boost framework, the controlling module at each BS mainly consists of State Manager, Caching Scheduler, Locator and Database Mapping, as well as Caching Demand Estimation. The State Manager collects the user states. A specific state is assigned to each mobile client to indicate its current operation (i.e., "caching" state, "requesting" state) for a given content item. Coupled with the location information, the client status is maintained by the locator and the database mapping unit. The fluid-based dynamic state model is built by monitoring the time variations of the client states. Further, the caching demand estimation module captures the evolution of the demand and supply capacity. Once the requested replicas for each content object have been determined, the content scheduler assigns the caching task to each in-area mobile client following the proposed allocation algorithm.

At the ME, the Caching Operation Module collects the caching content through requests to the BSs. To enable a ME-assisted video content distribution, the BSs based on 5G-D2D profiles, discover the D2D pairs for one-hop content delivery. Specifically, for received requests, the BSs first check the database to locate valid replicas of the requested content. If one or more content holders are located in the D2D communication range of a requester, a D2D link between the provider and requester is established for content delivery. Otherwise, the BS serves the request by utilizing its own bandwidth.

3.1. Video distribution through mobile edge caching

We divide the network into multiple areas, whereby each area corresponds to an MDC. Each area a consists of an MDC deployed at the BS and a set of mobile devices, denoted by B_a and M_a, respectively. Let K denote the universe of con-

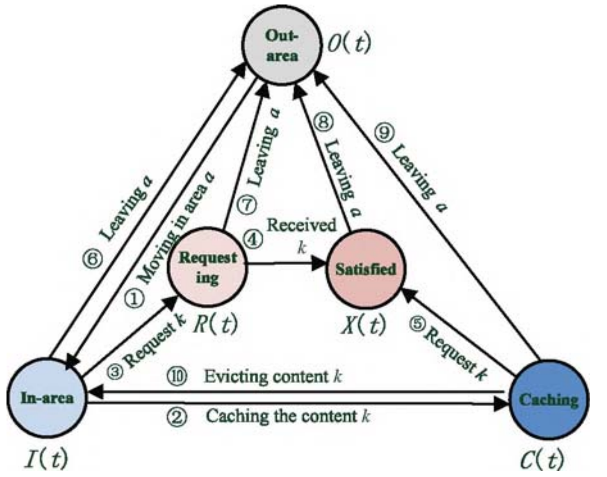

Fig. 2. Novel state transition design for edge caching video system

tent, according to the procedure of distributing the given content k ($k \in K$) in a, each mobile device can be defined by either of the following three roles: *normal nodes* are non-participants of the distribution k; a *consumer* requests the content k; *caching nodes* hold a copy of k. Based on the role assigned to MEs, we further define **five** different states:

1. *In-area state*: an ME in this state is a normal node entering the area a. We denote by $I(t)$ the population fraction of MEs in this state at time t.

2. *Requesting state*: an ME in this state is requesting the content k, we denote by $R(t)$ the population fraction of nodes in this state at time t.

3. *Caching state*: an ME in this state has been assigned a copy of k and can be used to serve other MEs. $C(t)$ denotes the population fraction of MEs in this state.

4. *Satisfied state*: a requesting ME that has received a requested content k enters this state, we define the $X(t)$ as the portion of MEs in this state.

5. *Out-area state*: an MEs in this state has moved out of the area a, $O(t)$ represents the population fraction of MEs in this state.

Each ME in network is in one of the five state; hence, the sum of all state population fraction should constantly equal one.

3.2. Fluid-based model for content distribution

Transitions between these five states are interpreted as shown in Fig.2. Owing to space limitations we only define the dynamics of transitions as follows:

Transition 1: If an out-area state ME enters the area a, it will convert to the $in - area$ state. We assume that ME movements follow the random way point (RWP) model [13].

Transition 2, 3, 4, and 5: If an in-area ME request content k, it enters the requesting state. Modeling the request arrival rates as Poisson processes, the probability of an ME requesting content k within a short time interval Δt can be denoted $\lambda_k \Delta t$, where λ_k is the Poisson rate. For short Δt, $\Delta t \to \mathrm{d}t$. We denote the rate of transition 2 by $\beta_k I(t)$. When an ME has been selected to cache k, it will convert to the "caching" status. We denote $\varphi_k(t)$ as the caching parameter to determine the proportions of agents to cache the content at time t. i.e., $\varphi_k(t)$ represents the caching policy. For instance, $\varphi_k(t) = 1$ indicates that all MEs in state $I(t)$ cache content k. When an ME obtains the requested k, its state converts to the satisfied state. Assuming the caching cluster (MDC) is transparent to end users, the video system at area a can be treated as an M/M/1 queue [14], whose arrival rate and service rate are $\beta_k I(t) B_d$ and $U + \varphi_k(t) I(t) B_u$, respectively, where U denotes the constant MDC service rate. B_d and B_u are the download and upload bandwidth, respectively. The average waiting delay before acquiring the content equals $1/U + \beta_k I(t) B_d - \varphi_k(t) I(t) B_u$, the probability of a node in $I(t)$ converting to state $X(t)$ equals to $U + \beta_k I(t) B_d - \varphi_k(t) I(t) B_u$. Further, the conversion rate equals $R(t)[U + I(t)(\beta_k B_d - \varphi_k(t) B_u)]$. An ME in the caching status may also become interested in the content. As the requested content is already local, it can directly convert to the satisfied status. Similar as $Transition$ 2, the conversion rate of *Transition 5* can be characterized by $C(t)\beta_k$.

Transitions 6, 7, 8, 9, and 10: As an ME in area a can move out arbitrarily, all states will transit to the out-area state. The conditional probability density function of ME with moving range δ_n at coordinates (x_n, y_n) is characterised by a piece-wise function $f(r)$, where r denotes the distance from the corresponding BS; $F_n(\delta_n) \,|\, (x_n, y_n)$ denotes the probability distribution of the BS area. In *Transition 10*, as the ME caching space is limited, a caching eviction occurs when the storage is full. We denote v_k for the eviction probability of content k which is the inverse of k's average cache lifetime

$$E(T_k) = \beta_k^{-1} e^{\beta_k} e^{\tau_k} - \frac{e^{-\beta_k \tau_k \left(\tau_k + \frac{1}{\beta_k}\right)}}{e^{\beta_k \tau_k}} + \tau_k, \qquad (1)$$

whereby τ_k is the eviction threshold time. The dynamics of all states is characterized by the following O.D.E functions with initial state[1] $(I(t_0), R(t_0), C(t_0), X(t_0), O(t_0))$:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = f(r)O(t) - I(t)\gamma(t) + E(T_k)^{-1} C(t) \qquad (2)$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t} = \beta_k I(t) - [F_l(x_0, y_0) + W_k(t)] R(t) \qquad (3)$$

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = W_k(t)R(t) + \beta_k C(t) - F_l(x_0, y_0) X(t) \qquad (4)$$

[1] without loss of generality, we assume that t_0=0, namely, the start-up time of the system

$$\frac{\mathrm{d}C(t)}{\mathrm{d}t} = \varphi_k(t)\, I(t) - L(t)\, C(t) \tag{5}$$

$$\frac{\mathrm{d}O(t)}{\mathrm{d}t} = F_l(x_0, y_0)\,[1 - O(t)] - f(r)O(t)\,, \tag{6}$$

with $\gamma(t) = [\sigma(t) + \beta_k + F_l(x_0, y_0)]$, $W(t) = I(t)[U + I(\beta_k B_d - \varphi_k(t) B_u)]$, and $L(t) = \beta_k + E^{-1}(T_k) + F_l(x_0, y_0)$.

4. PROBLEM FORMULATION

The optimal allocation policy for mobile edge scenario should consider the trade-off between service capacity and caching consumption. We formulate the objective function of the caching allocations as:

$$J(\varphi_k) = \alpha R(T_m) + \beta C(T_m)\,, \tag{7}$$

where $\alpha + \beta = 1$ and $T_m \triangleq \{T_m | R_k(T_m) = \max_{t \in T} R_k(t)\}$ indicates the peak load of the video system. Hence, the first term penalizes the system when there is a high peak load and second term penalizes when there is excessive caching redundancy[2]. Additionally, the mobile scenarios vary stochastically (node states and network topology), thus, it is necessary to formulate caching optimization as an online optimization problem that can adapt to the caching allocation dynamically. We assume that the time is slotted as $T \triangleq \{T(1), T(2), T(3), \ldots\}$. For simplicity, we assume that the slot time $\Delta T(i) = T(i+1) - T(i) = \Delta T$ is constant. Thus, the time varying form of the objective function (7) at ΔT_i can be expressed as:

$$J_k(\varphi_k(T(i))) = \alpha R_k(T_m(i)) + \beta C_k(T_m(i))\,, \tag{8}$$

where $R_k(T_m(i))$ and $C_k(T_m(i))$ are the $R(T_m)$ and $C(T_m)$ of content k during the time interval $(T(i), T(i+1))$. The caching optimization in BS area a is:

$$\text{Minimize} \quad \sum_{i=1}^{N} \sum_{T(i) \in T} J_k(\varphi_k(T(i))) \tag{9}$$

$$\text{S.t} \quad 0 \le \varphi_k(T(i)), \forall T(i) \in T. \tag{10}$$

5. ALGORITHM DESIGN

We address the user dynamics with an online optimal caching allocation mechanism based on solving problems. For simplicity, in solving Eqns. (9) and (10) we assume that each ME contributes the same cache storage space S. At each time slot $t, t \in T$, our proposed algorithm first calculates the $J(\psi)$ under different $\psi(t)$ via solving the ODE functions (2)–(6). Due to the problem complexity, we use Heun's method to compute numerical solutions for (2)–(6). According to the discussion in Section 3, the calculation of the numerical solution of (2)–(6) requires the initial value of the number of users in different states; moreover deriving $f(r)$ and $F_l(x_0, y_0)$ requires the average speed V, while deriving $E(T_k)^{-1}$ requires β_k.

We first discuss how to derive the initial state of EdgeBoost. Each ME in area a maintains a 4-tuple $\{\mathbb{N}_k(T(i)), \mathbb{R}_k(T(i)), \mathbb{C}_k(T(i)), \mathbb{S}_k(T(i))\}$ to identify its state for every content k, whereby $\mathbb{N}, \mathbb{R}, \mathbb{C}, \mathbb{S}$ are defined as the in-area, requesting, caching and satisfied state, respectively. Each value in this 4-tuple is a 0 or 1 indicating whether the ME is in the corresponding state or not. For example, when ME i has received video content k, the corresponding 4-tuple will be $\{0, 0, 0, 1\}$. Based on this 4-tuple design, the BSs are able to estimate the initial state at time t. Specifically, by collecting this 4-tuple from all MEs in the serving area, the BS of area a calculates the numbers of users in the *in-area*, *requesting*, *caching*, and *satisfied* state for all content items at time t. Each BS also shares the number of users in its area with other BSs in order to estimate the users in the *out-area* state. The requesting rate of β_k for each content can be derived as the ratio between the number of users in the requesting state and the total number of users in area a. MEs also upload the moving speed at time t to the BSs for average speed estimation.

To derive the optimal caching policy $\psi_k^*(T(i))$ for every content k at each time slot $T(i)$, each BS traverses through an interval $\psi \in [0, 1]$ and selects the ψ with the minimum $J_k(\psi)$ value as $\psi^*(T(i))$. We define the *caching gap* as the difference between the optimum number of replicas and the current number of replicas

$$G_k = \max\{0, \psi_k^*(T(i)) - C(T(i))\}. \tag{11}$$

G_k indicates how much caching space is required to achieve the optimum at $T(i)$. According to G_k, the caching space $\mathcal{C}_k(T(i))$ for each content k is allocated according to the rule:

$$\mathcal{C}_k(T(i)) = \frac{\mathcal{C} G_k}{\sum\limits_{k \in \mathcal{K}} G_k}, \tag{12}$$

where $\mathcal{K}$ is the set of video content, and $\mathcal{C}$ is the available caching space in area a, given by

$$\mathcal{C} = \sum_{i \in I_a(T(i))} \left(S - \sum_{k \in \mathcal{K}} (\mathbb{C}_k(t) + S_k(t)) \right). \tag{13}$$

The BS broadcasts caching replicas for each content following Eqn. (13). The caching policy is described in **Algorithm 1**. Lines 4 to 6 evaluate the J_k values and make assignment to G_k. Finally, C_k is computed.

6. PERFORMANCE EVALUATION

We perform our simulations with MATLAB on a 4 GB RAM, Intel Xeon system. We consider a 2000 × 2000 m^2 network

[2] As $C(T_m)$ indicates the caching overhead, this formulation inherently considers the energy consumption for caching

```
Algorithm 1: Caching Allocation Alg. at $T(i)$
Input: 4-tuples of users in $I_a(T(i))$, alloc. cache
    space $S$ for each user, search step size $\omega$
Output: $\mathcal{C}_k(T(i))$
while $k \in \mathcal{K}$ do
    $J_k(\psi_{\max}) = 0$;
    while $\psi \leq 1$ do
        Calculate the $J_k(\psi)$; if $J_k(\psi) > J_k(\psi_{\max})$
         then
            $J_k(\psi_{\max}) \leftarrow J_k(\psi)$;
        $G_k \leftarrow \max\{0, J_k(t) - C(T(i))\}$;
    $\mathcal{C}_k \leftarrow \frac{CG_k}{\sum_{k \in \mathcal{K}} G_k}$;
```

area with eight 5G-NR (new radio) BSs deployed at arbitrary locations. The communication range of each BS is set to 500 m. We simulate a total of 300 MEs, each equipped with a 5G-D2D communication module. The D2D communication range is set to 150 m and bandwidth to 30 Mbps. We use 20 different video instances with 200 s inter-request times per user. The video segments are 2 s long with a bitrate of 4000 Kbps, resulting in a chunk size of 1 MB. The arrival rate of each video request follows a Poisson distribution with parameter λ randomly chosen between [2, 20]. When the requested video is determined, the corresponding segments will be consequently accessed by ME. The simulation time is set to 1000 s and 95% confidence intervals are evaluated. ME movements follow the Random Way Point (RWP) model that independently chooses a destination and moving speed within given ranges. We consider the performance metrics:

- AAL (*average access latency*): The time interval between sending the request and receiving the first packet of the requested content is defined as the access latency. The mean value of the access latency in the simulation at time t is considered as the AAL at t.
- CHR (*cache hit ratio*): The ratio between the total number of requests and the number of satisfied requests at time t are defined as the CHR at t.

We compare our Edge-Boost with the state-of-the-art Random Cache policy in [10].

6.1. Simulation Results

Variation of Caching Size: Figs. 3(a) and (b) show the AAL and CHR as a function of the caching size. Each data point in Fig. 3(a) and (b) represents the AAL at 1000 s and the overall average CHR during the simulation. The caching space of each ME ranges from 1% to 4%. A decreasing AAL trend with the increase of caching size is observed. This is justified by the fact that the probability of accessing content within one hop increases with the caching capacity. In addition, Edge-Boost out-performs the random strategy [10] with upto 25% lower AAL. According to Fig. 3(b), the CHRs of both methods increase with growing caching capacity. This can be attributed to the fact that a larger caching size yields a higher cache hit probability. Due to the accurate and timely estimation of content demand, Edge-Boost has a higher CHR than random caching [10] across the entire range of considered caching space. For instance, for a cache size of 4%, Edge-Boost achieves a 28% higher CHR than random caching.

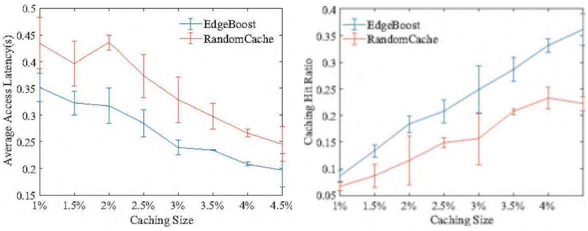

Fig. 3. Variation of caching size. (a) AAL vs. Caching Size (b) CHR vs. Caching Size

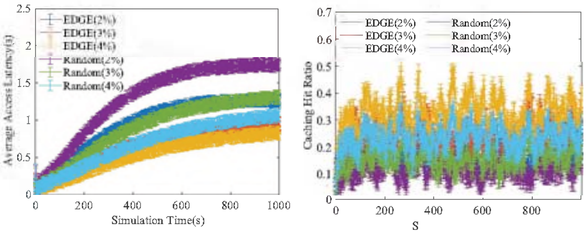

Fig. 4. Variation of simulation time. (a) AAL vs. Caching Size (b) CHR vs. Caching Size

Variation of Simulation Time: Figs. 4(a) and (b) show the AAL and CHR as a function of the simulation time for fixed caching spaces of 2, 3, and 4%. In Fig. 4(a), shows increasing AAL trends for both Edge-Boost and Random Cache. These ALL increases are due to the increasing scale of pending requests as the simulation time advances in the system of constant capacity. The overall Edge-Boost AAL is lower than that of state-of-the-art random caching [10]. For instance, the Edge-Boost curves corresponding to 3% and 4% caching space indicate a lower AAL than than the random caching curves for 4% caching space. The CHR fluctuates in every state during the simulation, which is mainly because of the dynamic request arrivals. In general, Edge-Boost achieves a higher CHR, mainly because Edge-Boost dynamically allocates the caching space for each content according to estimates of future demands.

Variation of Moving Velocity: Figs. 5(a) and (b) show the AAL and CHR of Edge-Boost and random caching as a function of the level of velocity. We consider five velocity range levels [1, 5] m/s, [5, 10] m/s, [10, 15] m/s, [15, 20] m/s, and

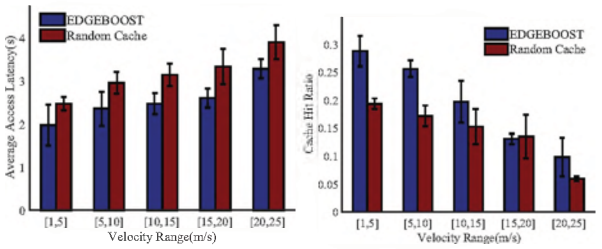

Fig. 5. Variation of velocity range. (a) AAL vs. Caching Size (b) CHR vs. Caching Size

[20, 25] m/s. As Fig. 5 (a) shows, the AAL increases with the increase in moving speed, which is mainly because of increased D2D link fluctuations caused by the increasing moving speed. As expected, we observe from Fig. 5(b) that the CHR decreases when the moving speeds of vehicles are accelerated. Further, Fig. 5 shows that Edge-Boost outperforms random caching for all velocity ranges. For example, for the moving speed [10, 15] m/s, Edge-Boost achieves about 40% higher CHR and 20% shorter AAL, respectively, than random caching [10].

7. CONCLUSION AND FUTURE WORK

We have studied the problem of optimally allocating caching in Mobile Device Clouds (MDCs). We have proposed the Edge-Boost framework for caching dynamics in MDCs. Further, a fluid-based model, which can provide timely estimates of caching demand constrained by the variation of content requests in mobile environment has been proposed. Additionally, an online caching optimization algorithm Edge-Boost based on fluid-based model is designed. Our evaluation results show that Edge-Boost achieves higher cache hit ratios (typically 20% higher) and shorter average access latencies (typically 15% shorter) than the state-of-the-art random caching strategy [10]. An important future research direction is to examine the energy constraints of the devices in MDCs. Moreover, future research should examine the MDC management of in hybrid SDN infrastructures that mix legacy non-SDN with SDN equipment [15].

8. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61871048 and 61872253.

9. REFERENCES

[1] V. Balasubramanian and A. Karmouch, "An infrastructure as a service for mobile ad-hoc cloud," in *Proc. IEEE Comp. Commun. Workshop and Conf. (CCWC)*, Jan 2017, pp. 1–7.

[2] C. Long, et al., "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE TMM*, vol. 20, no. 5, pp. 1126–1139, 2018.

[3] K. Habak, et al., "Workload management for dynamic mobile device clusters in edge femtoclouds," in *Proc. ACM/IEEE Symp. Edge Comp.*, 2017, pp. 6:1–6:14.

[4] X. Li, et al., "CaaS: Caching as a service for 5G networks," *IEEE Access*, vol. 5, pp. 5982–5993, 2017.

[5] Shantharama et al., "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing," *IEEE Access*, vol. 6, pp. 57545–57561, 2018.

[6] K. Zhang, et al., "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, 2018.

[7] D. Wu, et al., "Collaborative caching and matching for D2D content sharing," *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 43–49, 2018.

[8] Y. Li, M C. Gursoy, and S. Velipasalar, "A delay-aware caching algorithm for wireless D2D caching networks," *arXiv preprint arXiv:1704.01984*, 2017.

[9] G.S. Park, et al., "Smart base station-assisted partial-flow device-to-device offloading system for video streaming services," *IEEE Trans. Mob. Comp.*, vol. 16, no. 9, pp. 2639–2655, 2017.

[10] C. Xu, et al., "Optimal information centric caching in 5G device-to-device communications," *IEEE Trans. Mob. Comp.*, vol. 17, no. 9, pp. 2114–2126, Sep. 2018.

[11] Y. Li, et al., "Content-aware playout and packet scheduling for video streaming over wireless links," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 885–895, 2008.

[12] M. Reisslein and K. W. Ross, "High-performance prefetching protocols for VBR prerecorded video," *IEEE Network*, vol. 12, no. 6, pp. 46–55, 1998.

[13] J.-Y. Le Boudec, "On the stationary distribution of speed and location of random waypoint," *IEEE Trans. Mob. Comp.*, vol. 4, no. 4, pp. 404–405, 2005.

[14] T. L. Saaty, *Elements of Queueing Theory With Applications*, vol. 34203, McGraw-Hill New York, 1961.

[15] R. Amin, et al., "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surv. & Tut.*, vol. 20, no. 4, pp. 3259–3306, 2018.