Cooperative Caching and Video Characteristics in D2D Edge Networks

S. Sinem Kafıloğlu, Gürkan Gür and Fatih Alagöz

***Abstract*—Device-to-device (D2D) transmissions in wireless edge networks are promising for optimizing system-wide energy consumption and improving system service capacity. Cooperative content caching similarly serves efficiency goals for data-intensive applications in edge networks. In this work, we propose two cooperative cache replacement algorithms in D2D networks to support these techniques: *i)* distance-based *ii)* priority-class based. Video content dissemination in an edge network is our main use-case. In such content traffic, video characteristics have a significant impact on the system behavior. Therefore, we also investigate the effect of content scene change dynamics in our system. Distance based cooperation outperforms LRU, MIN-ACC and SXO in terms of goodput while priority-class based approach consumes less energy than MIN-ACC with almost the same consumption as LRU, especially under fast-changing scene regime. Besides, it is energy-wise slightly more rewarding than SXO in the fastest-changing scene regime.**

I. Introduction

Advanced multimedia services over mobile networks are leading to a tremendous surge in user traffic demand. Based on the Cisco predictions, mobile video will account for 79% of total mobile data traffic by the end of 2022 [1]. Besides, edge computing serves as one of the enablers for the realization of 5G requirements of low-latency and high spectrum efficiency. To realize this service setting, Multi-access Edge Computing (MEC) is standardized in 5G networks [2]. Such networks form a foundational element for wireless networks entailing heavy multimedia consumption. In that regard, various 5G vertical segments related to multimedia traffic such as Internet-of-Things (IoT), augmented reality and pervasive data sharing will be realized. D2D communications with edge computing infrastructure is a promising actuator towards that goal [3].

Caching is a broadly utilized performance amplifier for wireless network multimedia services [4]. It is also a facilitator tool for boosting performance improvement in D2D edge networks. In [4], the authors study edge caching with D2D offloading in terms of gain in transmission cost while Liu et. al. focus on the factors that determine the potential caching gain in D2D edge networks in [5]. For the caching techniques, cooperation adds another dimension for performance gains. In [6], Zhang et al. propose a mobility-aware cooperative edge caching scheme to reduce service latency. In [7], a cache replacement procedure is proposed where they consider cache contents that are also available in their cooperative group devices. These contents are evicted based on their request frequency over size values. Yin et. al. propose cooperative caching with i) *CacheData*, ii) *CachePath* [8]. In CacheData, nodes cache highly requested data along the transmission path, while CachePath caches the path data, particularly destination node, and only stores for devices closer to caching node. In our cache replacement procedure, we utilize contents with replicas in the reception range and their distance/availability. And also by applying $CPPC$ [9] for tie breaking, we utilize content popularity, layering and partitioning dimensions whereas they only use access frequency/order and content size In [10], Wu et al. propose a collaborative cache management in D2D networks where the cache controller receives neighbor cache information and decides according to the gap between popularity and caching proportion. They do not necessarily store a received content and focus on placement whereas we cache when a content is received and focus on the replacement problem from a variety of aspects.

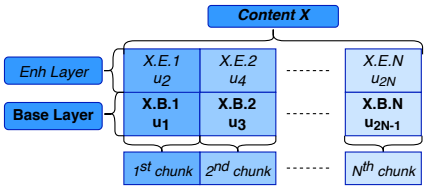


Fig. 1: An example layered content X ($N_l = 2$).

As cooperative caching is a promising technique to alleviate the multimedia traffic burden on wireless networks, we study multimedia caching in D2D networks from the cooperation aspect in this work. Our key contributions are: *i)* we propose two cooperative cache replacement algorithms based on the cache profile of neighbour devices in a D2D network. The first algorithm utilizes the prioritization of content unit availability in neighbours while the other one makes use of the neighbour proximity factor in determining the replacement units. *ii)* We integrate cooperative "lock-down" concept to counteract aggressive eviction in both techniques. *iii)* The impact of content scene change dynamics on layer-based content model is investigated. *iv)* We analyze the performance of cooperative caching techniques by investigating energy consumption and goodput through extensive simulations.

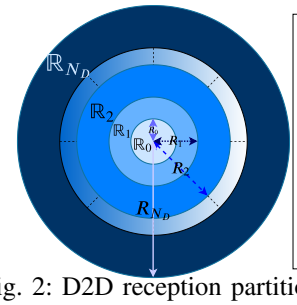
II. System Model

In our wireless edge network, the user devices are dispersed according to Poisson point process with density λ_D. Our main use case is multimedia content delivery in D2D networks. We utilize a content model where each content consists of N_l layers and N_p sized partitions. The utilization of content layering enables varying service qualities to different content consumers. The base layer is the essential core part of a content with its standard quality presentation. The other layers are added onto the base for high quality content rendering [9].The contents are not only divided in the layer domain but also in the time domain, into so-called *partitions*. According to [11], the initial partitions are more frequently requested by content

S. Sinem Kafıloğlu and Fatih Alagöz are with Bogazici University, Dept. of Computer Engineering, 34342, Istanbul, TR. e-mail:{sinem.kafiloglu, fatih.alagoz}@boun.edu.tr

Gürkan Gür is with Zurich University of Applied Sciences (ZHAW), Institute of Applied Information Technology, Switzerland. e-mail:gueu@zhaw.ch

This work was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under Grant 116E245.



- $\mathbb{R}_0$ is the inner circle area of radius R_0 with the requester at the origin.
- The ring area $\mathbb{R}_1$ is the exclusion of $\mathbb{R}_0$ from the circle area of radius R_1 with the requester at the origin.
- Continuing this way, we recursively define $\mathbb{R}_n := (\pi R_n^2) - \mathbb{R}_{n-1}$ with $R_{N_D} := R_{D2D}$.

Fig. 2: D2D reception partitions $\mathbb{R}_0$, $\mathbb{R}_1$,..., $\mathbb{R}_{N_D}$.

consumers. Besides, video content consumption requires large data retrieval and hence the partitioning is useful for flexible service and caching capabilities. We identify each content partition of some layer with a unique *content unit* id u_θ as in Fig. 1 with the example content X. The average base layer size s_b is 322.0 Mbits and enhancement layer size s_e is 152.1 Mbits [12]. The content request times are exponentially distributed of rate λ_u. For each user, the content request is generated based on the popularity following the Zipf distribution. For a content request, the partition requests are generated by the Weibull distribution. Proportional to the request probability p_{HQ} for high-quality video, the enhancement portions of requested content partition tuples are generated.

Users are capable of caching content units locally with cache capacity C. For content unit requests, the request manager first checks the local cache. In case of a local hit, no transmission occurs. Otherwise, the closest device in the reception range serves via D2D technique. The reception range $\mathbb{R}_{D2D}$ is the circle with radius R_{D2D} centered at the requester device. We assume the inter-user distances in a reception range are known by users via a signaling mechanism to discover neighbour devices [10] and cooperative calculation in this D2D network [13]. Thus, devices in the reception range of each other can exchange location data for determining distance between nodes. If there is not enough free space in the cache for the requested and received video content, the eviction takes place. The device cache replacement management is our *main use case* and thereof we specifically focus on device services and energy levels. For this purpose, we do not establish base station mode in our simulations. For D2D interference handling, no two D2D transmissions are allowed to operate simultaneously at a frequency in close proximity by meeting the following criteria: *i)* the new requester is at least R_{D2D} away from all active transmitters *ii)* the new transmitter is out of reception range of all active receivers. In the simulations, we consider D2D interference. At a frequency with simultaneous D2D transmissions, for each receiver device the received power strength of other transmitters directed for other receivers are summed as the corresponding interference. For each new D2D arrival, the interference of currently active transmitters to new receiver at that frequency is calculated and added dynamically till the D2D service completion by the interferer.

III. Cooperative Caching in the Edge Network

Both of our proposed techniques rely on the cooperation of devices within a vicinity of the requester. That cooperation is done for making eviction decisions based on the caching status of their neighboring devices in the reception range. In both techniques, the requester cache capacity C, the set of content units U in the requester cache and the newly retrieved unit u' are taken as system parameters.

Algorithm 1: $COOP_A$ **Algorithm**

INPUTS: ***U***: The list of content units available in the cache of device me,
u': Newly requested content unit by device me, ***C***: The cache capacity of device me

```
%select the set of units to be evicted
if (TotalSize(U) + size(u') > C) then
    sortedU_me ← sort("CPPC", U); %first sort on the CPPC for tie break %initially
    mark all units as out of reception range R_D2D
    sortedU_me ← [U, 0]; %priority-(N_D+1) class
    for (r=0:1:N_D) do
        for (i = 1:1:|U|) do
            if (u_i ∈ R_r) then
                %if u_i available in R_r pin as priority-r class member
                sortedU(i) ← [u_i, N_D + 1 − r];
            end
        end
    end
end
%sort units based on the priority class from priority-N_D (lowest) to priority-0 (highest) class
sortedU_me ← sortOnPriority(sortedU_me);
***
    %pin the highest eviction priority unit(last one) in sortedU_me
    w ← highestPriority(sortedU_me);
    %for all neighbours in R_D2D, select the devices that also have same highest prospective eviction
    priority unit w (for these devices in the very first new request w will be evicted next)
    wHolders ← selectNgh(R_D2D, w);
    %If this device is locked down for w eviction, decrease w's priority to the lowest and release
    lock down of w in me
    [lockDown, sortedU] ← LDUpdates(lockDown, w, me, sortedU);
    %start the eviction from the last unit (from higher priority) in sortedU_me and iterate until there
    is enough free space for u'
    sortedU_me ← cacheUpdate(sortedU_me, u');
    %if neighbours exist with w as the highest prospective eviction priority unit lock down these
    devices unless already locked down
    lockDown ← nghLDDecide(wHolders, lockDown, w);
    %multicast lockdown decisions to related neighbour devices in the reception range R_D2D for
    prospective requests
    multicast(wHolders, lockDown);
    %broadcast cache state updates to all devices in the reception range R_D2D for prospective requests
    broadcast(me, sortedU_me);
```

Algorithm 2: $COOP_D$ **Algorithm**

```
if (TotalSize(U) + size(u') > C) then
    sortedU_me ← sort("CPPC", U);%first sort on CPPC for tie break
    sortedU_me ← [U, Inf];%indicate all units as in-neighbour unavailable
    for (i = 1:1:|U|) do
        %get the set of devices that have u_i available in the reception range R_D2D of this u'
        requester device me
        curtxDevs ← getCandidateTxs(u_i);
        if (size(curtxDevs) > 0) then
            %among these devices having u_i choose the distance of the closest one to me
            dist ← closestDevDistance(curtxDevs);
            sortedU_me(i) ← [u_i, dist];
        end
    end
    %sort units based on the distance of the closest available neighbour starting from the distant
    (lowest priority) to closest (highest priority) one
    sortedU_me ← sortOnDistance(sortedU_me);
    ***
end
```

In $COOP_A$ in Algorithm 1, we construct priority-classes based on closeness to requesters and content unit availabilities in neighbours of the reception range. We assign each unit in the local cache to a priority-class. The requester device at the origin with radius R_{D2D} has reception range $\mathbb{R}_{D2D}$ that is partitioned into mutual exclusive areas $\{\mathbb{R}_0, \mathbb{R}_1, ..., \mathbb{R}_{N_D}\}$ (Fig. 2). We define the radius of each D2D reception partition by $R_n := R_{n-1} + R_0$, $R_0 := \frac{R_{N_D}}{N_D+1}$. For each area, we define a priority-class regarding eviction. Content units in the requester cache that are also available in areas closer to the requester have greater eviction priority. Content units already available in at least one neighbour in area $\mathbb{R}_0$ are *priority-0* class members with the *highest eviction priority*. The units already available in at least one neighbour in $\mathbb{R}_n$ are members of *priority-n* class having lower priority than priority-(n-1) class. The priority-(N_D+1) class is for the units out of D2D reception range $\mathbb{R}_{D2D}$ with the *lowest priority*. After the priority labeling, units are priority-sorted (lowest to highest) in $sortedU_{me}$.

For requests to devices in the reception range of each other

with the same unit w of the highest eviction priority, they will all evict that unit and erase it from the neighbourhood, which may lead to a caching performance degradation due to this bandwagon effect. To counteract this erasure problem, the first content requester device signals others to "lock down" w and then evict it. In that regard, aggressive eviction is postponed by cooperation and via this "lock-down" concept, we allow w to survive in the network longer. This action is realized in $selectNgh$ and $LDUpdates$ functions in Algorithm 1.

In function $cacheUpdate$, we start the eviction with high priority labeled units. In our range partitioning scheme, we label close vicinity area with a great eviction priority. Hence, we first evict already in-close neighbour available units. Thereof, prospective requests for such units at the same requester can be handled in D2D mode in a short range. Our scheme has another dimension for the service capacity improvement. It assigns lowest priority for local units unavailable in reception range and this leads to a greater probability to preserve such units in the cache and in the future serve them locally. When a prospective request for these units from some other device occurs, they are transmitted within the reception range via D2D with a great probability as well. Besides, by removing in-neighbour available units from the local, we open up more space for larger number of different units and this impacts the total system performance. For units in the same priority class, we utilize our previous proposal $CPPC$ [9] to break ties.

After $cacheUpdate$, by function $nghLDDecide$ we lock down in-range neighbors that have w as the next highest priority eviction candidate to block w eviction and allow w to survive longer. Next, these $lockDown$ decisions are multicast to those related neighbors for cooperation. We also broadcast cache state updates to neighbor devices in the reception range.

In the other proposed algorithm $COOP_D$ shown in Algorithm 2, the content units that are available in some neighbour of the reception range are sorted based on the distance from the closest available neighbour. These units are sorted from the closest to the distant one in $sortOnDistance$. When we consider $cacheUpdate$, the closest available content unit has the highest priority for the eviction and the priority descends with increasing distance. By prioritizing the closest in-neighbour available unit for eviction, we act to improve the D2D performance of our system. Higher priority is assigned to the unit which is available in a nearby device. When it is evicted from the local cache, the prospective D2D transmission of the request for the same unit will last for short and hence both the D2D energy consumption and service capacity will have gains compared to distance-unaware cooperative techniques. The units with the same distance have the same priority. To break ties, we utilize $CPPC$ again. In this algorithm, the lock-down cooperation mechanism and utilized functions are the same with the $COOP_A$ (marked as ***).

IV. Performance Metrics

The investigation of our caching proposals are basically focusing on *(i) energy consumption* and *(ii) goodput* with the given system parameters list in Table I [9].

Energy: The content unit local hits constitute energy consumption component E_{loc}. To calculate it, we sum the local energy consumption ($P_{loc} \cdot \frac{|s_u|}{C_{loc}}$) of all locally served requests ($r_u \in S_{(n,n)}$) as $E_{loc} := \sum_{u\in U} \sum_{n\in N} \sum_{r_u \in S_{(n,n)}} P_{loc} \cdot \frac{|s_u|}{C_{loc}}$.

The total D2D mode energy consumption E_{D2D} is another system energy consumption component consisting of transmission and reception energies. For D2D transmission of content unit u between devices n and m, we sum the reception energy $\frac{P^{rec}_{D2D}\cdot|s_u|}{C_D^{(n,m)}}$ and transmission energy $\frac{P^{tx}_{D2D}\cdot|s_u|}{C_D^{(n,m)}}$. We sum for content units transferred in D2D mode from all device pairs to get E_{D2D} as $\sum_{u\in U} \sum_{\substack{n,m\in N \\ n\neq m}} \sum_{r_u \in S_{(n,m)}} \frac{P^{tx}_{D2D}\cdot|s_u|}{C_D^{(n,m)}} + \frac{P^{rec}_{D2D}\cdot|s_u|}{C_D^{(n,m)}}$.

Some unit requests are blocked due to cache or network capacity limitations. A requester wakes up to idling state with $E_b(s_u)$ activation energy without successful transmission. E_b:=$\sum_{u\in U}\sum_{n\in N} E_b(s_u)$ is the total blocking energy for all such devices with unsuccessful unit services and the total system energy consumption E_{all} is equal to E_{loc}+E_{D2D}+E_b.

Goodput: The total number of locally served content bits is defined as G_{loc}:=$\{\sum_{u\in U} \sum_{n\in N} \sum_{r_u \in (Comp\cap S_{(n,n)})} |s_u|\}$. If a content unit request $r_u \notin Comp$, it has incomplete base chunk services leading to incomplete service and not contributing to local goodput. Therefore, we count for r_u's in the set $Comp$. G_{D2D}:=$\{\sum_{u\in U} \sum_{\substack{n,m\in N \\ n\neq m}} \sum_{\substack{r_u \in (Comp\cap S_{(n,m)}) \\ r_u \in \overline{Fail}}} |s_u|\}$ is the total goodput of D2D operations. By the same explanation above, a request r_u should be in $Comp$ for D2D goodput contribution. Such an r_u has all base chunks completely received but some enhancement chunk reception might fail meaning $r_u \in Fail$ and hence no D2D goodput contribution. In that regard, we count for no failure requests as well ($r_u \in \overline{Fail}$). The summation of local hit and D2D mode contributions gives the network goodput G_{all}:=$(G_{loc}$+$G_{D2D})/T_{sim}$.

In our D2D network, our motivation is to analyze our cooperative caching mechanisms and the impact of scene change dynamics for E_{all} and G_{all}. We compare our $COOP_D$, $COOP_A$ algorithms to LRU, $CPPC$[9], MIN-ACC[14], SXO[8]. We implemented our event-based simulator in MATLAB R2020 on a computer with Intel i7-8550U CPU@1.8GHz, 16GB RAM. In each case, we run simulations 10 times and average their results. The simulator processes incoming content unit requests (arrivals) and service completions. The content request rate is λ_u=8$\frac{user}{sec}$. Section II describes the request management. Each D2D service has B bandwidth for transmission with $N_{f_{ter}}$ frequencies in total. In service completions, frequencies are preempted. The system and simulation parameters are listed in Table I.

V. Performance Evaluation

A. Scene Change Dynamics

The dynamicity of multimedia has an impact on characteristics of video chunks. One such factor is the scene change (SC) dynamics. Some contents are more stable with small inter-frame changes. Others have rapid and evident changes like action movies. To investigate this factor, we utilize 30 fps temporal scalable video traces where I and P frames form the base layer and B frames form the enhancement in the temporal scalable encoding[12]. We use *terse traces* around 30 minutes of *Tokyo Olympics, The Silence of the Lambs, Star WarsIV* and *NBC News* videos of GoP size 16 with 3 consecutive B frames and quantizer 16. For content classification, *spatial*

TABLE I: System and simulation parameters.

Par.	Value	Explanation
C_{loc}	-	The service capacity of content
$C_D^{(n,m)}$	-	The channel capacity between the n^{th} and m^{th} devices
E_b	-	The activation energy of devices from the sleeping to the idling state
N	-	The total number of devices
U	-	The set of content units uniquely identifiable by content, chunk, and layer id
r_u	-	The request for the content unit u
s_u	-	The size of the content unit u
$S_{(n,m)}$	-	The set of services from the n^{th} device to m^{th} device
$Comp$	-	The set of requests for a content where all the base chunks are transmitted successfully (service completed successfully)
$Fail$	-	The set of requests for content units that have failed
T_{sim}	1800 s	The total simulation duration
N_l	2	The number of content layers
N_p	23.56 Mbits	The average partition size
N_D	1	The number of mutual exclusive area portions of the D2D reception range
λ_D	$1.46e^{-3}\frac{dev}{m^2}$	The mean density of device distribution in PPP
s	1	The skewness parameter of Zipf distribution
α	1	The Weibull distribution scale parameter
k	0.6	The Weibull distribution shape parameter
p_{HQ}	1	The high quality request probability
C	47.1 Mbits	The cache capacity of devices
R_{D2D}	120 m	The reception range radius of a requester
s_b	322.0 Mbits	The average base layer size
s_e	152.1 Mbits	The average enhancement layer size
P_{loc}	40 mW	The power consumption of local content unit retrieval
P_{D2D}^{tx}	80 mW	The transmission power consumption of a device
P_{D2D}^{rec}	16 mW	The reception power consumption of a device
N_c	250	The number of contents
N_{fter}	4	The total number of system frequencies
B	2 MHz	The operation bandwidth of each frequency
r_e	1	The interplay multiplier for changing average enhancement layer size

and/or *temporal* features are utilized and the frame difference variance is considered to project temporal multimedia dynamicity[15]. For a content c of the total number of frames F_c, the size of a given frame f $s_{(f)}$, the mean frame size difference is $M(c)$:=$\frac{1}{F_c-1}\sum_{f=1}^{F_c-1} s_{(f+1)}$-$s_{(f)}$. We calculate the variance of the frame size differences of aforementioned videos $Var(c)$:=$\frac{1}{F_c-1}\sum_{f=1}^{F_c-1}([s_{(f+1)}$-$s_{(f)}]$-$M(c))^2$.

We look at the ratio of the base layer size over the enhancement size for a content c, $R_c^{\frac{B}{E}}$:=$\frac{\sum_{b\in\mathbb{B}_c}(s_b)}{\sum_{i\in\mathbb{I}_c}(s_i)+\sum_{p\in\mathbb{P}_c}(s_p)}$ with $\mathbb{B}_c$,$\mathbb{I}_c$,$\mathbb{P}_c$ as the B-, I-, P-frame sets, respectively. We get the statistical information about that ratio by calculating $R_c^{\frac{B}{E}}$'s for the aforementioned videos with the provided video statistics in[16]. The correlation between the variance of frame size differences and ratio of layer sizes is observed by the Pearson correlation coefficient ρ=0.992 showing a high linear dependency. Thus, we can utilize the layer size ratios to analyze temporal SC dynamics. The base layer size is kept fixed to have the same SQ content compositions. For the SC adaptation, we change the enhancement size by s'_e:=s_e·r_e. In temporal content characteristics, the high variance of frame size differences are classified as high temporal activity [15]. As the layer size ratio and mentioned variance are highly linearly correlated, the enhancement layer with larger size for a fixed base size implies the content has higher temporal changing dynamic. In contrast, having smaller enhancement size reveals the content is more stable with little temporal variation.

B. Experimental Results

With increasing r_e the total system energy consumption increases for all techniques (Fig. 3). Intuitively, with increasing r_e the enhancement size and subsequently the total content size rises. Hence, the system requires greater energy consumption even for a local hit. Besides, the probability of local availability decreases and hence the content has to be transmitted in D2D mode and this requires larger energy than a local hit.

$CPPC$ outperforms other algorithms in terms of E_{all} (Fig. 3). As it utilizes content characteristics, the improvement over LRU (MIN-ACC) becomes more evident with larger enhancement content sizes. CPPC also outperforms SXO and

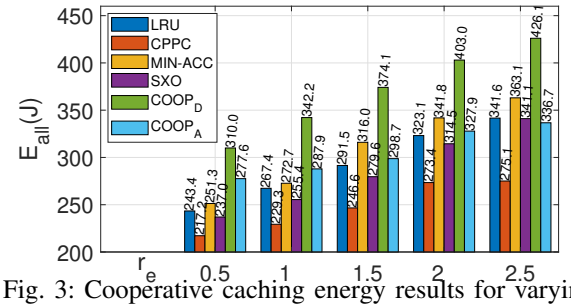


Fig. 3: Cooperative caching energy results for varying r_e.

the improvement rate rises up to 19.3% at r_e=2.5. SXO prioritizes large popular units for eviction, which are base units. Their prioritized eviction leads to more D2D transmissions instead of local hits and thereby more energy consumption compared to $CPPC$. When we compare $COOP_D$ to LRU (MIN-ACC), its consumption falls back for all r_e's in the inspection domain due to its prioritization nature of the replacement unit selection. It evicts in-neighbour available units based on closeness prioritization. The more popular a content unit is, more likely to have it in a nearby device and then selected for eviction. The prospective requests for them can be served in D2D mode instead of local hits. Thus, it has higher energy consumption than LRU (MIN-ACC) ranging from 24.7% to 28.3% (17.4% to 25.5%). $COOP_D$ also falls back from SXO for all r_e's in the inspection domain. As explained above in $COOP_D$ popular units are given a greater priority for eviction and requests for them can be served by D2D transmission rather than locally. SXO considers the local device cache and selects highly popular large units that are of type base for eviction. SXO will also cover up for the requests of these units via D2D mechanism but $COOP_D$ will also require D2D transmission for popular enhancement units available in a close vicinity and thereof consuming more energy than SXO ranging from 24.9% to 34.0%. $COOP_A$ has also higher results compared to LRU (SXO) except for the largest r_e=2.5. When we compare to MIN-ACC, $COOP_A$ achieves better energy performance for $r_e \geq 1.5$. The highest improvement is observed from 363.1 to 336.7 J at r_e=2.5.

Next, we compare $COOP_A$ and $COOP_D$ to $CPPC$. In both cooperative techniques, in-neighbour available units are to be evicted first and such units have higher popularity compared to ones not available in neighbors. When prospective requests for such more popular units occur again, this time they are not found locally but served in D2D mode. However, in $CPPC$ such units are not necessarily evicted and those requests are handled locally. Thus, cooperative techniques result in an energy consumption increase in contrast to $CPPC$.

In Fig. 3, we also observe that $COOP_A$ has better energy performance than distance based $COOP_D$ for all r_e's. This improvement reaches up to 21.0% from 426.1 to 336.7 J at r_e=2.5. In distance based cooperative cache replacement technique, local content units are sorted based on the distance from the closest unit-storing neighbour. The more popular a unit is, the higher probability it is found in a close device and more popular units are assigned higher eviction priority. Prospective requests for such units can be handled via D2D links instead of low-cost local hits. However, in $COOP_A$

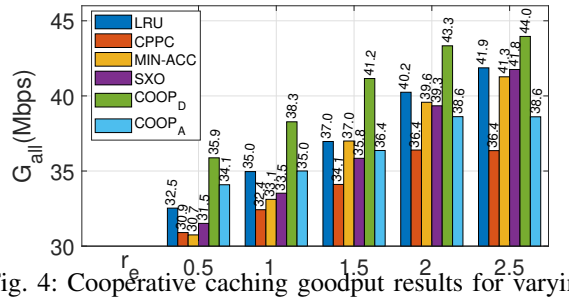


Fig. 4: Cooperative caching goodput results for varying r_e.

with N_D=1 unit availability in a neighbour is sufficient to be selected for eviction. Among these available units, $CPPC$ selects the least popular as the prioritized eviction candidate. Thereof, $COOP_A$ preserves popular units among in-neighbour units and prospective requests for them can be low energy cost local hits. This explains the effective improved performance of $COOP_A$ over distance based $COOP_D$.

The goodput investigation results are presented in Fig. 4. With increasing r_e, the requested content size rises and the system serves with larger capacity in all mechanisms. For all r_e's in the investigation range, $COOP_D$ has better goodput values than LRU (MIN-ACC) with improvement reaching to 11.3% at r_e=1.5 (16.7% at r_e=0.5). The erasure of high popular content units that are available in the neighbourhood opens up caching opportunity to new units and increasing the unit diversity. Thus, more requests can be served and the rise in overall goodput G_{all} is observed. $COOP_D$ also has improved goodput over SXO for all r_e's in the investigated range. This improvement reaches to 14.8% at r_e=1.5. $COOP_D$ gives high eviction priority to popular units in all types. On the contrary, SXO gives high eviction priority to popular base units compared to enhancements with smaller size. The low rate of highly popular enhancement unit eviction in SXO results in less vacancy for new units and lower unit diversity compared to $COOP_D$. $COOP_A$ has lower goodput values than LRU for $r_e >$1.0. At r_e=0.5, this cooperative technique becomes better than LRU and 4.8% improvement from 32.5 to 34.1 Mbps is observed in Fig. 4. $COOP_A$ has lower goodput values than MIN-ACC (SXO) for $r_e >$1.0 ($r_e >$1.5). With decreasing r_e, this cooperative technique performs better than MIN-ACC (SXO). At r_e=0.5, its improvement reaches to 10.9% (8.2%) from 30.7 to 34.1 (31.5 to 34.1) Mbps (Fig. 4).

The cooperative approaches have improved goodput than $CPPC$ for all r_e's (in Fig. 4). The improvement rate of $COOP_D$ ($COOP_A$) over $CPPC$ reaches to 20.9% from 36.4 to 44.0 Mbps at r_e=2.5 (10.3% from 30.9 to 34.1 Mbps at r_e=0.5). Cooperative techniques open up caching opportunity to new units during replacement leading to higher service capacity than $CPPC$. When we compare $COOP_D$ to $COOP_A$, $COOP_D$ has better goodput for all r_e's. Distance based technique opens up more space to new units as it starts eviction from units commonly found in neighbors while non-distance based starts eviction from units available in vicinity but not so popular due to $CPPC$ based sorting. Thus, an in-neighbour unpopular unit is a direct replacement candidate and if the neighbour also similarly evicts it before a prospective request, then content unit diversity is not preserved as in $COOP_D$ and such requests are not served. This results in lower system capacity than $COOP_D$. At r_e=0.5, the improvement of $COOP_D$ over $COOP_A$ is 5.3%. The improvement rises up to 13.9% at r_e=2.5 from 38.6 to 44.0 Mbps.

VI. Conclusion

In this work, we investigated D2D networks from caching aspect. We proposed two cooperative cache replacement techniques and performed simulations for varying scene change dynamics in terms of energy consumption and goodput. According to the simulation results, our distance based cooperative cache replacement technique outperforms LRU, MIN-ACC, SXO and $CPPC$ in terms of system goodput for all scene change dynamics in the investigated regime. However, it has an increase in the energy consumption. To alleviate this energy problem, our other proposal $COOP_A$ can be utilized that has much the same energy consumption as LRU (less energy consumption than MIN-ACC) for large r_e case (the regime where the content has rapidly changing characteristic). $COOP_A$ consumes only at most 6.8% higher energy than SXO for 1.0$< r_e \leq$2.0 and it also performs slightly more energy rewarding than SXO for the largest r_e=2.5.

References

[1] C. W. Paper, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022 white paper," Feb. 2019.

[2] ETSI White Paper, "MEC in 5G networks," June 2018.

[3] V. Balasubramanian, M. Wang, M. Reisslein, and C. Xu, "Edge-boost: Enhancing multimedia delivery with mobile edge caching in 5G-D2D networks," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, July 2019, pp. 1684–1689.

[4] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6399–6410, 2017.

[5] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[6] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 80–87, June 2018.

[7] S. Ghandeharizadeh and S. Shayandeh, "Cooperative caching techniques for continuous media in wireless home networks," in *Proc. of the 1st Int. Conference on Ambient Media and Systems*, ser. Ambi-Sys '08, 2008.

[8] Liangzhong Yin and Guohong Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 1, pp. 77–89, 2006.

[9] S. S. Kafiloğlu, G. Gür, and F. Alagöz, "Multidimensional content modeling and caching in D2D edge networks," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2019, pp. 1–6.

[10] D. Wu, L. Zhou, Y. Cai, and Y. Qian, "Collaborative caching and matching for D2D content sharing," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 43–49, 2018.

[11] S. Lim, Y. Ko, G. Jung, J. Kim, and M. Jang, "Inter-chunk popularity-based edge-first caching in content-centric networking," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1331–1334, Aug. 2014.

[12] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 6, no. 3, pp. 58–78, Third 2004.

[13] P. Zhang, J. Lu, Y. Wang, and Q. Wang, "Cooperative localization in 5G networks: A survey," *ICT Express*, vol. 3, no. 1, pp. 27 – 32, 2017.

[14] S. Jin and L. Wang, "Content and service replication strategies in multi-hop wireless mesh networks," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'05)*. ACM, 2005, p. 79–86.

[15] A. Khan, L. Sun, E. Ifeachor, J. O. Fajardo, F. Liberal, and H. Koumaras, "Video quality prediction models based on video content dynamics for H.264 video over UMTS networks," *International Journal of Digital Multimedia Broadcasting*, vol. 2010, 04 2010.

[16] "Trace files and statistics: H.264/SVC video trace library." [Online]. Available: http://trace.eas.asu.edu/h264svc/