On the Caching Policy and Cooperation Distance Design in Base Station Assisted Wireless D2D Networks

Ming-Chun Lee, Student Member, IEEE, and Andreas F. Molisch, Fellow, IEEE
Ming Hsieh Department of Electrical Engineering
University of Southern California
Los Angeles, CA, USA
Email: mingchul@usc.edu, molisch@usc.edu

Abstract—This work investigates the caching policy and cooperative distance designs for throughput and energy efficiency (EE) in base station (BS) assisted device-to-device (D2D) caching networks. To conduct the investigation in joint consideration of BS-, D2D-, and self-caching and the impact of cooperation distance, we configure a clustering network with specifically designed power control and resource reuse policies. After analyzing the throughput and EE of the network, their design problems and solving approaches are provided. By simulations, we validate our analyses and evaluate the proposed designs. Moreover, we show that the throughput and EE designs can significantly conflict with each other, and a trade-off design that provides a compromise between them is thus necessary for improving the system.

I. INTRODUCTION

Demand for video delivery services has dramatically increased in recent years and is expected to continue to grow [1]. One of the most promising approaches to resolve the resulting challenge to network throughput is caching at the wireless edge [2], [3]. In contrast to conventional approaches for throughput increase, such as network densification or use of more spectrum, wireless caching leverages the unique traffic characteristics, namely asynchronous reuse of video content, and cheap storage, to improve the system capacity [2], [3].

A. Literature Review

Wireless caching has been investigated in various scenarios. Femtocaching was first proposed based on low-cost helper nodes with no or limited backhaul [4] and has been generalized to heterogeneous networks in [5], [6]. The combinations of femtocaching and other techniques, such as multiple-input multiple-output techniques and coded multicast, have also been widely explored [7], [8]. Self-caching is another approach that naturally leverages existing storage resource in devices [11], [12]. With device-to-device (D2D) communications becoming widely available [10], wireless D2D caching networks have been widely explored [11]–[17]. The theoretical scaling laws of D2D caching networks were in [14]–[16]. In consideration of user mobility, caching designs with uncertainty were investigated in [18]. Recent research has started to consider heterogeneous user preference modeling in designing caching networks [19], [20]. We note that, while there are many papers investigating self-caching and D2D caching, their interaction has not been well explored. Furthermore, although self-caching could be influential to the system [11], [12], its impact was occasionally overlooked [21], [22].

Caching policies in wireless D2D caching networks have been designed in pursuit of different objectives: cache hit rate, i.e., successful access probability or outage [12], download delay [11], throughput [15], [21], and energy efficiency (EE) [22]. However, different objectives generally conflict with one another and have their own disadvantages. When using cache hit rate, the designs aim to maximize the probability that a user can reach the desired file through D2D communications, while ignoring the potential help from BS. Also, an optimal hit rate does not actually mean that the system throughput can indeed be optimal [12]. Similarly, when considering network throughput, delay and EE of users then could not be guaranteed. On the other hand, when focusing on optimizing delay or EE, the network throughput might spontaneously be sacrificed. As a result, in order to improve the design of the system, it is necessary to comprehensively explore the trade-offs between different objectives.

The impact of cooperation distance on the D2D networks has been discussed in [11], [15], [21], [22]. Generally speaking, a larger cooperation distance can provide better caching cooperation between users, i.e., the user can have a higher chance to obtain the desired content via D2D links, while on the downside it leads to the higher power consumption and lower frequency reuse gain [11], [21]. This trade-off motivates the interests in exploring the effect of cooperation distance. In [11], the effect of cooperation distance on the average delay was studied for both deterministic and random caching schemes. In [21], the influence of cooperation distance was considered, and it characterized the optimal throughput–outage trade-off under given assumptions. Apart from [21], throughput–outage trade-offs of different caching schemes were compared in [15]. It demonstrates that, by well designing the cooperation distance, the simple decentralized caching policy can achieve a near-optimal throughput scaling. The relationship of offloading gain and energy consumption was investigated in [22], and, via numerical simulations, it pro-
B. Main Contribution

From the literature review, it can be concluded that a comprehensive understanding of different optimal designs and their trade-offs is necessary. In addition, the investigation of designs that provide the best compromise between different objectives is still far from providing conclusive results. In this paper, we aim to address the following issues of previous literature: (i) lack of joint design of caching policy and cooperation distance in throughput and EE aspects; (ii) relying on numerical results and/or simplified models; (iii) misinterpretation or disregard of the interactions of BS-, D2D-, and self-caching. We note that the trade-off design between the throughput and EE will be investigated in our future work [23].

In this work, a BS-assisted wireless D2D caching network is considered. We focus on finding optimal caching policy and cooperation distance designs in terms of network throughput and EE, respectively. Specifically, our contributions are:

- We establish a network configuration that is able to provide the flexibility in investigating different types of optimal designs with joint consideration of interactions between BS-, D2D-, and self-caching and the effects of cooperation distance and interference between D2D communications.
- We conduct the network throughput analyses of two closely-related schedulers, i.e., the pull-based and push-based scheduler, and propose a throughput optimization design. We note that, although the pull-based scheduler is optimal for the cluster network, it suffers from a complicated formulation that is intractable for further optimization. In contrast, the push-based throughput provides an easily tractable convex form and can serve as an effective lower bound for the pull-based throughput. Thus, it is used for optimization.
- We formulate the EE of the typical user and propose the optimal EE design. Note that, by analysis, we prove that the EE optimization problem can be cast into a quasi-concave program, which is solvable by converting to sequential concave problems.
- We use computer simulations to evaluate the proposed designs and analyses. Also, we numerically validate that the self-caching effect is influential and should not be ignored. Finally, by comparing between the throughput and EE designs, we demonstrate the necessity of proposing their trade-off design.

The remainder of the paper is organized as follows. In Sec. II, the adopted network and system models are introduced. In Sec. III, throughput analysis and the corresponding optimization are provided. The EE formulation and its optimization are proposed in Sec. IV. Numerical results are provided in Sec. V. Conclusions are provided at the end of this paper.

II. CONTENT CACHING AND SYSTEM MODELS OF BASE STATION ASSISTED WIRELESS D2D CACHING NETWORKS

A. Network and System Models

In this work, a BS-assisted cache-enabled wireless D2D network is considered, and the clustering presented in [11], [21] is adopted. In the network, a square cell with side length \(D\) is served by a BS and is split into several equal-sized square clusters with side length \(d\), where D2D communication is allowed between two devices within the same cluster. Then the number of clusters in a cell is

\[ N = \frac{D^2}{d^2}, \]

where \(d\) is denoted as the cluster size. With slight loss of practicality, a fractional number of clusters is allowed for mathematical tractability and simplicity. We consider two non-overlapping frequency bands for establishing BS communications and D2D communications, respectively. For communications between the BS and the devices, the time-frequency resources of the BS band are shared by all clusters via using an orthogonal multiple access approach, such as FDMA. For simplicity, we equally split the BS resources between different clusters leading to a transmission rate of the BS links inversely proportional to \(N\) — in other words, the set of users within a cluster gets \(1/N\) of the available resources.\(^1\)

D2D communications are considered only between users within the same cluster. Consequently, we call (with slight abuse of definition) \(d\) also the cooperation distance; in fact, "cluster size" and "cooperation distance" will be used interchangeably throughout this paper. Resources for D2D communications are spatially reused between the clusters. Such a reuse scheme evenly applies \(K\) colors to the clusters, and only the clusters with the same color can be active on the same time-frequency resource for D2D communications. Note that the adopted reuse scheme is analogous to the spatial reuse scheme in conventional cellular networks, and \(K\) is the reuse factor. Since the resources of both the BS and D2D communications are shared in a cluster-based manner, we implicitly indicate that only a single user in a cluster can be activated to use the BS and D2D resources belonging to the cluster at the same time, i.e., only one user in a cluster is allowed to communicate through BS link at a time and similarly for the D2D link.

In this work, we adopt a simplified channel model in which only the path-loss effect is considered for mathematical tractability. The path-loss model is

\[ 20 \log_{10} \frac{4\pi d_0}{\lambda_c} + 10\alpha \log_{10} \left( \frac{r}{d_0} \right) \quad \text{[dB]}, \]

where \(r\) is the distance between transmitter and receiver, \(\lambda_c\) is the wavelength of the carrier frequency, \(\alpha\) is the pathloss exponent, \(d_0\) is the break point distance. To restrict the interference between different clusters, a power control policy is adopted such that

\[ E_D = \left[ \sqrt{2}(\sqrt{\nu} - 1) \frac{d}{d_0} \right]^\alpha \cdot \frac{A}{\lambda_c} \cdot \frac{d_0^2}{\lambda_c^2} \cdot \nu, \]

\(^1\)Of course, the BS resources per user do not scale with the cluster size; but as we will below analyze the throughput per cluster, our above formulation is more convenient.
where $E_D$ is the transmission power for D2D communications and $\nu$ is the maximum allowable interference between two clusters using the same resource. By fixing $\nu$ to be sufficiently small, interference can be effectively avoided. Besides, by adopting the proposed policy, in which $E_D$ scales with the cluster size, the average received power of users in the cluster can be maintained even if the cluster size is adjusted for optimization purposes; this also is the reason for the later assumption that the throughput for a user employing the D2D link is invariant with respect to the change of cluster size. Note that this power control policy depends only on system parameters, and no attempt is made to adapt it to the channel states/distances between transmitters and receivers. Hence, given the system parameters, the transmission powers of all D2D links in the network are identical. Also note that, since the interference is avoided when $\nu$ is sufficiently small, the interference between clusters will be ignored in the remainder of the paper.

In this work, users can obtain the desired content via their own caches, D2D communications, or BS communications with different transmission qualities and costs. We denote the throughput for a user to access the content via a BS link as $T_B$; the throughput for a user to access the content using a D2D link as $T_D$; the throughput for a user to access the content from its own cache as $T_S$; and consider $T_S \geq T_D > T_B$. Note that we generally assume $T_D$ and $T_S$ to be invariant with respect to the cluster size $d$ and $T_B \propto \frac{1}{\nu}$, and these assumptions are reasonable when the power control policy in (2) is adopted, and the BS resources are evenly split. Furthermore, we assume that the throughput of the user is independent of the actual distance between the transmitter and receiver, and this is practical when we have only one modulation-and-coding scheme format. Similar to the throughput case, we denote the power consumption for a user to access the content using a BS link as $E_B$; the power consumption for a user to access the content using a D2D link is by definition $E_D$ in (2); we consider only $E_B > E_D$. Zero power consumption is assumed if the user can access the desired content from its own cache. Note that we assume that the BS is equipped with an unlimited backhaul connected to repositories containing all contents in the library. Thus, the request from a user can always be satisfied if the BS link is available for that user.

In this paper, we consider only active users, denoted as the type of users who place requests that need to be satisfied and participate in the D2D cooperation (i.e., send files to other users that request them), and consider the active users to be independently distributed according to homogeneous point Poisson processes (HPPPs) with user density $\lambda_u$. In our future work [23], we will extend to include another type of users, inactive users, who do not place requests of their own but still participate in the D2D cooperation, and all results in this paper will be extended.

The library consists of $M$ files with all files having the same size. Each user is assumed to be able to cache $S$ files in the device. A random caching policy [11], [12] is used by users, and all users adopt the same caching policy. Denoting $b_m$ as the probability for the user to cache file $m$, the caching policy is expressed as $\{b_m\}_1^M$, where $\sum_{m=1}^M b_m = S \leq M$. All users follow the identical request probability distribution. The request probability of a user for file $m$, i.e., the probability that a user wants file $m$ in the future is denoted as $a_m$ with $0 \leq a_m \leq 1, \forall m$, and $\sum_{m=1}^M a_m = 1$.

B. Elementary Access Probability Analysis

Here the elementary access probabilities of using different approaches are analyzed. The results will serve as the foundation for acquiring further results in the subsequent sections.

Consider the caching policy $\{b_m\}_1^M$. The self-access probability of a user is defined as the probability that the desired file of the user can be found in its own cache:

$$P_S = \sum_{m=1}^M a_m b_m. \quad (3)$$

Then considering there are $k$ users in a cluster, the probability that a user cannot find the desired content through self-cache and D2D communications is

$$P_{B,k} = \left[\sum_{m=1}^M a_m (1 - b_m)^k\right], \quad (4)$$

where $(1 - b_m)^k$ is the probability that file $m$ is not in the caches of users of the cluster, and therefore $a_m (1 - b_m)^k$ is the probability that the user wants file $m$ but file $m$ is not in the caches of users of the cluster. Finally, when both BS and D2D links are available for the user, the probability that the user obtains the desired file via the D2D link is

$$P_{D,k} = 1 - P_{B,k} - P_S = 1 - \sum_{m=1}^M a_m (1 - b_m)^k - \sum_{m=1}^M a_m b_m. \quad (5)$$

III. Caching Policy and Cooperation Distance Design for Throughput Optimization

In this section, the caching policy and cooperation distance design is investigated for the goal of optimizing network throughput. We first analyze the network throughput considering two different schedulers, i.e., the pull-based and push-based schedulers. Then the throughput optimization design is proposed. The reason for considering two schedulers is to gain mathematical tractability. In fact, we will later show that the throughput of the push-based system can serve as an effective lower bound for the throughput of the pull-based system while it benefits from better mathematical tractability. Note that, due to page limitation, we will omit most of the derivations and proofs and simply provide the final results in this paper. For the details please refer to our future work [23].

The pull-based system functions as follows. In a cluster, all the users first check whether their desired contents can be found from their own cache. If yes, the users’ requests are satisfied and they remain online to provide potential D2D communications to other users; otherwise, the users send requests to the BS to ask for service. Once the BS receives requests from users, the BS randomly chooses one to serve. If
the chosen user can find the desired content from other users, the user accesses the desired content by D2D communications; otherwise, the BS will satisfy the user’s request directly via the BS link. This approach is called pull-based because the users proactively attempt to pull the desired content down when their requests cannot be satisfied by self-cache.

Now we analyze the throughput of the pull-based system. Since we consider HPPP, the number of users in a cluster $k$ follows the Poisson distribution with density $\lambda_k d^2$. Then after several algebraic manipulations, the network throughput of the pull-based system is

$$T_{\text{c,pull}, k} = N \sum_{n=0}^{\infty} P_k \sum_{k=1} T_{\text{c,pull}, k},$$

where $T_{\text{c,pull}, k}$ is the throughput per cluster given by

$$[\sum_{n=0}^{k-1} B_k(n, P_k) (T_D + (T_B - T_D) \frac{P_{B,k}}{1 - P_S} + nT_S)] + B_k(k, P_k) \cdot kT_S;$$

and $B_k(n, P_k)$ is the probability mass function of the binomial distribution $B(n, P_k)$. Note that the details of the derivations will be provided in [23].

Since the throughput of the pull-based system is a very complicated function of cooperation distance $d$ and caching policy $\{b_m\}_M$, an alternative system, i.e., the push-based system, is considered, as it provides an easily tractable throughput formulation. The push-based system functions differently from the pull-based system as follows. For each cluster, the BS randomly chooses a user to serve without considering whether the user can obtain its desired content from its own cache. If the selected user can obtain the desired content from its own cache, the self-cache is used by the user; otherwise, the BS checks whether the desired content can be found through D2D links. If yes, the D2D communication is used; otherwise, the BS will serve the selected user by a BS link. This approach is called push-based because the BS tends to push the content to the selected user without considering whether the content has already been cached by this user. Note that the throughput of a push-based system is obviously a lower bound for the pull-based system because the number of users satisfied by the push-based system is always less and equal to the number of users satisfied by the pull-based system.

By the system description and after some algebraic manipulations, the throughput of the push-based system is

$$T_{\text{c,push}} = N \cdot T_{\text{c,push}},$$

where $T_{\text{c,push}}$ is in (9). Observe that, when fixing $d$, $T_{\text{c,push}}$ is a concave function with respect to the caching policy $\{b_m\}_M$. This therefore leads to the high tractability when conducting optimizations. The concavity will be formally elaborated in [23]. Moreover, we will numerically and analytically demonstrate that (8) is an effective lower bound of (6) in the Sec. V and in the future work [23], respectively.

Following the analysis in Sec. III.A, we design the optimal caching policy for the system by solving the optimization problem:

$$\begin{align*}
\max_{d, b_m, \forall m=1, \ldots, M} & \quad T_{\text{c,push}} \\
\text{subject to} & \quad \sum_{m=1}^{M} b_m \leq S, \quad 0 \leq b_m \leq 1, \forall m.
\end{align*}$$

To solve (10), we first observe that, if we can solve its sub-problem with any given $d$, the problem then becomes a simple one-dimensional problem with small range. Note that $d > 0$ is generally within 1000 meters in practice, and, given the optimal solution is attainable when fixing $d$, the problem is solvable even by simple quantization without significant effort. We then provide the following Proposition:

Proposition 1: When given a fixed $d$, problem (10) becomes a concave optimization problem and its optimal solution must be tight at the equality of the sum constraint, i.e., for the optimal solution $(b_m)^*, \forall k, m$, we have

$$\sum_{m=1}^{M} (b_m)^* = S.$$

By Proposition 1, the problem becomes a standard concave optimization problem, and any convex solver can be used to solve the problem. However, general convex solvers need to find the Hessian matrix which requires a high computational cost as the dimension of the solution space is large. Besides, (9) involves the exponential function which is unable to be dealt with efficiently by certain widely used convex solvers. We will hence provide a specifically designed algorithm in the future [23] to efficiently obtain the optimal solution.

IV. CACHING POLICY AND COOPERATION DISTANCE DESIGN FOR ENERGY EFFICIENCY OPTIMIZATION

In this work, EE of the system is defined as the EE of the typical user who can choose between using its own cache, D2D link, and BS link to obtain the desired content. By this definition, EE is considered in the sense of long-term average. Note that for any given user, it must obtain the desired content either by self-cache or by the D2D and BS communications, i.e., it must be served to satisfy the request when the desired content is not in its own cache. Then since all requests of users should ultimately be satisfied when considering the long-term behavior of the system, the long-term average EE of the system should be equal to the EE of a typical user.

Conditioning on the existence of the typical user, by using the Slivnyak-Mecke theorem [24], and after some derivations, the EE of a typical user is expressed as in (12). The derivations will again be provided later in [23]. By (12), the EE optimization problem is:

$$\begin{align*}
\max_{d, b_m, \forall m=1, \ldots, M} & \quad EE_T \\
\text{subject to} & \quad \sum_{m=1}^{M} b_m \leq S, \quad 0 \leq b_m \leq 1, \forall m.
\end{align*}$$
\[ T_{c,push} = T_D + (T_B - T_D) \left[ \sum_{m=1}^{M} a_m \exp(-\kappa_a b_m) \right] + (T_S \kappa_a - T_D (1 - \exp(-\kappa_a))) \left[ \sum_{m=1}^{M} a_m b_m \right] - T_B \exp(-\kappa_a). \] (9)

\[ EE_{ty} = \frac{T_y}{E_{C_{ty}}} = \frac{T_D + (T_B - T_D) \left[ \sum_{m=1}^{M} a_m (1 - b_m) \exp(-\kappa_a b_m) \right] + (T_S - T_D) \sum_{m=1}^{M} a_m b_m}{E_D + (E_B - E_D) \left[ \sum_{m=1}^{M} a_m (1 - b_m) \exp(-\kappa_a b_m) \right] - E_D \sum_{m=1}^{M} a_m b_m}. \] (12)

Here we again use same trick as in Sec. III.B in which we solve the sub-problems with different fixed \( d \) and then conduct the one-dimensional search for the optimal \( d \). When given \( d \), the following proposition is exploited:

**Proposition 2:** Suppose \( d \) is fixed. Then \( T_y \) is a concave function and \( E_{C_{ty}} \) is a convex function for the feasible domain \( B = \{ 0 \leq b_m \leq 1, \forall m \} \). Hence \( EE_{ty} \) is a quasi-concave function for the feasible domain \( B \), and (13) becomes a quasi-concave problem. Besides, \( T_y \) is non-decreasing and \( E_{C_{ty}} \) is non-increasing with respect to the increase of \( b_m, \forall m \), respectively. This implies that the optimal solution should be tight at the equality of the sum constraint.

For the proof see [23]. By Proposition 2, we know that, when fixing \( d \), we aim to solve a quasi-concave optimization with convex feasible set. Consequently, the standard solving procedure can be extended to obtain the optimal solution. Moreover, due to the similar reason as in throughput optimization, we will provide a specifically designed algorithm in [23] to efficiently solve the quasi-concave problem.

V. NUMERICAL RESULTS

Numerical results are provided to validate our analyses and evaluate the proposed throughput and EE designs. For all simulations in this paper, we consider the following parameters and setup: \( D = 1000 \) meters, \( K = 16 \) and \( \lambda_c = 1000 \) km\(^{-2} \). Also, we consider \( d_0 = 5 \) meters, \( \alpha = 3.8 \), \( \lambda_c = \frac{3 \times 10^{5}}{f_c} \), and \( f_c = 2 \) GHz in the path-loss model. The maximum allowable interference is set to be equal to the noise power, i.e., \( \nu = N_0 W \), where \( N_0 = -174 \) dBM/Hz is the noise power density and \( W = 50 \) MHz is the bandwidth for a D2D link. For the throughput and power consumptions, we consider \( T_B = \frac{T}{N} \), \( T_D = 50 \) Mbits/Sec, and \( T_S = T_D \); \( E_B = 42 \) dBm and \( E_D \) is computed by (2). We consider \( M = 1000 \), \( S = 10 \), and the request probabilities follow a Zipf distribution with Zipf factor \( \gamma \):

\[ a_m = \frac{m^{-\gamma}}{\sum_{m=1}^{M} n^{-\gamma}}. \] (14)

In Fig. 1, \( \gamma = 0.6 \) is considered, and the throughput analysis results are evaluated. The throughput caching policy design proposed in Sec. III is adopted when acquiring the results. From the figure, we can observe that the analytical results are consistent with the Monte Carlo results. Moreover, it can also be observed that the proposed push-based throughput is an effective lower bound of the pull-based throughput.

In Fig. 2, we consider \( \gamma = 0.6 \) and compare different caching policies in terms of throughput. In the figure, the throughput-based and EE-based designs correspond to the proposed designs in Secs. III and IV, respectively; the max-hit-rate design is to maximize the probability of the desired content to be found in the D2D network; and the selfish design is that all users selfishly cache the \( S \) most popular files. From the figure, it can be observed that the throughput-based design can provide the optimal throughput as expected. Besides, the selfish design can provide good performance. This is because, with self-caching, users unable to be served by D2D and BS links could still have the chance to access the desired files via their own caches, and those users in fact account for a large portion of the network throughput. This implies the importance of considering self-caching when aiming to optimize throughput. We note that these results are related to the assumption that all users are active, and, when we extend to consider both active and inactive users [23], the performance difference between the selfish design and the throughput-based design increases if the density of inactive users increases. In contrast to the selfish policy, EE-based and
the file popularity as compared to $\gamma$ as $\gamma$ self-caching rises when not being shown here, we can observe that the importance of more on enhancing the D2D cooperation. Finally, although this is because, rather than being selfish, they emphasize max-hit-rate designs perform poorly in terms of throughput. From the figure, we can observe that the throughput- and EE-based designs in the system with $\gamma = 0.6$; this thus leads to a higher impact of self-caching.

In Fig. 3, we consider $\gamma = 0.6$ and compare different caching policies in terms of EE. From the figure, we can observe that the proposed EE-based design provides the best (optimal) performance. Besides, the max-hit-rate design can also provide the near-optimal performance. Moreover, unsurprisingly, the throughput-based and selfish designs provide poor EE performance. The results indicate that, in contrast to being selfish, to have effective D2D cooperation is more important for EE. In Fig. 4, we consider $\gamma = 1.3$ and also compare different caching policies in terms of EE. From this figure, we can again observe that the EE-based design provides the best EE. However, the max-hit-rate design in this case is no longer near-optimal. Moreover, it provides the worst performance when the cluster size, i.e., the cooperation distance, is large. This is because, in contrast to the EE-based design, the max-hit-rate design ignores the effect of self-caching, and thus cannot balance between using D2D-caching and self-caching.

Finally, in Fig. 5, we focus on comparing the throughput- and EE-based designs in the system with $\gamma = 0.6$. From the figure, we can observe that the throughput- and EE-based designs conflict with each other significantly. Besides, their optimal cooperation distances are evidently different (one is at $d = 50$ and the other is at $d = 190$). As a result, their optimal trade-off design which provides the best compromise between throughput and EE is necessary; this topic will be investigated in [23].

VI. CONCLUSIONS

By jointly considering the interactions of BS-, D2D-, and self-caching and the impact of the cooperation distance, we provide the near-optimal and optimal caching policy and cooperation distance designs in terms of throughput and EE, respectively, in BS-assisted wireless D2D caching networks. With effective designs on hand, we then numerically show that considering the self-caching effect is important and the throughput-EE trade-off design is necessary for further improving the system. We will thus extend this work and investigate the optimal trade-off design in our future work [23].

ACKNOWLEDGEMENT

Part of this work was supported by the National Science Foundation (NSF).

REFERENCES


