Cooperative caching for dynamic shared spectrum networks

Dibakar Das Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: dasd2@rpi.edu
Alhussein A. Abouzeid Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180
Email: abouzeid@ecse.rpi.edu

Abstract

This paper considers cooperation between primary and secondary users in shared spectrum radio networks via caching. A network consisting of a single macro (primary) base-station and multiple small (secondary) base-stations is considered. Secondary base-stations can cache some primary files and thereby satisfy content-requests generated from nearby primary users. For this cooperative scenario, we develop two caching and scheduling policies under which the set of primary and secondary request generation rates that can be supported is expanded from the case without cooperation. The first of these algorithms provide maximum gain in the set of supportable primary and secondary request generation rates. However under this algorithm primary packet transmissions from secondary base-stations do not have higher priority of access than that of secondary packets. As a result, we propose another suboptimal algorithm wherein primary file transmissions from secondary base-stations have high priority of access than that of secondary files. Extensive simulations are conducted to compare the performance of both algorithms with that of a non-cooperative algorithm that is optimal, with respect to set of supportable request generation rates, among all non-cooperative policies.

I. Introduction

With the huge increase in number of wireless devices, there has been an increasing demand for new spectrum. However, at any time, the licensed spectrum is often under-utilized. This has prompted the widespread study of dynamic shared spectrum networks. In such networks some unlicensed or secondary users are allowed to opportunistically access and transmit on a given channel provided the primary or licensed users of that channel are not active. Traditionally, primary and secondary networks have been assumed to be non-cooperative i.e. the primary and secondary users do not assist in each other's transmissions. However, if secondary users choose to assist transmission of primary users, then it may reduce the overall duration for which the primary user is active on that

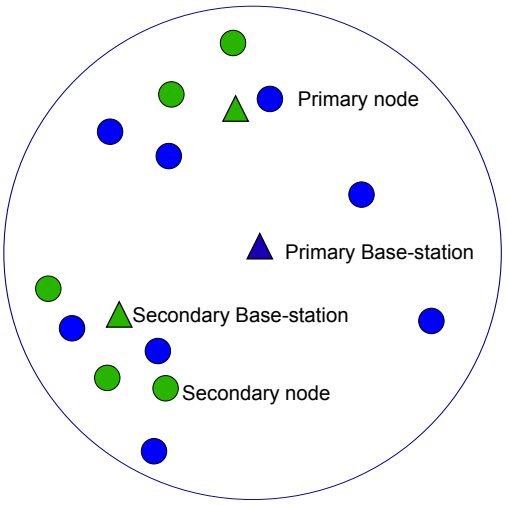


Fig. 1. A primary network with one base-station co-existing with two secondary base-stations. Some primary nodes are located closer to the secondary base-stations than the primary base-station. The outermost circle indicates the transmission region of the primary base-station.

channel. This in turn can also benefit the secondary users because it increases their own transmission opportunities. Cooperation between primary and secondary networks have been widely studied from a physical-layer perspective. Some of these works (eg- [1]–[3]) study it as an information-theoretic problem. Other recent works - [4]–[8] study network-layer aspects of cooperation such as queuing and prioritized scheduling.

Another way to support increasing mobile network traffic is to encourage localized communication by using small base-stations. This leads to higher spatial re-use of spectrum. However, limited capacity of backhaul links at the base-stations diminish the impact of this approach [9]. As a result, it was suggested in [9] to cache popular files at nearby base-stations which leads to lower back-haul usage as well as improves delay performance for the users. Caching is also tempting since storage-capacity is relatively inexpensive compared to other network resources.

In this chapter we explore cooperation between primary and secondary networks via caching. We consider a primary network consisting of a single base-station that serves primary users. Co-existing with the primary network is a set of small secondary base-stations that serve secondary users. An example of such a network is shown in Fig. 1. Both primary and secondary users request content from their respective base-stations where these requests are enqueued. The base-stations serve these requests by transmitting packets corresponding to requested files from their cache. If the requested file is not currently present in cache, then the file is fetched after sometime so as to satisfy the content-request. We assume that content is fetched at base-stations periodically and we refer to this period as the cache-refresh period. Since some of the small base-stations might be located close to primary users they may have better downlink channel to some primary users than the primary base-station. Therefore, if some of the content-requests from primary users are served via small base-stations then it might free up spectrum resources for the secondary users.

For the above network scenario we address the important problem of developing caching and scheduling policies

with performance guarantees. In particular we design algorithms under which a centralized network controller determines (a) which files to cache at small base-stations in every cache-refresh period and (b) how to schedule transmissions in each time-slot. The goal of the network controller is to maintain stability of request-queues i.e., to keep length of all request-queues in the network bounded. Accordingly, our performance measure is the set of all primary and secondary request generation rates for which every request-queue in the network is stable.

We develop two algorithms: DCSP and MCSP. Both algorithms are developed using Lyapunov-drift techniques that makes decisions based on length of request-queues as well as popularity distribution of files requested by primary users. The set of primary and secondary request generation-rate vectors for which the network is stable under each of these algorithms is greater than that under any non-cooperative algorithm. Under the DCSP algorithm each small base-station caches only one secondary file in every period while filing up rest of the cache with primary files. However, only primary packet transmissions from the primary base-station enjoy higher priority of channel access. In the MCSP algorithm however, requests from primary users enqueued at any small base-station are always served with higher priority than that from secondary users. Further, in every caching period, the network-controller dynamically varies the number of cached primary files while ensuring system stability. As a result small base-stations can serve more than one type of secondary file requests in each period. Simulation results show this algorithm may have better delay performance than the DCSP algorithm when request generation rates are low. In such scenarios it is not necessary for system stability to cache as much primary files as possible in every small base-station. However, guaranteed stability region of the network under MCSP is less than that under the DCSP policy. *Related work:* There has been a lot of work on caching in non-cognitive wireless networks. Some results include [9]–[11] and the references therein. Caching in cognitive networks has been studied recently in [12] and [13]. However they did not consider primary-secondary cooperation. Our periodic cache-refresh policy and the use of Lyapunov drift to develop a scheduling policy is motivated by [14] and [15]. However, their results are not directly applicable in the cognitive network setting since here primary users have higher priority of channel access than secondary nodes. While a simple Lyapunov drift-based algorithm tries to serve the queues with higher backlogs first, in our model the request-queues in the primary base-station would have to be served before that at the secondary base-stations even when it has relatively lower backlog. Such complications resulting from higher priority of service for primary users is addressed in this work.

Our periodic cache-refresh policy and the use of Lyapunov drift to develop a scheduling policy is motivated by [14] and [15]. However, their results are not directly applicable in the cognitive network setting since here primary users have higher priority of channel access than secondary nodes. While a simple Lyapunov drift-based algorithm tries to serve the queues with higher backlogs first, in our model the request-queues in the primary base-station would have to be served before that at the secondary base-stations even when it has relatively lower backlog (except possibly when the latter is transmitting a secondary packet). Such complications resulting from higher priority of service for primary users is addressed in this work.

In Section II we describe our system model. In Section III, we define the capacity-region for this network consisting of supportable primary and secondary request generation-rate vectors. In Section IV we propose the

DCSP policy and show that it is throughput-optimal with respect to the capacity region. In Section V we propose the MCSP policy and find a guaranteed stability region for it. Section VI discusses simulation results. Section VII summarizes the chapter.

II. System Model

The network consists of a macrocell wherein a single primary base-station PB serves $N^{(p)}$ primary users: PU_1,...,$\text{PU}_{N^{(p)}}$. It also contains M secondary small-cells $\text{SC}_1, \ldots, \text{SC}_M$ with small base-stations SB_1,...,SB_M at their centres respectively. There are $N^{(s)}$ secondary users which are denoted as SU_1,...,$\text{SU}_{N^{(s)}}$. Each secondary user is located in exactly one of those small-cells. We consider a discrete time-slotted model. Every file request is served by successfully transmitting C packets of equal size. A base-station attempts a new packet transmission only at the beginning of a time-slot; it can attempt at most one such transmission at any slot. At the end of the time-slot the transmission is either successful and the packet is successfully received by the desired user or the transmission fails and the packet needs to be re-transmitted at some other slot. Next we present details about our transmission scheme, interference model, caching and scheduling constraints.

A. Transmission Model for Primary and Secondary Base-stations

The primary and secondary base-stations transmit at fixed power. All secondary base-stations have identical transmission range which is lower than that of PB. Some primary users are located relatively far away from PB but close to one or more secondary base-stations (i.e., within a small cell). As a result, the probability of a successful transmission from PB to such primary users is lower than that from adjacent secondary base-stations. For simplicity, we assume

1) At every slot, transmission from PB to any primary user succeeds with probability p (where $0 < p \leq 1$) while transmissions from any secondary base-station to an user within its transmission range succeeds with probability 1.
2) Any primary user is within transmission range of atmost one small-cell.

We denote the set of primary and secondary users in SC_i (where $1 \leq i \leq M$) by $\phi_i^{(p)}$ and $\phi_i^{(s)}$ respectively. All secondary users and base-stations are within the transmission range of PB.

B. Caching Model

Every time a secondary base-station caches a set of files it incurs some cost. This cost is modeled by requiring that secondary base-stations only cache files periodically. A higher frequency of caching reflects a higher cost. A cache-refresh period is defined to be the number of slots between two successive caching events. A cache refresh period may also represent the frequency with which contents in files become outdated thereby requiring newer versions to be fetched. It consists of T slots with the very first caching event being at slot $t = 1$.

C. Generation of Content-requests From Primary and Secondary Users

We consider the case where primary and secondary networks cater to different types of users. This can occur, for example, if the small-cells serve an industrial or academic environment while users of the macrocell are the general public who are typically interested in video content. We denote the library of files requested by primary users as $F^{(p)}$ with individual files in set being referred to as $\mathrm{F}_1^{(p)}$,...,$\mathrm{F}_{|F^{(p)}|}^{(p)}$. Similarly we denote the library of files requested by secondary users as $F^{(s)}$ with individual files in set being referred to as $\mathrm{F}_1^{(s)}$,...,$\mathrm{F}_{|F^{(s)}|}^{(s)}$. In this work we assume these two sets are mutually exclusive. Henceforth we will call files in $F^{(p)}$ and $F^{(s)}$ as primary and secondary files respectively. Similarly, we call packets corresponding to primary and secondary files as primary and secondary packets respectively. All files are assumed to be of equal size.

Files are requested by users according to a popularity distribution which varies from period to period in an identical and independently distributed (iid) fashion reflecting the change in user's preference. We model this by assuming that in each period the network is in one of "popularity states" $r \in \tilde{r}$, where $\tilde{r}$ denotes a finite collection of such states. Let q_r denote the probability of the network being in state r at any period. Typically in works on caching, Zipf distribution is used to model the popularity distribution of files. Given a primary user requests a file when the network is in state r, the file $\mathrm{F}_i^{(p)}$ is requested with probability $P_{i,r}^{(p)}$. Similarly, given a secondary user requests a file when the network is in state r, the file $\mathrm{F}_i^{(s)}$ is requested with probability $P_{i,r}^{(s)}$.

Every slot primary user PU_i requests a file with probability $\lambda_i^{(p)}$. Similarly, at every slot a secondary user SU_j requests files with probability $\lambda_j^{(s)}$. All request generation process are assumed to be iid from slot to slot.

D. Serving Requests From Primary Users

File-request from a primary user is served by either a secondary base-station (if the primary user is located in a small-cell and the associated secondary base-station contains the file) or by the primary base-station. Every base-station maintains *request-queues* corresponding to each type of file-requests. Items in each request-queue correspond to packet-requests that need to be satisfied in response to a file-request. For every file-request at a base-station, C such packet-requests are created and enqueued at the appropriate request-queue. These packet-requests are satisfied in a first-come first serve (FCFS) manner from respective request-queues.

In order to focus only on the effect of cooperative caching by the secondary network, we assume the primary base-station can cache all the $|F^{(p)}|$ primary files. On other hand, the size of cache at each secondary base-station is finite. Each such cache can store at most B files where $0 \leq B \leq \min(|F^{(p)}|, |F^{(s)}|)$. Further, we assume the following:

1) In each caching period a secondary base-station caches at least one secondary file i.e. the maximum number of primary files that can be cached at each base-station in any period is B-1. This ensures that in each period any secondary base-station can serve at least one type of secondary packet-requests, provided such a transmission opportunity arises.
2) A secondary base-station only admits primary user requests if the corresponding file is present in its cache.

This assumption prevents scenarios wherein a secondary base-station cannot serve unsatisfied primary file-requests due to absence of the requested file in its cache.

We indicate a request generated from PU_i at slot t for file $f \in F^{(p)}$ by variable $A^{(p)}_{f,i}(t)$. This variable equals C if such a request is indeed generated at t; otherwise it equals 0.

We denote the length of the request-queue of primary file f at SB_k at slot t as $U^{(p)}_{f,k}(t)$. If PU_i is within SC_k and the base-station SB_k contains the file f requested by PU_i at t, then C packet-requests will be enqueued at SB_k and $U^{(p)}_{f,k}(t)$ is incremented by C. Otherwise, this request will be served by PB and the length of the associated request-queue, denoted as $U^{(p)}_{\text{f,0}}(t)$, will be incremented by C. We assume the system begins at slot t=1 with all queues being initially empty.

On successful transmission of a packet corresponding to file f from PB at t, $U^{(p)}_{\text{f,0}}(t)$ is decremented by 1. We denote the transmission rate offered to base-station a for packets of file f at slot t by the binary variable $\mu_{f,a}(t) \in \{0, 1\}$. Therefore the length of request-queues of primary files at PB is updated as follows:

$$U^{(p)}_{\text{f,0}}(t+1) = U^{(p)}_{\text{f,0}}(t) - \mu_{f,\text{PB}}(t) I_{f,\text{PB}}(t) + \sum_{\substack{i:\text{PU}_i \notin \cup_j \phi^{(p)}_j \text{ or, } \text{PU}_i \in \phi^{(p)}_k \\ \text{and } f \text{ is not present in } \text{SB}_k\text{'s cache}}} A^{(p)}_{f,i}(t) \quad \forall f \in F^{(p)} \tag{1}$$

where $I_{f,\text{PB}}(t)$ is an indicator variable representing whether the transmission from PB to the primary user at slot t was succesful or not; it equals 1 if the transmission is successful and is zero otherwise. We assume that no transmission-rate is offered to PB for transmitting packets corresponding to an empty request-queue.

The length of request-queues of primary files at secondary base-stations are updated as follows:

$$\begin{aligned} U^{(p)}_{f,k}(t+1) &= \max\{U^{(p)}_{f,k}(t) - \mu_{f,\text{SB}_k}(t), 0\} \\ &\quad + \sum_{i:\text{PU}_i \in \phi^{(p)}_k} A^{(p)}_{f,i}(t) \quad \forall 1 \le i \le N^{(p)}, 1 \le k \le M, f \in F^{(p)} \end{aligned} \tag{2}$$

E. Serving Requests From Secondary Users

Requests from secondary users are submitted to the unique secondary base-station associated with each such user. Similar to the case of primary files, each secondary base-station maintains a request-queue for every secondary file. We indicate a request generated from SU_j at slot t for a secondary file $f \in F^{(s)}$ by a variable $A^{(s)}_{f,j}(t)$. It equals C if such a request is indeed generated at t; otherwise it equals 0. For every $k = 1, \ldots, M$ we denote the length of request-queue in SB_k at t of file f as $U^{(s)}_{f,k}(t)$. The length of this queue is incremented (or decremented) by C (or one) every time SB_k receives (or serves) a request for (or transmits a packet of) file f. The queue is updated as,

$$U^{(s)}_{f,k}(t+1) = \max\{U^{(s)}_{f,k}(t) - \mu_{\text{SB}_k,f}(t), 0\} + \sum_{j \in \phi^{(s)}_k} A^{(s)}_{j,f}(t) \quad \forall 1 \le k \le M, \quad f \in F^{(s)} \tag{3}$$

We refer to requests queues corresponding to primary and secondary files as primary and secondary request-queues respectively.

F. Interference Model

We model co-channel interference among secondary base-stations by using a modified version of the protocol model of interference described in [16]. We call two secondary base-stations "interfering neighbors" if they lie within twice the transmission range of each other. A transmission from a secondary base-station to any user is feasible only if none of its interfering neighbors is transmitting at the same slot. We represent the set of all base-stations that can transmit simultaneously by an *activation* vector of length M. An activation vector E is binary and its m'th (where $1 \leq m \leq M$) component corresponds to SB_m. It is set to 1 if SB_m is transmitting in that slot; otherwise it is set to zero. The set of all feasible activation vectors is denoted as $\tilde{E}$. No secondary base-station can transmit when PB is transmitting.

III. Stability Objective

In this section we describe the capacity region corresponding to the system model described in Section II. The capacity region contains all primary and secondary request generation-rate vectors for which the network is stable under any possible algorithm. It is defined by establishing the stability constraints at the primary and secondary base-stations similar to description of capacity region in [17]. In next section we will develop an algorithm that stabilizes the network of request-queues for all such rate vectors in the interior of this region.

First we introduce some new notations. We use a binary matrix to indicate the availability of primary files at secondary base-stations at any given slot. Each such matrix, referred to as *primary availability matrix*, contains $|F^{(p)}|$ rows and M columns corresponding to the total number of primary files and number of secondary base-stations respectively. The (n, j)'th component of a primary availability matrix $D^{(p)}$ equals 1 if the file $\text{F}_n^{(p)}$ is present at the cache of SB_j; otherwise it equals zero. Given a primary availability matrix $D^{(p)}$, we indicate whether or not a file f requested by PU_i is present in a nearby secondary base-station by the binary variable $Q(i, f|D^{(p)})$. In particular, for every primary file f, $Q(i, f|D^{(p)}) = 1$ if there exists j s.t. $D_{l,j}^{(p)} = 1$ and $\text{PU}_i \in \phi_j^{(p)}$; it is set to 0 otherwise.

Since every secondary base-station caches at least one secondary file in each period we consider only the set of primary availability matrices for which sum of every column is less than B. We denote this set as $\tilde{D}^{(p)}$.

We denote primary and secondary request generation-rate vectors:
$(\lambda_1^{(p)}, \ldots, \lambda_{N^{(p)}}^{(p)})^T$ and $(\lambda_1^{(s)}, \ldots, \lambda_{N^{(s)}}^{(s)})^T$ as $\boldsymbol{\lambda}^{(p)}$ and $\boldsymbol{\lambda}^{(s)}$ respectively. The capacity region Λ is the set of all possible primary and secondary request generation-rate vectors-$(\boldsymbol{\lambda}^{(p)}, \boldsymbol{\lambda}^{(s)})$ for which there exists primary availability matrix $D^{(p)}$ and variables π_0, R_1,..., R_M s.t.

$$\pi_0 = C \left\{ \sum_{n=1}^{N^{(p)}} \lambda_n^{(p)} - \sum_{i=1}^{N^{(p)}} \lambda_i^{(p)} \sum_{r \in \tilde{r}} \sum_{l=1}^{|F^{(p)}|} P_{l,r}^{(p)} Q(i, F_l^{(p)}|D^{(p)}) q_r \right\} \tag{4}$$

$$\frac{\pi_0}{p} \leq 1 \tag{5}$$

$$C \left\{ \sum_{i:\text{SU}_i \in \phi_j^{(s)}} \lambda_i^{(s)} + \sum_{r \in \tilde{r}} \sum_{k:\text{PU}_k \in \phi_j^{(p)}} \sum_{l=1}^{|F^{(p)}|} P_{l,r}^{(p)} \lambda_k^{(p)} D_{l,j}^{(p)} q_r \right\} \leq R_j \quad \forall 1 \leq j \leq M \tag{6}$$

for some $(R_1, \ldots, R_M)^T \in \Gamma(D^{(p)})$ where

$$\Gamma(D^{(p)}) = \Bigg\{ 1 - \frac{\sum_{n=1}^{N^{(p)}} C\lambda_n^{(p)}}{p} - \frac{C \sum_{i=1}^{N^{(p)}} \lambda_i^{(p)} \sum_{r \in \tilde{r}} \sum_{l=1}^{|F^{(p)}|} P_{l,r}^{(p)} Q(i, F_l^{(p)} | D^{(p)}) q_r}{p} \textbf{conv}(\tilde{E}) \tag{8}$$

The term π_0 in (4) represents the average rate of packet transmissions by PB corresponding to the availability matrix $D^{(p)}$. The right hand side (RHS) of (4) is the product of the rate of primary user requests served by PB and C, the number of packet transmissions corresponding to every served file request. The constraint (5) is the stability constraint at PB. The inequality constraint (6) represents the stability requirement at secondary base-stations. The left hand side (LHS) indicates total arrival-rate into all the request-queues in a given secondary base-station. The set $\Gamma(D^{(p)})$ in (7) characterizes the set of feasible secondary transmission-rate vectors given a primary availability matrix $D^{(p)}$. This set is defined in average sense as the convex hull of all feasible activation vectors when PB is not transmitting, multiplied by the probability that PB is not transmitting. The latter is equal to the term $\{1 - \frac{\sum_{n=1}^{N^{(p)}} C\lambda_n^{(p)} - C\sum_{i=1}^{N^{(p)}} \lambda_i^{(p)} \sum_{r \in \tilde{r}} \sum_{l=1}^{|F^{(p)}|} P_{l,r}^{(p)} Q(i, F_l^{(p)} | D^{(p)}) q_r}{p}\}$. The **conv** of a set of vectors is the set of all possible convex combinations of its elements.

IV. DCSP Algorithm

In this section we propose the DCSP algorithm that stabilizes the network of request-queues for all primary and secondary request generation-rate vectors in the interior of the capacity region Λ. Under this policy, every secondary base-station caches $B-1$ most popular primary files (averaged over all popularity states) and the secondary file with highest instantaneous request-queue length. In each period, whenever the primary base-station is not transmitting, the secondary base-stations transmit according to a backpressure type policy. Intuitively, caching $B-1$ most popular primary files maximizes the rate of primary user requests generated in each small-cell that is served by the corresponding secondary base-stations. Since the primary and secondary base-stations take on average $\frac{1}{p}$ and 1 slots respectively to successfully transmit a packet, secondary base-stations create more transmission opportunities to serve their own users by serving higher fraction of primary user requests.

Before we describe the algorithm we introduce new notations. We denote the j'th slot in r'th period as $t_{r,j}$, where $r = 1, 2, \ldots$ and $j = 1, 1, \ldots, T$. We refer to aggregate of all primary request-queues in SB_k (where $k = 1, \ldots, M$) as the *primary queue* of SB_k. Clearly, the length of primary queue in SB_k, denoted as $U_k^{(p)}(t)$, is the sum of length of all primary request-queues i.e., $U_k^{(p)}(t) = \sum_{f \in F^{(p)}} U_{f,k}^{(p)}(t)$. We denote the number of primary packets transmitted by SB_k at slot t by the binary variable $\mu_k^{(p)}(t) \triangleq \sum_{f \in F^{(p)}} \mu_{\text{SB}_k, f}(t)$.

Under the DCSP algorithm for every $r = 1, 2, \ldots,$ following steps are performed in r'th period:

1) *Caching Scheme:* At the beginning of r'th period every secondary base-station caches the $B-1$ most popular primary files where popularity of i'th primary file is $\tilde{P}_i^{(p)} = \sum_{r\in\tilde{r}} P_{i,r}^{(p)} q_r$. Without loss of generality we assume $\tilde{P}_1^{(p)} \geq \tilde{P}_2^{(p)} \ldots \geq \tilde{P}_{|F^{(p)}|}^{(p)}$. In addition every secondary base-station caches the secondary file corresponding to maximum back-log at $t_{r,1}$. If we denote the secondary file to be cached at SB_k in r'th period as $f_k^*(r)$ then

$$f_k^*(r) \quad \in \quad \underset{f\in F^{(s)}}{\text{argmax}}\, U_{f,k}^{(s)}(t_{r,1}) \quad \forall k = 1, 2, \ldots, M \tag{9}$$

2) *Scheduling for PB:* PB retains highest priority of transmission in the network. At any slot t it transmits a packet corresponding to the Head-of-Line (HOL) packet-request at the request-queue of highest length (in case of ties, pick one arbitrarily). This packet-request is removed from the queue if the transmission is successful. Mathematically,

$$\mu_{\text{PB},f}^{\text{DCSP}}(t) = \begin{cases} 1, & \text{if } 0 < U_{f,0}^{(p)}(t) \geq U_{\bar{f},0}^{(p)}(t) \quad \forall \bar{f} \in F^{(p)} - f \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mu_{a,f}^{X}(t)$ denotes the transmission rate offered to base-station a under policy X to transmit a packet of file f.

3) *Scheduling Policy at secondary base-stations:* If PB is not transmitting at slot $t_{r,j}$, then obtain the set of secondary base-stations that are allowed to transmit at $t_{r,j}$, denoted as $E^*(t_{r,j})$ by solving the following max-weight problem:

$$E^*(t_{r,j}) \quad \in \quad \underset{E\in\tilde{E}}{\text{argmax}} \left(\left(U_k^{(p)}(t_{r,j}) + U_{f_k^*(r),k}^{(s)}(t_{r,j}) \right)_{k=1}^{M} \right)^T E \tag{11}$$

Suppose, according to $E^*(t_{r,j})$ some SB_k is allowed to transmit in this slot. Then transmit the packet corresponding to the HOL packet-request at the request-queue of file $f_k^*(r)$ if $U_k^{(p)}(t)$ is less than $U_{f_k^*(r),k}^{(s)}(t_{r,j})$. Otherwise transmit the packet corresponding to the HOL packet-request at the primary request-queue of highest length in SB_k. If this queue is empty, transmit a dummy packet.

Mathematically, given k'th component of $E^*(t_{r,j})$ is 1, $\mu_{\text{SB}_k}^{\text{DCSP},(p)}(t_{r,j}) = 1$, $\mu_{f_k^*(r),k}^{\text{DCSP}}(t_{r,j}) = 0$ if $U_k^{(p)}(t) \geq U_{f_k^*(r),k}^{(s)}(t_{r,j})$ and $\mu_{\text{SB}_k}^{\text{DCSP},(p)}(t_{r,j}) = 0$, $\mu_{f_k^*(r),k}^{\text{DCSP}}(t_{r,j}) = 1$ otherwise.

Note, the scheduling policy only uses knowledge of instantaneous queue-lengths and not the request-generation rates.

It can be shown that DCSP is throughput-optimal. theorem DCSP stabilizes the network of request-queues for all ($\boldsymbol{\lambda}^{(p)}$,$\boldsymbol{\lambda}^{(s)}$)$\in$ Interior(Λ).

In order to prove Theorem IV, we first state and prove Lemmas IV and IV.

In Lemma IV we show that Λ can be characterized by considering only the set of policies in which $B-1$ most popular primary files are cached at each secondary base-station in every caching period. As mentioned earlier intuitively this represents that maximum transmission opportunities for secondary users are created when as much

primary user requests are served by secondary base-stations as possible. Next we propose an alternate policy ALT1 which, in every period, minimizes the T-slot conditional drift[1] of a Lyapunov function of length of request-queues in secondary base-stations. In Lemma IV we compare the values of a utility function that corresponds to the conditional Lyapunov drift of the system under DCSP and ALT1 respectively. Finally we use these results in the proof of Theorem IV. lemma Λ can be obtained by restricting $D^{(p)}$ in (4)-(6) only to the primary availability matrix D^* for which

$$D^*_{n,j} = \begin{cases} 1, & \text{if } 1 \leq n \leq B-1, \quad 1 \leq j \leq M \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

We first show that it is sufficient, in description of Λ, to consider only those $D^{(p)}$ which has $B-1$ non-zero components in each column. Consider the set of primary availability matrices in which at least at one secondary base-station, less than $B-1$ primary files are cached. Let $D'^{(p)}$ denote one such matrix in which there exists at least one column j with $\sum_{n=1}^{|F^{(p)}|} D'^{(p)}_{n,j} = B' < B-1$. Suppose $D'^{(p)}$ is the primary availability matrix in description of Λ. Then we can construct a new matrix $D''^{(p)}$ by setting one of the zero components in j'th column of $D'^{(p)}$, denoted as $D'^{(p)}_{i,j}$, to 1. Then replacing $D'^{(p)}$ with $D''^{(p)}$ in (4)-(6) we observe that the L.H.S of (6) increases by $\sum_{k:\text{PU}_k \in \phi_j^{(p)}} \lambda_k^{(p)} \tilde{P}_i^{(p)}$ while the R.H.S of (6) increases by $\frac{\sum_{k:\text{PU}_k \in \phi_j^{(p)}} \lambda_k^{(p)} \tilde{P}_i^{(p)}}{p}$. Therefore, (6) is still satisfied by replacing $D'^{(p)}$ with $D''^{(p)}$. Similarly, by extending $D''^{(p)}$ until there are $B-1$ non-zero elements in each column of the extended matrix, we can show Λ can be obtained by considering only those $D^{(p)}$ with $B-1$ non-zero components in each column.

Next, we assume there is some $D''^{(p)}$ which has $B-1$ non-zero components in each column but at least one component in $D''^{(p)}_{1,j}$,...,$D''^{(p)}_{B-1,j}$ is zero for some j. Suppose $D''^{(p)}$is the primary availability matrix in description of Λ.. Therefore, by replacing $D''^{(p)}$ with D^* in (4)-(6) we once again observe the L.H.S of (6) increases by less amount than the R.H.S. Hence, the capacity region Λ can be achieved by replacing $D''^{(p)}$ with D^*. This proves the Lemma.

Next we consider a caching and scheduling policy ALT1 that performs the following in every period:

1) *Caching policy:* Cache files in same way as DCSP.
2) *Scheduling at PB:* Transmit primary packets as in DCSP.
3) *Scheduling at secondary base-station:* If at some slot in r'th period, $t_{r,j}$, PB is not transmitting any file, obtain $E^{\text{ALT1}}(t_{r,j})$ as

$$E^{\text{ALT1}}(t_{r,j}) \quad \in \quad \underset{E \in \tilde{E}}{\text{argmax}} \left(\left(U_k^{(p)}(t_{r,1}) + U^{(p)}_{f_k^*(r),k}(t_{r,1}) \right)_{k=1}^{M} \right)^T E \tag{13}$$

Suppose, according to $E^*(t_{r,j})$ some SB_k is allowed to transmit in this slot. Then transmit the packet corresponding to the HOL packet-request at one of the primary requests queues in SB_k if $U_k^{(p)}(t_{r,1})$ is greater

[1]The k-slot conditional drift of a Lyapunov function of instantaneous queue-lengths, $V(t)$ is $E[V(t+k) - V(t) | \{U^{(p)}_{f,n}(t), U^{(s)}_{f,n}(t)\}]$ [17].

than or equal to $U^{(s)}_{f^*_k(r),k}(t_{r,1})$. Otherwise, transmit the the packet corresponding to the HOL packet-request at request-queue of file $f^*_k(r)$. In case the primary queue is empty, transmit a dummy packet.

Thus ALT1 is different from DCSP in that in ALT1 scheduling decisions at secondary base-stations are made based on request-queue lengths in the beginning of the period while in DCSP they are made based on their instantaneous values.

For any policy X that caches $B-1$ most popular primary files at each secondary base-station in every period, we define an utility function $\psi^X(t_{r,1})$ (where $r=1,2,\ldots$) as

$$\begin{aligned}\psi^X(t_{r,1}) &\triangleq \sum_{k=1}^{M} U^{(p)}_k(t_{r,1}) E[\sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu^{X,(p)}_k(\tau) | (U^{(s)}_{f,k}(t_{r,1}))^T, (U^{(p)}_k(t_{r,1}))^T] \\ &+ \sum_{k=1}^{M} \sum_{f\in F^{(s)}} U^{(s)}_{f,k}(t_{r,1}) E[\sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu^{X,(s)}_{f,k}(\tau) | (U^{(s)}_{f,k}(t_{r,1}))^T, (U^{(p)}_k(t_{r,1}))^T] \end{aligned} \quad (14)$$

It can be easily seen that ALT1 maximizes $\psi^X(t_{r,1})$ among the set of all policies Φ that

1) Caches $B-1$ most popular primary files at every secondary base-station in each period, and
2) In each slot first select an activation vector. If a secondary base-station is allowed to transmit then it transmit a packet corresponding to one of the request-queues of cached files. If the request-queue is empty transmit a dummy packet.

Then the following result can be shown: lemma $\psi^{\text{DCSP}}(t_{r,1}) \geq \psi^{\text{ALT1}}(t_{r,1}) - K_1$ where $K_1 \geq 0$ is a finite constant independent of queue-lengths.

Let $U^{\Phi,(s)}_{f,k}(t_{r,j})$, $U^{\Phi,(p)}_k(t_{r,j})$ denote the secondary request-queue length of file f and primary queue-lengths respectively in SB_k under policy Φ at slot $t_{r,j}$ (where $j=1,\ldots,T$). First we note that location of transmission-opportunities for secondary base-stations are identical under both DCSP and ALT1. Now, there exists a finite constant $\tilde{T} > 0$ such that

$$U^{(p)}_k(t_{r,j}) \leq U^{(p)}_k(t_{r,1}) + \tilde{T} \text{ and } U^{(p)}_k(t_{r,j}) \geq U^{(p)}_k(t_{r,1}) - \tilde{T}, \quad (15)$$

$$U^{(s)}_{f,k}(t_{r,j}) \leq U^{(s)}_{f,k}(t_{r,1}) + \tilde{T} \text{ and } U^{(s)}_{f,k}(t_{r,j}) \geq U^{(s)}_{f,k}(t_{r,1}) - \tilde{T} \quad (16)$$

as maximum number of arrivals and departures of requests from any station in any slot is finite.

Then for $j=1,\ldots,T$

$$\begin{aligned} & E[\{\sum_{k=1}^{M} U^{(p)}_k(t_{r,1}) \mu^{\text{DCSP},(p)}_k(t_{r,j}) \\ & + \sum_{k=1}^{M} \sum_{f\in F^{(s)}} U^{(s)}_{f,k}(t_{r,1}) \mu^{\text{DCSP},(s)}_{f,k}(t_{r,j})\} | (U^{(s)}_{f,k}(t_{r,1}))^T, (U^{(p)}_k(t_{r,1}))^T] \\ \geq\ & E[\sum_{k=1}^{M} U^{\text{DCSP},(p)}_k(t_{r,j}) \mu^{\text{DCSP},(p)}_k(t_{r,j}) | (U^{(s)}_{f,k}(t_{r,1}))^T, (U^{(p)}_k(t_{r,1}))^T] \\ & + E[\sum_{k=1}^{M} \sum_{f\in F^{(s)}} U^{\text{DCSP},(s)}_{f,k}(t_{r,j}) \mu^{\text{DCSP},(s)}_{f,k}(t_{r,j}) | (U^{(s)}_{f,k}(t_{r,1}))^T, (U^{(p)}_k(t_{r,1}))^T] \end{aligned}$$

$$-\tilde{T}M(1+|F^{(s)}|) \tag{17}$$

$$\geq E[\sum_{k=1}^{M} U_k^{\text{DCSP},(p)}(t_{r,j})\mu_k^{\text{ALT1},(p)}(t_{r,j})|(U_{f,k}^{(s)}(t_{r,1}))^T,(U_k^{(p)}(t_{r,1}))^T]$$

$$+E[\sum_{k=1}^{M}\sum_{f\in F^{(s)}} U_{f,k}^{\text{DCSP},(s)}(t_{r,j})\mu_{f,k}^{\text{ALT1},(s)}(t_{r,j})|(U_{f,k}^{(s)}(t_{r,1}))^T,(U_k^{(p)}(t_{r,1}))^T]$$

$$-\tilde{T}M(1+|F^{(s)}|) \tag{18}$$

$$\geq E[\sum_{k=1}^{M} U_k^{(p)}(t_{r,1})\mu_k^{\text{ALT1},(p)}(t_{r,j})|(U_{f,k}^{(s)}(t_{r,1}))^T,(U_k^{(p)}(t_{r,1}))^T]$$

$$+E[\sum_{k=1}^{M}\sum_{f\in F^{(s)}} U_{f,k}^{(s)}(t_{r,1})\mu_{f,k}^{\text{ALT1},(s)}(t_{r,j})|(U_{f,k}^{(s)}(t_{r,1}))^T,(U_k^{(p)}(t_{r,1}))^T]$$

$$-2\tilde{T}M(1+|F^{(s)}|) \tag{19}$$

(18) follows because DCSP maximizes among all policies in Φ the following expression for every slot τ in r'th period:

$$\sum_{k=1}^{M} U_k^{(p)}(\tau)E[\mu_k^{\Phi,(p)}(\tau)|(U_{f,k}^{(s)}(\tau))^T,(U_k^{(p)}(\tau))^T]$$

$$+\sum_{k=1}^{M}\sum_{f\in F^{(s)}} U_{f,k}^{(s)}(\tau)E[\mu_{f,k}^{\Phi,(s)}(\tau)|(U_{f,k}^{(s)}(\tau))^T,(U_k^{(p)}(\tau))^T] \tag{20}$$

Combining the above for all $j=1,\ldots,T$ we prove the Lemma. [Proof of Theorem IV] We denote the request-queue lengths in each secondary base-station at the beginning of r'th period as,

$$Z_{f,k}^{(s)}(r) \triangleq U_{f,k}^{(s)}(t_{r,1}) \tag{21}$$

$$Z_k^{(p)}(r) \triangleq U_k^{(p)}(t_{r,1}) \tag{22}$$

Then under policy Φ

$$Z_{f,k}^{(s)}(r+1) = \max\{Z_{f,k}^{(s)}(r) - \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1}\mu_{f,k}^{\Phi,(s)}(\tau),0\} + \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} A_{k,f}^{(s)}(\tau) \tag{23}$$

$$Z_k^{(p)}(r+1) = \max\{Z_k^{(p)}(r) - \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1}\mu_k^{\Phi,(p)}(\tau),0\} + \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} A_k^{(p)}(\tau) \tag{24}$$

For any $(\boldsymbol{\lambda}^{(p)},\boldsymbol{\lambda}^{(s)})\in$ Interior(Λ) there exists some constant $\epsilon>0$ s.t. $(\boldsymbol{\lambda}^{(p)}+\boldsymbol{\epsilon}^{(p)},\boldsymbol{\lambda}^{(s)}+\boldsymbol{\epsilon}^{(s)})\in\Lambda$ where $\boldsymbol{\epsilon}^{(p)},\boldsymbol{\epsilon}^{(s)}$ are vectors of lengths $N^{(p)}$ and $N^{(s)}$ respectively and whose each component is ϵ.

It can be easily shown that there exists a stationary policy STAT in Φ that stabilizes the network of request-queues without knowledge of request-queue lengths at secondary base-stations, for all request generation-rate vectors in Λ. Since, $(\boldsymbol{\lambda}^{(p)}+\boldsymbol{\epsilon}^{(p)},\boldsymbol{\lambda}^{(s)}+\boldsymbol{\epsilon}^{(s)})\in\Lambda$, therefore STAT stabilizes the network for this request-generation vector as well.

We define a Lyapunov function $L(\{Z_{f,k}^{(s)}(r), Z_k^{(p)}(r)\}) \triangleq \sum_{k=1}^{M} \sum_{f\in F^{(s)}} (Z_{f,k}^{(s)}(r))^2 + \sum_{k=1}^{M} (Z_k^{(p)}(r))^2$. The conditional drift is defined as

$$\begin{aligned}\Delta(r) &\triangleq E[L(\{Z_{f,k}^{(s)}(r+1), Z_k^{(p)}(r+1)\}) \\ &- L(\{Z_{f,k}^{(s)}(r), Z_k^{(p)}(r)\})|\{Z_{f,k}^{(s)}(r), Z_k^{(p)}(r)\}] \end{aligned} \tag{25}$$

Number of request-arrivals and satisfied by each station in every period is upper bounded by some finite positive number $\hat{T}$. Therefore, we have

$$\begin{aligned}\Delta(r) &\leq \hat{T}^2 M(1+|F^{(s)}|) \\ &- 2\sum_{k=1}^{M} \sum_{f\in F^{(s)}} E[Z_{f,k}^{(s)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_{f,k}^{\text{DCSP},(s)}(\tau) - A_{k,f}^{(s)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \\ &- 2\sum_{k=1}^{M} E[Z_k^{(p)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_k^{\text{DCSP},(p)}(\tau) - A_k^{(p)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \qquad (26) \\ &\leq K_1 + \hat{T}^2 M(1+|F^{(s)}|) \\ &- 2\sum_{k=1}^{M} \sum_{f\in F^{(s)}} E[Z_{f,k}^{(s)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_{f,k}^{\text{ALT1},(s)}(\tau) - A_{k,f}^{(s)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \\ &- 2\sum_{k=1}^{M} E[Z_k^{(p)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_k^{\text{ALT1},(p)}(\tau) - A_k^{(p)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \qquad (27) \\ &\leq K_1 + \hat{T}^2 M(1+|F^{(s)}|) \\ &- 2\sum_{k=1}^{M} \sum_{f\in F^{(s)}} E[Z_{f,k}^{(s)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_{f,k}^{\text{STAT},(s)}(\tau) - A_{k,f}^{(s)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \\ &- 2\sum_{k=1}^{M} E[Z_k^{(p)}(r) \sum_{\tau=t_{r,1}}^{t_{r,1}+T-1} \mu_k^{\text{STAT},(p)}(\tau) - A_k^{(p)}(\tau)|(Z_{f,k}^{(s)}(r))^T, (Z_k^{(p)}(r))^T] \qquad (28) \\ &\leq K_1 + \hat{T}^2 M(1+|F^{(s)}|) - 2T\epsilon \sum_{k=1}^{M} \sum_{f\in F^{(s)}} Z_{f,k}^{(s)}(r) - 2T\epsilon \sum_{k=1}^{M} Z_k^{(p)}(r) \qquad (29)\end{aligned}$$

The inequality (27) follows from Lemma IV. The inequality (28) follows because ALT1 maximizes $\psi^{\Phi}(t_{r,1})$ among all policies in Φ including STAT. Therefore the network is strongy stable by Theorem 4.1 in [17].

V. MCSP Algorithm

In this section we propose the MCSP algorithm under which the network controller dynamically determines the number of primary files to be cached in each period. Further, at every secondary base-station primary packet-requests are satisfied ahead of secondary ones. In this algorithm only a fixed set of non-interfering secondary base-stations, denoted without loss of generality as $\text{SB}_1, \ldots, \text{SB}_G$ respectively (where $G \leq M$), cooperatively transmit primary packets. In every period, the network controller determines the number of most popular primary files to cache by maximizing an expression consisting of two terms. The first term decreases with the number of cached primary

files while the second term increases with increasing number of cached primary files as well as length of secondary request-queues. The expression is a standard Lyapunov drift plus penalty expression for renewal frames, a brief description of which is presented next.

A. Renewal Frame Techniques for Shared Spectrum Networks

In renewal-frame based optimization methods, the time-line is partitioned into contiguous collection of slots with each collection referred to as a frame. Each frame is defined in terms of a "system state"; a new frame begins whenever the system-state is refreshed. The length of each frame is variable and is determined by the control decision taken at the beginning of the frame. At the beginning of each frame a controller selects a policy based on average rewards collected at previous frames as well as instantaneous values of some network parameters (such as queue-length). The selected policy remains fixed during the entire frame. Rewards are utility function of some network parameter of interest that we want to optimize (such as average power consumption, throughput etc.) subject to some feasibility constraints (such as stability of queues, maximum transmission power etc.). Near-optimal control decisions are obtained by maximizing a Lyapunov drift plus penalty expression. The drift term corresponds to the feasibility constraints while the penalty function corresponds to the utility function. Typically, such expressions represent a trade-off between satisfying the constraints versus achieving optimal value, with the extent of the trade-off determined by a constant penalty parameter.

In our case we define the system-state to be the total length of primary request-queues in PB and all secondary base-stations, similar to the way primary user's channel occupancy process was used as system-state in [8][2]. Each frame begins when the sum of queue-lengths of primary request-queues transitions from zero to a non-zero value. Fig 2 shows an example of such a partition. Note, each frame consists of two distinct phases: one in which at least one of the primary request-queues in the entire network in non-empty followed by one in which all primary request-queues in the network are empty.

B. Minimum Caching and Scheduling Policy

In the MCSP algorithm we attempt to minimize the number of primary files to be cached in each period subject to the constraint that all request-queues are stable and primary packet-requests are satisfied ahead of secondary ones in each secondary base-station. The algorithm is constructed using the renewal-frame structure with system-state being the sum of primary request-queue lengths in the whole network, as mentioned before. This allows us to prioritize transmission of primary packets over that of secondary packets in each cooperative base-station. Unlike typical renewal-frame based algorithms the MCSP algorithm makes caching decisions only at the beginning of every period rather than at the beginning of each frame[3] in the following manner.

At the beginning of r'th period (where $r = 1, \ldots,$) we first estimate the number of primary files ($\hat{n}_k^*(r)$) that would be be cached at the cooperative basestation SB_k by a renewal-frame based method *had* the beginning of the

[2]Note, in that work the authors did not study cooperative caching in cognitive radios.

[3]Recall, according to our system model, files can be cached only at beginning of the period.

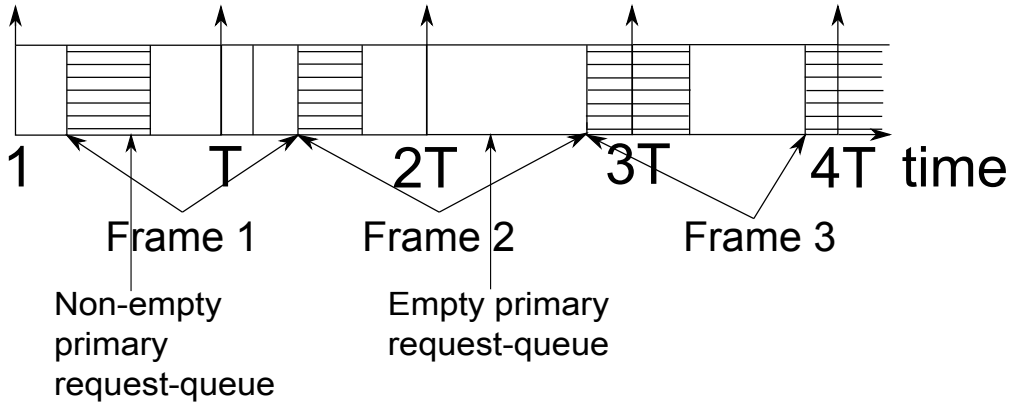


Fig. 2. Partition of timeline into frames. Shaded region shows slots for which at least one primary request-queue in the network is non-empty. Each frame consists of one such shaded region followed by period for which all primary request-queues are empty. An arrow indicates beginning of a new period.

period coincided with the beginning of a frame (the expression of $\hat{n}_k^*(r)$ is provided in (31)). The actual number of cached primary files ($n_k^*(r)$) at SB$_k$ can be higher than $\hat{n}_k^*(r)$. This is because, at the beginning of the period some primary request-queues in a secondary base-station might be non-empty and we need to ensure that all primary files corresponding to those queues are cached at that base-station. This is guaranteed as follows:

1) Suppose at least one primary request-queue was non-empty for all slots in previous period i.e. the $(r-1)$'th period. In this case, at $t_{r,1}$ cache at least as many most popular primary files at SB$_k$ as cached in the $(r-1)$'th period i.e., $n_k^*(r) = \max\{n_k^*(r-1), \hat{n}_k^*(r)\}$.
2) Otherwise, suppose at some slot $t_{r-1,j}$ (where $j = 1, 2, \ldots, T$) all primary request-queues become empty for the first time. Then from slot $t_{r-1,j+1}$ till the end of the $(r-1)$'th period SB$_k$ only admits primary file requests corresponding to $\hat{n}_k^*(r-1)$ most popular primary files[4]. At $t_{r,1}$ SB$_k$ caches atleast $\hat{n}_k^*(r-1)$ most popular primary files i.e., $n_k^*(r) = \max\{\hat{n}_k^*(r-1), \hat{n}_k^*(r)\}$, if at least one primary queue is non-empty at $t_{r,1}$; otherwise, it caches $\hat{n}_k^*(r)$ most popular primary files i.e, $n_k^*(r) = \hat{n}_k^*(r)$.

The overall MCSP algorithm consists of the following steps in r'th period:

1) *Caching Policy:* At the beginning of the period for every cooperative secondary base-station SB$_k$ (where $1 \leq k \leq G$) compute $\hat{n}_k^*(r)$ and then $n_k^*(r)$. In SB$_l$ (where $1 \leq l \leq M$) cache the $n_l^*(r)$ most popular primary files, with $n_l^*(r)$ being zero for non-cooperative base-stations; in the remaining positions cache the $B - n_l^*(r)$ secondary files with longest request-queue lengths in SB$_l$ at $t_{r,1}$. We denote the set of secondary files cached at SB$_l$ in r'th period as $H_l^{(s)}(r)$.

[4]This requires small modification in the network model described in Section II. Specifically, the network controller may not admit primary file-requests at a cooperative secondary base-station even if the file is present in its cache.

2) *Request Admission and Scheduling Policy for Primary Users:* Suppose, atleast one primary request-queue is non-empty at $t_{r,1}$. Consider all slots in r'th period until either the end of the period or the first slot in which all primary queues become non-empty, whichever is earliest. In these slots, for every $k = 1, \ldots, G$, at SB_k admit requests corresponding to $n_k^*(r)$ most popular primary files from the $\phi_k^{(p)}$ primary users. Suppose at slot t_{r,j^*} all primary request-queues become empty for the first time. Then from slots t_{r,j^*} until end of the period, in SB_k admit requests corresponding to $\hat{n}_k^*(r)$ most popular primary files from the $\phi_k^{(p)}$ primary users. In any slot $t_{r,j}$ (where $j = 0, 1, \ldots, T$) transmit the packet corresponding to the HOL packet-request at the primary request-queue of highest length SB_k. If the primary queues in all secondary base-stations are empty and the request-queue in PB is non-empty, transmit the packet corresponding to HOL packet-request at the request-queue of highest length in PB.

3) *Scheduling Policy for Secondary Users:* Suppose, at $t_{r,j}$ all the primary request-queues of cooperative base-station SB_k are empty but some other cooperative base-station is transmitting a primary packet. Then identify the request-queue of the cached secondary file with largest instantaneous backlog in SB_k and then transmit the packet corresponding to its HOL packet-request i.e., $\mu_{\text{SB}_k,f}(t_{r,j}) = 1$ if $f \in \text{argmax}_{f \in H_k^{(s)}(r)} U_{f,k}^{(s)}(t_{r,j})$. In case the request-queue is empty, transmit a dummy packet.
Otherwise, if all the primary request-queues are empty at slot $t_{r,j}$ then schedule secondary packet transmission according to a modified backpressure algorithm. First find the activation vector $E^{\text{MCSP}}(t_{r,j})$ as

$$E^{\text{MCSP}}(t_{r,j}) \in \underset{E \in \tilde{E}}{\text{argmax}} \left(\left(\max_{f \in H_l^{(s)}(r)} U_{f,k}^{(s)}(t_{r,j}) \right)_{l=1}^{M} \right)^T E \tag{30}$$

If base-station SB_l is scheduled to transmit then transmit a packet corresponding to the request-queue in SB_l of the cached secondary file with largest instantaneous backlog i.e. $\mu_{\text{SB}_l,f}(t_{r,j}) = 1$ if $f \in \text{argmax}_{f \in H_l^{(s)}(r)} U_{f,l}^{(s)}(t_{r,j})$. In case the request-queue is empty, transmit a dummy packet.

C. Expression to Compute $\hat{n}_k^*(.)$ at Every Cooperative Secondary Base-station SB_k

For every $r = 1, \ldots,$ and $k = 1, \ldots, G$ the variable $\hat{n}_k^*(r)$ is computed by maximizing a drift plus penalty expression as follows:

$$\begin{aligned} (\hat{n}_1^*(r), \ldots, \hat{n}_G^*(r))^T \quad \in \quad & \underset{(n_1,\ldots,n_G)^T : 0 \le n_k \le B-1}{\text{argmax}} \; -V \sum_{k=1}^{G} n_k \\ + \quad & \frac{\sum_{k=1}^{G} \alpha_k(n_1, \ldots, n_G) \hat{U}_k^{(s)}(r) + \frac{1}{\lambda_{\text{tot}}} \left(\left(\hat{U}_l^{(s)}(r) \right)_{l=1}^{M} \right)^T E^*(t_{r,1})}{\kappa(n_1, \ldots, n_G)} \end{aligned} \tag{31}$$

where

$$\kappa(n_1,\ldots,n_G) = \frac{\nu(n_1,\ldots,n_G)}{\lambda_{\text{tot}}(1-\nu(n_1,\ldots,n_G))}+\frac{1}{\lambda_{\text{tot}}} \tag{32}$$

$$\lambda_{\text{tot}} = \sum_{k=1}^{N^{(p)}}\lambda_k^{(p)}C \tag{33}$$

$$\nu(n_1,\ldots,n_G) = \frac{\lambda_{\text{tot}}-\lambda_{\text{rel}}(n_1,\ldots,n_M)}{p}+\nu_0(n_1,\ldots,n_G) \tag{34}$$

$$\nu_0(n_1,\ldots,n_G) = \sum_{k_1=1}^{G}\hat{\nu}_{k_1}(n_{k_1})-\sum_{k_1,k_2:1\le k_1<k_2\le G}\hat{\nu}_{k_1}(n_{k_1})\hat{\nu}_{k_2}(n_{k_2})+$$

$$\ldots+(-1)^G\sum_{k_1,\ldots,k_G:1\le k_1<k_2<\ldots<k_G\le G}\hat{\nu}_{k_1}(n_{k_1})\ldots\hat{\nu}_{k_G}(n_{k_G}) \tag{35}$$

$$\lambda_{\text{rel}}(n_1,\ldots,n_G) = \sum_{k=1}^{G}\sum_{\text{PU}_j\in\phi_k^{(p)}}\sum_{n=1}^{n_k}\tilde{P}_n^{(p)}\lambda_j^{(p)}C \tag{36}$$

$$\hat{\nu}_k(n_k) = \begin{cases}\sum\limits_{j:\text{PU}_j\in\phi_k^{(p)}}\lambda_j^{(p)}\sum\limits_{n=1}^{n_k}\tilde{P}_n^{(p)}C, & \forall 1\le k\le G, 1\le n_k\le B-1\\ 0, & \text{otherwise}\end{cases} \tag{37}$$

$$\alpha_k(n_1,\ldots,n_G) = \kappa(n_1,\ldots,n_G)\{\nu_0(n_1,\ldots,n_G)-\hat{\nu}_k(n_k)\} \quad \forall 1\le k\le G \tag{38}$$

$$\hat{U}_l^{(s)}(r) \in \max_{f\in F^{(s)}}U_{f,k}^{(s)}(t_{r,1}) \quad \forall l=1,\ldots,M \tag{39}$$

In (31) the first term $-V\sum_{k=1}^{G}n_k$ corresponds to the penalty function that decreases with increasing number of cached primary files with V being a fixed penalty parameter. Higher V implies lower number of primary files to be cached and vice-versa. The second term represents conditional drift of a Lyapunov function of secondary request-queue lengths at the beginning of r'th period divided by the expected duration of a frame. The term is non-decreasing with higher number of cached primary files as well as higher secondary request-queue lengths. The conditional drift is calculated under the assumption that a new frame begins at slot $t_{r,1}$ and caching decisions happen only at beginning of the frame rather than at beginning of each period. Next we describe the derivation of the conditional drift term and the denominator $\kappa(n_1,\ldots,n_G)$.

First we consider the denominator $\kappa(n_1,\ldots,n_G)$ which is the expected frame-length if n_k (where $1\le k\le G$) most popular primary files are cached at every SB_k at the beginning of the frame. Each frame consists of three distinct segments: one in which all primary request-queues are empty, one in which at least one primary request-queue in some cooperative secondary base-station is non-empty and one in which all primary request-queues in the cooperative secondary base-stations are empty but at least one of the primary request-queues in PB is non-empty. Clearly, duration of the first segment is a geometric random variable with parameter equal to the total generation rate of primary user requests λ_{tot} and its mean is therefore, $\frac{1}{\lambda_{\text{tot}}}$. Expected duration of the second segment, is the average proportion of time at least one secondary base-station is transmitting a

primary packet multiplied by the expected length of the frame. Now, the proportion of the time each secondary base-station transmits a primary packet is independent of each other and according to Little's law its mean equals $\hat{\nu}_k(n_k)$ for SB_k (for a stable system). The expected proportion of time some secondary base-station transmits a primary packet therefore equals $\nu_0(n_1,\ldots,n_G)$ in (35). Similarly, the expected duration of the third segment is the proportion of time PB transmits a primary packet multiplied by the expected length of the frame. Again, according to Little's law this expected proportion is obtained as $\frac{\lambda_{tot}-\lambda_{rel}(n_1,\ldots,n_M)}{p}$ since p is the probability of successful transmission from PB and λ_{rel} denotes the average rate of primary packet transmission by secondary base-stations. Note, $\nu(n_1,\ldots,n_G)$ denotes the average proportion of time some base-station is transmitting a primary packet. By adding the expected duration of the three segments, noting that the sum equals $\kappa(n_1,\ldots,n_G)$ and then equating both sides we obtain the value of $\kappa(n_1,\ldots,n_G)$ in (32).

Next we consider the numerator:
$\sum_{k=1}^{G}\alpha_k(n_1,\ldots,n_G)\hat{U}_k^{(s)}(r)+\frac{1}{\lambda_{\text{tot}}}\left(\left(\hat{U}_l^{(s)}(r)\right)_{l=1}^{M}\right)^T E^*(t_{r,1})$. Given $n_1,\ldots,n_G$ most popular primary files are cached at $\text{SB}_1,\ldots,\text{SB}_G$ respectively this term corresponds to conditional drift of a Lyapunov function of secondary request-queue lengths over a frame assuming that $t_{r,1}$ is the first slot in the frame. It is the solution to the problem of maximizing the sum of the dot-product of the vector of secondary request-queue lengths at $t_{r,1}$ i.e.,$\left(\left(U_{f,l}^{(s)}(t_{r,1})\right)_{l,f}\right)^T$ and the corresponding feasible transmission-rate offered at all slots in the frame i.e., $\left(\left(\mu_{f,\text{SB}_l}^{(s)}(.)\right)_{l,f}\right)^T$. The maximization is performed among all possible scheduling policies under the assumption that

a) $t_{r,1}$ is the first slot in a new frame and
b) no non-cooperative secondary base-station simultaneously transmits secondary packets when some SB_k is transmitting a primary packet.

Accordingly, the term $\sum_{k=1}^{G}\alpha_k(n_1,\ldots,n_G)\hat{U}_k^{(s)}(r)$ represents the contribution of secondary packet transmissions from cooperative base-stations towards this drift expression, during those slots in which some secondary base-station is transmitting a primary packet. This follows as the term $\alpha_k(n_1,\ldots,n_G)$ in (38) represents the expected duration of those slots in the frame in which some cooperative secondary base-station other than SB_k is transmitting a primary packet. The term $\frac{1}{\lambda_{\text{tot}}}\left(\left(\hat{U}_l^{(s)}(r)\right)_{l=1}^{M}\right)^T E^*(t_{r,1})$ represents the contribution of secondary packet transmissions from all base-stations towards this drift expression during those slots in which no base-station is transmitting any primary packet.

For the special case when all secondary base-stations are non-interfering, 31 can be simplified and $\hat{n}_k^*(r)$ is obtained using a simple expression for every $k=1,\ldots,G$:

$$\begin{aligned}\hat{n}_k^*(r) \quad \in \quad & \underset{n_k:0\le n_k\le B-1}{\operatorname{argmax}} \; -Vn_k \\ & +\frac{\hat{\nu}_k(n_k)}{p}\sum_{k'\neq k}\hat{U}_{k'}^{(s)}(r)+\hat{\nu}_k(n_k)(\frac{1}{p}-1)\hat{U}_k^{(s)}(r) \end{aligned} \tag{40}$$

Next we find a guaranteed stability region for the MCSP algorithm.

D. Guaranteed Stability Region

Let $\Lambda_0^{(p)}$ denote the set of primary request generation-rate vectors which can be satisfied even in absence of co-operation. Consider the region Λ^{MCSP} defined as the set $\left\{(\boldsymbol{\lambda}^{(p)}, \boldsymbol{\lambda}^{(s)}) : \boldsymbol{\lambda}^{(p)} \in \Lambda_0^{(p)}, \boldsymbol{\lambda}^{(s)} \in \quad \text{Interior}\left(\Lambda^{(s)}(\boldsymbol{\lambda}^{(p)})\right)\right\}$ where $\Lambda^{(s)}(\boldsymbol{\lambda}^{(p)})$ consists of all $\boldsymbol{\lambda}^{(s)}$ that satisfies

$$C\lambda_k^{(s)}(n_k) \leq \begin{cases} R_k, & \forall G+1 \leq k \leq M \\ R_k + \nu(B-1,\ldots,B-1) - \hat{\nu}_k(B-1), & \text{otherwise} \end{cases} \tag{41}$$

for some $(R_1, R_2, \ldots, R_M)^T \in \Gamma(\boldsymbol{\lambda}^{(p)})$ where

$$\Gamma(\boldsymbol{\lambda}^{(p)}) \quad = \quad \{1 - \nu(B-1,\ldots,B-1)\}\,\textbf{conv}(\tilde{E}) \tag{42}$$

For a given primary request generation-rate vector $\boldsymbol{\lambda}^{(p)}$ the set $\Lambda^{(s)}(\boldsymbol{\lambda}^{(p)})$ defines the set of all secondary request-generation rates that can be supported if $B-1$ most popular primary files are cached at each cooperative secondary base-station in each period and, if only cooperative secondary base-stations can simultaneously transmit secondary packets when some cooperative base-station is transmitting a primary packet. Note, $\nu(B-1,\ldots,B-1) - \hat{\nu}_k(B-1)$ denotes the probability of the event: "SB$_k$ does not transmit any primary packet but some cooperative base-station transmits a primary packet". The term $1-\nu(B-1,\ldots,B-1)$ denotes the probability of the event: " no one is transmitting any primary packet".

theorem Under the MCSP algorithm the network is stable for all request generation-rate vectors in the set Λ^{MCSP}.

Since the conditional drift term in (31) increases with increasing secondary request-queue lengths, there exists a finite constant K_0' s.t. if $\hat{U}_k^{(s)}(r) > K_0'$ for even one SB$_k$ then (31) is maximized when $n_1 = n_2 = \ldots = n_G = B-1$.

We define $\tilde{Z}_{f,k}^{(s)}(r) \stackrel{\triangle}{=} U_{f,k}^{(s)}(\tilde{t}_{r,1})$ where $\tilde{t}_{r,j}$ denotes the j'th (where $j = 1,\ldots,\tilde{t}_{r+1,1}-1$) slot in r'th ($r = 1,2,\ldots$) frame for every $f \in F^{(s)}$. Let the first frame begin at slot $\tilde{T}_0$. We first consider the case when $\tilde{T}_0 < \infty$ and $\tilde{Z}_{f,k}^{(s)}(r) > K_0' + T$ for at least one $f \in F^{(s)}$ and SB$_k$ where $k = 1,\ldots,M$. The case where $\tilde{T}_0 = \infty$ is equivalent to special case of scheduling for the general network without any primary packet arrival and is thereby skipped.

Consider an algorithm ALT2 that, at the beginning of r'th frame caches $B-1$ most popular primary files and the secondary file with maximum request-queue length at each cooperative base-station and B secondary files with maximum request-queue lengths in other secondary base-stations. Caching occurs only at the beginning of every frame. Scheduling of primary and secondary packets take place as in MCSP. Clearly, for given vector of secondary request-queue lengths at beginning of r'th frame $(\tilde{Z}_{f,k}^{(s)}(r))$ ALT2 maximizes $E[\sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \sum_{k=1}^{M} \sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r)\mu_{f,k}^{(s),\Phi}(\tau)]$ among all policies Φ_2 that

1) caches $B-1$ primary files and one secondary file *only* in the beginning of r'th frame in every cooperative secondary base-station; caches B secondary files in other secondary base-stations.

2) Only cooperative base-stations can simultaneously transmit secondary packets when some cooperative secondary base-station s is transmitting a primary packet. No base-station transmits a secondary packet if one of its primary request-queues is non-empty.

Note, the length of r'th frame remains the same under both ALT2 and MCSP.

Assume length of request-queue of file f at beginning of r'th frame, $U_{k,f}^{(s)}(\tilde{t}_{r,1})$ is given. Since the number of arrivals and departure of secondary requests at an SB_k in every slot is bounded by $N^{(s)}$ we have for all $\tau \in [\tilde{t}_{r,1}, \ldots, \tilde{t}_{r+1,0} - 1]$

$$U_{f,k}^{\text{ALT2},(s)}(\tau), U_{f,k}^{\text{MCSP},(s)}(\tau) \geq U_{f,k}^{(s)}(\tilde{t}_{r,1}) - \tau N^{(s)} \tag{43}$$

$$U_{f,k}^{\text{ALT2},(s)}(\tau), U_{f,k}^{\text{MCSP},(s)}(\tau) \leq U_{f,k}^{(s)}(\tilde{t}_{r,1}) + \tau N^{(s)} \tag{44}$$

Let $\mathbf{U^{(s)}}(\tau)$ denote the vector of secondary request-queue lengths at slot τ. Let $\boldsymbol{\mu}^{\Phi,(s)}(\tau)$ denote the vector of secondary file transmissions at slot τ under policy Φ. We have for every τ in r'th frame,

$$\begin{aligned}
(\mathbf{U^{(s)}}(\tilde{t}_{r,1}))^T \boldsymbol{\mu}^{\text{MCSP},(s)}(\tau) &\geq (\mathbf{U}^{\text{MCSP},(s)}(\tau))^T \boldsymbol{\mu}^{\text{MCSP},(s)}(\tau) - \tau M |F^{(s)}| N^{(s)} && (45)\\
&\geq (\mathbf{U}^{\text{MCSP},(s)}(\tau)^T \boldsymbol{\mu}^{\text{ALT2},(s)}(\tau) - \tau M |F^{(s)}| N^{(s)} && (46)\\
&\geq (\mathbf{U}^{(s)}(\tilde{t}_{r,1}))^T \boldsymbol{\mu}^{\text{ALT2},(s)}(\tau) - 2\tau M |F^{(s)}| N^{(s)} && (47)
\end{aligned}$$

The relation (46) follows from definition of MCSP. Summing over all τ we obtain

$$\begin{aligned}
E[\sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} (\mathbf{U}^{(s)}(\tilde{t}_{r,1}))^T \boldsymbol{\mu}^{\text{MCSP},(s)}(\tau) | \mathbf{U^{(s)}}(\tilde{t}_{r,1})] \geq E[\sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} (\mathbf{U}^{(s)}(\tilde{t}_{r,1}))^T \boldsymbol{\mu}^{\text{ALT2},(s)}(\tau)\\
- E[(\tilde{t}_{r+1,0} - \tilde{t}_{r,1})^2] M |F^{(s)}| N^{(s)} && (48)
\end{aligned}$$

Similar to the proof in [8] it can be shown that there exists finite non-zero constants W_1 and W_2 s.t. $W_0 \geq E[\tilde{t}_{r+1,0} - \tilde{t}_{r,1}] > W_1$ and $E[(\tilde{t}_{r+1,0} - \tilde{t}_{r,1})^2] < W_2$ irrespective of policy.

Next we define the Lyapunov function $L(r) \stackrel{\triangle}{=} \sum_{k=1}^{M} \sum_{f \in F^{(s)}} (\tilde{Z}_{f,k}^{(s)}(r))^2$. The conditional drift under policy MCSP is given as,

$$\Delta(r) \stackrel{\triangle}{=} E[L(r+1) - L(r) | \mathbf{U^{(s)}}(\mathbf{r})] \tag{49}$$

Then

$$\begin{aligned}
\Delta(r) &\leq E[(\tilde{t}_{r+1,0} - \tilde{t}_{r,1})^2] 2M |F^{(s)}| (N^{(s)})^2\\
&\quad - E[\sum_{k=1}^{M} \sum_{f \in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \{\mu_{f,k}^{\text{MCSP},(s)}(\tau) - A_{k,f}^{(s)}(\tau)\} | \mathbf{U^{(s)}}(\mathbf{r})] && (50)\\
&\leq 4M |F^{(s)}| W_2 (N^{(s)})^2 - E[\sum_{k=1}^{M} \sum_{f \in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \{\mu_{f,k}^{\text{ALT2},(s)}(\tau)\\
&\quad - A_{k,f}^{(s)}(\tau)\} | \mathbf{U^{(s)}}(\mathbf{r})] && (51)
\end{aligned}$$

For any secondary request-generation-rate vector $\boldsymbol{\lambda}^{(s)} \in$ Interior$(\Lambda(\boldsymbol{\lambda}^{(p)}))$ there exists $\epsilon > 0$ s.t. $\boldsymbol{\lambda}^{(s)}+\boldsymbol{\epsilon} \in \Lambda(\boldsymbol{\lambda}^{(p)})$. For secondary request-generation rate-vector $\boldsymbol{\lambda}^{(s)} + \boldsymbol{\epsilon}$ there exists a stabilizing stationary policy STAT that caches $B-1$ primary files at the beginning of every frame and stochastically selects the secondary file to cache at each frame, independent of queue-length. Now since ALT2 maximizes $\sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \sum_{k=1}^{M} \sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) E[\mu_{\mathrm{SB}_k,f}^{(s),\Phi}(\tau)|\mathbf{U^{(s)}}(\tilde{t}_{r,1})]$ among all policies Φ_2 of which STAT is one, we have from (51)

$$\begin{aligned}
\Delta(r) &\leq 4M|F^{(s)}|W_2(N^{(s)})^2 - E[\sum_{k=1}^{M}\sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \{\mu_{f,k}^{\mathrm{STAT},(s)}(\tau) \\
&\quad -A_{k,f}^{(s)}(\tau)\}] \qquad (52)\\
&\leq 4M|F^{(s)}|W_2(N^{(s)})^2 - \epsilon W_1 \sum_{k=1}^{M}\sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \qquad (53)
\end{aligned}$$

Now consider the case when $\tilde{Z}_{f,k}^{(s)}(r) \leq K_0' + T$ for every $k = 1, \ldots, M$ and $f \in F^{(s)}$. Then the conditional drift under MCSP policy over r'th frame is,

$$\begin{aligned}
\Delta(r) &\leq E[(\tilde{t}_{r+1,0} - \tilde{t}_{r,1})^2]2M|F^{(s)}|(N^{(s)})^2 \\
&\quad -E[\sum_{k=1}^{M}\sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} \{\mu_{f,k}^{\mathrm{MCSP},(s)}(\tau) - A_{k,f}^{(s)}(\tau)\}|\mathbf{U^{(s)}}(\mathbf{r})] \qquad (54)\\
&\leq 4M|F^{(s)}|W_2(N^{(s)})^2 \\
&\quad +E[\sum_{k=1}^{M}\sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \sum_{\tau=\tilde{t}_{r,1}}^{\tilde{t}_{r+1,0}-1} A_{k,f}^{(s)}(\tau)|\mathbf{U^{(s)}}(\mathbf{r})] \qquad (55)\\
&\leq 4M|F^{(s)}|W_2(N^{(s)})^2 + (K_0'+T)M|F^{(s)}|E[\tilde{t}_{r+1,0} - \tilde{t}_{r,1}] \qquad (56)\\
&\leq K_3 \qquad (57)
\end{aligned}$$

where K_3 is a finite constant since $E[\tilde{t}_{r+1,0} - \tilde{t}_{r,1}], K_0'$ are bounded. Combining (53) and (57) we get

$$\Delta(r) \leq 2M|F^{(s)}|W_2(N^{(s)})^2 + K_3 - \epsilon W_1 \sum_{k=1}^{M}\sum_{f\in F^{(s)}} \tilde{Z}_{f,k}^{(s)}(r) \qquad (58)$$

Therefore the network is stable by Theorem 4.1 in [17].

Clearly, the guaranteed stability region is greater than the capacity region without cooperation. However, with respect to request generation-rates from primary users, the guaranteed stability region under the MCSP algorithm remains the same as for the case without cooperation.

Special Case: We observe that when all secondary base-stations are non-interfering i.e., G=M, the guaranteed stability region and the set Λ are related as follows.

When $G = M$, the set $\Lambda^{(s)}(\boldsymbol{\lambda}^{(p)})$ reduces to the following

$$\Lambda^{(s)}(\boldsymbol{\lambda}^{(p)}) = \{\boldsymbol{\lambda}^{(s)} : (\boldsymbol{\lambda}^{(p)}, \boldsymbol{\lambda}^{(s)}) \in \Lambda\} \qquad (59)$$

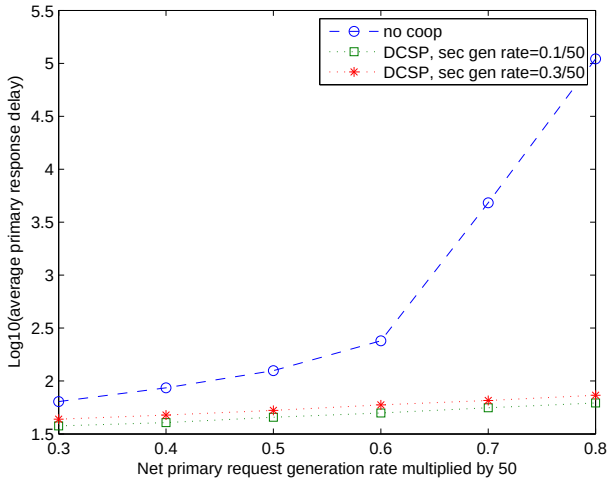


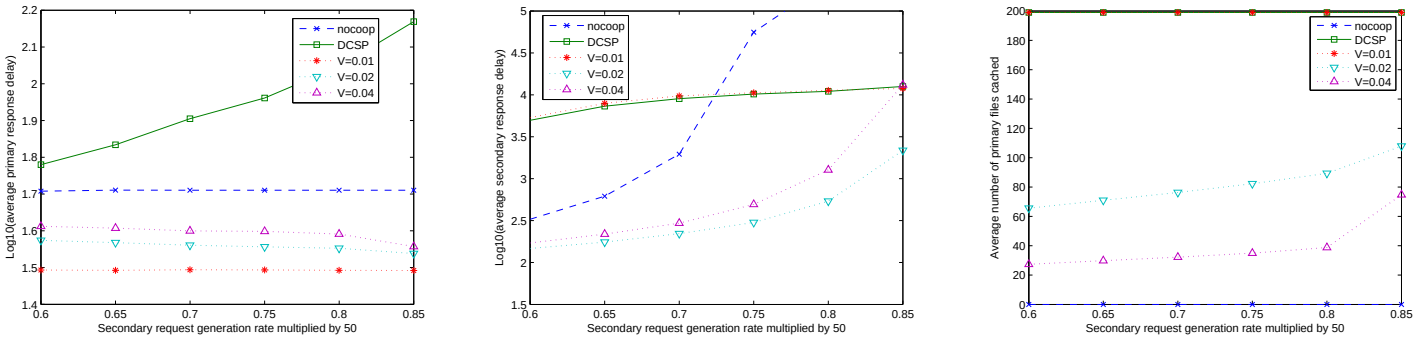
Fig. 3. Plot of base-10 logarithms of time-averaged primary response delay versus net primary request generation rate $50\lambda^{(p)}$, when $\lambda^{(s)}$ is $\frac{0.1}{50}$, $\frac{0.3}{50}$ under no cooperation and the DCSP algorithm.

VI. Simulation Results

In this section we observe the performance of the DCSP and MCSP algorithms through simulations in C-programming language. We consider a network with 3 small secondary base-stations that are non-interfering to each other. We assume there is one primary and secondary user in each of these 3 small cells; there is no other primary user in the network. We consider symmetric request generation: requests generation rates by primary (and secondary) users in each small cell are equal. For convenience, we denote the sum of request-generation rates of all primary users in the entire network as $\lambda^{(p)}$ and the request generation rates of every secondary user as $\lambda^{(s)}$ respectively. We use the following parameters: cache size (B) is 200, number of successful packet transmissions corresponding to a given file transmission (C) is 50, caching period (T) is 100, probability of successful transmission by primary base-station (p) is 0.7. Total number of primary and secondary files i.e., $|F^{(p)}|$ and $|F^{(s)}|$ respectively, are both equal to 400. There is one popularity state and the popularity of primary and secondary files have a Zipf distribution with parameter 0.8. All simulations are run for 2,000,000 slots.

For comparison we use a non-cooperative protocol similar to DCSP wherein all primary user requests are served by the primary base-station (PB). The algorithm is described as follows. At every slot, PB transmits a packet corresponding to the request-queue in PB of highest length. A secondary base-station transmits only if all request-queues in PB are empty. Every secondary base-station caches B files with highest request-queue lengths at the beginning of every cache refresh period. In every slot it transmits a packet corresponding to the requeust-queue of highest length, among the set of all request-queues whose files are currently cached in that base-station.

In Figure 3 we compare the base-10 logarithm of time-averaged response delay for primary packets under both



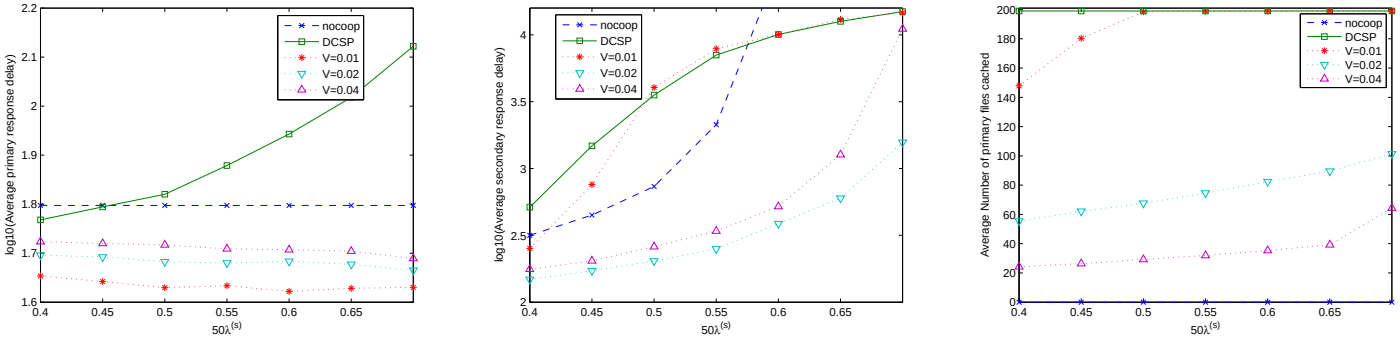
(a) Average primary response delay (b) Average secondary response delay (c) Average number of primary files cached

Fig. 4. Plot of base-10 logarithms of time-averaged primary and secondary response delay and the average number of primary files cached versus $50\lambda^{(s)}$ when $\lambda^{(p)}$ is $\frac{0.2}{50}$ under no cooperation, DCSP algorithm, and the MCSP algorithm for parameter V=0.01, 0.02 and 0.04.

DCSP and the case without cooperation for different values of primary user request generation rates[5]. The response delay for every transmitted packet is measured as the time between generation of the corresponding file request by a user and the slot when the packet is successfully transmitted to that user. Average primary (resp. secondary) response delay is obtained by averaging response delays of all primary (resp. secondary) packets that are transmitted during the simulation run-time. Plotting base-10 logarithm values allow us to view relatively high values of the time-averaged delay alongside smaller values. Two values of $\lambda^{(s)}$: $\frac{0.1}{50}$ and $\frac{0.3}{50}$ are used; for each value of $\lambda^{(s)}$, the value of $\lambda^{(p)}$ is varied from $\frac{0.1}{50}$ to $\frac{0.8}{50}$. Note, maximum $\lambda^{(p)}$ that can be satisfied without cooperation is $\frac{0.7}{50}$; as a result average delay is very high without cooperation for $\lambda^{(p)} = \frac{0.7}{50}$ and $\frac{0.8}{50}$. Fom Figure 3 we observe that for these primary user request generation rates, average primary response delay is lowered under DCSP since the system is stable under DCSP. The difference in average delays is more pronounced as $\lambda^{(p)}$ increases or $\lambda^{(s)}$ decreases.

In Fig. 4 we plot base-10 logarithm of time-averaged primary and secondary response delay and the average number of primary files cached versus $\lambda^{(s)}$ when $\lambda^{(p)}$ is $\frac{0.2}{50}$. We plot these values for the case without cooperation, under the DCSP algorithm, and under the MCSP algorithm with different values of the penalty parameter V. From Fig. 4a we observe the primary users do not benefit under the DCSP algorithm as compared to the case of no cooperation. This is due to relatively high request generation rates by secondary users as compared to $\lambda^{(p)}$ because of which primary user requests are not served very often under the DCSP algorithm. However, under the MCSP algorithm, average primary response delay is small as primary user requests are served with higher priority by secondary users over secondary user requests. Further, this delay improves with higher $\lambda^{(s)}$. This is because with higher $\lambda^{(s)}$ there is higher backlog of secondary user requests queued at each secondary base-station. As a result each secondary base-station caches more primary files with increasing $\lambda^{(s)}$. We observe from Fig. 4b that when $\lambda^{(s)}$ is less than 0.75, average secondary response delay decreases under cooperation when DCSP or MCSP with

[5]Note, response delay is directly proportional to length of request-queues, by Little's law. Lower request-queue length implies lower response delay and vice-versa.



(a) Average primary response delay (b) Average secondary response delay (c) Average number of primary files cached

Fig. 5. Plot of base-10 logarithms of time-averaged primary and secondary response delay and the average number of primary files cached versus $50\lambda^{(s)}$ when $\lambda^{(p)}$ is $\frac{0.2}{50}$, $\lambda_0^{(p)}$ is $\frac{0.1}{50}$ under no cooperation, DCSP algorithm, and the MCSP algorithm for parameter V=0.01, 0.02 and 0.04.

penalty parameter $V = 0.01$ are used. This is because, as observed from Fig. 4c under both these algorithms 199 out of every 200 cache positions at each secondary base-station are filled by primary files[6]. This in turn reduces opportunities for secondary base-stations to satisfy secondary file requests. For higher $\lambda^{(s)}$ however, the system is unstable without cooperation and both these algorithms perform better than the case without cooperation. With respect to average secondary response delay MCSP with $V = 0.02$ outperforms other algorithms. As can be observed from Fig. 4c for this parameter the MCSP algorithm strikes a balance between number of primary files cached versus system stability and is beneficial for both primary and secondary users in terms of average response delay. Similarly from Fig. 5 we observe that MCSP with $V = 0.5$ outperforms DCSP and MCSP with V=0.01, 0.05 and 0.1. Note, in Fig. 4b and 5b, we do not plot average secondary response delay for the case without cooperation when $\lambda^{(s)}$ is greater than 0.75 and 0.45 respectively because the secondary request-queues are unstable for those parameters. In Fig. 5 we study the behavior of the above algorithms when there are primary users which are not in any small-cell. We use the same network parameters as in Fig. 4; in addition we consider one primary user, not within transmission range of any secondary base-station, which requests contents at rate $\frac{0.1}{50}$. We observe similar results as in Fig. 4. In this case MCSP with $V = 0.02$ outperforms DCSP and MCSP with V=0.01 and 0.04. Note, in Fig. 4b and 5b, we do not plot average secondary response delay for the case without cooperation when $\lambda^{(s)}$ is greater than 0.75 and 0.55 respectively because the secondary request-queues are unstable for those parameters.

VII. Conclusion

In this work we studied cooperative caching in cognitive radio networks. Using Lyapunov drift techniques we proposed a throughput-optimal caching and scheduling policy. We also proposed an alternative algorithm whereby

[6]Recall, lower V implies higher number of primary files to be cached with exactly 199 primary files being cached when V is 0.

secondary base-stations serve primary files requests with higher priority. Both algorithms increase the set of request generation rate vectors which can be served by the base-stations from the case with no cooperation.

REFERENCES

[1] I. Maric, R. Yates, and G. Kramer, "Capacity of interference channels with partial transmitter cooperation," *Information Theory, IEEE Transactions on*, vol. 53, no. 10, pp. 3536–3548, 2007.

[2] I. Marić, A. Goldsmith, G. Kramer *et al.*, "On the capacity of interference channels with one cooperating transmitter," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 405–420, 2008.

[3] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *Information Theory, IEEE Transactions on*, vol. 51, no. 9, pp. 3037–3063, 2005.

[4] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Stable throughput of cognitive radios with and without relaying capability," *Communications, IEEE Transactions on*, vol. 55, no. 12, pp. 2351–2360, 2007.

[5] I. Krikidis, N. Devroye, and J. Thompson, "Stability analysis for cognitive radio with multi-access primary transmission," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 1, pp. 72–77, 2010.

[6] A. Fanous and A. Ephremides, "Stable throughput in a cognitive wireless network," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 3, pp. 523–533, 2013.

[7] A. El-Sherif, A. Sadek, and K. Liu, "Opportunistic multiple access for cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 4, pp. 704–715, 2011.

[8] R. Urgaonkar and M. Neely, "Opportunistic cooperation in cognitive femtocell networks," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 3, pp. 607–616, 2012.

[9] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *INFOCOM, 2012 Proceedings IEEE*, pp. 1107–1115.

[10] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Wireless device-to-device communications with distributed caching," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, 2012, pp. 2781–2785.

[11] N. Golrezaei, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 7077–7081.

[12] J. Zhao, W. Gao, Y. Wang, and G. Cao, "Delay-constrained caching in cognitive radio networks," in *INFOCOM, 2014 Proceedings IEEE*, pp. 2094–2102.

[13] J. Zhao and G. Cao, "Spectrum-aware data replication in intermittently connected cognitive radio networks," in *INFOCOM, 2014 Proceedings IEEE*, pp. 2238–2246.

[14] M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-aware caching and traffic management in content distribution networks," in *INFOCOM, 2011 Proceedings IEEE*, pp. 2858–2866.

[15] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless broadcast networks with elastic and inelastic traffic," in *WiOpt*, May 2011, pp. 125–132.

[16] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000.

[17] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.