# Stochastic Content-Centric Multicast Scheduling for Cache-Enabled Heterogeneous Cellular Networks

Bo Zhou[1], Ying Cui[1] and Meixia Tao[1,2]
[1]Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]Cooperative Medianet Innovation Center, Shanghai, China
{b.zhou, cuiying, mxtao}@sjtu.edu.cn

## ABSTRACT

Caching at small base stations (SBSs) has demonstrated significant benefits in alleviating the backhaul requirement in heterogeneous cellular networks (HetNets). While many existing works focus on what contents to cache at each SBS, an equally important but much less investigated problem is what contents to deliver given the cache status and user requests. In this paper, we study the optimal content delivery strategy in cache-enabled HetNets by taking into account the inherent multicast capability of wireless medium. We establish a content-centric request queue model and then formulate a stochastic multicast scheduling problem to jointly minimize the average network delay and power costs. This stochastic optimization problem is an infinite horizon average cost Markov decision process (MDP), which is well known to be challenging. By using *relative value iteration algorithm* and the special properties of the request queue dynamics, we characterize some properties of the value function of the MDP. Based on these properties, we show that the optimal multicast scheduling policy, which is adaptive to the request queue state, is of the threshold type. Finally, we propose a low complexity optimal algorithm by exploiting the structural properties of the optimal policy.

## CCS Concepts

•**Networks** → **Network performance modeling; Wireless access networks;** *Network design principles; Network management;*

## Keywords

HetNets, wireless caching, content-centric, multicast, Markov decision process, structural properties, queueing.



*CCDWN'15, December 01-04, 2015, Heidelberg, Germany*

DOI: http://dx.doi.org/10.1145/2836183.2836190

## 1. INTRODUCTION

The rapid proliferation of smart mobile devices has trigged an unprecedented growth of the global mobile data traffic. One promising approach to meet the dramatic traffic growth is to deploy small base stations (SBSs) together with traditional macro base stations (MBSs) in a heterogeneous network paradigm [1, 2]. However, the main drawback of this approach is the requirement of expensive high-speed backhaul links for connecting all the SBSs to the core network. The backhaul capacity requirement can be enormously high during peak traffic hours.

Recently, caching at base stations (BSs) has been proposed as an effective way to alleviate the backhaul capacity requirement in wireless networks [3–6]. Many existing works have focused on cache placement (i.e., what contents to cache), which is essential in cache-enabled wireless networks. These works, however, pay little attention to content delivery (i.e., what contents to deliver upon user requests) which is equally important towards system optimization. Few attempts have been made on the optimization of content delivery or the joint optimization of cache placement and content delivery. Note that the major distinction of wireless communication from wireline communication is the inherent broadcast nature of wireless medium. Therefore, when multiple users request a same content in cache-enabled wireless networks, the broadcast or multicast capability can be exploited for efficient content delivery. In [7], the authors consider multicasting for inelastic services (with strict deadline) in a cache-enabled small cell network and propose a heuristic caching scheme to minimize the service cost. In [8], the authors consider multicasting for inelastic services in a cache-enabled multi-cell network, and propose joint throughput-optimal caching and scheduling algorithms to maximize the service rates of the inelastic services. In our recent work [9], we consider optimal multicast scheduling to jointly minimize the average delay and service costs of elastic services (delay-sensitive services but without strict deadlines) for a cache-enabled single-cell network. However, it remains unknown how to design optimal multicast scheduling for elastic services in cache-enabled heterogeneous cellular networks (HetNets).

In this paper, we consider a cache-enabled HetNet with one MBS, $N$ SBSs, $K$ users and $M$ contents (with possi-

bly different content sizes). The SBS coverage areas are assumed to be disjoint. Assume that the MBS and the SBSs are not allowed to operate concurrently, to avoid excessive interference, while the SBSs are allowed to operate at the same time. Each SBS is equipped with a cache storing a certain number of contents, depending on the sizes of the cached contents and the cache size. The MBS stores all contents in the network. In each slot, each BS either schedules one cached content for multicasting to serve the pending requests from the users in its coverage area, or keeps idle, i.e., does not transmit any content.

We consider the optimal dynamic multicast scheduling to jointly minimize the average network delay and power costs. We formulate this stochastic optimization problem as an infinite horizon average cost Markov decision process (MDP) [10], which is well-known to be challenging. Although dynamic programming provides a systematic approach for MDPs, there generally exist only numerical solutions. These solutions do not typically offer many design insights and are usually impractical due to the curse of dimensionality [10]. Thus, it is highly desirable to study the structural properties of the optimal policy. Specifically, our problem can be viewed as a problem of scheduling a single broadcast server (the MBS) or multiple broadcast servers (the SBSs) to parallel (request) queues with general arrivals. However, existing works have only studied the problems of scheduling a single broadcast server to parallel queues (see [9] and references therein). Therefore, the structural analysis of the optimal multicast scheduling of a single broadcast server or multiple broadcast servers to parallel queues with general arrivals remains unknown and cannot be straightforwardly extended from the existing solutions.

By using *relative value iteration algorithm* (RVIA) [10, Chapter 4.3.1] and the special properties of the request queue dynamics, we characterize some properties of the value function of the MDP. Based on these properties, we show that the optimal multicast scheduling policy, which is adaptive to the request queue state, is of the threshold type. This reveals the tradeoff between the delay cost and the power cost. Finally, we propose a low complexity optimal algorithm by exploiting the structural properties of the optimal policy.

# 2. NETWORK MODEL

Consider a cache-enabled HetNet with one MBS, $N$ SBSs, $K$ users and $M$ contents, as illustrated in Fig. 1. Let $\mathcal{N} \triangleq \{0, 1, 2, \cdots, N\}$ denote the set of all BSs, where BS 0 refers to the MBS and BS $n = 1, 2, \cdots, N$ refers to SBS $n$. Let $\mathcal{N}^+ \triangleq \{1, 2, \cdots, N\}$ denotes the set of $N$ SBSs. The SBS coverage areas are assumed disjoint. Let $\mathcal{K} \triangleq \{1, 2, \cdots, K\}$ denote the set of $K$ users in the network. Let $\mathcal{K}_n \subseteq \mathcal{K}$ denote the set of users within the coverage area of SBS $n \in \mathcal{N}^+$. Each user $k \in \mathcal{K}_n$ ($n \in \mathcal{N}^+$) can be served by the MBS and the SBS $n$. Let $\mathcal{K}_0 \triangleq \mathcal{K} - \bigcup_{n\in\mathcal{N}^+} \mathcal{K}_n$ denote the set of users not covered by any SBS. Each user $k \in \mathcal{K}_0$ can only be served by the MBS. Let $\mathcal{M} \triangleq \{1, 2, \cdots, M\}$ denote the set of $M$ contents (with possibly different content sizes) in the network. Each BS is equipped with a cache storing a certain number of contents, depending on the cache size and the sizes of the cached contents. Let $\mathcal{M}_n \subseteq \mathcal{M}$ denote the set of cached contents in BS $n \in \mathcal{N}$. We assume $\mathcal{M}_0 = \mathcal{M}$, i.e., the MBS stores all contents in the network. This assumption can be easily removed by allowing the MBS to fetch any uncached content from the core network via a backhaul link with a fetching cost [9]. Let $\mathcal{N}_m \triangleq \{n | n \in \mathcal{N}^+ \text{ and } m \in \mathcal{M}_n\}$ denote the set of SBSs caching content $m \in \mathcal{M}$. We assume that the contents stored in the caches are given (as in [11]) and consider multicast scheduling for a given caching design. Notice that caching is in a much larger time-scale (e.g., on a weekly or monthly basis) while multicast scheduling is in a shorter time-scale [3, 5, 12]. Consider time slots of unit length (without loss of generality) indexed by $t = 1, 2, \cdots$.

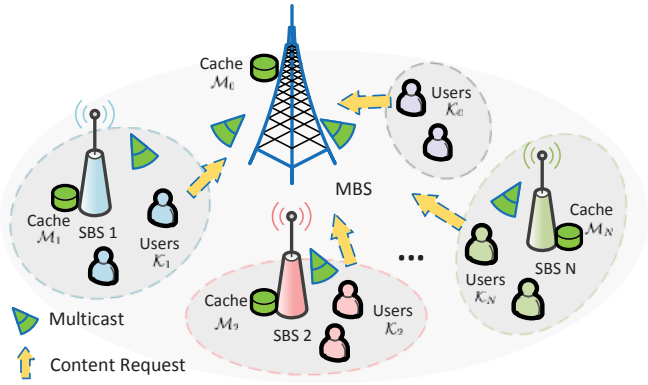



Figure 1: Cache-enabled heterogeneous cellular network.

## 2.1 Request Arrival Traffic

In each slot, each user submits content requests to the MBS. Note that each user can represent a group of users in the same location. Let $A_{m,k}(t) \in \{0, 1, \cdots\}$ denote the number of the new request arrivals for content $m$ from user $k$ at the end of slot $t$, where $m \in \mathcal{M}$ and $k \in \mathcal{K}$. Let $\mathbf{A}(t) \triangleq (A_{m,k}(t))_{m\in\mathcal{M},k\in\mathcal{K}}$ denote the request arrival matrix at the end of slot $t$. We assume that the request arrival processes $\{A_{m,k}(t)\}$ ($m \in \mathcal{M}, k \in \mathcal{K}$) are mutually independent with respect to $m$ and $k$; and $\{A_{m,k}(t)\}$ are i.i.d. with respect to $t$ for all $m \in \mathcal{M}$ and $k \in \mathcal{K}$. The MBS maintains separate request queues for each BS $n \in \mathcal{N}$ and each associated cached content $m \in \mathcal{M}_n$. The request queue model will be further illustrated in Section 2.3.

## 2.2 Service Model

We consider multicast service for content delivery in the network. In each slot, each BS $n \in \mathcal{N}$ either schedules one cached content for multicasting to serve the pending requests from all users in its coverage area, or keeps idle (i.e., does not transmit any content). Let $p(n, m)$ denote the minimum transmission power required by BS $n$ for successfully delivering one cached content $m$ to all users in its coverage area within a scheduling slot, where $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. We set $p(n, 0) = 0$ for all $n \in \mathcal{N}$. If BS $n$ multicasts content $m$ with transmission power $p(n, m)$, all pending requests for

content $m$ from all users in the coverage area of BS $n$ are satisfied. Let $u_n(t) \in \mathcal{U}_n \triangleq \mathcal{M}_n \cup \{0\}$ denote the scheduling action of BS $n \in \mathcal{N}$ at slot $t$, where $u_n(t) \neq 0$ indicates that BS $n$ multicasts the cached content $u_n(t)$ with transmission power $p(n, u_n(t))$ at slot $t$ and $u_n(t) = 0$ indicates that BS $n$ does not transmit any content at slot $t$. Let $\boldsymbol{u}(t) \triangleq (u_n(t))_{n\in\mathcal{N}}$ denote the multicast scheduling action in the network at slot $t$.

The MBS is assumed to operate at much higher transmission power levels than the SBSs, for providing full coverage of the network. To avoid excessive interference, we therefore do not allow the MBS and the SBSs to operate concurrently. On the other hand, since the SBSs are spatially separated and use much lower powers, we allow the SBSs to operate at the same time. Mathematically, we require, for all $t$,

$$u_0(t) \sum_{n\in\mathcal{N}^+} u_n(t) = 0. \tag{1}$$

Let $\boldsymbol{\mathcal{U}} \triangleq \{(u_n)_{n\in\mathcal{N}} | u_n \in \mathcal{U}_n \ \forall n \in \mathcal{N}$ and $u_0 \sum_{n\in\mathcal{N}^+} u_n = 0\}$ denote the feasible multicast scheduling action space. The network power cost $p(\mathbf{u})$ associated with $\boldsymbol{u} \in \boldsymbol{\mathcal{U}}$ is given by

$$p(\mathbf{u}) \triangleq \sum_{n\in\mathcal{N}} p(n, u_n). \tag{2}$$

## 2.3 Request Queue Model

As illustrated above, for each SBS $n \in \mathcal{N}^+$, the requests for cached content $m \in \mathcal{M}_n$ from user $k \in \mathcal{K}_n$ can be served by both the MBS and SBS $n$, while the requests for uncached content $m \in \mathcal{M}_0 \setminus \mathcal{M}_n$ can only be served by the MBS. On the other hand, the MBS can serve the requests for any content $m \in \mathcal{M}_0$ from any user $k \in \mathcal{K}_n$. Therefore, the request queues maintained by the MBS are constructed as follows. For each $n \in \mathcal{N}^+$ and each $m \in \mathcal{M}_n$, we construct a separate request queue, referred to as queue $(n, m)$, storing the requests for content $m$ from all users in $\mathcal{K}_n$. Let $Q_{n,m}(t)$ denote the length of queue $(n, m)$ at the beginning of slot $t$, where $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. For each $m \in \mathcal{M}_0$, we also construct a separate request queue, referred to as queue $(0, m)$, storing the requests for content $m$ from all users in $\bigcup_{n\in\mathcal{N}^+\setminus\mathcal{N}_m} \mathcal{K}_n$ (the set of users covered by the SBSs where content $m$ is not cached) and $\mathcal{K}_0$. Let $Q_{0,m}(t)$ denote the length of queue $(0, m)$ at the beginning of slot $t$, where $m \in \mathcal{M}_0$. Let $\mathbf{Q}_n(t) \triangleq (Q_{n,m}(t))_{m\in\mathcal{M}_n} \in \boldsymbol{\mathcal{Q}}_n \triangleq \prod_{m\in\mathcal{M}_n} \mathcal{Q}_{n,m}$ denote the request queue state vector for BS $n \in \mathcal{N}$ at the beginning of slot $t$, where $\mathcal{Q}_{n,m} \triangleq \{0, 1, \cdots, N_{n,m}\}$. Here, we assume $N_{n,m}$ to be finite (can be arbitrarily large) for technical tractability. Let $\mathbf{Q}(t) \triangleq (\mathbf{Q}_n(t))_{n\in\mathcal{N}} \in \boldsymbol{\mathcal{Q}}$ denote the request queue state of the network at the beginning of slot $t$, where $\boldsymbol{\mathcal{Q}} \triangleq \prod_{n\in\mathcal{N}} \boldsymbol{\mathcal{Q}}_n$ denotes the request queue state space. Note that the request queues can be implemented using counters and no data is contained in these queues.

For each $n \in \mathcal{N}^+$ and each $m \in \mathcal{M}_n$, all the pending requests in queue $(n, m)$ are satisfied, if content $m$ is scheduled for multicasting by the MBS (i.e., $u_0(t) = m$) or by SBS $n$ (i.e., $u_n(t) = m$) at slot $t$. Thus, for each $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$, the request queue dynamics is as follows:

$$Q_{n,m}(t+1) = \min\{\mathbf{1}(u_0(t) \neq m \ \&\ u_n(t) \neq m) Q_{n,m}(t) + B_{n,m}(t), N_{n,m}\}, \tag{3}$$

where $B_{n,m}(t) \triangleq \sum_{k\in\mathcal{K}_n} A_{m,k}(t)$ denotes the total number of the new request arrivals for content $m$ from all users in $\mathcal{K}_n$ at the end of slot $t$, and $\mathbf{1}(\cdot)$ denotes the indicator function. For each $m \in \mathcal{M}_n$, all the pending requests in queue $(0, m)$ are satisfied, if content $m$ is scheduled for multicasting by the MBS at slot $t$ (i.e., $u_0(t) = m$). Thus, for each $m \in \mathcal{M}_0$, the request queue dynamics is as follows:

$$Q_{0,m}(t+1) = \min\{\mathbf{1}(u_0(t) \neq m) Q_{0,m}(t) + B_{0,m}(t), N_{0,m}\}, \tag{4}$$

where

$$B_{0,m}(t) \triangleq \sum_{n\in\mathcal{N}^+\setminus\mathcal{N}_m} \sum_{k\in\mathcal{K}_n} A_{m,k}(t) + \sum_{k\in\mathcal{K}_0} A_{m,k}(t)$$

denotes the total number of the new request arrivals for content $m$ from all users in $\bigcup_{n\in\mathcal{N}^+\setminus\mathcal{N}_m} \mathcal{K}_n$ and $\mathcal{K}_0$ at the end of slot $t$. Note that the request arrivals for each $m \in \mathcal{M}$ from each $k \in \mathcal{K}$ at the end of slot $t$ are stored in only one queue.

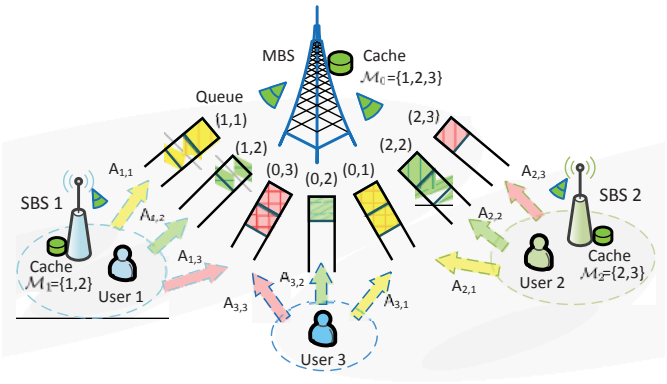



Figure 2: An example with 1 MBS, 2 SBSs, 3 users and 3 contents.

## 2.4 Motivating Example

As illustrated in Fig. 2, consider a network with 1 MBS, 2 SBSs ($\mathcal{N}^+ = \{1, 2\}$), 3 users ($\mathcal{K} = \{1, 2, 3\}$) and 3 contents ($\mathcal{M} = \{1, 2, 3\}$). We set $\mathcal{K}_1 = \{1\}$, $\mathcal{K}_2 = \{2\}$, $\mathcal{K}_0 = \{3\}$, $\mathcal{M}_1 = \{1, 2\}$, $\mathcal{M}_2 = \{2, 3\}$ and $\mathcal{M}_0 = \{1, 2, 3\}$. According to Section 2.3, the MBS maintains seven request queues, i.e., queues $(0, 1)$, $(0, 2)$, $(0, 3)$, $(1, 1)$, $(1, 2)$, $(2, 2)$ and $(2, 3)$. Our goal is to design the optimal multicast scheduling so as to jointly minimize the network delay cost and power cost. This involves two challenging and coupled tasks.

First, at each time slot, shall we operate the MBS or the SBSs? If we schedule the MBS to multicast, then the pending requests for one content in the whole network can be satisfied with a higher power cost, e.g., clear queues $(0, 2)$, $(1, 2)$ and $(2, 2)$ with power $p(0, 2)$. Otherwise, if we schedule the SBSs to multicast, the pending requests for (possibly different) contents in different SBS coverage areas can be satisfied

with a lower power cost, e.g., clear queues $(1,2)$ and $(2,3)$ with power $p(1,2)+p(2,3)$.

Second, at each time slot, for each BS $n \in \mathcal{N}$, which content should we schedule for multicasting, or should we keep BS $n$ idle? Take SBS 1 for an example. Suppose at certain time slot t we have $Q_{1,1}(t) > Q_{1,2}(t)$ and $p(1,1) > p(1,2)$. Should we schedule Content 1 for satisfying more requests with a higher power cost or schedule Content 2 for satisfying fewer requests with a lower power cost or do nothing at the current slot?

We can see that, it is the flexibility of multicast transmission, the elasticity of services, and the heterogeneity of the network that make it difficult to design the optimal multicast scheduling. In the sequel, we formalize the multicast scheduling problem and try to tackle these challenges.

# 3. PROBLEM FORMULATION AND OPTIMALITY EQUATION

## 3.1 Problem Formulation

Given an observed request queue state $\mathbf{Q}$, the multicast scheduling action $\mathbf{u}$ is determined according to a stationary policy defined below.

DEFINITION 1 (STATIONARY POLICY). *A feasible stationary multicast scheduling policy $\mu$ is a mapping from the request queue state $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$ to the feasible multicast scheduling action $\mathbf{u} \in \boldsymbol{\mathcal{U}}$, where $\mu(\mathbf{Q}) = \mathbf{u}$.*

By the queue dynamics in (3) and (4), the induced random process $\{\mathbf{Q}(t)\}$ under policy $\mu$ is a controlled Markov chain. We restrict our attention to stationary unichain policies[1]. For a given stationary unichain policy $\mu$, the average network delay cost is defined as

$$\bar{d}(\mu) \triangleq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[d\left(\mathbf{Q}(t)\right)\right], \tag{5}$$

where the expectation is taken w.r.t. the measure induced by the policy $\mu$ and $d\left(\mathbf{Q}\right)$ is a monotonically non-decreasing function of $\mathbf{Q}$. For example, if $d\left(\mathbf{Q}\right) = \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}_n} Q_{n,m}$, then $\bar{d}(\mu)$ reflects the average waiting time in the network under policy $\mu$ according to Little's law. For a given stationary unichain policy $\mu$, the average network power cost is given by

$$\bar{p}(\mu) \triangleq \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[p(\mathbf{u}(t))\right]. \tag{6}$$

Therefore, under a given stationary unichain policy $\mu$, the average network cost is defined as

$$\begin{aligned} \bar{g}(\mu) &\triangleq \bar{d}(\mu) + w\bar{p}(\mu) \\ &= \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[g(\mathbf{Q}(t), \mathbf{u}(t))\right], \end{aligned} \tag{7}$$

where $w$ is the weight for the power cost and $g(\mathbf{Q},\mathbf{u}) \triangleq d\left(\mathbf{Q}\right) + wp(\mathbf{u})$ is the per-stage network cost.

We wish to find an optimal multicast scheduling policy to minimize the average network cost $\bar{g}(\mu)$ in (7).

PROBLEM 1 (NETWORK COST MINIMIZATION).

$$\bar{g}^* \triangleq \min_{\mu} \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[g(\mathbf{Q}(t), \mathbf{u}(t))\right], \tag{8}$$

*where $\mu$ is a stationary unichain policy in Definition 1 and $\bar{g}^*$ denotes the minimum average network cost achieved by the optimal policy $\mu^*$.*

Note that, Problem 1 is an infinite horizon average cost MDP, which is well known to be very challenging.

## 3.2 Optimality Equation

The optimal multicast scheduling policy $\mu^*$ can be obtained by solving the following Bellman equation.[2]

LEMMA 1 (BELLMAN EQUATION). *There exist a scalar $\theta$ and a real-valued function $V(\cdot)$ satisfying*

$$\theta + V(\mathbf{Q}) = \min_{\mathbf{u}\in\boldsymbol{\mathcal{U}}} \left\{g(\mathbf{Q},\mathbf{u}) + \mathbb{E}\left[V(\mathbf{Q}')\right]\right\}, \ \forall \mathbf{Q} \in \boldsymbol{\mathcal{Q}}, \tag{9}$$

*where the expectation is taken over the distribution of request arrival $\mathbf{A}$ and $\mathbf{Q}' \triangleq (Q'_{n,m})_{n\in\mathcal{N},m\in\mathcal{M}_n}$ with $Q'_{0,m} \triangleq \min\{\mathbf{1}(u_0 \neq m)Q_{0,m} + B_{0,m}, N_{0,m}\}$ for all $m \in \mathcal{M}_0$ and $Q'_{n,m} \triangleq \min\{\mathbf{1}(u_0 \neq m \ \& \ u_n \neq m)Q_{n,m} + B_{n,m}, N_{n,m}\}$ for all $n \in \mathcal{N}^+$ and $m \in \mathcal{M}_n$. $\theta = \bar{g}^*$ is the optimal value to Problem 1 for all initial state $\mathbf{Q}(1) \in \boldsymbol{\mathcal{Q}}$ and $V(\cdot)$ is called the value function. Furthermore, the optimal policy achieving the optimal value $\bar{g}^*$ is given by*

$$\mu^*(\mathbf{Q}) = \arg\min_{\mathbf{u}\in\boldsymbol{\mathcal{U}}} \left\{g(\mathbf{Q},\mathbf{u}) + \mathbb{E}\left[V(\mathbf{Q}')\right]\right\}, \ \forall \mathbf{Q} \in \boldsymbol{\mathcal{Q}}. \tag{10}$$

From Lemma 1, we can observe that $\mu^*$ given by (10) depends on $\mathbf{Q}$ through the value function $V(\cdot)$. Obtaining $V(\cdot)$ involves solving the Bellman equation in (9) for all $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$, which does not admit a closed-form solution in general [10]. Brute force numerical solutions such as value iteration and policy iteration are usually computationally impractical to implement in practical systems, and do not typically yield many design insights [10]. Thus, it is highly desirable to study the structural properties of the optimal policy $\mu^*$.

# 4. OPTIMALITY PROPERTIES

Problem 1 can be viewed as a problem of scheduling a single broadcast server (the MBS) or multiple broadcast servers (the SBSs) to parallel (request) queues with general arrivals. The structural analysis is more challenging than the existing structural analysis for the scheduling of a single broadcast server. First, by RVIA and the special structure of the request queue dynamics, we can prove the following property of the value function.

LEMMA 2 (MONOTONICITY OF $V(\mathbf{Q})$). *For any $\mathbf{Q}^1, \mathbf{Q}^2 \in \boldsymbol{\mathcal{Q}}$ such that $\mathbf{Q}^2 \succeq \mathbf{Q}^1$, we have $V(\mathbf{Q}^2) \geq V(\mathbf{Q}^1)$.*[3]

[1] A unichain policy is a policy, under which the induced Markov chain has a single recurrent class (and possibly some transient states) [10].

[2] All the proofs can be found in the full version in [13].

[3] The notation $\succeq$ indicates component-wise $\geq$.

Next, we introduce the state-action cost function:

$$J(\mathbf{Q},\mathbf{u}) \triangleq g(\mathbf{Q},\mathbf{u}) + \mathbb{E}\left[V(\mathbf{Q}')\right]. \quad (11)$$

Note that $J(\mathbf{Q},\mathbf{u})$ is related to the R.H.S. of the Bellman equation in (9). Then, based on $J(\mathbf{Q},\mathbf{u})$, we introduce:

$$\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \triangleq J(\mathbf{Q},\mathbf{u}) - J(\mathbf{Q},\mathbf{v}). \quad (12)$$

Note that $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) = -\Delta_{\mathbf{v},\mathbf{u}}(\mathbf{Q})$. Action $\mathbf{u}$ is said to dominate $\mathbf{v}$ at state $\mathbf{Q}$ if $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q}) \leq 0$. In particular, by Lemma 1, if $\mathbf{u}$ dominates any $\mathbf{v} \in \boldsymbol{\mathcal{U}}$ at state $\mathbf{Q}$, then $\mu^*(\mathbf{Q}) = \mathbf{u}$. Based on Lemma 2, we have the following property of the function defined in (12).

LEMMA 3 (MONOTONICITY OF $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$). *For any $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$ and $\mathbf{u},\mathbf{v} \in \boldsymbol{\mathcal{U}}$, $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ has the following properties.*

1. *If $u_0 = m \in \mathcal{M}_0$, then $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ is monotonically non-increasing with $Q_{0,m}$ and $Q_{n,m}$ for all $n \in \mathcal{N}_m$.*
2. *If $u_n = m \in \mathcal{M}_n$ for some $n \in \mathcal{N}^+$, then $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ is monotonically non-increasing with $Q_{n,m}$.*

Lemma 3 indicates that, if $\mathbf{u}$ dominates $\mathbf{v}$ at some state $\mathbf{Q}$, then by increasing $Q_{0,u_0}$ and $Q_{n,u_0}$ for any $n \in \mathcal{N}_{u_0}$ and $u_0 \neq 0$ or by increasing $Q_{n,u_n}$ for any $n \in \mathcal{N}^+$ and $u_n \neq 0$, $\mathbf{u}$ still dominates $\mathbf{v}$. The properties of $\Delta_{\mathbf{u},\mathbf{v}}(\mathbf{Q})$ in Lemma 3 is similar to the diminishing-return property of submodular functions used in the existing structural analysis [14]. Lemma 3 stems from the special properties of multicasting and is essential to characterize the optimality properties. By Lemma 3, we can characterize the structural properties of the optimal policy $\mu^*$. We start with several definitions. Define:

$$\Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \triangleq \{Q_{n,m} | Q_{n,m} \in \mathcal{Q}_{n,m} \text{ and } \Delta_{\mathbf{u},\mathbf{v}}(Q_{n,m},\mathbf{Q}_{-n,-m}) \leq 0 \ \forall \mathbf{v} \in \boldsymbol{\mathcal{U}} \text{ and } \mathbf{v} \neq \mathbf{u}\},$$

where $\mathbf{Q}_{-n,-m} \triangleq (Q_{i,j})_{i\in\mathcal{N},j\in\mathcal{M}_i,(i,j)\neq(n,m)}$. Based on $\Phi_{\mathbf{u}}(\cdot)$, we define:

$$\phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \triangleq \begin{cases} \max \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}), & \text{if } \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \neq \emptyset \\ -\infty, & \text{otherwise} \end{cases}$$

$$\psi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \triangleq \begin{cases} \min \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}), & \text{if } \Phi_{\mathbf{u}}(\mathbf{Q}_{-n,-m}) \neq \emptyset \\ +\infty, & \text{otherwise} \end{cases}$$

Let $\mathbf{0}_n$ denote the $1 \times n$ vector with all entries 0. Then, we have the following theorem.

THEOREM 1 (STRUCTURAL PROPERTIES OF $\mu^*$). *For any $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$, the optimal policy $\mu^*(\mathbf{Q}) = \mathbf{u}^*$ has the following structural properties.*

1. *$\mathbf{u}^* = \mathbf{0}_{N+1}$, if $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}_0 \triangleq \{\mathbf{Q} | Q_{n,m} \leq \phi_{\mathbf{u}^*}(\mathbf{Q}_{-n,-m}), \forall n \in \mathcal{N} \text{ and } m \in \mathcal{M}_n\}$.*
2. *$\mathbf{u}^* \neq \mathbf{0}_{N+1}$ and $u_n^* = m$ if*

$$Q_{n,m} \geq \psi_{\mathbf{u}^*}(\mathbf{Q}_{-n,-m}), \quad (13)$$

*where $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. Moreover, $\psi_{\mathbf{u}^*}(\mathbf{Q}_{-0,-m})$ is monotonically non-increasing with $Q_{n,m}$ for all $n \in \mathcal{N}_m$.*

We verify the analytical results of Theorem 1 in Fig. 3, where the optimal policy is computed numerically using *policy iteration algorithm* (PIA) [15, Chapter 8.6]. We can observe from Fig. 3(a) that, if the queue state falls in the region of blue squares (i.e., $\boldsymbol{\mathcal{Q}}_0$), the optimal control is $(0,0)$, i.e., both the MBS and the SBS keep idle. Hence, we refer to $\boldsymbol{\mathcal{Q}}_0$ as the idle region of the optimal policy. From Fig. 3(b)-3(d), we can observe that given $\mathbf{Q}_{-n,-m}$, the scheduling for content $m \in \mathcal{M}_n$ by BS $n \in \mathcal{N}$ is of the threshold type (Property 2 of Theorem 1). This indicates that, it is not efficient to schedule content $m$ by BS $n$ when $Q_{n,m}$ is small, (i.e., the delay cost is small) as a higher power cost per request is consumed. This shows the tradeoff between the delay cost and the power cost. Fig. 3(c) illustrates the monotonically non-increasing property of $\psi_{\mathbf{u}^*}(\mathbf{Q}_{-0,-1})$ in terms of $Q_{1,1}$. This reveals that the MBS is more willing to multicast content 1 when $Q_{1,1}$ is large. The reason is that the MBS can satisfy more requests than any SBS. These optimality properties provide design insights for multicast scheduling in practical cache-enabled HetNets.

# 5. LOW COMPLEXITY OPTIMAL ALGORITHM

The results in Theorem 1 can be exploited to substantially reduce the computational complexity of solving the Bellman equation for (9) in obtaining $\mu^*$. In particular, by Property 2 in Theorem 1, we know that, for all $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$,

$$\mu^*(\mathbf{Q}) = \mathbf{u} \ \Rightarrow \ \mu^*(\mathbf{Q}') = \mathbf{u}, \quad (14)$$

where $\mathbf{Q}' = (\mathbf{Q}'_n)_{n\in\mathcal{N}}$ and

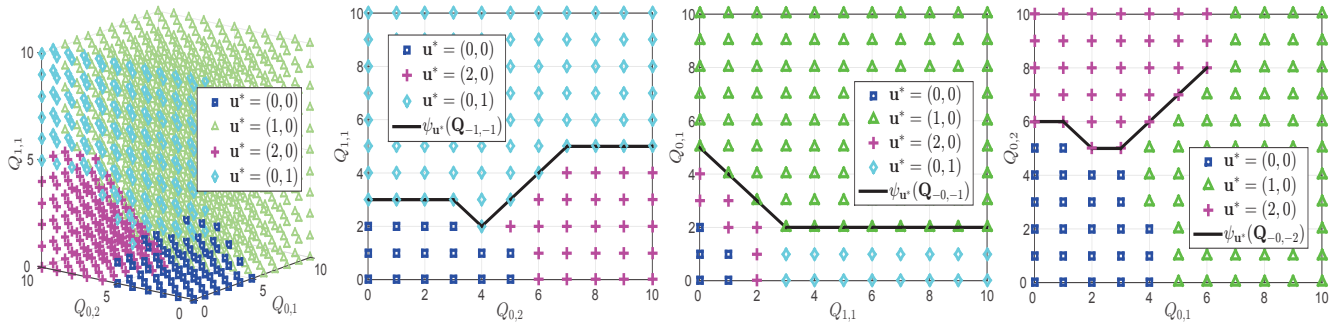
$$\mathbf{Q}'_n = \begin{cases} \mathbf{Q}_n + \mathbf{e}_{n,u_n} \text{ or } \mathbf{Q}_n & \text{if } u_n \neq 0 \\ \mathbf{Q}_n, & \text{otherwise} \end{cases}.$$

Here, $\mathbf{e}_{n,m}$ denotes the $1 \times |\mathcal{M}_n|$ vector with all entries 0 except for a 1 in its $m$-th entry, where $n \in \mathcal{N}$ and $m \in \mathcal{M}_n$. Therefore, by incorporating the property in (14) into the standard PIA, we develop a low complexity algorithm in Algorithm 1, which is referred to as the structured policy iteration algorithm (SPIA). According to Theorem 8.6.6 and Chapter 8.11.2 in [15], we know that SPIA converges to the optimal policy $\mu^*$ in (10) within a finite number of iterations, and hence is an optimal algorithm.

Note that, in Step 3 (structured policy improvement) of Algorithm 1, we do not need to perform the minimization over $\boldsymbol{\mathcal{U}}$ when the condition is satisfied (which is the case for a large amount of queue states in $\boldsymbol{\mathcal{Q}}$). This can be seen in Fig. 3 as an example. While, in the standard policy improvement step of PIA, the new policy $\mu_{l+1}^*$is obtained by:

$$\mu_{l+1}^*(\mathbf{Q}) = \arg\min_{\mathbf{u}\in\boldsymbol{\mathcal{U}}} \left\{ g(\mathbf{Q},\mathbf{u}) + \mathbb{E}\left[V_l(\mathbf{Q}')\right] \right\}, \ \forall \mathbf{Q} \in \boldsymbol{\mathcal{Q}}. \quad (15)$$

By (15), obtaining $\mu_{l+1}^*$ requires a brute-force minimization over $\boldsymbol{\mathcal{U}}$ for each $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$, which can be very computationally expensive when the numbers of the contents $M$ and the SBSs $N$ are large. By comparing the structured policy improvement step of SPIA with the standard policy improve-




(a) Whole space. (b) Fixed $Q_{0,1}$. (c) Fixed $Q_{0,2}$. (d) Fixed $Q_{1,1}$.

Figure 3: Structure of the optimal scheduling. $N = 1$, $\mathcal{K} = \{1,2,3,4\}$, $\mathcal{M} = \{1,2\}$, $\mathcal{K}_1 = \{1,2\}$ and $\mathcal{M}_1 = \{1\}$.

ment step of PIA, we can see that SPIA can achieve considerable computational saving.

**Algorithm 1** Structured Policy Iteration Algorithm

1: Set $\mu_0^*(\mathbf{Q}) = \mathbf{0}_{N+1}$ for all $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$, select reference state $\mathbf{Q}^\dagger$, and set $l = 0$.

2: (Policy Evaluation) Given policy $\mu_l^*$, compute the average cost $\theta_l$ and value function $V_l(\mathbf{Q})$ from the linear system of equations[4]

$$\begin{cases} \theta_l + V_l(\mathbf{Q}) = g(\mathbf{Q}, \mu_l^*(\mathbf{Q})) + \mathbb{E}\left[V_l(\mathbf{Q}')\right], \ \forall \mathbf{Q} \in \boldsymbol{\mathcal{Q}} \\ V_l(\mathbf{Q}^\dagger) = 0 \end{cases} \tag{16}$$

where $\mathbf{Q}'$ is defined in Lemma 1.

3: (Structured Policy Improvement) Obtain a new policy $\mu_{l+1}^*$, where for each $\mathbf{Q} \in \boldsymbol{\mathcal{Q}}$, $\mu_{l+1}^*(\mathbf{Q})$ is such that:
**if** $\exists n \in \mathcal{N}$ and $m \in \mathcal{M}_n$ such that $\mu_l^*(\mathbf{Q}') = \mathbf{u}$ where $u_n = m$, $\mathbf{Q}'_n = \mathbf{Q}_n - \mathbf{e}_{n,m}$ and $\mathbf{Q}'_i = \mathbf{Q}_i$ for all $i \in \mathcal{N}, i \neq n$ **then**

$$\mu_{l+1}^*(\mathbf{Q}) = \mathbf{u}.$$

**else**

$$\mu_{l+1}^*(\mathbf{Q}) = \arg\min_{\mathbf{u} \in \boldsymbol{\mathcal{U}}} \left\{ g(\mathbf{Q}, \mathbf{u}) + \mathbb{E}\left[V_l(\mathbf{Q}')\right] \right\}.$$

**endif**

4: Go to Step 2 until $\mu_{l+1}^* = \mu_l^*$.

# 6. CONCLUSION

In this paper, we study the optimal content delivery strategy in a cache-enabled HetNet by taking into account the inherent multicast capability of wireless medium. We establish a content-centric request queue model and then formulate a stochastic multicast scheduling problem to jointly minimize the average network delay and power costs. This stochastic optimization problem is an infinite horizon average cost MDP. We show that the optimal multicast scheduling policy, which is adaptive to the request queue state, is of the threshold type. The optimality properties provide design insights for practical cache-enabled HetNets. Finally, we propose a low complexity optimal algorithm by exploiting the structural properties of the optimal policy.

[4]The solution to (16) can be obtained directly using Gaussian elimination or iteratively using the relative value iteration method [10].

# 7. ACKNOWLEDGMENTS

This work was supported by the NSF of China under grants 61322102, 61571299 and 61401272.

# 8. REFERENCES

[1] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T.A. Thomas, J.G. Andrews, P. Xia, H.S. Jo, H.S. Dhillon, and T.D. Novlan. Heterogeneous cellular networks: From theory to practice. *IEEE Commun. Mag.*, 50(6):54–64, 2012.

[2] S.V. Hanly, C. Liu, and P. Whiting. Capacity and stable scheduling in heterogeneous wireless networks. *IEEE J. Sel. Areas Commun.*, 33(6):1266–1279, June 2015.

[3] K. Shanmugam, N. Golrezaei, A.G. Dimakis, A.F. Molisch, and G. Caire. Femtocaching: Wireless content delivery through distributed caching helpers. *IEEE Trans. Inf. Theory*, 59(12), Dec 2013.

[4] M.A. Maddah-Ali and U. Niesen. Fundamental limits of caching. *IEEE Trans. Inf. Theory*, 60(5):2856–2867, May 2014.

[5] A. Liu and V. Lau. Exploiting base station caching in MIMO cellular networks: Opportunistic cooperation for video streaming. *IEEE Trans. Signal Process.*, 63(1):57–69, Jan 2015.

[6] E. Bastug, M. Bennis, and M. Debbah. Living on the edge: The role of proactive caching in 5G wireless networks. *IEEE Commun. Mag.*, 52(8):82–89, Aug 2014.

[7] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas. Multicast-aware caching for small cell networks. In *Proc. IEEE WCNC*, April 2014.

[8] N. Abedini and S. Shakkottai. Content caching and scheduling in wireless networks with elastic and inelastic traffic. *IEEE/ACM Trans. Netw.*, 22(3):864–874, June 2014.

[9] B. Zhou, Y. Cui, and M. Tao. Optimal dynamic multicast scheduling for cache-enabled content-centric wireless networks. In *Proc. IEEE ISIT*, June 2015.

[10] D. P. Bertsekas. *Dynamic programming and optimal control, 3rd edition, volume II*. Belmont, MA: Athena Scientific, 2011.

[11] E. Bastug, M. Bennis, M. Kountouris, and M. Debbah. Cache-enabled small cell networks: modeling and tradeoffs. *EURASIP Journal of Wireless Communications and Networking*, 2015:1, 2015.

[12] A. F. Molisch, G. Caire, D. Ott, J. R. Foerster, D. Bethanabhotla, and M. Ji. Caching eliminates the wireless bottleneck in video aware wireless networks. *Advances in Electrical Engineering*, 2014.

[13] B. Zhou, Y. Cui, and M. Tao. Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks. [Online.] Available: http://arxiv.org/abs/1509.06611.

[14] G. Koole. *Monotonicity in Markov reward and decision chains: Theory and applications*. Now Publishers Inc, 2007.

[15] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*, volume 414. John Wiley & Sons, 2009.