UNIVERSITY CARLOS III OF MADRID

Department of Telematics Engineering

Master of Science Thesis

Adaptive Modulation for Finite Horizon Multicasting of Erasure-coded Data

Author: **Gek Hong, Sim**
Master of Engineering Science

Supervisor: **Albert Banchs, Ph.D.**
Co-supervisor: **Joerg Widmer, Ph.D.**

Leganés, October 2012

Abstract

We design an adaptive modulation scheme to support opportunistic multicast scheduling in wireless networks. Whereas prior work optimizes capacity, we investigate the finite horizon problem where (once or repeatedly) a fixed number of packets has to be transmitted to a set of wireless receivers in the shortest amount of time – a common problem, e.g., for software updates or video multicast.

In the finite horizon problem, the optimum coding and modulation schemes critically depend on the recent reception history of the receivers and requires a fine balance between maximizing overall throughput and equalizing individual receiver throughput. We formulate a dynamic programming algorithm that optimally solves this scheduling problem. We then develop two low complexity heuristics that perform very close to the optimal solution and are suitable for practical online scheduling in base stations. We further analyze the performance of our algorithms by means of simulation in a wide range of wireless scenarios. They substantially outperform existing solutions based on throughput maximization or favoring the user with the worst channel, and we obtain a 25% performance improvement over the former and a 40% improvement over the latter in a scenario with Rayleigh fading.

Table of Contents

Chapter 1

Introduction

In recent years, multicasting data to mobile users (e.g., for the purpose of video streaming, video conferencing, IPTV, distribution of news and alerts, or application and operating system updates) has gained in popularity and importance. As an example, the most recent mobile network architecture LTE includes the evolved Multimedia Broadcast Multicast Service (eMBMS) specifically for the purpose of distributing data and mobile TV content in a cellular network. Since the amount of such traffic in cellular networks is increasing rapidly and wireless resources are usually scarce and costly, improving the efficiency of wireless multicast is of high practical relevance.

The most common method for wireless multicasting is broadcasting. The base station (BS) transmits at some fixed low rate or the rate supported by the worst receiver to ensure that all receivers are able to receive the multicast transmission. It exploits the wireless broadcast gain whereby a single transmission simultaneously serves all receivers. Opportunistic multicast scheduling (OMS) improves over plain broadcast by exploiting multiuser diversity [1]. Having the BS transmit at a rate higher than the broadcast rate to the subset of receivers that can receive at this rate improves overall throughput capacity and minimizes broadcast delay [2]. The intuition is that in an environment with variable channels, receivers that are not served in the current slot since their channel conditions are bad will be served in later slots when their conditions improve, and thus over time all receivers will eventually receive all the data. Hence, there is a tradeoff between multicast gain and multiuser diversity gain. As an extreme case, the BS may even unicast data to the receiver that supports the highest rate, serving receivers one-by-one and entirely foregoing the broadcast gain for the largest possible multiuser diversity gain. Selecting the transmission rate and thus the subset of receivers to multicast to is a complex problem that has been the focus of a range of OMS algorithms [3]. To simplify the scheduling problem and improve performance when multicasting data, such algortithms often use erasure codes that ensure that with high probability, each packet received by a receiver is useful (unless the receiver has decoded all of the packets that exist in the system) [4], i.e., the identity of the received packets is unimportant. Fixed rate LDPC [5] or rateless LT or Raptor codes [6] are examples for such erasure codes that work well in practice and have good performance.

Most of the existing literature considers OMS algorithms for the infinite-horizon multicast problem, where the sender has an infinite number of packets to send and the goal of the optimization is to maximize throughput capacity. This setting is also a good approximation for the case of multicasting very large files. In practice, however, multicasting data with a size on the order of hundreds to several thousands of packets much more common, particularly for mobile networks. Mobile apps and operating system updates often have a size of several to several tens of MB, corresponding to thousands of packets of size 1kB. When streaming video, it is common to apply erasure coding to blocks consisting of one or several groups of pictures (GOP) [7], where a GOP usually consists of a few hundreds of packets, depending on the video rate. Since erasure coded blocks can only be decoded after they have been fully received, coding over larger blocks of video data would unecessarily increase the playout delay of the video.

In this thesis we therefore consider the finite horizon multicast problem, where a fixed amount of erasure coded data has to be delivered to a set of wireless receivers. Whenever the BS transmits a packet, the optimization algorithm has to select a suitable modulation and coding scheme (MCS). The MCS determines the amount of data transmitted per time slot and thus the data rate. At the same time, the MCS influences the packet loss probability, where more robust MCSs that transport less data are more likely to be decodable at a receiver. (We assume that the transmit power is fixed.) Our main objective is to minimize the completion time, i.e., the time needed for all receivers to successfully receive the data.

The finite horizon multicast problem is inherently more complex than the infinite horizon counterpart. When multicasting an infinite amount of data among a homogeneous group of receivers (i.e., with the same average channel conditions and receive rates), the optimum tradeoff between multiuser diversity and multicast gain only depends on the number of receivers and their current channel conditions. In expectation, differences in the amount of data received by the different receivers will even out over time and therefore do not have to be taken into account. In contrast, the optimum decision in the finite horizon case also depends on the amount of data received thus far by each receiver (or, more accurately, on the amount of data each receiver still needs to obtain in order to decode the full block of data and thus complete). Intuitively, in case a receiver is lagging behind but many other receivers are also still for from completing, the lagging receiver may catch up by itself and jointly maximizing throughput for all these receivers may be the optimum decision. If, however, all other receivers are close to completion, optimizing the MCS (and hence the transmit rate) for the lagging receiver only may be the optimum choice to minimize overall completion time, given that all other receivers are likely to complete before the lagging receiver in any case.

The main contributions of our thesis are as follows:

- We formalize the finite horizon OMS problem and propose a dynamic programming (*Dyn-Prog*) based solution, that optimally adapts the MCS to minimize the *completion time*, the time at which all receivers successfully receive the required amount of data.
- The high complexity of *Dyn-Prog* renders this approach unsuitable for many practical scenarios. We therefore propose a simple state-based heuristic that selects the MCS that maximizes the instantaneous throughput for the receiver with the minimum number of packets which thus had the lowest throughput so far (called *Max-Min*).
- We further design a slightly more complex adaptive algorithm that selects the MCS that results in an expected system state (given by numbers of packets received by the different receivers) that has the lowest expected completion time. Due to the complexity of accurately calculating completion time, we use a weighted Euclidian distance metric (called weighted

completion time or *Weighted-CT*). It measures the distance between the different possible future states and the final state where all receivers completed, with weights based on average throughput estimates of the receivers.

• We compare the performance of our two low-complexity heuristics to the optimal *Dyn-Prog* solution as well as to existing approaches that greedily maximize the throughput for all receivers and a broadcasting scheme that always transmits to all receivers. We analyze scenarios with homogeneous and heterogeneous receiver sets under a basic multi-state channel model as well as Rayleigh fading. Under Rayleigh fading, the *Max-Min* algorithm provides a performance gain of 15% over the throughput maximization scheme and a gain of 20% over the broadcasting scheme. At a very slight increase in complexity, the *Weighted-CT* heuristics performs very close to the optimal *Dyn-Prog* strategy in almost all scenarios, achieving performance gains of 25% and 40% over the throughput maximization and broadcast scheme, respectively. For both our heuristics, the gains that can be obtained under the basic multi-state channel model are even larger.

Our thesis is organized as follows. In Chapter 2, we review existing work on opportunistic multicast scheduling. Chapter 3 provides an overview of our system model including the channel model and the MCS dependent packet loss model. In Chapter 4, we detail the optimal scheduling algorithm based on a dynamic programming formulation. To address the problem of state space explosion and high complexity of the dynamic programming solution, we propose two low-complexity heuristics in Section Chapter 5. We also discuss some basic scenarios to provide an intuition into how the heuristics trade off receiver throughput depending on the current state of the system. Simulation results that compare the relative performance of the different algorithms are presented in Chapter 6. The possible extension on our current work is presented in Chapter 7 and finally we conclude our thesis in Chapter 8.

Chapter 2

Related Work

The idea of OMS was pioneered by Gopala and Gamal [1] who studied the tradeoff between multiuser diversity and multicast gain. They studied the performance of three different scheduling mechanisms that adapt the transmit rate to the user with the best channel, the worst channel, and the median channel, respecively. In their follow-up paper [8], they analyzed the performance achieved by serving a fixed fraction of users. This restriction is relaxed in [4], where the authors show that dynamic selection ratios that select more than 50% of the users can achieve higher throughput. Furthermore, a throughput maximizing scheme for erasure-coded multicast (F-OMS) is presented, where the user selection ratio depends only on the set of multicast users in the system.

The authors of [9] propose algorithms with a static selection ratio (fixed for all transmissions) and a dynamic selection ratio (adapted to the instantaneous channels at each transmission) that maximize overall throughput. In [10], the authors extend their work of [9] from homogeneous to heterogeneous scenarios, composed of different groups of homogeneous users. A similar optimization algorithm for multicast throughput maximization in homogeneous to heterogeneous networks is proposed in [11]. While all of these works target the infinite horizon case, in [2] the authors do consider scenarios with a finite number of multicast packets. Using extreme value theory, they derive the optimal selection ratio for each transmission that minimizes completion time, i.e., the time period during which each user is selected often enough to receive the whole block of data. In contrast to our work, their optimization algorithm does not consider the state of the receivers in terms of number of packets received.

All of the above papers use a simple outage based channel model, where packet errors are deterministic. Receivers with channel conditions better than a certain threshold are guaranteed to receive the packet and all others are guaranteed to lose the packet. In real wireless systems, packet errors are much more random and depend on noise and interference. In our model, we explicitly take the relationship between the channel conditions, the chosen MCS, and the probability of error into account. Furthermore, all of the above papers – except [2] – focus on the infinite-horizon scenario and thus the performance of the proposed algorithms is sub-optimal in the finite-horizon problem we consider here.

The problem of minimizing the overall delay for all users to receive a certain number of packets is studied in [12] through a dynamic programming approach. This work does not consider erasure coding over a larger block of data but repeatedly multicasts a single packet until each receiver has obtained it. The BS then multicasts the next packet in the same manner, and so on. The goal of the optimization algorithm is therefore to minimize the number of transmission required to multicast a single packet to all receivers, and the state of the system is the number of receivers that did not yet receive the packet. The algorithm adapts its decision to the changes in the set of users that still need to receive the packet and maximizes the throughput for those users. The approach is mainly suitable for a single homogeneous group of users, since its complexity increases exponentially with the number of user groups in heterogeneous scenarios. The method of multicasting a single packet repeatedly is less efficient than multicasting blocks of erasure coded packets as is done above.

The most basic scheme against which we compare our proposed algorithms is the *broadcast* algorithm (called LCG user rate in [3]), where the transmission rate is limited by the receiver that currently has the worst channel. This scheme ensures successful transmission to all receiver at all times but may sacrifice a lot of throughput when channels are highly variable. We further compare against a scheme called *greedy* that optimizes the selection ratio at each transmission opportunistically based on the current channel states of all receivers so as to maximize total throughput. This mechanism has a performance that is indicative of the different selection ratio based throughput maximization algorithms above.

For reliable broadcasting in multicast applications, we use erasure coding as in [2, 4, 10, 11]. The main difference that distinguishes our work is the optimization of completion time in finite horizon multicasting. Further, we do not assume deterministic packet errors, i.e., channel outage, but model the packet error rate based on the channel conditions and the chosen MCS. We evaluate the proposed algorithm under realistic channel assumptions that include the effects of small-scale Rayleigh fading.

Chapter 3

System Model

We model the system as a time-slotted broadcasting system with a single Base Station (BS) and N mobile users scattered within the coverage radius of the cell. Each user must receive a block of data of B bytes and we assume that, thanks to erasure coding, each packet transmitted by the BS and received by a mobile user is useful if the user has received less than B bytes. In case multiple blocks of data are to be transmitted in succession, the BS will start transmitting the next data block only after all the receivers fully received the current block.

A time slot is of fixed duration. Thus, the BS broadcasts the same number of symbols, which, depending on the MCS corresponds to a variable number of bytes. We assume that the BS can select one of M MCSs, indexed by $m = 1, \dots, M$. The number of bytes per slot that can be transmitted using MCS m is denoted by R_m.

Perfect Channel State Information (CSI) and knowledge of the number of bytes a user has successfully received is assumed to be available at the base station prior to the transmission in each time slot. The users see independent channel instances $h_i[k]$ at each time slot k, and the discrete-time channel model for the received signal at user i is given by:

$$s_i^{\text{rx}}[k] = h_i[k] s^{\text{tx}}[k] + n_i[k], \tag{3.1}$$

where $s_i^{\text{rx}}[k]$ is the signal received by user i at time slot k, $s^{\text{tx}}[k]$ is the signal broadcast from the BS at time slot k, and $n_i[k]$ is the additive white Gaussian noise term with power spectral density N_0.

For the analysis, we assume a discrete set of C channel realizations $\mathcal{H}_i$ for user i. The probability of user i seeing channel coefficient $h_i \in \mathcal{H}_i$ in a slot is assumed to be $\alpha_i(h_i)$. The vector of channels perceived by all users, also referred to as the channel combination, is denoted by $\mathbf{h} = \{h_i, i = 1 \dots, N\}$ where $h_i \in \mathcal{H}_i$. With some abuse of notation, we denote by $\mathcal{H}$ the set of all possible channel combinations, and by $\alpha(\mathbf{h}) = \prod_{i=1}^{N} \alpha_i(h_i)$, the probability of a channel combination $\mathbf{h} \in \mathcal{H}$. Note that the the total number of channel combinations is $H = C^N$.

Different from the prior work discussed in Section 2, we do not assume deterministic channel outage but use the probability of error (PER) for a given channel and MCS from [13] in order to model the erasure probability. Therefore, for a channel instance $h_i \in \mathcal{H}_i$, the PER for user i under MCS m is represented by $p_i^m(h_i)$, and the probability of success is given by $q_i^m(h_i) = 1 - p_i^m(h_i)$.

In this thesis, we use the following terms:
(1) A *strategy* g specifies the MCS $g(\mathbf{h})$ for each channel combination $\mathbf{h} \in \mathcal{H}$. Hence, the total number of strategies is $S = M^H$. We denote by $\mathcal{G}$, the set of all possible strategies. (2) The *state* consists of the vector of the number of bytes received by each user i denoted by $\mathbf{x} = \{x_i, i = 1, \ldots, N\}$. The state space $\mathcal{X}$ consists of all states where the number of bytes received by all users is positive and less than or equal to B. The *initial state* where none of the users have any information is $\mathbf{x}^0$ and the end state where all the users have received B bytes is denoted by $\mathbf{x}^B$.[1]
(3) A *policy* μ maps any given state $\mathbf{x} \in \mathcal{X}$ to the strategy $g^\mu_\mathbf{x}$ to be used in that state.
(4) The *expected completion time* $D_\mu(\mathbf{x})$ is the mean time required to get from state $\mathbf{x}$ to the end state $\mathbf{x}^B$ under policy μ.

[1] To reduce the state space, we can use a normalized block size by replacing the block size of B bytes by the greatest common divisor of the MCS rates. Thus, each MCS corresponds to transmitting a discrete number of bytes in a slot. As an example, for $M = 2$ and MCSs with rates of 6 Mbps and 9 Mbps, the normalized transmitted data per slot is $R_1 = 2$ and $R_{2=}3$, respectively (in units of 3kb if we assume a slot duration of 1ms).

Chapter 4

Optimization Problem

In this section, we consider the case of memoryless channels and formulate the problem as a stochastic shortest path problem [14] with cost per stage equal to 1 (time needed per slot is fixed, $\tau = 1$) and no terminal cost. We assume that the probability of successfully receiving a packet is non-zero for every combination of modulation scheme and channel condition, though it might be extremely low for some combinations.

4.1 Dynamic programming based solution (*Dyn-Prog*)

Let $\mathcal{E} = \{\mathbf{e} | \, |\mathbf{e}| = N, e_i \in \{0,1\}\}$ be the set of all vectors of size N whose components take values 0 or 1. The transition probability from state $\mathbf{x} \in \mathcal{X}$ to state $\mathbf{y} \in \mathcal{X}$ when MCS m is used under channel combination $\mathbf{h}$ is given by:

$$\rho_{\mathbf{h}}^{m}(\mathbf{x},\mathbf{y}) = \sum_{\substack{\min(\mathbf{x}+R_m\mathbf{e},B)=\mathbf{y} \\ \mathbf{e}\in\mathcal{E}}} \left(\prod_{i=1}^{N} p_i^m(h_i)^{e_i} q_i^m(h_i)^{1-e_i} \right), \tag{4.1}$$

where the above minimization is defined element-wise. Note that in the case of every user experiencing an erasure, the state remains unchanged.

The state space is finite, and there clearly exists a finite integer K such that there is a positive probability of terminating after K steps irrespective of the policy. Thus, the optimal policy μ^* satisfies Bellman's equations at every state $\mathbf{x}$:

$$D_{\mu^*}(\mathbf{x}) = \min_{g\in\mathcal{G}} \left(1 + \sum_{\mathbf{h}\in\mathcal{H}} \alpha(\mathbf{h}) \sum_{\mathbf{y}\in\mathcal{X}} \rho_{\mathbf{h}}^{g(\mathbf{h})}(\mathbf{x},\mathbf{y}) D_{\mu^*}(\mathbf{y}) \right), \tag{4.2}$$

and the optimal strategy at state $\mathbf{x}$

$$g^{\mu^*}(\mathbf{x}) = \operatorname*{argmin}_{g\in\mathcal{G}} \left(\sum_{\mathbf{h}\in\mathcal{H}} \alpha(\mathbf{h}) \sum_{\mathbf{y}\in\mathcal{X}} \rho_{\mathbf{h}}^{g(\mathbf{h})}(\mathbf{x},\mathbf{y}) D_{\mu^*}(\mathbf{y}) \right). \tag{4.3}$$

Since the state space is finite, there are several options to solve for the optimal policy as well as the minimum expected completion time. We choose a simple value iteration approach.

Starting from the end state $\mathbf{x}^B$, we use Bellmann's equation Eq. 4.2 to determine the completion times of the states that only depend on the end state (for which the completion time is known to be 0). We then proceed in the same manner to determine the expected completion times of states that only depend on states for which the completion time is already known, until the completion times for all states are computed. This process also yields the optimum policies from Eq. 4.3.

4.2 A simple two user example

Consider a scenario with two users ($N = 2$), with identically distributed channels. Let $\mathcal{H}_1 = \mathcal{H}_2 = \{L, H\}$, and $\mathcal{H} = \{HH, HL, LH, LL\}$, where L and H denote channels with low and high channel quality, respectively. The base station can choose one of three modulation schemes in each slot. The probability of packet error when MCS m is used is denoted by $p^m(L)$ and $p^m(H)$ for both users under the low and high channel respectively. A strategy is defined by specifying the modulation scheme to be used for each vector channel in $\mathcal{H}$.

In this example, Bellman's equation at state $\{x_1, x_2\}$ is:

$$
\begin{aligned}
D_{\mu^*}(\{x_1, x_2\}) \quad &= 1 + \min_{g \in \mathcal{G}} \sum_{\mathbf{h} \in \mathcal{H}} \alpha(\mathbf{h}) \Big(p^{g(\mathbf{h})}(h_1) p^{g(\mathbf{h})}(h_2) D_{\mu^*}(\{x_1, x_2\}) \\
&+ p^{g(\mathbf{h})}(h_1) q^{g(\mathbf{h})}(h_2) D_{\mu^*}(\{x_1, \min(x_2 + R_{g(\mathbf{h})}, B)\}) \\
&+ q^{g(\mathbf{h})}(h_1) p^{g(\mathbf{h})}(h_2) D_{\mu^*}(\{\min(x_1 + R_{g(\mathbf{h})}, B), x_2)\}) \\
&+ q^{g(\mathbf{h})}(h_1) q^{g(\mathbf{h})}(h_2) D_{\mu^*}(\{\min(x_1 + R_{g(\mathbf{h})}, B), \\
&\qquad \min(x_2 + R_{g(\mathbf{h})}, B)\}) \Big)
\end{aligned}
$$

We evaluate the optimal policy in a scenario where the H and L channels for the users are $\mathcal{H}_1 = \mathcal{H}_2 = \{5dB, 28dB\}$. The probability of L and H are $\alpha_1(L) = \alpha_2(L) = 0.75$ and $\alpha_1(H) = \alpha_2(H) = 0.25$. We choose such a highly variable channel, since it makes it easier to demonstrate the decision tradeoffs that the algorithm makes in the different regions of the state space. The probability of each channel combination in $\mathcal{H}$ can be easily obtained by multiplying the respective channel probabilities. For simplicity, we only use $M = 3$ MCSs with normalized rates of $R_{m=1} = 1$, $R_{m=2} = 4$ and $R_{m=3} = 9$ and the PER for each MCSs and channel instant is listed in Table 4.1.

Table 4.1: PER for different MCS and SNR value pair

$p_i^m(h_i)$	$m = 1$	$m = 2$	$m = 3$
$h_i = 28dB$	0	0	0.08
$h_i = 5dB$	0.23	0.97	1

In Figure 4.1, each drift vector (arrow) reflects optimal policy at that state. It shows the expected future state given the optimum MCSs chosen for the different channel instances, and hence the length of a vectors is indicative of the throughput obtained by the correspond-

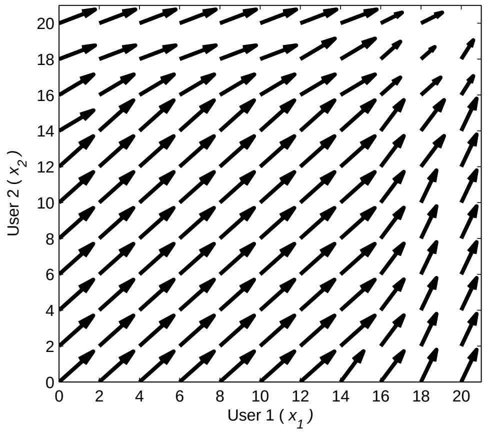


Figure 4.1: Optimal policy with the dynamic programming algorithm (*Dyn-Prog*)

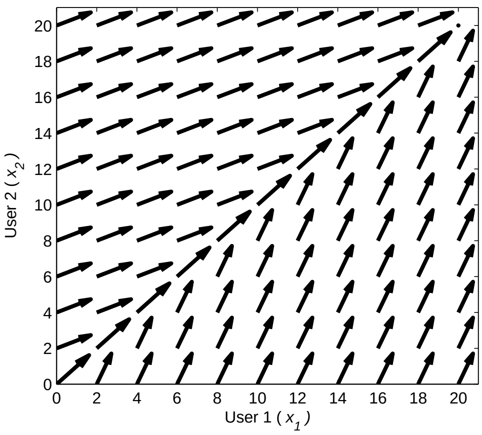


Figure 4.2: Max-Min Algorithm

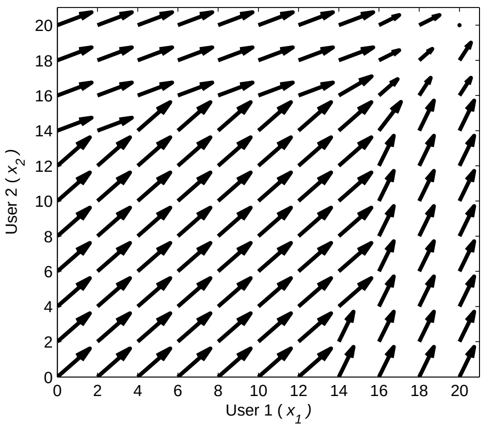


Figure 4.3: Weighted-CT algorithm

ing policy.[1] We set the (normalized) block size $B = 20$. At the initial state $\mathbf{x}^0 = \{0, 0\}$ the optimal policy is $g(\mathbf{h}) = \{1, 3, 3, 3\}$, i.e., MCS $m = 1$ is used for channel combination LL and MCS $m = 3$ is used for channel combinations LH,HL, and HH. This particular policy is a greedy policy which gives the maximum throughput to both users. This policy is also used in almost all states up to $\mathbf{x} = \{14, 14\}$. Closer to the borders of the state space, the policy changes from greedy to more and more favoring the user that is lagging behind. This accounts for the fact that the leading user is likely to finish before the trailing user even if MCS decisions are optimized for the trailing user and aims to prevent the loss of multiuser diversity caused by a user finishing early. For $\{14 < x_1 \leq 16, x_2 < 14\}$ and $\{x_1 < 14, 14 < x_2 \leq 16\}$, the predominant policies are $g(\mathbf{h}) = \{1, 3, 2, 3\}$ and $g(\mathbf{h}) = \{1, 2, 3, 3\}$, respectively, where a more conservative coding modulation scheme is chosen when the trailing user has a bad channel. This policy sacrifices throughput to pre-

[1]For better readability we plot only every other policy vector and double their lengths.

vent the trailing user from falling further behind. Even closer to the borders the policies are $g(\mathbf{h}) = \{1, 3, 1, 3\}$ and $g(\mathbf{h}) = \{1, 1, 3, 3\}$, even further trading off overall throughput for a higher packet reception probability for the trailing user. When both users received a similar number of bytes and are close to the end state $\mathbf{x}^B$, the algorithm also chooses a more conservative MCS indicated by a shorter arrow length to avoid overshooting (i.e., unnecessarily delivering more than B bytes to both users).

While solving the stochastic shortest path problem minimizes average completion time and provides the optimal policy, the size of the state space and the computational complexity increase exponentially with N, the number of users. Therefore, despite providing the optimal solution, the above approach is not, in general, practical for actual network implementation.

Chapter 5

State-Aware Heuristics

Due to the high complexity of the *Dyn-Prog* algorithm introduced in the previous section, we propose two low-complexity heuristics that mimic the characteristic of *Dyn-Prog* algorithm.

5.1 Maximize minimum throughput heuristic (*Max-Min*)

This heuristic is based only on the current state (the number of bytes that each user has successfully received), and the current channel conditions. At each slot, the user with the least number of received bytes is identified as the worst user who is most likely to require the highest number of slots to receive all data. Note that this is indeed the case when the users are homogeneous and perceive identical channel distributions. In the case of heterogeneous users, those users that perceive channel conditions that are worse (on average) are highly likely to also be the trailing users and thus most likely to finish last. The algorithm uses in each slot the MCS that maximizes the throughput for the trailing user. If both users have the same number of bytes, the algorithm greedily maximizes sum throughput for both.

Figure 4.2 depicts the average drift resulting from such a policy in a scenario with the same parameter setting as explained in Section for homogeneous users. There are two predominant strategies that are used for all states off the diagonal. As *Max-Min* sacrifices overall throughput in favor of the trailing user as soon as a user falls behind, the resulting strategies are $g(\mathbf{h}) = \{1, 3, 1, 3\}$ and $g(\mathbf{h}) = \{1, 1, 3, 3\}$. On the diagonal, *Max-Min*'s sum throughput maximization leads to the same strategy as in the *Dyn-Prog* solution, except for the last state before finising. Since in contrast to the *Dyn-Prog*, *Max-Min* does not explicitly take expected completion time into account, it does not switch to more conservative symmetric strategies of $g(\mathbf{h}) = \{1, 2, 2, 2\}$ and $g(\mathbf{h}) = \{1, 1, 1, 1\}$, respectively, that deliver the required number of bytes to finish with a lower packet loss probability compared to using the highest MCS $m = 3$ (which would result in delivering more bytes than necessary to the receivers).

Overall, we note that compared to the optimal *Dyn-Prog* the *Max-Min* algorithm is more conservative and ensures that the progression of state is with high probability along the diagonal where both users have the same number of bytes.

5.2 Weighted completion time heuristic (*Weighted-CT*)

In many cases, favoring the trailing user is overly conservative. In particular, when the number of pending bytes is large for all users and the relative lag is small, the probability that the currently trailing user finishes last is small. We now present a heuristic that more closely models the decisions taken by the *Dyn-Prog* algorithm to achieve a better tradeoff between instantaneous sum throughput and balancing the number of bytes pending for different users.

At slot k, with state $\mathbf{x}$ and channel $\mathbf{h}$, we evaluate the average drift and determine the expected next state, $\mathbf{y}^m$, conditioned on using modulation scheme m as:

$$y_i^m = x_i + q_i^m(h_i)R_m,\ i = 1, \ldots, N \tag{5.1}$$

We then estimate the additional time required, on average, for all users to receive B bytes relative to other states. Since computing the average remaining time under the optimal policy is computationally intensive, we use a weighted Euclidean distance measure in order to characterize the difference in completion times from different states. The metric $\tau_{\mathbf{y}}$associated with state $\mathbf{y}$ is:

$$\tau_{\mathbf{y}} = \sqrt{\sum_{i=1}^{N} \left(\frac{(B - y_i)}{w_i}\right)^2} \tag{5.2}$$

Here, the weights reflect the average channel conditions perceived by each user, with a user that perceives poor channels on average associated with a lower weight. The modulation scheme chosen at slot k is $m^* = \arg\min_m \tau_{\mathbf{y}^m}$.

5.2.1 Choice of weights

We choose weights w_i that are proportional to the average throughput achieved by the user under a hypothetical policy that chooses the MCS uniformly at random, i.e.,

$$w_i = \sum_{\mathbf{h} \in \mathcal{H}} \sum_{m=1}^{M} \alpha(\mathbf{h}) q_i^m(h_i) R_m.$$

As the actual choice of MCS depends on the state as well as the channels of the other users and cannot be determined in advance (sort of using the optimum decisions given by the *Dyn-Prog*), this hypothetical policy is a very simple method to capture the relative throughput differences among the users.

In practical scenarios, the channel distribution of individual users may not be known in advance. Further, the channel statistics of a mobile user may change over time, albeit on time scales that are slow with respect to average completion times. In such settings, we use exponentially weighted averaging in order to track the user weights. The estimated weight of user i at slot k, $\hat{w}_i[k]$, when the perceived channel is $\mathbf{h}$ is given by:

$$\hat{w}_i[k] = (1 - \beta)\hat{w}_i[k-1] + \beta \sum_{m=1}^{M} q_i^m(h_i) R_m, \tag{5.3}$$

where β is a constant that is chosen to be sufficiently small.

Figure 4.3 shows the drifts for the proposed *Weighted-CT* algorithm. Our choice of weights indeed captures well the relative desirability of the different states. While the set of strategies used is not as rich as in the *Dyn-Prog* approach, in particular at the transition between the greedy throughput maximization strategy and the more conservative border strategies, the strategies in the majority of state space are almost the same. In particular, this holds for states around the diagonal which are much more likely to occur in practice than states far off the diagonal where the number of bytes for the two users differs a lot.

Chapter 6

Results

In this section, we evaluate and study the performance of our proposed algorithms in homogeneous and heterogeneous scenarios and compared them to the existing *broadcast* and *greedy* schemes. For small scenarios ($N = 2$) we also compare our results to the optimal *Dyn-Prog* solution. First of all, we study the performance in a simple scenario, i.e., $N = 2$ users, $C = 2$ channel instances to build an intuition towards more complicated scenarios. Later, we study the impact of block size B as well as number of user N and finally, we analyze performance under multipath Rayleigh fading channels with path loss model from COST207 [15] and Winner Model II [16], respectively.

In all of the simulations, we consider 3 modulation types BPSK, QPSK and QAM with channel codes code with code rates $1/2$, $2/3$ and $3/4$, resulting in seven different MCS. These correspond to rates of 6, 12, 24, 36, 48, and 54 Mbps. The corresponding PER with respect to the instantaneous channel quality for each MCS is obtained from [13]. For the performance metric we use average throughput which we compute as follows:

$$\eta = \frac{B}{D} \quad Mbps \tag{6.1}$$

where η is the average throughput, B is the normalized block size and D is the completion time. Here, we consider normal MTU sized packets of $1500 Bytes$. In this Chapter, block size and normalized block size are use interchangeably but both of them represent the size normalized block size corresponding to the greatest common divisor of the MCS rates.

6.1 Comparison to optimal *Dyn-Prog* solution

In this section, we first analyse the performance of the different algorithms in a simple $N = 2$ with $C = 2$ channel instances and $M = 7$ modulations scenarios. This simple scenario allows us to obtain the optimum *Dyn-Prog* solution. We study both homogeneous and heterogeneous user scenarios.

6.1.1 Homogeneous network

In the homogeneous scenario, both users have the same channel parameters (but the channel instances themselves are independent for both users). Here, we also present the result of increasing the channel variability δ of the users. Channel variability δ is the difference between the H and L channel of each user. For instance, in this case the lowest channel variability is $\delta = 4$ (H = 11dB and L = 6.7dB) and the highest channel variability is $\delta = 20.8$ (H = 20dB and L = 0.2dB). Each channel variability pair is chosen such that the average throughput for each pair remains fixed, which facilitates the comparison of the results. In addition, we also fix the stationary channel probability for high ($\alpha(H) = 0.25$) and low ($\alpha(L) = 0.75$) channel. The normalized block size is $B = 200$.

Figure 6.1 depicts the average throughput of each scheme as the channel variability δ increases. As expected, *greedy* scheme performs well since the users have the same average channel conditions. *Greedy* scheme optimizes the throughput therefore, in this scenario it is able to equally serves both users. The *Weighted-CT* scheme similarly optimizes throughput and simultaneously serves all the user in a fair manner while ensuring low delay. On the other hand, the *Max-min* scheme is overly conservative towards the trailing user as it tries to balance the users even if the trailing user is only lagging by a small number of bits. As the *broadcast* scheme is limited by the user with the worst channel, it is transmitting at the lowest MCS most of the time. While it is expected that the performance of both *Max-min* and *broadcast* decreases since the L channel degrades as the variability increases, it is interesting that the performance of *greedy*, *Weighted-CT* and *Dyn-Prog* first decreases but then slightly improve later. This is because of the MCS chosen which gives the highest possible throughput. For instance, when $\delta = 4dB$, the throughput optimization choices are $R_m = \{4, 2, 2$ and $2\}$ for channel combination $\mathcal{H}$ as mentioned in Chapter 6.1, but the throughput optimization choice at $\delta = 10dB$ is $R_m = \{4, 4, 4$ and $2\}$. It can easily be calculated that the normalized throughput at $\delta = 10dB$ is lower than that at $\delta = 4dB$. Here, the *Weighted-CT* scheme behaves like the *greedy* scheme in the sense that it tries to give as much as possible to both users. Since the average channels of the users are the same, the weights w for both the users are the same as well. Another unusual performance for *Max-Min* is when it downperfroms all the schemes including *broadcast* because giving at higher rate while the trailer has H channel is not optimal (see per slot throughput comparison in Figure 6.2(a)) as compared to giving to both users as an average rate. On the contrary, *broadcast* scheme performs worse than *Max-min* because the bad channel quality became worst when δ increases and it is unreasonable to serve the worst channel user while sacrificing a lot of throughput. Figure 6.2(b) shows the average achievable sum throughput for *broadcast* scheme is consistently lower than the rest because it ignore the opportunity of sending at a much higher rate that results in higher sum throughput.

6.1.2 Heterogeneous network

In the heterogeneous channel model, we assume that one user does not move (fixed user) while the other user is moving towards BS (mobile user) with an initial distance equivalent to that of the fixed user. In another word, the mobile user have an increment in its H and L channel's SNR value. The first extreme in the graphs is when both users are homogeneous and having high channel $H_{fixed} = H_{mobile} = 15dB$ and low channel $L_{fixed} = L_{mobile} = 5dB$. On the other extreme, the fix user's channel instances is $H_{fixed} = 15dB$ and $L_{fixed} = 5dB$ while the channel instances for the mobile user is $H_{mobile} = 29dB$ and $L_{mobile} = 19dB$. It is clear that, we ensure the channel variability between the H and L channel is

$\delta = 10dB$ and the corresponding probability of H and L channels are $\alpha(H) = 0.25$ and $\alpha(L) = 0.75$.

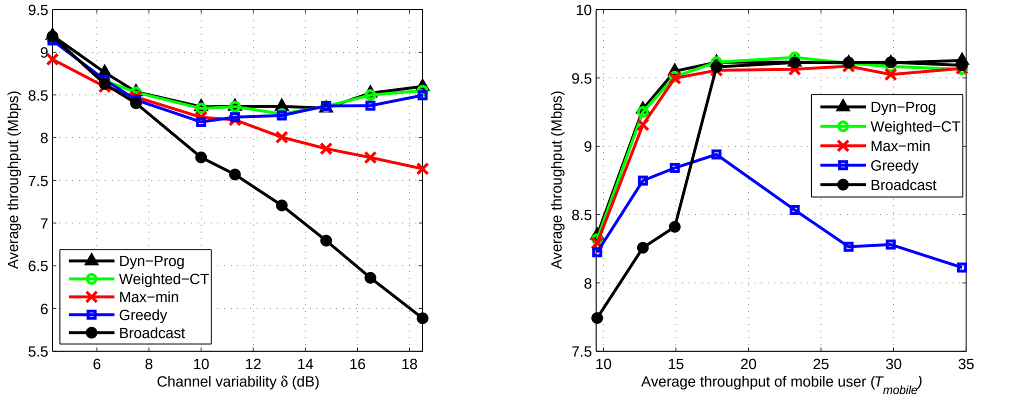
In Figure 6.1(a), the performance of *broadcast* scheme is bad in homogeneous scenario, it also applies when the heterogeneity of the users is not high. As the heterogeneity increases where the average throughput of the mobile user increases, *broadcast* and *Max-min* scheme which always favour the worst or trailing users have advantage over *greedy* scheme. Since the parameters setting in for $\delta = 10dB$ Figure 6.1(a) and first point in Figure 6.1(b) is similar, it gives the same average throughput. As other scheme gradually increase in their performance, *greedy's* throughput also initially improves however it degrades as the heterogeneity increase because the rates used for transmitting a packet is higher and *greedy* tends to give to the mobile user when its channel improves and leaves the fixed user with a higher backlog which cost *greedy* with a higher completion time. However, *Weighted-CT* cares to serve both user under all circumstances, here it behaves like *Max-min* as it assigns lower weight to fixed user than mobile user. In addition, we include the per timeslot throughput performance for low heterogeneity (see Figure 6.2(c)) and high heterogeneity (see Figure 6.2(d)) networks. In these figures, the average sum throughput is computed based on the chosen MCS and channel combination at each time slot. For the purpose of clear comparison between scheme, we take a sample once every 5 time slots. In Figure 6.2(a), we observe a clear two level of average sum throughput. For number of user $N = 2$, on average the first user receive the complete data within 120 slots and the second user complete in an approximate of 150 slots for all the schemes. While Figure 6.2(a) and Figure 6.2(b) only show a two levels of sum average throughput, Figure 6.2(c) and Figure 6.2(d) show more than two levels of sum average throughput. In heterogeneous network, users are distinct, on average, the mobile (good channel quality) user will finish much quicker than the fixed (bad channel quality) user. There exist a clear second and third levels in Figure 6.2(d) especially for *greedy* scheme because the fixed user remains in the system for some times before it receives all the desired information. As we only have two types of user (i.e., mobile and fixed), we cobserve that *greedy* has a much higher throughput than the other scheme because it greedily serves the good users (i.e., users with better channel quality) while ignoring the bad user and when all the good users received all the intended packets, it has a longer time slots with lower throughput because the users left are the bad users with high and low channel of $15dB$ and $5dB$, respectively. We also perform simulations for these scenarios by using a larger block size $B = 5000$ and it results in the same performance for both homogeneous and heterogeneous scenarios.

6.2 Larger scenarios

In this section, we present the simulation results for the scheme in two different higher complexity settings (i.e., impact of increasing the number of users in the system and impact of increasing the block size B and study the performance.

6.2.1 Impact of increasing the number of users

For the scenario where the number of user increases, we fixed the block size to $B = 200$ in both homogeneous and heterogeneous settings. Here, we use the same channel parameters



(a) Homogeneous network with increasing channel variability δ (Point 4 corresponds to Point 1 in Heterogenous graph Figure 6.1(b)).

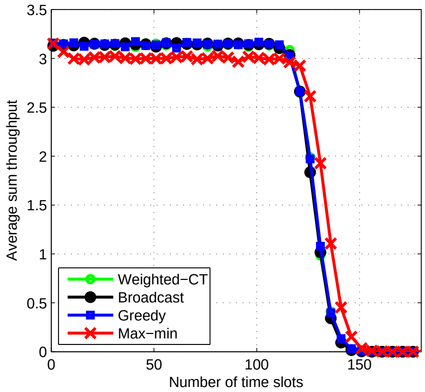
(b) Heterogeneous network with increasing heterogeneity.

Figure 6.1: 2-user scenario with block size $B = 200$

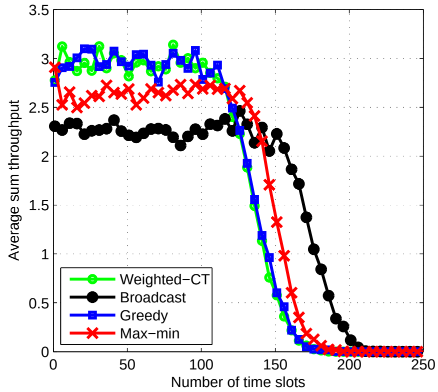
for the H and L channel as in Chapter 6.1. For the homogeneous scenario, the corresponding H and L channels are $19dB$ and $2.5dB$, respectively. In the heterogeneous user case, the corresponding H and L for fixed and mobile users are $H_{fixed} = 15dB, L_{fixed} = 5dB$ and $H_{mobile} = 27dB, L_{mobile} = 17dB$, respectively. The number of user increases exponentially from $N = 2$ to $N = 64$.

Figure 6.3 shows that the average throughput for all the schemes decreases as the number of user increases because it is more difficult to find decisions that jointly optimize performance for a larger number of users. At the same time, since completion time is determined by the slowest user, as the number of user increases, it becomes more probable that some users see a high number of bad channels and lag far behind. In the homogeneous scheme (see Figure 6.3(a)) it is expected that *broadcast* scheme performs the worst in this scenario because it tends to be very conservative and always transmit at the lowest rate at least 90% of the time. *Max-min* however, performs better although it tries to keep both users as close as possible regardless of the relative difference in the number of bits obtained. As explained in the Chapter 6.1, homogeneous user distribution gives advantage to *greedy* scheme because it tries to give to all users at the highest possible throughput which also explains the good performance of *Weighted-CT*.

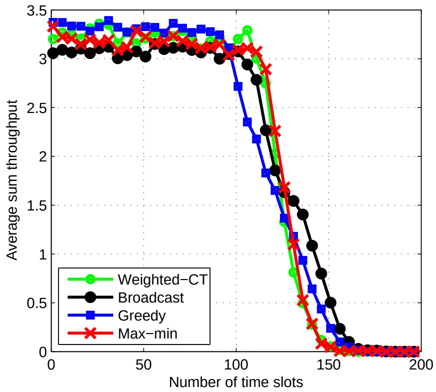
In Figure 6.3(b), we see the same pattern where the average throughput decreases with an increasing number of user for the same reasons as above. It is interesting to note that the *broadcast* scheme performs better than *Max-min* for a higher number of users. This is due to the fact that optimizing for the trailing user is similar to broadcasting to all users. *Max-min* loses throughput when it favors the trailing user, without taking into account that this user may have a better channel later. When the channel of the trailing user becomes better, *Max-min* will transmits at a higher rate causing high PER for low channel user who might continuously see bad channel in the next few slots therefore causing unwanted backlog. Our proposed *Weighted-CT* performs the best since based on its knowledge of the average channel it can ensure that the right tradeoff between user throughput is achieved through



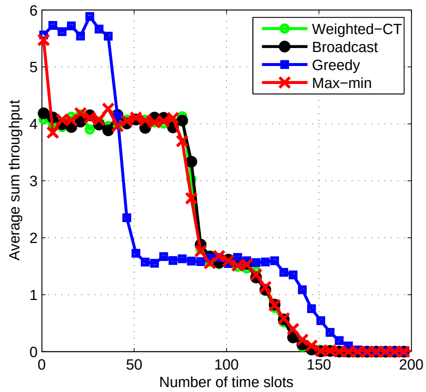
(a) $\delta = 4.3dB$, $H = 11dB$, $L = 6.7dB$



(b) $\delta = 14.8dB$, $H = 18dB$, $L = 3.2dB$



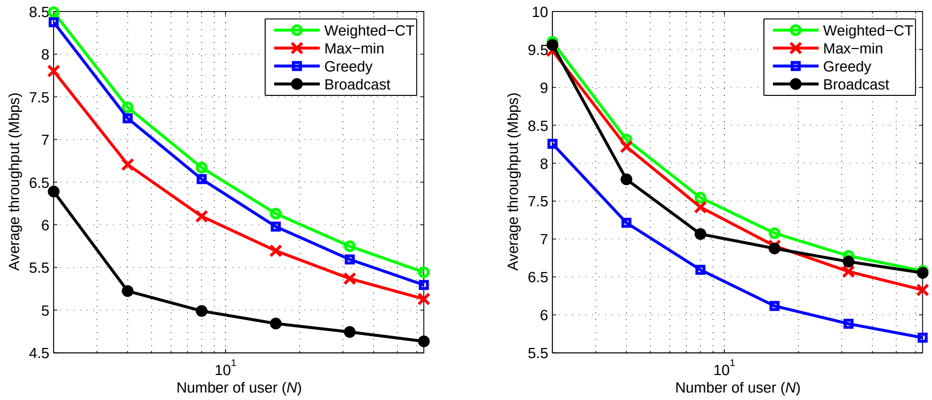
(c) Average throughput = 12.7, $H = 17dB$, $L = 7dB$



(d) Average throughput =29.9, $H = 27dB$, $L = 17dB$

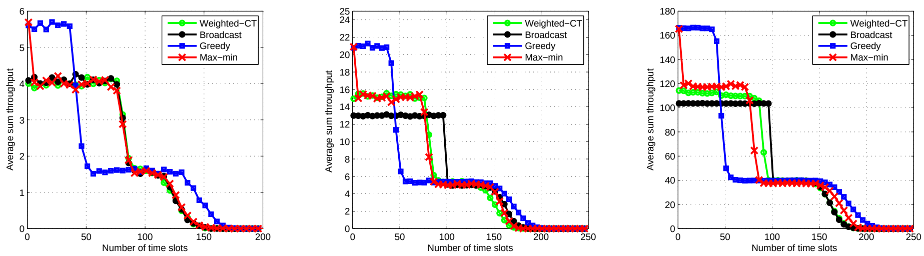
Figure 6.2: Throughput comparison between schemes in homogeneous (Figure 6.2(a) and Figure 6.2(b)) and heterogeneous scenario (Figure 6.2(c) and Figure 6.2(d))

their different weights. It is also interesting to note that, the performance of *broadcast* scheme is unusual when compared to other schemes as the number of user increases. We compare the per-slot sum throughput for different number of user in Figure 6.4 and notice that performance in Figure 6.4(b), as compared to that in Figure 6.4(a) and Figure 6.4(c), the sum throughput difference between *Weighted-CT* and *broadcast* is greater in Figure 6.4(b). *Broadcast* throughput is bounded by the worst channel while *Weighted-CT* and *Max-min* have better chance to transmit at higher rate as the number of user increases while the choice is limited when the number of user is as small as $N = 2$. Therefore for case with $N = 2$, *Weighted-CT* and *Max-min* perform similar to *broadcast*. By equally serving the fixed and mobile users as N increases, *Weighted-CT* obtains lower sum throughput as compared to *Max-min* at the beginning of the transmission but it reduces the time taken for the fixed user to finish because it has obtained more useful information during this period. This as well applies to *broadcast* that try to transmit to all users at each slot. Therefore, *broadcast* achieves a performance close to *Weighted-CT* schemes.



(a) Homogeneous network. (b) Heterogeneous network,

Figure 6.3: Fixed block size of $B = 200$ with increase number of user



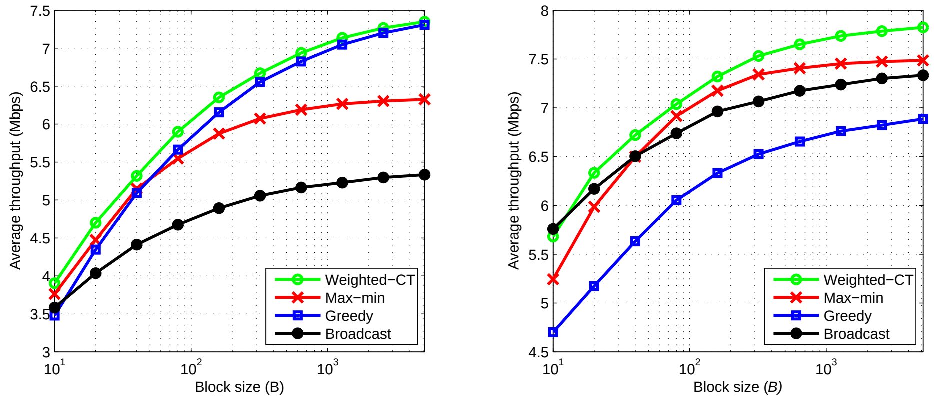
(a) $N = 2$ users. (b) $N = 8$ users. (c) $N = 64$ users.

Figure 6.4: Throughput comparison between schemes for different N in heterogeneous network.

6.2.2 Impact of increasing the block size

We use similar parameters as in Chapter 6.2.1 except that here we set the number of user to $N = 10$ and investigate the performance as the block size increases exponentially from $B = 10$ to $B = 5120$. Different from the performance in Figure 6.3, throughput increases as the block size increases (see Figure 6.5). At smaller block size, it is easier for the good user to finish but this leaves the bad user in the system and it will take as much time as the worst user to finish. With the probability of being in bad channel 3 times greater than good channel, it is very likely to encounter bad channel within the next time slots. However, when the block size increases, users will stay in the system for a longer time. As we compare the drop off time in Figure 6.7(a) and Figure 6.7(b), the fraction of drop off time with respect to the total completion time is larger (approximately 85%) for small block size $B = 10$, and smaller (approximately 60%) for a larger block size $B = 80$. In another word, the longer is the time all the user stay in the system together, the more likely will the time to broadcast to all user be shorter. From all the previous results, we already know that both *broadcast* and *Max-min* downperforms the rest in homogeneous scenario and *greedy* is expected to perform close to *Weighted-CT*. Nevertheless, *greedy* downperforms other schemes at small block size

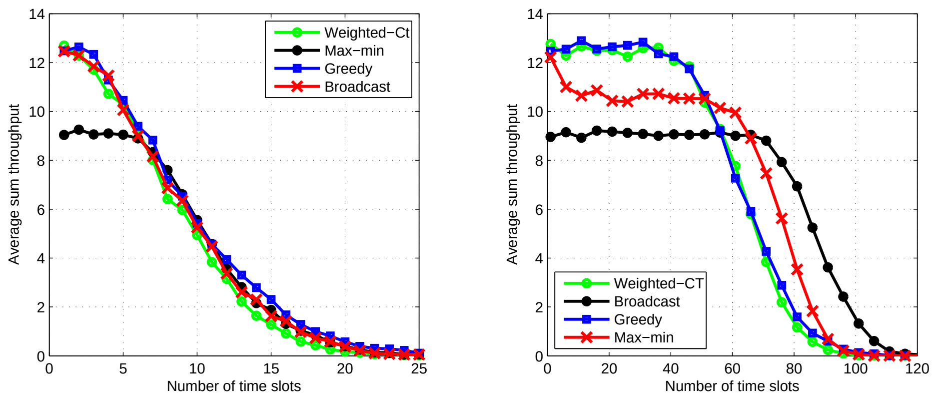
$B = 10$ because its greedy decision for optimizing per slot sum throughput does not allow the users who are currently in $L = 5dB$ channel to obtain any new data byte and cost it a higher completion time (see Figure 6.7(a)).



(a) Homogeneous network. (b) Heterogeneous network.

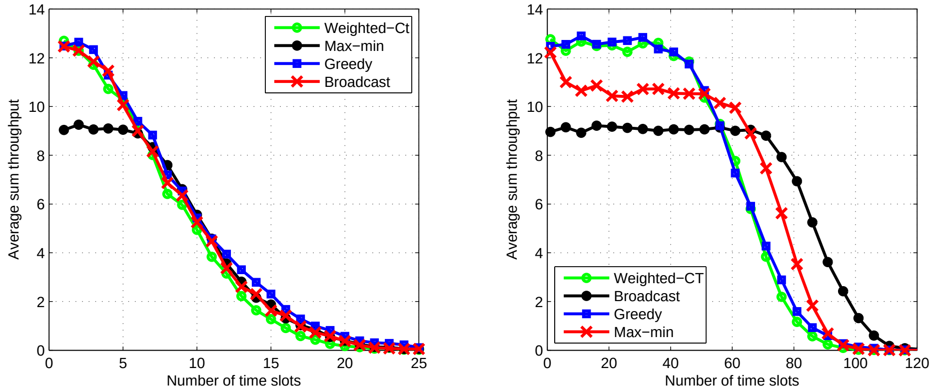
Figure 6.5: Fixed 10 users scenario with increase number of packets

In the heterogeneous users scenario, *broadcast* performs as good as *Weighted-CT* because when the number of packet is as small as $B = 10$ and the highest normalized rate that we have is 9, it is only important to serve the users in the bad channel especially for the case of heterogeneous users when there exists one clear bad user (fixed user). This is also the reason for which *greedy* scheme underperforms the rest of the schemes as it only opportunistically optimize for the mobile (good) users. It can give a much higher instantaneous throughput than favoring the fixed (bad) user. *Max-min* is expected to outperform *broadcast* scheme as block size increases because *broadcast* ignores the opportunity of sending at high rate which caused the lost in throughput.



(a) Small block size ($B = 20$). (b) Large block size ($B = 80$).

Figure 6.6: Average sum throughput per time-slot for different block size in homogeneous network.



(a) Small block size ($B = 10$). (b) Large block size ($B = 80$).

Figure 6.7: Average sum throughput per time-slot for different block size in homogeneous network.

6.3 Mutipath Rayleigh Fading networks

All previously shown results are the baseline for the performance simulated in this section. In the following, we include the actual multipath model (from COST207 model [15]) and path loss model (from Winner Model II [16]) as our channel model. In the subsequent, we evaluate the performance in both homogeneous and heterogeneous user's network for different block size B and number of user N scenarios with different transmit power from the BS. In the homogeneous scenario, we place the users at an equidistant from the base station while in the heterogeneous scenario, we place them at a different distance from the base station. The multipath Rayleigh channel distribution of the users are independent and identically distributed (non i.i.d) in homogeneous scenario and non-i.i.d in heterogeneous scenario. The lowest transmit power is set such that the edge user is still able to receive the transmitted data at its average channel. We also analyse the performance of the effect of having low and high power transmitter. For the different transmission powers (low & high), we show the results for the impact of exponentially increasing block size with fixed number of user. In addition, the performance is also evaluated for exponentially increase in number of user with fixed block size.

6.3.1 Impact of increasing the number of users for low transmit power at BS

Here, we fix the block size to $B = 200$ and observe the impact of increasing number of user from $N = 2$ to $N = 64$ towards the average throughput in both homogeneous and heterogeneous network scenarios with low transmit power at the BS. Figure 6.8 shows the impact of increasing the number of user in both homogeneous and heterogeneous scenarios. The general looks of the figure (i.e., as the number of user increases, the average throughput decreases) is consistent with the results we obtained in the simple scenario with multiple users (see Figure 6.3). However, there are a few differences between the characteristic of the figures which will be highlighted in the following.

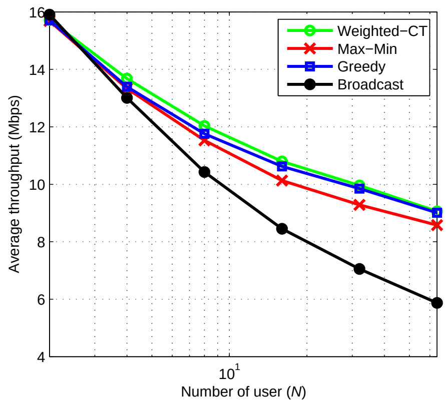
Homogeneous network. In Figure 6.8(a), all the schemes tend to perform the same when $N = 2$ which is in contrast to that in Figure 6.3(a). This is because of the unlikelihood of staying in only two channel instantaneous (i.e., the simple case in Chapter 6.1 and Chapter 6.2 with $C = 2$) due to the Rayleigh distributed channel. In addition, as the number of user in the system is small, the users are not fully homogeneous because the transmission period is too short for the user to see the average channel. Later, the performance converges to what we expected which is comparable to that in the simple case (see Figure 6.3(a)) because the homogeneity among users is achievable as the number of user increases, i.e., the transmission period needed to serves higher N is much longer. Overall, each scheme still behaves in the same way (i.e., *broadcast* scheme still performs the worst among all the other scheme).

Heterogeneous network. In the heterogeneous scenario, we observe from Figure 6.8(b) that the *greedy* scheme performs as expected. However the *broadcast* scheme performs better than the rest of the schemes when the number of user is small. When there are only $N = 2$ users in the system, it is clear in heterogeneous case that the users are having very distinct channel instances (i.e., high SNR for one and low for the other) and the optimal scheme is a scheme that always gives to the user who constantly has the lower channel SNR among the two. Therefore, in this situation, *Max-min* and *broadcast* schemes perform the best. The channel variation δ between H and L channel is always $\delta = 10dB$ in Chapter 6.1.2 and the lowest channel SNR is $5dB$. In contrast, δ in multipath Rayleigh fading from one channel instant to another can be very small (e.g. $\delta = 0.1dB$) and the lowest channel SNR in deep fades can be below $0dB$. Therefore, the smooth transition between the highest and lowest channel SNR gives throughput avantage to *Max-min* over *broadcast* scheme for the *trailer* (trailing user) does not always has the lowest channel SNR. Therefore, *Max-min* has high chance of transmitting at a higher rate as compared to *broadcast* scheme. In addition, the deep fade caused by multipath Rayleigh fading induced a high throughput lost to *broadcast* scheme over all the other schemes in a system with higher number of user (e.g. $N = 64$) for transmitting at a lowest possible rate. In contrast, the *trailer* is not always the user with the worst channel in *Max-min* scheme. Therefore deep fade channel does not greatly affect the throughput performance of *Max-min*.

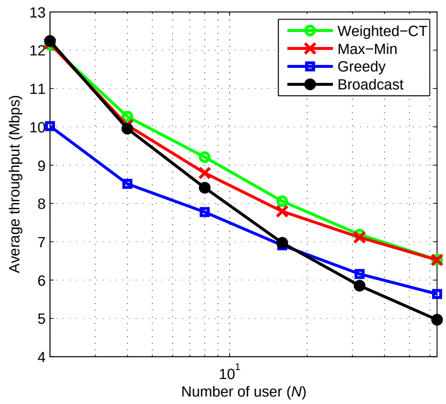
6.3.2 Impact of increasing block size for low transmit power at BS

Here, we fix the number of user to $N = 10$ and observe the impact of increasing block size from $B = 10$ to $B = 5120$ towards the average throughput in both homogeneous and heterogeneous network scenarios for which BS is transmitting at low power.

Homogeneous network. As the block size increases, we observe in Figure 6.10(a) that the performance curve is very similar to that in the simple scenario case (see Figure 6.5(a)). However, the performance of *broadcast* scheme is better than all the other schemes including our proposed schemes. First of all, it only consumes about 12 (see Figure 6.9(a)) and 19 (see Figure 6.9(b)) time slots (as needed by *greedy* scheme) to complete the transmission of a block of size $B = 10$ and $B = 20$, respectively. Within this short period of time, it is hardly possible that two users are completely homogeneous. Therefore, having distinct instantaneous channel users resulting in good performance for both *broadcast* and *Max-min*



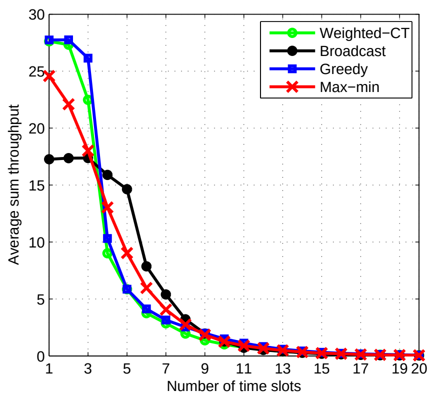
(a) Homogeneous network.



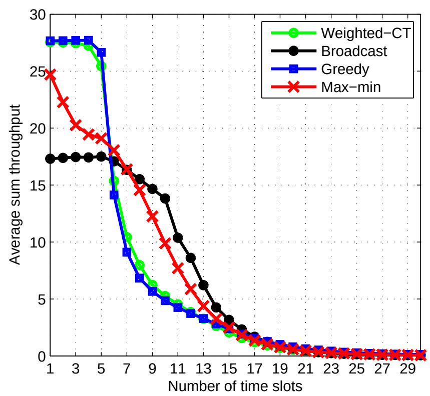
(b) Heterogeneous network

Figure 6.8: Increasing the number of user in multipath network with low transmit power at BS.

schemes because the most important user is the one user who is having the worst channel instances within the entire transmission slots. As the block size increases, the users achieve almost similar average channel SNR (i.e., users are close or almost homogeneous) within a longer transmission slots, therefore, the performance in Figure 6.10(a) is now comparable to that obtained in Figure 6.5(a).



(a) $B = 10$.

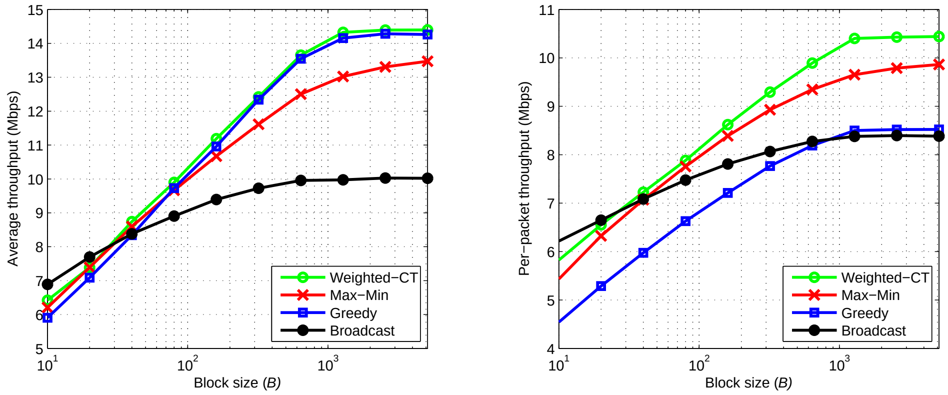


(b) $B = 20$.

Figure 6.9: Per time slot average sum throughput performance in homogeneous network.

Heterogeneous network. Figure 6.10(b) depicts the impact of increasing block size towards the performance of each scheme. For the same reason explained above, the existing of a clear worse user caused *broadcast* scheme to outperform the rest of the scheme when the block size is very small. *Weighted-CT*, *Max-min* and *greedy* schemes perform as expected where *greedy* scheme performs the worst among other scheme in heterogeneous network. *Broadcast* scheme performs unusually due to the occurance of deep fade in the multipath

Rayleigh fading channel model and the impact is previously explained in Chapter 6.3.1.



(a) Homogeneous multipath network.

(b) Heterogeneous multipath network.

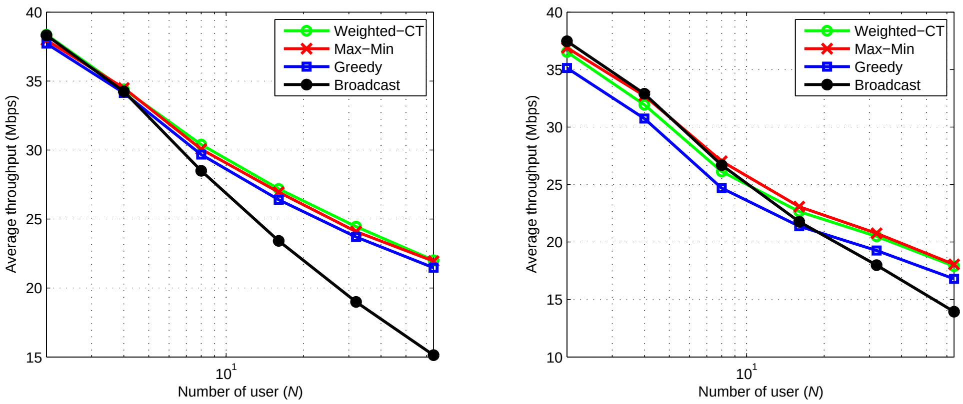
Figure 6.10: Increasing the block size (*B*) in multipath rayleigh fading network with low transmit power at BS.

6.3.3 Impact of increasing the number of user for high transmit power at BS

Different from Chapter 6.3.1, here we consider a scenario where the transmit power of the BS is increased. Similarly, we can also place the users closer to the BS as compared to that in Chapter 6.3.1 instead of increasing the transmit power at the BS. Like the setting in Chapter 6.3.1, we fix the normalized block size to $B = 200$ and observe the impact of increasing number of user from $N = 2$ to $N = 64$ towards the average throughput in both homogeneous and heterogeneous network scenarios.

Homogeneous network. Figure 6.11(a) shows that the overall performance between schemes is very similar to that in Figure 6.8(a) when users are homogeneous. It is interesting to note that *greedy* scheme downperforms *Max-min* scheme, conflicting the result obtained in Figure 6.11(a). At a certain channel instantaneous, there will be one or more users having channel SNR much higher than the others (channel SNR ranges between $15dB$ and $30dB$) which means transmitting at a much higher rate gives a very high throughput but it burdens the bad channel users. *Greedy* scheme is suboptimal because the maximum average number of time slots needed is less than 80 for *broadcast* scheme (i.e., other schemes need much lower number of time slots). This means that, the users does not see enough channels to obtain the same average channel among users. It is clear that, under this situation, *Max-min* scheme that optimizes for the trailer performs very close to *Weighted-CT* which also pays attention to the trailer. As for *broadcast* scheme, it still performs badly although the bad user does not see deep fade channel as low as that in the low transmit power at BS case (e.g. The average SNR is between $20dB$ and $27dB$ but the deep fade can go from $15dB$ to as low as $0dB$). *Weighted-CT* performs the best among all other schemes because it tries to transmit at the highest possible rate without causing the trailer to have a high backlog.

Heterogeneous network. As mentioned earlier in Chapter 6.3.3, the number of time slot needed for every user in the multicast network to receive the entire block is short when the transmit power is high. When users are heterogeneous, giving more data to the good channel user that results in a slightly shorter average completion time D is the optimal strategy for *Weighted-CT* because the gain from transmitting at high rate for good user (with high channel SNR) is higher than the lost by the low channel SNR's user. This has leads to higher throughput at the earlier time slots and introduces some backlog for the low channel user. Therefore, in this scenario (heterogeneous network with $B = 200$) *Max-min* performs very well. As mentioned in Chapter 6.3.1, short transmission time gives opportunity to *broadcast* and *Max-min* to perform better than the other scheme especially when the number of user low and only requires an average of 30 to 40 time slots to complete the multicast transmission.



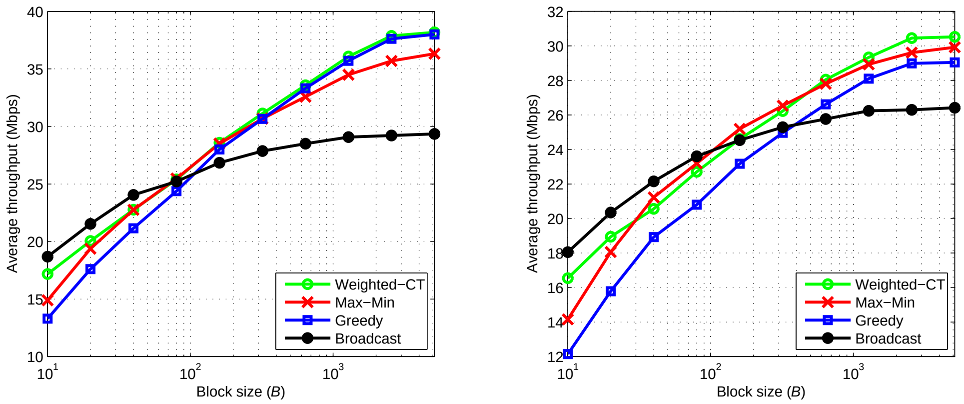
(a) Homogeneous users (b) Heterogeneous users

Figure 6.11: Increasing the number of user in high power transmitter network

6.3.4 Impact of increasing block size for high transmit power at BS

Here, we fix the number of user $N = 10$ and observe the impact of increasing block size from $B = 10$ to $B = 5120$ towards the average throughput in both homogeneous and heterogeneous network scenarios for which BS is transmitting at high power.

Homogeneous network. From Figure 6.12(a), it is expected that *broadcast* outperforms the other scheme for two reasons: (1) There is a clear worst user within a short average time slots of 20 (corresponding to $B = 40$), therefore it will take as much time as the worst user to finish. (2) The users are not completely homogeneous due to the short transmission session. It is interesting to note that *Max-min* performs worst than *Weighted-CT* for small block size. As the block size increases, it performs close to *Weighted-CT* and finally at a large block size, it again underperforms *Weithted-CT*. It is clear that, with a very short average transmission slot, the users are heterogeneous. In the first time slot, *Max-min* treats users equally therefore transmitting at the *greedy* rate. Since the number of time slot is very small, the impact of the bad decision in one time slot is 20% on the completion time. This also explains the performance of *greedy* scheme for small block size. Increasing the block size stabilize the performance of each scheme because the homogeneity between users is achievable. Therefore, *Weighted-CT* and *greedy* that optimize for overall throughput perform equally well over *Max-min* scheme. *Broadcast* scheme performs badly because it sacrifices a lot of throughput as it tries to serve all the users with the least transmission rate limited by the worst channel at a time slot. The performance of *Max-min* lies between *Weighted-CT* and *broadcast* because it not only serves the trailer but it also obtains chances to serve at a high rate when the trailer has a good channel (a chance to achieve higher throughput).



(a) Homogeneous network. (b) Heterogeneous users

Figure 6.12: Increasing the number of packets in in high power transmitter network

Heterogeneous network. Figure 6.12(b) also inherited the same performance of *broadcast* scheme as in Figure 6.10 and it is as expected. It is also clear that *Max-min* performs close to *Weighted-CT* in heterogeneous users scenario. One major difference between transmitting at high power (see Figure 6.12(b)) and low power (see Figure 6.10(b)) in heterogeneous network is the performance of *greedy* scheme at large block size. With high power, *greedy* performs much closer to *Weithted-CT* because the users are having better channel SNR and even if the BS is transmitting at a higher rate, the worst user still manage to receive the transmitted data bytes with a reasonable success rate. In contrast, transmitting at a high rate with low transmission power at BS leads to high PER for the user who are close to the cell's edge. This however does not apply if the block size is small because short transmission window (multicast session) does not allow the user who sees a low channel SNR now to see good channels in the near future.

Chapter 7

Future work

In this thesis, our main objective is to derive an optimal solution with two other heuristics (*Max-min* and *Weighted-CT*) that minimize the total time required for all the multicast users to successfully receive a common set of data which is based on assumptions such as perfect channel and state knowledge, fixed number of users for the duration of transmitting the common set of data bytes to all multicast users, no constraint on energy and etc. In addition, we only consider a single hop multicast from the BS to the end user. An actual implementation however should not make the aforementioned assumptions. Following is the list of our major focuses of research in the near future:

1. Eliminating the assumption of perfect channel and state knowledge at the BS by taking into consideration the feedback from the end user. The channel quality is estimated based on the received signal strength of the feedback packet from the end user while the state knowledge is directly provided by the end user. Acquiring the feedback from the end user at each time slot is a burden to the network. Therefore, we aim to design a model whereby feedback is only acquired from the user whose channel quality is higher than a certain threshold to avoid serving the user when its channel quality is too low. In addition, this allow the reduction on the feedback load.

2. Joint network coding and cooperative transmission between users in a network coding based sensor network can greatly improve the throughput performance of our scheme. We intend to extend our single hop model to a cooperative network coding model so that a user with more data can perform coding to the received data and forward it by broadcasting it to the users in the same multicast group. We will use our *Weighted-CT* algorithm at the relay station for it to choose the optimal MCS once the end nodes' or relay nodes' channel and state are known.

3. The ability of OMS choosing the set of users to transmit to may allow us to reduce the energy consumed by the mobile station. When a mobile station is in bad channel state, serving this mobile station reduces the per-user throughput and it also does not allow this user to go to its idle state for the next consecutive slots (i.e., in multipath Rayleigh fading channel, once a user observes a bad channel, it commonly takes several time-slots before it sees a good channel) where energy can be saved.

Chapter 8

Conclusions

In this thesis we investigated the finite horizon opportunistic multicast scheduling problem, where a wireless base station has to transmit a fixed amount of erasure coded data to a set of receivers. We designed an algorithm based on dynamic programming that, to the best of our knowledge, is the first to explicitly take into account the system state in terms of received amount of data at each receiver for the selection of the optimum modulation and coding scheme. In addition to the well known tradeoff between broadcast gain and multiuser diversity gain that is inherent to opportunistic multicast scheduling, the finite horizon nature of our problem introduces an interesting further tradeoff, namely that of equalizing the finish times of the users versus the total system throughput. While it has been shown that serving a subset of users with good channels improves throughput, it also increases the likelihood that those users finish before the other users. The more users leave the system early, the lower the multiuser diversity gain that can be obtained through opportunistic scheduling from then on. This tradeoff is state dependent. Intuitively, throughput maximization is a reasonable strategy as long as users are far from finishing, whereas the closer users are to finishing, the more important it becomes to allow lagging users to catch up rather than optimizing throughput for all.

Based on these insights, we designed two simple and pratical heuristics that perform close to the more complex proposed dynamic programming solution and that outperform existing approaches that do not consider state. We performed an extensive range of simulations for homogeneous and heterogeneous user scenarios and showed that our heuristics outperform existing schemes by as much as 40% in realistic Rayleigh fading scenarios.

References

[1] P. K. Gopala and H. E. Gamal, “Opportunistic multicasting,” in *Asilomar Conference on Signals, Systems and Computers*, Nov. 2004.

[2] T.-P. Low, M.-O. Pun, Y.-W. P. Hong, and C.-C. J. Kuo, “Optimized opportunistic multicast scheduling (oms) over wireless cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 791 –801, Feb. 2010.

[3] A. Richard, A. Dadlani, and K. Kim, “Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey,” *IEEE Communications Surveys Tutorials*, pp. 1–15, 2012.

[4] U. Kozat, “On the throughput capacity of opportunistic multicasting with erasure codes,” in *IEEE Infocom*, Apr. 2008.

[5] A. Shokrollahi, “An Introduction to Low-Density Parity-Check Codes,” ser. Lecture Notes in Computer Science. Springer, 2002.

[6] ——, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[7] D. L. Gall, “MPEG: a video compression standard for multimedia applications,” *Communications of the ACM - Special issue on digital multimedia systems*, vol. 34, no. 4, pp. 46–58, Apr. 1991.

[8] P. K. Gopala and H. E. Gamal, “On the throughput-delay tradeoff in cellular multicast,” in *International Conference on Wireless Networks, Communications and Mobile Computing*, Jun. 2005.

[9] T.-P. Low, M.-O. Pun, and C.-C. J. Kuo, “Optimized opportunistic multicast scheduling over cellular networks,” in *IEEE Globecom*, Dec. 2008.

[10] T.-P. Low, M.-O. Pun, Y.-W. P. Hong, and C.-C. J. Kuo, “Optimized opportunistic multicast scheduling (OMS) over heterogeneous cellular networks,” in *IEEE ICASSP*, Apr. 2009.

[11] W. Huang and K. L. Yeung, “On maximizing the throughput of opportunistic multicast in wireless cellular networks with erasure codes,” in *IEEE ICC*, Jun. 2011.

[12] ——, “Optimal opportunistic multicast for minimizing broadcast latency in wireless networks,” in *IEEE ICC*, May 2010.

[13] S. Choudhury and J. D. Gibson, "Payload length and rate adaptation for multimedia communications in wireless lans," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 796–807, 2007.

[14] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*, 3rd ed. Athena Scientific, 2005.

[15] C. . M. Committee, *COST 207: Digital Land Mobile Radio Communications: Final Report*, 1989.

[16] J. MeinilÃď, P. KyÃűsti, T. JÃďmsÃď, and L. HentilÃď, "WINNER II channel models," in *Radio Technologies and Concepts for IMT-Advanced*. John Wiley & Sons Ltd, 2009, pp. 39–92.