Optimal Distributed Scheduling Algorithm for Cooperative Communication Networks

by

MEHDI SALEHI HEYDAR ABAD

Submitted to the Graduate School of Engineering
and Natural Sciences in partial fulfillment of the requirements
for the degree of Master of Science

Sabancı University
January 2015

Optimal Distributed Scheduling Algorithm for Cooperative Communication Networks

by Mehdi Salehi Heydar Abad

APPROVED BY

Assoc. Prof. Dr. Özgür Erçetin ....
(Thesis Advisor)

Prof. Dr. Eylem Ekici Eylem Ekici ....
(Thesis Co-Advisor)

Assoc. Prof. Dr. Özgür Gürbüz Özgür Gürbüz ....

Assoc. Prof. Dr. Müjdat Çetin ....

Assoc. Prof. Dr. Kerem Bülbül ....

DATE OF APPROVAL: January 5, 2015

To my dad and mum

Optimal Distributed Scheduling Algorithm for Cooperative Communication Networks

Mehdi Salehi Heydar Abad

MSc Thesis, 2015

Thesis Advisor: Assoc. Prof. Dr. Özgür Erçetin
Thesis Co-Advisor: Prof. Dr. Eylem Ekici

Keywords: ***Wireless Scheduling, Resource Allocation, Cooperative communication, Throughput Optimal***

There has been an enormous interest towards cooperative communication in recent years. Cooperative communication plays a significant role in providing a reliable communication in wireless networks. Cooperative communication helps overcome fading and attenuation in wireless networks. Its main purpose is to increase the communication rates across the network and to increase reliability of time-varying links. It is known that wireless communication from a source to a destination can benefit from the cooperation of nodes that overhear the transmission.

In this thesis we consider problem of resource allocation in cooperative network consisting of Primary User (PU) and $(N-1)$ Secondary Users (SUs), operating in a shared wireless medium. In our network scenario, PU's dedicated channel suffers from fading. PU, in order to overcome fading and attenuation, grants access of its dedicated channel to other SUs conditioned on their cooperation. Whenever PU's dedicated channel is OFF, its packet can be relayed through SU's. Our ultimate goal is to design a distributed algorithm to achieve optimal throughput properties.

Maximum Weight Scheduling can achieve throughput optimality by exploiting opportunistic gain in general network topology with fading channels. Despite the advantage of opportunistic scheduling, this mechanism requires that the existing central scheduler is aware of network conditions such as channel state and queue length information of users. We break this assumption by considering that only individual information is available at each user. We design a Carrier Sense Multiple Access (CSMA) based algorithm which only uses individual queue length information. We derive exact capacity region of the cooperative network for two user scenario thus establishing superiority of the cooperative network over non cooperative network. Then we prove throughput optimality of our proposed algorithm for two scenarios; first being a cooperative network consisting of N users with only PU having fading channel and second a two user scenario where all existing links suffer from fading.

Acknowledgments

My first acknowledgment is to my supervisor, Dr. Özgür Erçetin. Although this is no surprise I must emphasize that Dr. Özgür Erçetin has done a good deal more for me than most supervisors would and a great more than his job description would suggest. He has begun to teach me that it is possible to appreciate the big picture and the minute details of a complex problem at the same time. Without this unique set of attributes I have no doubt that I would not have lasted long in my research field. Dr. Özgür Erçetin has guided me through a crisis of confidence when I felt that I had nothing to contribute. He has been both friend and mentor. However, it is in the day-to-day supervisory capacity that he has excelled most. When I look back at the range of errors and shortcuts I have attempted to get past him it is bewildering how he has managed to supervise me with a smile and to gently guide me back towards the correct path. I could not ask for a better supervisor and mentor.

I would also like to thank my co adviser Dr. Eylem Ekici for my stay in the Ohio State University and valuable research discussions and guidance. I also would like to express my gratitude to my M.Sc oral examination committee members Dr. Özgür Gürbüz, Dr. Müjdat Çetin, and Dr. Kerem Bülbül for taking their time serving on defense exam committee, and I thank them for kindly reading the thesis and their valuable comments.

I would like to thank Sabanci University for supporting this research. This thesis is also supported in part by European Commission grant FP7-MC-PIRSES-269132 AGILENet.

Contents

List of Figures

Chapter 1

Introduction

There has been an enormous interest towards cooperative communication in recent years. Cooperative communication plays a significant role in providing a reliable communication in wireless networks. Cooperative communication helps overcome fading and attenuation in wireless networks. Its main purpose is to increase the communication rates across the network and to increase reliability of time-varying links. It is known that wireless communication from a source to a destination can benefit from the cooperation of nodes that overhear the transmission. Early works on cooperation of this type, i.e., using a relay as a cooperative element is considered, in [1] which exemplifies this situation. Most of the early works on cooperation scheme focus on the physical layer and on information-theoretic considerations (e.g. [2, 3]). In these settings, information bit streams have been modeled as continuous data flows, while the rate regions have been defined as Shannon rate regions which can be sometimes characterized when symbol length and packet delay are allowed to approach infinity.

In this thesis we consider the problem of resource allocation in cooperative networks consisting of a Primary User (PU) and $(N-1)$ Secondary Users (SUs) operating in a shared wireless medium. In our network scenario, PU's dedicated channel suffers from fading. PU in order to overcome fading and attenuation, grants access of its dedicated channel to the SUs conditioned on their cooperation. Whenever PU's dedicated channel is OFF, its packets can be relayed through SU's. Our ultimate goal is to design a distributed algorithm to achieve optimal throughput properties. The capacity region of the network should be distinguished from the capacity region of a specific policy. The latter being the collection of all traffic load matrices that are sustainable by the specific policy. If a policy achieves capacity region of the network, then the policy is throughput optimal.

Similar problem is considered in [4–7] where a SU acts as a relay to a PU while maintaining its own packets. Stationary policies are developed in those with non throughput optimal properties, while we develop an optimal throughput algorithm by dynamically exploiting stochastic process associated with the network.

Optimal throughput algorithm, known as Maximum Weight Scheduling (MWS) for a general network topology, first proposed in [8]. However, MWS requires the network to select a max-weight schedule in every time slot (the weight of a schedule is the sum of the weights of the scheduled links), which corresponds to finding a max-weight independent set in the interference graph. This is known to be NP-hard for general interference graphs [9] and [10]. In addition, MWS is not amenable to distributed implementation. A centralized equalizer needs to gather necessary information at the beginning of the slot which introduces overhead and as a result degrades efficiency of a time slot and compromises the throughput [11].

Most of the low-complexity scheduling schemes have been designed for networks with nonfading channels. Maximal scheduling is a low-complexity alternative to MWS that is amenable to parallel and distributed implementation [12] and [13]. However, maximal scheduling may only achieve a small fraction of the capacity region [14–16] while the complexity is $O(\log N)$ [16] and N denotes the number of nodes. Greedy Maximal Scheduling (GMS), also known as Longest-Queue-First (LQF), is another natural low-complexity alternative to MWS [17–19] with complexity that grows linearly with the total number of links L [20]. Its performance has been observed to be close to optimal in a variety of wireless network simulations [21] and [22]. The Constant-time scheduling algorithms, instead, can achieve a comparable capacity with $O(1)$ complexity, i.e., the complexity does not grow with the network size [23]. Another class of scheduling policies called Pick-and-Compare has been developed in [24–28] with $O(L)$ complexity. A policy in this class picks a schedule at random, evaluates this and the current schedule by comparing their queue weighted rate sum, and chooses the one with the larger sum as the next schedule. One weakness of this approach is that the comparison process often needs network-wide computations, which incur high complexity. Another class of distributed scheduling, called Queue-length-based Random Access Scheduling policies, uses local message exchanges to resolve contention [29–31]. By adjusting each link's contention probability from the link's local queue information, it provides explicit tradeoffs between efficiency, complexity, and the contention period. Complexity problem in schedulers has been solved by recently developed Carrier-Sensing-Multiple-Access (CSMA)-based scheduling policies [32, 33], which simplify the comparison process by exploiting carrier-sensing. These schedulers also have $O(1)$ complexity.

1.1 Contributions and the Outline of the Thesis

These results indicate that good throughput performance may be attained for non-fading environments using algorithms with very low complexity. We attack the problem of scheduling in a fading environment. Recently, there have been a few other low-complexity schemes that are provably efficient with fading channels [34, 35]. These algorithms use only local information which is defined to be information available to nodes in the same neighborhood. For small networks, where every nodes are assumed to be in each others neighborhood, theses algorithms can achieve optimal throughput performance, given availability of queue sizes and channel states. However, we assume that each user only has access to its individual information (individual queue length information) without any explicit message passing between the nodes.

In Chapter 2 we start by giving a thorough literature review regarding our network model. Then general definitions regarding network model and network layer queuing is described. Then we describe MWS algorithm and how it works in a general network topology. And finally Q-CSMA as a base for our algorithm is described and intuition and idea behind it is stated clearly.

In Chapter 3 we derive MWS for our cooperative network. We assume two different cooperative scenarios. First, we assume that There are N users consisting of a PU and $(N-1)$ SUs with only PU having a fading channel. Then we extend our model in to a network with two users consisting of a PU and a SU in which all possible links suffer from fading. Exact capacity region of the network for case of two users is derived and superior performance of cooperative network over non cooperative network is established.

In Chapter 4 we establish why Q-CSMA is not an appropriate choice in our cooperative model. Then we propose a distributed algorithm which generates time reversible Discrete Time Markov Chain (DTMC) with product form stationary distribution. Using this result we show that if some conditions are satisfied, our algorithm leads to a scheduling policy sufficiently close to MWS, which guarantees the throughput optimality. Then we extend the algorithm to the case with two users with multiple fading channels and prove the throughput optimality.

Finally in Chapter 5 we compare our algorithm numerically with Q-CSMA, simple 802.11 and MWS in terms of of average sum of queues in the network and also average queue size evolution of individual queues. As expected, numerical results are consistent with the analytical results, suggesting throughput optimality of our algorithm. Also, as expected numerical results suggest that Q-CSMA is not throughput optimal.

Chapter 2

Fundamentals

2.1 Literature Review

Cooperative communication helps overcome fading and attenuation in wireless networks. Its main purpose is to increase the communication rates across the network and to increase reliability of time-varying links. It is known that wireless communication from a source to a destination can benefit from the cooperation of nodes that overhear the transmission. Early work on cooperation with means of relaying can be found in [1] which exemplifies this situation. Further work on the relay channel in [36] and [37] has enabled substantial performance improvement. Most of the early works on cooperation scheme focus on the physical layer and on information-theoretic considerations [2, 3, 38–41]. In these settings, information bit streams have been modeled as continuous data flows, while the rate regions have been defined as Shannon rate regions which can be sometimes characterized when symbol length and packet delay are allowed to approach infinity.

Additional improvements can be achieved through network layer design even without any physical layer consideration. In [42] cognitive multiple-access strategy in the presence of a cooperating relay is proposed and its advantages in terms of maximum stable throughput region and the delay performance, over conventional relaying strategies such as selection and incremental relaying has been studied. Benefits of user cooperation again in terms of maximum stable throughput region and the delay performance over the non-cooperative situation is established in [43].

Delay trade-offs in systems with cooperation has been considered in [44] where a secondary user acts as a relay for a primary user. In [45] by dynamically and opportunistically exploiting spatial diversity among the source users, a packet is delivered to the common destination through either a direct link or through cooperative relaying by intermediate source nodes that have a statistically better channel to the destination. The results establish that the stable throughput region strictly contains the stable throughput region achieved without cooperation.

In [4], benefits of using one user of a two-user random access system to relay traffic of the other user is evaluated. A measure of capacity region is maximized by optimizing packet acceptance probability by secondary user. A similar network model is considered in [5] characterizes the stable-throughput region in a two user cognitive shared channel with multi-packet reception, where the primary (higher priority) user transmits whenever it has packets to transmit while the secondary (cognitive) node transmits its packets with probability p. Therefore, in [4] and [5], the secondary link is allowed to share the channel along with the primary link, in contrast to the traditional notion of cognitive radio, in which the secondary user is required to relinquish the channel as soon as the primary is detected.

In [6] and [7] Markovian game solution is adopted to solve problem of throughput optimization in relay networks where all users transmit their packets on a multiple-access channel . In these works, maximization of the system throughput with minimum transmission delay and power consumption cost is considered.

The capacity region of the network should be distinguished from the capacity region of a specific policy. The latter being the collection of all traffic load matrices that are sustainable by the specific policy [46]. A control policy that is optimal in the sense of having a capacity region that coincides with the network capacity region and is therefore a super set of the capacity region of any other policy was introduced in [8] and [47]. That policy, the max weight adaptive back-pressure policy, was generalized later in several ways [48–51] and it is an essential component of policies that optimize other performance objectives. The back pressure policy consists in giving priority in forwarding through a link to traffic classes that have higher backlog differentials.

The stochastic optimal control problem where the objective is the optimization of a performance functional of the system is considered in [49, 52–56]. The development of optimal policies for these cases relies on a number of advances including extensions of Lyapunov techniques to enable simultaneous treatment of stability and performance optimization, introduction of virtual cost queues to transform performance constraints into queuing stability problems and introduction of performance state queues to facilitate optimization of time averages.

As mentioned above, Max Weighted Scheduling (MWS) algorithm is throughput-optimal. However, MWS requires the network to select a max-weight schedule in every time slot (the weight of a schedule is the sum of the weights of the scheduled links), which corresponds to finding a max-weight independent set in the interference graph. This is known to be NP-hard for general interference graphs [9] and [10]. In addition, MWS is not amenable to distributed implementation. Even in small networks, MWS can require quite a lot of operations because its complexity is tied to the number of maximal schedules of the network [57]. Therefore, it is of interest to find simple, distributed scheduling algorithms that can achieve optimal or near-optimal performance.

Most of the low-complexity scheduling schemes have been designed for networks with nonfading channels. Maximal scheduling is a low-complexity alternative to MWS that is amenable to parallel and distributed implementation [12] and [13]. However, maximal scheduling may only achieve a small fraction of the capacity region [14–16] while the complexity is $O(\log N)$ [16] and N denotes the number of nodes. Greedy Maximal Scheduling (GMS), also known as Longest-Queue-First (LQF), is another natural low-complexity alternative to MWS [17–19] with complexity that grows linearly with the total number of links L [20]. Its performance has been observed to be close to optimal in a variety of wireless network simulations [21] and [22]. New bounds on the throughput efficiency of GMS is derived in [58]. The Constant-time scheduling algorithms, instead, can achieve a comparable capacity with $O(1)$ complexity, i.e., the complexity does not grow with the network size [23]. Another class of scheduling policies called Pick-and-Compare has been developed in [24–28] with $O(L)$ complexity. A policy in this class picks a schedule at random, evaluates this and the current schedule by comparing their queue weighted rate sum, and chooses the one with the larger sum as the next schedule. One weakness of this approach is that the comparison process often needs network-wide computations, which incur high complexity. Another class of distributed scheduling, called Queue-length-based Random Access Scheduling policies, uses local message exchanges to resolve contention [29–31]. By adjusting each link's contention probability from the link's local queue information, it provides explicit tradeoffs between efficiency, complexity, and the contention period. Limitations of randomization on the efficient scheduling in wireless networks has been studied in [59, 60]. This framework models many existing schedulers operating under a time-scale separation assumption as special cases and identifies necessary and sufficient conditions on the network topology and on the functional forms used in the randomization for throughput-optimality. Complexity problem in schedulers has been solved by recently developed Carrier-Sensing-Multiple-Access (CSMA)-based scheduling policies [32, 33], which simplify the comparison process by exploiting carrier-sensing. These schedulers also have $O(1)$ complexity. Nonetheless, these results indicate that good throughput performance may be attained for non-fading environments using algorithms with very low complexity.

In practice, however, most wireless systems experience some level of channel fading. When link rates vary across time due to fading, the system throughput can be further improved by scheduling links when their rate are high. This is known as the opportunistic gain [61]. For wireless networks, the MaxWeight algorithm can exploit this opportunistic gain and in fact achieve the optimal throughput even with fading. However, many of the low- complexity scheduling algorithms described in the previous paragraph cannot exploit the opportunistic gain, and their performance in fading environments will be much worse [34, 35, 62].

Recently, there have been a few other low-complexity schemes that are provably efficient with fading channels [34, 35]. These algorithms use only local information which is defined to be information available to nodes in the same neighborhood. For small networks, where every nodes are assumed to be in each others neighborhood, theses algorithms can achieve optimal throughput performance, given availability of queue sizes and channel states.

2.2 Preliminaries and General Definitions

Consider a general network with a set $\mathcal{N}$ of nodes and a set $\mathcal{L}$ of transmission links. We denote by N and L respectively the number of nodes and links in the network. Each link i represents a communication channel for direct transmission from a given node n to another node m, corresponding ordered node pair $(n,\ m)$ (where $n,\ m \in \mathcal{N}$). Note that link $(n,\ m)$ is distinct from link $(m,\ n)$. In a wireless network, direct transmission between two nodes may or may not be possible and this capability, as well as the transmission rate, may change over time due to weather conditions, mobility or node interference [63]. Hence in the most general case one can consider that $\mathcal{L}$ consists of all ordered pairs of nodes, where the transmission rate of link i is zero if direct communication is impossible. However, in cases where direct communication between some nodes is never possible, it is helpful to consider that $\mathcal{L}$ is a strict subset of the set of all ordered pairs of nodes.

The network is assumed to operate in slotted time with slots normalized to integral units, so that slot boundaries occur at times $t \in \{0, 1, 2, 3, \cdots\}$. Hence, slot t refers to the time interval $[\ t,\ t+1\ )$. Let $\boldsymbol{\mu}(t) = (\mu_l(t))$ represent the vector of transmission rates offered over each link l during slot t. By convention, we define $\mu_i(t) = 0$ for all time t whenever a physical link i does not exist in the network. The link transmission rates are determined by a link transmission rate function $R(I, S)$, so that:

$$\boldsymbol{\mu}(t) = R(I(t), S(t))$$

where $S(t)$ represents the network topology state during slot t, and $I(t)$ represents a link control action taken by the network during slot t.

The topology state process $S(t)$ represents all uncontrollable properties of the network that influence the set of feasible transmission rates. For example, the network channel conditions and interference properties might change from time to time due to user mobility, wireless fading, changing weather locations, or other external environmental factors. In such cases, the topology state $S(t)$ might represent the current set of node locations and the current attenuation coefficients between each node pair.

The link control input $I(t)$ takes values in a general state space $\mathcal{I}_{S(t)}$, which represents all of the possible resource allocation options available under a given topology state $S(t)$. For example, in a wireless network where certain groups of links cannot be activated simultaneously, the control input $I(t)$ might specify the particular set of links chosen for activation during slot t, and the set $\mathcal{I}_{S(t)}$ could represent the collection of all feasible link activation sets under topology state $S(t)$.

Every time slot the network controller observes the current topology state $S(t)$ and chooses a transmission control input $I(t) \in \mathcal{I}_{S(t)}$, according to some transmission control policy. This enables a transmission rate vector of $\boldsymbol{\mu}(t) = \mathbf{R}(I(t), S(t))$.

2.2.1 Network layer queuing

We assume that, all data that enters the network is associated with a common destination. Let $A_n(t)$ represent the amount of data that exogenously arrives to source node n during slot t (for all $n \in \mathcal{N}$). We assume that $A_n(t)$ takes units of packets. The arrival vector $(A_n(t))$ is i.i.d. over slots, where $A_n(t)$ take integer units of packets. The arrival rates are given by $\lambda_n = \mathbb{E}\left\{A_n(t)\right\}$. It is assumed that $A_n(t) \leq A_{max}$ for all n and t. The second moments $\mathbb{E}\left\{A_n(t)^2\right\}$ and are assumed to be finite.

Let $Q_n(t)$ represent the current backlog, stored in a network layer queue at node n. Consequently, Let $Q_i(t) = Q_n(t) - Q_m(t)$, represent queue size of link i connecting ordered pair of nodes $(n,\ m)$. We assume that all network layer queues have infinite buffer storage space. Primary goal for this layer is to ensure that all queues are stable, so that time average backlog is finite. A queue is strongly stable if:

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{t-1}^{\tau=0} \mathbb{E}\left\{Q(t)\right\} < \infty \tag{2.1}$$

That is, a queue is strongly stable if it has a bounded time average backlog. So naturally, A network is strongly stable if all individual queues of the network are strongly stable [46].

A network layer control algorithm makes decisions about routing, scheduling, and resource allocation in reaction to current topology state and queue backlog information. The resource allocation decision $I(t) \in \mathcal{I}_{S(t)}$ determines the transmission rates $\mu_l(t) = R_l(I(t), S(t))$ offered over each link i on time slot t. We assume that only the data currently in node n at the beginning of slot t can be transmitted during that slot. Hence, the slot-to-slot dynamics of the queue backlog $Q_n(t)$ satisfies the following equality [64]:

$$Q_n(t+1) = \max\left[Q_n(t) - \sum_{m=1, n\neq m}^{N} \mu_{nm}, 0\right] + \sum_{n=1, n\neq m}^{N} \mu_{mn} + A_n(t) \tag{2.2}$$

where $\mu_{nm}(t)$ denotes the actual amount of data transmitted from node n to node m (i.e., over link $(n,\ m)$) on slot t.

2.2.2 Maximum Weight Scheduling (MWS)

Next, we describe below an algorithm for resource allocation and routing which stabilizes the network whenever the vector of arrival rates lies within the capacity region of the network. The network layer capacity region $\mathbf{\Lambda}$ is the closure of the set of all arrival rate vectors (λ_n) that can be stably supported by the network, considering all possible strategies for choosing the control variables to affect routing, scheduling, and resource allocation [8, 64].The notion of controlling the system to maximize its stability region and the following algorithm that achieves it was introduced in [8, 47] and generalized further in [48, 49, 51]. MWS algorithm works as follow:

Every time slot t, the network controller observes the queue backlog vector $\mathbf{Q}(t) = (Q_n(t))$ and the topology state variable $S(t)$ and performs the following actions for routing and resource allocation.

- Resource Allocation: For each link i, define $\omega_i(t)$ as the weight:

$$\omega_i(t) = \max\left[Q_n(t) - Q_m(t), 0\right] \quad i \text{ connects } n \text{ to } m \tag{2.3}$$

  Choose the control action $I(t)$ that solves the following optimization:

$$\begin{aligned} &\max_{I(t)} \sum_i \omega_i(t) R_i\left(I(t), S(t)\right) \\ &\text{Subject to} : I(t) \in \mathcal{I}_{S(t)} \end{aligned} \tag{2.4}$$

- Routing: For each link i offer a transmission rate of $\mu_i(t) = R_l(I(t), S(t))$.

The weights $\omega_i(t)$ can be determined at each node provided that nodes are aware of the backlog sizes of their neighbors. However, the optimization problem 2.4 that must be solved at the beginning of each time slot requires in general knowledge of the whole network state.

2.2.3 Queue-Length Based CSMA/CA (Q-CSMA)

In [33], a discrete time distributed randomized algorithm is proposed to achieve the full capacity region in a single non-fading channel network. The algorithm of [33] is based on a generalization of Glauber dynamics in statistical physics. In Glauber dynamics, only one link has a state update within a time slot. In scheduling, a state update can be interpreted as a transition of a link from "transmitting" to "idle" or from "idle" to "transmitting". The incremental state update in every time slot leads to a scheduling policy sufficiently close to MWS, which guarantees the throughput optimality. In the following we will briefly describe Q-CSMA as in [33].

2.2.3.1 Assumptions and the idea behind Q-CSMA

We further simplify the network model by assuming that none of the links i account for fading. This implies that network topology state $S(t)$ only accounts for interference model. Also, there exists a directed link $(n, m) \in \mathcal{L}$ if node n can hear the transmission of node m. We assume that if $(n, m) \in \mathcal{L}$, then $(m, n) \in \mathcal{L}$. For interference model, let us denote $\mathcal{C}(i)$ as the set of conflicting links (called conflict set) of i for any $i \in \mathcal{L}$. This means, $\mathcal{C}(i)$ is the set of links such that if any one of them is active, then link i cannot be active. Conflict set $\mathcal{C}(i)$ includes; *node-exclusive constraint* and *radio interference constraint*, where, the first constraint accounts for nodes sharing a common node with i (i.e., two links sharing a common node cannot be active simultaneously) and the latter accounts for nodes that are close to each other (i.e., links that will cause interference to link i when transmitting). There is symmetry in the conflict set so that if $i \in \mathcal{C}(j)$ then $j \in \mathcal{C}(i)$. A feasible schedule(Collision free) is a set of links that can be active at the same time according to the conflict set constraint, i.e., no two links in a feasible schedule conflict with each other. A schedule is represented by a vector $\mathbf{x} \in \{0,1\}^L$. The i^{th} element of $\mathbf{x}$ is equal to 1 (i.e., $x_i = 1$) if link i is included in the schedule; $x_i = 0$ otherwise. According to the conflict set constraint a feasible schedule $\mathbf{x}$ satisfies the following condition:

$$x_i + x_j \leq 1, \text{ for all } i \in \mathcal{L} \text{ and } j \in \mathcal{C}(i)$$

Let $\mathcal{M}$ be the set of all feasible schedules of the network.

For this interference model MWS selects a maximum weight schedule $\mathbf{x}^*(t)$ in every time slot t such that

$$\sum_{i \in \mathbf{x}^*(t)} \omega_i(t) = \max_{\mathbf{x} \in \mathcal{M}} \sum_{i \in \mathbf{x}(t)} \omega_i$$

The key step of the MaxWeight algorithm is to find a feasible schedule which its links have the maximum weight. In the original MaxWeight algorithm, weight of a link i is defined to be the queue length Q_i of that link, i.e., $\omega_i(t) = Q_i(t)$. This result was generalized in [65] as follows. For all $i \in \mathcal{L}$, let link weight $\omega_i(t) = f_i(Q_i(t))$, where $f_i : [0,\ \infty] \rightarrow [0,\ \infty]$ are functions that satisfy the following conditions:

1. $f_i(Q_i)$ is a non decreasing, continuous function with $\lim_{Q_i \to \infty} f_i(Q_i) = \infty$.
2. Given any $M_1 > 0$, $M_2 > 0$ and $0 < \epsilon < 1$, there exists a $Q < \infty$, such that for all $Q_i > Q$ and $\forall i$, we have

$$(1-\epsilon) f_i(Q_i) \leq f_i(Q_i - M_1) \leq f_i(Q_i + M_2) \leq (1+\epsilon) f_i(Q_i). \tag{2.5}$$

Q-CSMA implements MaxWeight scheduling in a distributed fashion when link weights change slowly over time, so is throughput optimal. The key idea behind Q-CSMA is to select feasible schedules according to the following distribution:

$$\pi(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathbf{x}} e^{\omega_i(t)} = \frac{e^{\sum_{\in \mathbf{x}} \omega_i(t)}}{Z} \tag{2.6}$$

where, $\omega_i(t)$ is the associated weight of link i and

$$Z = \sum_{\mathbf{x} \in \mathcal{M}} \prod_{i \in \mathbf{x}} e^{\omega_i(t)}$$

The reason to choose such a distribution, is that if an algorithm generates schedules according to (2.6), then the following results can be applied to [65]:

Let $\omega^*(t) := \max_{x \in \mathcal{M}(t)} \sum_{i \in x(t)} \omega_i(t)$, where $\mathcal{M}(t)$ is the set of all feasible schedules at time t. For a scheduling algorithm, if given any $0 < \epsilon, \delta < 1$, there exists $\beta > 0$ such that: if $\omega^*(t) > \beta$, the scheduling algorithm chooses a schedule $\mathbf{x}(t) \in \mathcal{M}(t)$ that satisfies

$$Pr\left\{ \sum_{i \in \mathbf{x}(t)} \omega_i(t) \geq (1-\epsilon)\omega^*(t) \right\} \geq 1 - \delta \tag{2.7}$$

then the scheduling algorithm is throughput optimal.

2.2.3.2 Q-CSMA

Next, we describe a distributed algorithm (i.e., QCSMA) that generates schedules according to distribution (2.6). We assume that ω_i's are fixed and do not change with time. In reality, ω_i will change, but if it changes very slowly, for example, if $f_i(Q_i)$ is chosen to be slightly smaller than $\log(1+Q_i)$; one can show that the stability results will not be affected, in manner that can be precisely described. We will describe a DTMC whose states are the feasible schedules $\mathbf{x}$, and show that the steady-state distribution of this DTMC has the desired form. We will then describe a distributed algorithm under which the MAC layer behaves like the DTMC. Now, we describe the basic scheduling algorithm.

Let us divide each time slot t into a control slot and a data slot. The purpose of the control slot is to generate a collision-free transmission schedule $\mathbf{x}(t) \in \mathcal{M}$ used for data transmission in the data slot. To achieve this, the network first selects a set of links that do not conflict with each other, denoted by $\mathbf{m}(t)$. Note that these links also form a feasible schedule, but it is not the schedule used for data transmission. We call $\mathbf{m}(t)$ the *decision schedule* in time slot t.

Let $\mathcal{M}_0 \subset \mathcal{M}$ be the set of possible decision schedules. The network selects a decision schedule according to a randomized procedure, i.e., it selects $\mathbf{m}(t) \in \mathcal{M}_0$ with positive probability $\alpha(\mathbf{m}(t))$, where $\sum_{\mathbf{m}(t) \in \mathcal{M}_0} \alpha(\mathbf{m}(t)) = 1$. Then, the transmission schedule is determined as follows. For any link i in $\mathbf{m}(t)$, if no links in $\mathcal{C}(i)$ were active in the previous data slot, then link i is chosen to be active with an activation probability p_i and inactive with probability $\bar{p}_i = 1 - p_i$ in the current data slot. If at least one link in $\mathcal{C}(i)$ was active in the previous data slot, then i will be inactive in the current data slot. Any link not selected by $\mathbf{m}(t)$ will maintain its state (active or inactive) from the previous data slot.

A necessary and sufficient condition for the DTMC of the transmission schedules to be irreducible and aperiodic can be found in Proposition 1 of [33]. Also this algorithm generates a DTMC which is reversible and has the following product form stationary distribution:

$$\pi(\mathbf{x}) = \frac{1}{Z}\prod_{l\in\mathbf{x}}\frac{p_i}{\bar{p}_i} Z = \sum_{\mathbf{x}\in\mathcal{M}}\prod_{l\in\mathbf{x}}\frac{p_i}{\bar{p}_i} \tag{2.8}$$

This suggests that if we choose:

$$p_i = \frac{e^{\omega_i(t)}}{1+e^{\omega_i(t)}}, \quad \forall i \in \mathcal{L} \tag{2.9}$$

Then, (2.6) and (2.8) are equivalent. Proof of optimality can be found in Proposition 2 of [33].

To implement Q-CSMA in a distributed manner control slot is further divided into control mini-slots. Recall that in the control slot, a collision-free transmission schedule is generated and used for data transmission in the data slot. Note that once a link knows whether it is included in the decision schedule, it can determine its state in the data slot based on its carrier sensing information (i.e., whether its conflicting links were active in the previous data slot) and activation probability. Detailed algorithm is described in Algorithm 1 [33]:

Algorithm 1 Q-CSMA Algorithm

```
(at Link $i$ in Time slot $t$)
Link $i$ selects a random (integer) back-off time $T_i$ uniformly in $[0, W-1]$ and waits
  for $T_i$ control mini-slots.
IF link $i$ hears an INTENT message from a link in $\mathcal{C}(i)$ before the $(T_i+1)$-th control
  mini-slot, $i$ will not be included in $\mathbf{m}(t)$ and will not transmit an INTENT message
  anymore. Link $i$ will set $x_i(t) = x_i(t-1)$.
IF link $i$ does not hear an INTENT message from any link in $\mathcal{C}(i)$ before the $(T_i+1)$-
  th control mini-slot, it will send (broadcast) an INTENT message to all links in $\mathcal{C}(i)$
  at the beginning of the $(T_i+1)$-th control mini-slot.
if there is a collision then
    link $i$ will not be included in $\mathbf{m}(t)$ and will set $x_i(t) = x_i(t-1)$.
if there is no collision then
    link $i$ will be included in $\mathbf{m}(t)$ and and decide its state as follows:
    if no links in $\mathcal{C}(i)$ were active in the previous data slot then
        $x_i(t) = 1$ with probability $p_i$
        $x_i(t) = 0$ with probability $1 - p_i$
    else
        $x_i(t) = 0$
IF $x_i(t) = 1$, link $i$ will transmit a packet in the data slot.
```

Chapter 3

Cooperative Network Model

3.1 Problem Definition

Cooperative communication helps overcome fading and attenuation in wireless networks. Its main purpose is to increase the communication rates across the network and to increase reliability of time-varying links. Due to the broadcasting nature of wireless medium, nodes can overhear each others messages and consequently can benefit from this characteristic. This message overhearing by other nodes, enables cooperation. We consider a cooperative network as in Figure 3.1, where a Secondary User (SU) acts as a relay for a Primary User (PU). PU is the owner of channel and SU can access the channel conditioned on cooperating with PU by relaying its packets. We assume that PU channel to destination suffers from fading while SU has a reliable channel. As we will show, this cooperation will result in a better performance for PU.

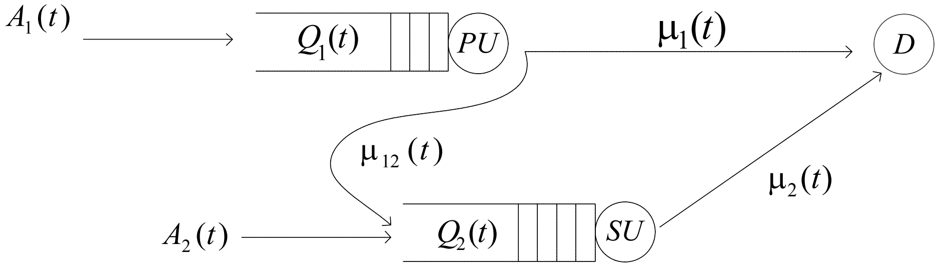


FIGURE 3.1: System Model.

3.2 System Model

We consider a time slotted system where a primary user and a secondary user are uploading their packets to a common destination as in Figure 3.1. Primary user, secondary user and destination are denoted by PU, SU and D respectively. Links have unit capacity and only one packet can be sent in any slot whenever possible.

The packet arrival process at PU and SU are denoted by $A_1(t)$ and $A_2(t)$ respectively. Arrival process is i.i.d. over slots with Poisson distribution. The arrival rates are given by $\lambda_1 = \mathbb{E}\left\{A_1(t)\right\}$ and $\lambda_2 = \mathbb{E}\left\{A_2(t)\right\}$. It is assumed that $A_i(t) \leq A_{max}$ for all i and t. The second moments $\mathbb{E}\left\{A_1(t)^2\right\}$ and $\mathbb{E}\left\{A_2(t)^2\right\}$ are assumed to be finite. To capture the effect of fading on the link $PU - D$, we denote ρ_1 as the probability that a packet transmitted by PU is successfully decoded at D. We denote channel state process on link $PU - D$ by $s_1(t)$. We assume channel state process are $i.i.d$ over slots and takes values on $\{0,\ 1\}$ with $Pr\left(s_1(t) = 1\right) = \rho_1$. We assume that, only PU's channel (i.e., link between PU and D) suffers from fading. At each time slot t, three actions are defined for the network. The scheduler based on the optimal policy which will be described later, schedules a packet from PU to D (denoted by $\mu_1(t)$), or from PU to SU (denoted by $\mu_{12}(t)$) to be relayed later, or from SU to D (denoted by denoted by $\mu_2(t)$). Only one packet can be transmitted at any time slot t. A packet from PU can be transmitted to SU only if the channel is OFF (i.e., $s_1(t) = 0$).

3.3 Centralized Algorithm

This section describes the centralized algorithm to maximize the capacity region. We assume that there is a centralized scheduler observing the network and scheduling the actions. The scheduler has the all necessary information to take decisions. The ultimate goal of the scheduler is to optimize $\mu_1(t)$, $\mu_{12}(t)$ and $\mu_2(t)$ such that the network capacity region is maximized. The cooperative network capacity region Λ^c for our model is the closure of the set of all arrival rate vectors $(\lambda_1,\ \lambda_2)$ that can be stably supported by the network, considering all possible strategies for choosing scheduling variables, $\mu_1(t)$, $\mu_{12}(t)$ and $\mu_2(t)$.

It is known that Maximum Weight Scheduling (MWS) [8, 46, 64] can stabilize the network whenever the arrival vector $(\lambda_1,\ \lambda_2)$ lies strictly inside the network capacity region (i.e., $(\lambda_1,\ \lambda_2) \in \Lambda^c$). In other words, MWS is throughput optimal. Next we will describe how to implement MWS to our network model and specify the constraints.

Let $\boldsymbol{\mu}(t) = (\mu_1(t), \mu_{12}(t), \mu_2(t))$ be the transmission decision on slot t taking only integer values $\{0, 1\}^3$ with the following constraint:

$$\mu_1(t) + \mu_{12}(t) + \mu_2(t) \leq 1 \tag{3.1}$$

Constraint 3.1 captures our interference model, where only one node can transmit at any time slot t. Simultaneous transmission of nodes will result in collision. And also *PU* transmits, only to D or *SU* and not both of them at the same time slot. The queuing dynamics are given by:

$$Q_1(t + 1) = \max\left[Q_1(t) - \mu_1(t) - \mu_{12}(t), 0\right] + A_1(t) \tag{3.2}$$

$$Q_2(t + 1) = \max\left[Q_2(t) - \mu_2(t), 0\right] + A_2(t) + \mu_{12}(t) \tag{3.3}$$

Following Lyapunov drift theorem it can be shown that [64] MWS algorithm achieves optimal throughput by opportunistically maximizing the following optimization problem at each time slot t.

$$\begin{aligned} \max_{\boldsymbol{\mu}(t)} \quad & \omega_1(t)\left(\mu_1(t) + \mu_{12}(t)\right) + \omega_2(t)\mu_2(t) \\ \text{subject to} \quad & \mu_1(t) + \mu_{12}(t) + \mu_2(t) \leq 1 \end{aligned} \tag{3.4}$$

where, $\omega_1(t)$ and $\omega_2(t)$ are the weights associated with *PU* and *SU* respectively, at time slot t as follow:

$$\begin{aligned} \omega_1(t) =& Q_1(t)s_1(t) + (1 - s_1(t))\left(Q_1(t) - Q_2(t)\right)^+ \\ \omega_2(t) =& Q_2(t) \end{aligned} \tag{3.5}$$

where, $(x)^+ = \max\{x, 0\}$.

For non-cooperative network ($\mu_{12}(t) = 0$ for all t) the network capacity Λ^{nc}, can be written as follow [46]:

$$\Lambda^{nc} = \{(\lambda_1,\ \lambda_2)\,|\lambda_1 < \rho_1,\ \lambda_2 < 1,\ \lambda_1 + \lambda_2 < 1\} \tag{3.6}$$

It can be seen from Λ^{nc} that, in a non-cooperative network, SU can limit achievable rate of *PU*. Also maximum supportable rate by PU is ρ_1 at most, regardless of presence or absence of *SU*. As mentioned above, *PU* is the licensed user of the channel and *SU* cannot operate on the channel without *PU* permission. In the following theorem we will prove capacity region of the cooperative network Λ^c and state motivation of *PU* in permitting access to *SU*.

Theorem 1. The cooperative network capacity region Λ^c is as follows:

$$\begin{aligned} \Lambda^c = \{\, \boldsymbol{\lambda} | \lambda_2 < 1 - \rho_1,\ 2\lambda_1 + \lambda_2 < 1 + \rho_1, \\ 1 - \rho_1 \leq \lambda_2 < \rho_2,\ \lambda_1 + \lambda_2 < 1 \end{aligned} \tag{3.7}$$

Proof. We prove the first segment of the capacity region, $\{\lambda_2 < 1 - \rho_1,\ 2\lambda_1 + \lambda_2 < 1 + \rho_1\}$. The second segment (i.e., $\lambda_1 < \rho_1$) is the same as the network with no cooperation as defined in 3.6. We denote $\mu_1^t := \mu_1(t) + \mu_{12}(t)$

1. $Q_1(t) < Q_2(t)$

$$\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} = 0 \tag{3.8}$$
$$\mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = 1 \tag{3.9}$$
$$\mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} = 0 \tag{3.10}$$

2. $Q_2(t) \leq Q_1(t) \leq 2Q_2$

$$\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} = \rho_1 \tag{3.11}$$
$$\mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = 1 - \rho_1 \tag{3.12}$$
$$\mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} = 0 \tag{3.13}$$

3. $Q_1(t) > 2Q_2(t)$

$$\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} = 1 \tag{3.14}$$
$$\mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = 0 \tag{3.15}$$
$$\mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} = 1 - \rho_1 \tag{3.16}$$

As we are concentrating on $\lambda_2 < 1-p, 2\lambda_1+\lambda_2 < 1+\rho_1$, it can be seen from service rates that whenever, $Q_1(t) > 2Q_2(t)$, PU gets an exceeding amount of service rate while SU gets non. So Q_2 starts to grow while Q_1 decrease in size. Consequently Network makes a transition from $Q_1(t) > 2Q_2(t)$ to $Q_2(t) \leq Q_1(t) \leq 2Q_2$. In this state PU gets a mean service rate less than its arrival rate and starts to grow while Q_2 starts to decrease so the network returns to $Q_1(t) > 2Q_2(t)$. So at any slot t, $Pr\left(Q_2(t) < Q_1(t) \leq 2Q_2(t)\right) + Pr\left(Q_1(t) > 2Q_2(t)\right) = 1$. We define the Lyapunov function as $q(t) = 2Q_1(t) + Q_2(t)$ and show that expected drift has a negative value. For small positive value of ϵ we have $2\lambda_1 + \lambda_2 + \epsilon = 1 + \rho_1$.

- $Q_2(t) < Q_1(t) \leq 2Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =&2\lambda_1 - 2\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} - \\ &\mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = 2\lambda_1 + \lambda_2 - \rho_1 - 1 = -2\epsilon < 0 \end{aligned} \tag{3.17}$$

- $Q_1(t) > 2Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =&2\lambda_1 - 2\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} \\ &- \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = 2\lambda_1 + \lambda_2 - \rho_1 - 1 = -2\epsilon < 0 \end{aligned} \tag{3.18}$$

Using total probability law we have:

$$\begin{aligned}&\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} \\ =&\mathbb{E}\left\{q(t+1)-q(t)|Q_2(t) < Q_1(t) \leq 2Q_2(t)\right\} Pr\left(Q_2(t) < Q_1(t) \leq 2Q_2(t)\right) \\ &+ \mathbb{E}\left\{q(t+1)-q(t)|Q_1(t) > 2Q_2(t)\right\} Pr\left(Q_1(t) > 2Q_2(t)\right) = -2\epsilon < 0 \end{aligned} \tag{3.19}$$

□

The cooperative network capacity region is depicted in Figure 3.2. It can be seen that in the cooperative network, PU's maximum supportable rate is $\frac{1+\rho_1}{2}$ which strictly greater than ρ_1, (i.e., PU's maximum supportable rate when there is no cooperation whenever $\rho_1 < 1$.

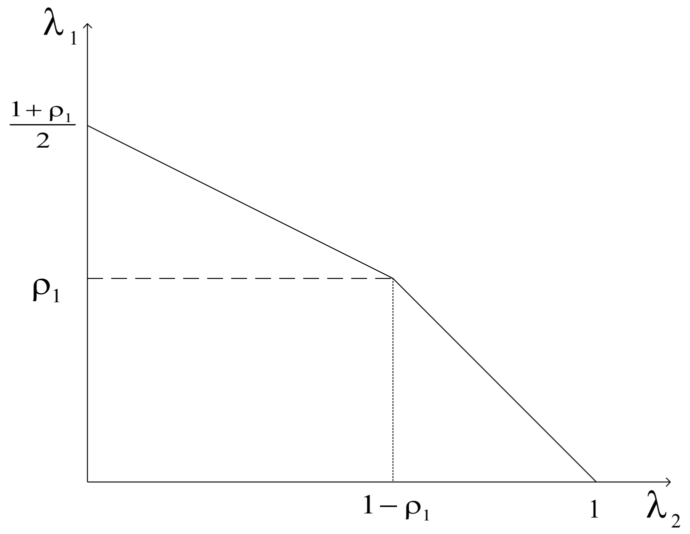


FIGURE 3.2: Cooperative capacity region Λ^c.

3.4 Extension to N users

Consider the same problem with N users consisting of a PU and $N-1$ SUs as depicted in Figure 3.3. All SU's acts as a relay for PU. The packet arrival process at users are denoted by $A_i(t)$ (for $i \in \{1, \cdots, N\}$) with $A_1(t)$ being arrival process of PU. Arrival process is i.i.d. over slots with Poisson distribution. The arrival rates are given by $\lambda_i = \mathbb{E}\{A_i(t)\}$. It is assumed that $A_i(t) \leq A_{max}$ for all i and t. The second moments $\mathbb{E}\left\{A_i(t)^2\right\}$ are assumed to be finite. Similarly, to capture the effect of fading on the link $PU - D$, we denote ρ_1 as the probability that a packet transmitted by PU is successfully decoded at D. We denote channel state process on link $PU - D$ by $s_1(t)$. We assume channel state process are $i.i.d$ over slots and takes values on $\{0,\ 1\}$ with $Pr\left(s_1(t) = 1\right) = \rho_1$.

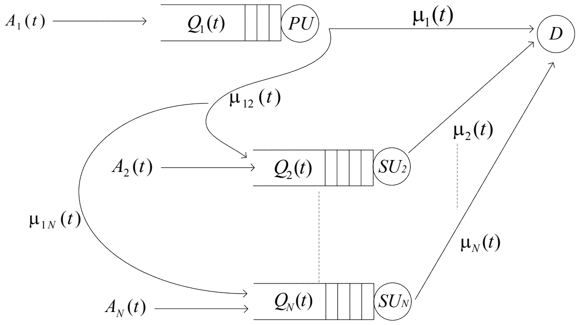


FIGURE 3.3: System model for N users

Next we describe MWS with extension to N users. Let $\boldsymbol{\mu}(t) = (\mu_i(t), \mu_{1j}(t))$ for all $i \in \{1, \cdots, N\}$ and $j \in \{2, \cdots, N\}$ be the transmission decision on slot t. Where, $\mu_i(t)$denotes decision on direct transmission of user i to D at time slot t and $\mu_{1j}(t)$ denotes decision on transmitting a packet from PU to SU_j.

$\boldsymbol{\mu}(t)$ takes only integer values $\{0,1\}^{2N-1}$ with the following constraint:

$$\sum_{i=1}^{N} \mu_i(t) + \sum_{i=2}^{N} \mu_{1i}(t) \leq 1 \tag{3.20}$$

Constraint 3.20 captures our interference model, where only one node can transmit at any time slot t. Simultaneous transmission of nodes will result in collision. Also it implies that PU can transmit a packet to only D or one of the SUs at any time slot.

The queuing dynamics are given by:

$$Q_1(t+1) = \max\left[Q_1(t) - \mu_1(t) - \sum_{i=2}^{N} \mu_{1i}(t), 0\right] + A_1(t) \tag{3.21}$$

$$Q_i(t+1) = \max\left[Q_i(t) - \mu_i(t), 0\right] + A_2(t) + \mu_{1i}(t) \;\; i \in \{2, \cdots, N\} \tag{3.22}$$

Following Lyapunov drift theorem it can be shown that [64] MWS algorithm achieves optimal throughput by opportunistically maximizing the following optimization problem at each time slot t.

$$\begin{aligned} \max_{\boldsymbol{\mu}(t)} \quad & \omega_1(t)\left(\mu_1(t) + \sum_{i=2}^{N} \mu_{1i}(t)\right) + \sum_{i=2}^{N} \omega_i(t)\mu_i(t) \\ \text{subject to} \quad & \sum_{i=1}^{N} \mu_i(t) + \sum_{i=2}^{N} \mu_{1i}(t) \le 1 \end{aligned} \tag{3.23}$$

where, $\omega_i(t)$ is the weight associated with user i, at time slot t as follow:

$$\begin{aligned} \omega_1(t) =& Q_1(t)s_1(t) + (1 - s_1(t))\left[\max_i \left(Q_1(t) - Q_i(t)\right)\right]^+ \\ \omega_i(t) =& Q_i(t) \end{aligned} \tag{3.24}$$

where, $(x)^+ = \max\{x, 0\}$. Note that, if the scheduler decides on relaying a packet, the packet only goes to SU_i with $i = \arg\max_j (Q_1(t) - Q_j(t))$. The exact capacity region of the MWS when the number of users exceed two in fading channels is unknown. But MWS as a optimal scheduler can stabilize the network whenever arrival rate vector $\boldsymbol{\lambda}$ lies strictly inside the network capacity region.

3.5 Extension to multiple fading channels (when, N=2)

Consider the same cooperative model as in Figure 3.1 with only difference in channel modeling. In this section we assume all channels $PU - D$, $PU - SU$ and $SU - D$ denoted by $s_1(t)$, $s_{12}(t)$ and $s_2(t)$ respectively, suffer from fading. We capture fading effect by assuming ON, OFF channels with $Pr\,(s_1(t) = 1) = \rho_1$, $Pr\,(s_{12}(t) = 1) = \rho_{12}$ and $Pr\,(s_2(t) = 1) = \rho_2$. Channels state vector is i.i.d. over slots and is denoted by $\mathbf{s}(t) = (s_1(t),\ s_{12}(t),\ s_2(t))$.

3.5.1 MWS for multiple fading channels (with N=2)

Let $\boldsymbol{\mu}(t) = (\mu_1(t),\ \mu_{12}(t),\ \mu_2(t))$ be the transmission decision on slot t. Where, $\mu_1(t)$ denotes decision on direct transmission of PU to D at time slot t, $\mu_{12}(t)$ denotes decision on transmitting a packet from PU to SU and $\mu_2(t)$ is the decision on transmission a packet from SU to D

$\boldsymbol{\mu}(t)$ takes only integer values $\{0, 1\}^3$ with the following constraint:

$$\mu_1(t) + \mu_{12}(t) + \mu_2(t) \leq 1 \tag{3.25}$$

Constraint 3.25 captures our interference model, where only one node can transmit at any time slot t. Simultaneous transmission of nodes will result in collision. It also implies that at any time slot, PU can only transmit to D or SU. The queuing dynamics are given by:

$$Q_1(t+1) = \max\left[Q_1(t) - \mu_1(t) - \mu_{12}(t), 0\right] + A_1(t) \tag{3.26}$$

$$Q_i(t+1) = \max\left[Q_i(t) - \mu_2(t), 0\right] + A_2(t) + \mu_{12}(t) \tag{3.27}$$

Following Lyapunov drift theorem it can be shown that [64] MWS algorithm achieves optimal throughput by opportunistically maximizing the following optimization problem at each time slot t.

$$\begin{aligned} \max_{\boldsymbol{\mu}(t)} \quad & \omega_1(t)\left(\mu_1(t) + \mu_{12}(t)\right) + \omega_2(t)\mu_2(t) \\ \text{subject to} \quad & \mu_1(t) + \mu_{12}(t) + \mu_2(t) + \mu_2 \leq 1 \end{aligned} \tag{3.28}$$

where, $\omega_i(t)$ is the weight associated with user i, at time slot t as follow:

$$\begin{aligned} \omega_1(t) =& Q_1(t)s_1(t) + (1 - s_1(t))s_{12}(t)\left(Q_1(t) - Q_2(t)\right)^+ \\ \omega_2(t) =& Q_2(t)s_2(t) \end{aligned} \tag{3.29}$$

where, $(x)^+ = \max\{x, 0\}$.

3.5.2 Capacity region for multiple fading channels (when, N=2)

In this section, by calculating exact capacity region of our cooperative network with multiple fading channels, we will show that even in the case of multiple fading channels cooperation is possible. We start our analysis by computing the expected service rates for each user as follow:

$$
\begin{aligned}
&\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} = \mathbb{E}\left\{\mu_1(t)|\mathbf{Q}(t)\right\} + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} \\
&= \rho_1(1-\rho_2) + \rho_1\rho_2 \mathbb{1}_{\{Q_1(t) \geq (Q_2(t)\}} + (1-\rho_1)\rho_{12}(1-\rho_2)\mathbb{1}_{\{Q_1(t)-Q_2(t)>0\}} \\
&+ (1-\rho_1)\rho_{12}\rho_2 \mathbb{1}_{((Q_1(t)-Q_2(t))^+ > Q_2(t))} && (3.30) \\
&\mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} = (1-\rho_1)(1-\rho_{12})\rho_2 + (1-\rho_1)\rho_{12}\rho_2 \mathbb{1}_{\{(Q_1(t)-Q_2(t))^+ \leq Q_2(t)\}} \\
&+ \rho_1\rho_2 \mathbb{1}_{\{Q_1(t) < Q_2(t)\}} && (3.31) \\
&\mathbb{E}\left\{A_2^t(t)|\mathbf{Q}(t)\right\} = A_2(t) + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} \\
&= A_2(t) + (1-\rho_1)\rho_{12}(1-\rho_2)\mathbb{1}_{\{Q_1(t)-Q_{12}(t)>0\}} + (1-\rho_1)\rho_{12}\rho_2 \mathbb{1}_{\{(Q_1(t)-Q_2(t))^+ > Q_2(t)\}} \\
& && (3.32)
\end{aligned}
$$

where, expectation is taken with respect to channel state vector $\mathbf{s}(t)$; $\mu_1^t(t)$ and $A_2^t(t)$ is the total service rate of PU and total arrival of SU, respectively. Note that expected service rates exactly follow from MWS algorithm.

Following theorems establish the capacity region associated with our network model.

Theorem 2. For $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$, the optimal capacity region is

$$\Lambda^c = \{\boldsymbol{\lambda}|\lambda_2 < \rho_2, \lambda_1 + \lambda_2 < \rho_1 + \rho_2(1-\rho_1)\} \tag{3.33}$$

Proof. We will focus on the improved section of capacity region over a non-cooperative model. It is well known that in a similar network without cooperation the capacity region is $\Lambda^{nc} = \{\boldsymbol{\lambda}|\lambda_1 < \rho_1, \lambda_2 < \rho_2, \lambda_1 + \lambda_2 < \rho_1 + \rho_2(1-\rho_1)\}$. So we will provide the proof for $\rho_1 \leq \lambda_1 < \rho_1 + (1-\rho_1)\rho_2$. when $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$, for $\rho_1 \leq \lambda_1 < \rho_1 + (1-\rho_1)\rho_2$ and any possible value of λ_2 that satisfies (3.33), states where $Q_1(t) > 2Q_2(t)$, will be transient and not likely to happen. The reason is that in this state $\lambda_1 < \mathbb{E}\left\{\mu_1^t(t)|Q_1(t) > 2Q_2(t)\right\}$ and $\lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|Q_1(t) > 2Q_2(t)\right\} > \mathbb{E}\left\{\mu_2(t)|Q_1(t) > 2Q_2(t)\right\}$, meaning that Q_1 tends to decrease in size, while Q_2 increases in size. So given that the network is in state $Q_1(t) > 2Q_2(t)$, it will have a transition to $Q_2(t) < Q_1(t) \leq 2Q_2$ and with a similar analysis, in state $Q_2(t) < Q_1(t) \leq 2Q_2$, Q_1 tends to decrease in size, while Q_2 increases in size so the system will have a transition to $Q_1(t) \leq Q_2(t)$. The story there is different and SU has a better service rate. Briefly the system will have a transition from $Q_1(t) \leq Q_2(t)$ to $Q_2(t) < Q_1(t) \leq 2Q_2$ and the reverse transition will happen. We define $q(t) = Q_1(t) + Q_2(t)$ and show that for the defined range of (λ_1, λ_2) the expected Lyapunov drift is strictly negative in every possible case leading to the strong stability of the network. For small positive value of ϵ we have $\lambda_1 + \lambda_2 + \epsilon = \rho_1 + (1-\rho_1)\rho_2$

- $Q_1(t) < Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =& \lambda_1 - \mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 - \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\}\\ =& \lambda_1+\lambda_2-\rho_1(1-\rho_2)-\rho_2 = -\epsilon < 0 \end{aligned} \tag{3.34}$$

- $Q_1(t) = Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =& \lambda_1 - \mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 - \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\}\\ =& \lambda_1+\lambda_2-\rho_1-(1-\rho_1)\rho_2 = -\epsilon < 0 \end{aligned} \tag{3.35}$$

- $Q_2(t) < Q_1(t) \leq 2Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =& \lambda_1 - \mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\}\\ &- \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\}\\ =& \lambda_1+\lambda_2+(1-\rho_1)\rho_{12}(1-\rho_2)-\rho_1\\ &-(1-\rho_1)\rho_{12}(1-\rho_2)-\rho_2(1-\rho_1) = -\epsilon < 0 \end{aligned} \tag{3.36}$$

Using total probability law we have:

$$\begin{aligned}&\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\}\\ &= \mathbb{E}\left\{q(t+1)-q(t)|Q_1(t)<Q_2(t)\right\} Pr\left(Q_1(t)<Q_2(t)\right)\\ &+ \mathbb{E}\left\{q(t+1)-q(t)|Q_1(t)=Q_2(t)\right\} Pr\left(Q_1(t)=Q_2(t)\right)\\ &+ \mathbb{E}\left\{q(t+1)-q(t)|Q_2(t)<Q_1(t)\leq 2Q_2(t)\right\} Pr\left(Q_2(t)<Q_1(t)\leq 2Q_2(t)\right) = -\epsilon < 0 \end{aligned} \tag{3.37}$$

□

Capacity region of the network for $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$ can be seen in Figure 3.4.

Theorem 3. For $\frac{\rho_{12}}{1+\rho_{12}} < \rho_2 \leq \frac{\rho_{12}}{1-\rho_{12}}$, the optimal capacity region is

$$\begin{aligned}\Lambda^c = \{\, \boldsymbol{\lambda}|\lambda_2 < (1-\rho_1)\left[\rho_2-\rho_{12}(1-\rho_2)\right], 2\lambda_1+\lambda_2 < 2\rho_1+(1-\rho_1)\left[\rho_{12}(1-\rho_2)+\rho_2\right],\\ (1-\rho_1)\left[\rho_2-\rho_{12}(1-\rho_2)\right] \leq \lambda_2 < \rho_2, \lambda_1+\lambda_2 < \rho_1+(1-\rho_1)\rho_2 \,\}\end{aligned} \tag{3.38}$$

Proof. For $(1-\rho_1)\left[\rho_2-\rho_{12}(1-\rho_2)\right] \leq \lambda_2 < \rho_2$ the analysis is the same as in Theorem 2. and states where $Q_1 > 2Q_2(t)$ will be transient and unlikely to happen. And it can be shown that $q(t) = Q_1(t)+Q_2(t)$ has a negative drift exactly as in Theorem 2. For $\lambda_2 < (1-\rho_1)\left[\rho_2-\rho_{12}(1-\rho_2)\right]$ and λ_1 close to the boundary defined in

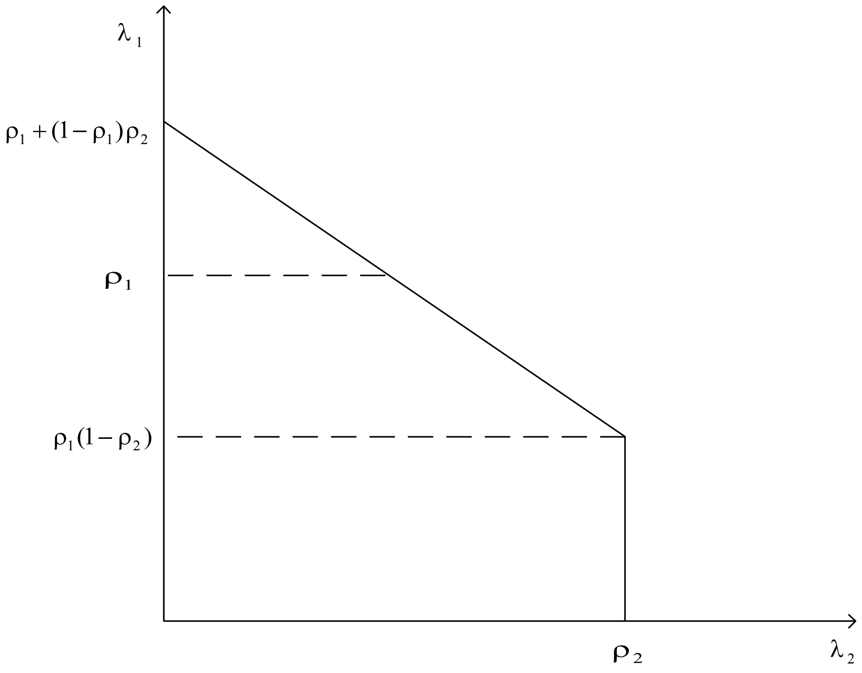


FIGURE 3.4: Cooperative capacity region for $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$.

(3.38), given that the network is operating in the states of $Q_2(t) < Q_1(t) \leq 2Q_2(t)$, it can be seen that $\lambda_1 > \mathbb{E}\left\{\mu_1^t(t)|Q_2(t) < Q_1(t) \leq 2Q_2(t)\right\}$ and $\lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} < \mathbb{E}\left\{\mu_2(t)|Q_2(t) < Q_1(t) \leq 2Q_2(t)\right\}$ which results in increment of Q_1 and decrement of Q_2. Note that, it is unlikely for the network to have a transition into $Q_1(t) \leq Q_2(t)$. Then the network will have transition into $Q_1(t) > 2Q_2(t)$ where $\lambda_1 < \mathbb{E}\left\{\mu_1^t(t)|Q_1(t) > 2Q_2(t)\right\}$ and $\lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|Q_1(t) > 2Q_2(t)\right\} > \mathbb{E}\left\{\mu_2(t)|Q_1(t) > 2Q_2(t)\right\}$ which results in decrement of Q_1 and increment of Q_2. Then the network will have a transition back to $Q_2(t) < Q_1(t) \leq 2Q_2(t)$. We define the Lyapunov function as $q(t) = 2Q_1(t) + Q_2(t)$ and show that expected drift has a negative value. For small positive value of ϵ we have $2\lambda_1 + \lambda_2 + \epsilon = 2\rho_1 + (1-\rho_1)\left[\rho_{12}(1-\rho_2) + \rho_2\right]$

- $Q_2(t) < Q_1(t) \leq 2Q_2(t)$

$$\begin{aligned}\mathbb{E}\left\{q(t+1) - q(t)|\mathbf{Q}(t)\right\} =& 2\lambda_1 - 2\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} \\ &- \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} \\ &= 2\lambda_1 + \lambda_2 + (1-\rho_1)\rho_{12}(1-\rho_2) - 2\rho_1 \\ &- 2(1-\rho_1)\rho_{12}(1-\rho_2) - \rho_2(1-\rho_1) = -2\epsilon < 0 \quad (3.39)\end{aligned}$$

- $Q_1(t) > 2Q_2(t)$

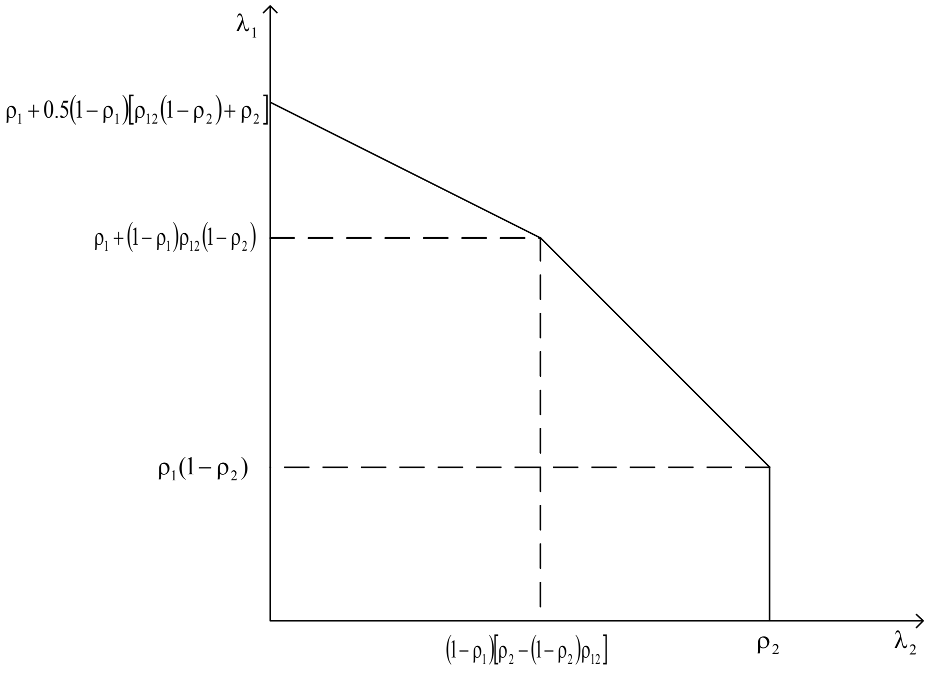
$$\begin{aligned}\mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} =& 2\lambda_1 - 2\mathbb{E}\left\{\mu_1^t(t)|\mathbf{Q}(t)\right\} + \lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|\mathbf{Q}(t)\right\} \\ & - \mathbb{E}\left\{\mu_2(t)|\mathbf{Q}(t)\right\} \\ & = 2\lambda_1 + \lambda_2 + (1-\rho_1)\rho_{12} - 2\rho_1 \\ & - 2(1-\rho_1)\rho_{12} - (1-\rho_1)(1-\rho_{12})\rho_2 = -2\epsilon < 0 \end{aligned} \tag{3.40}$$

Using total probability law we have:

$$\begin{aligned} & \mathbb{E}\left\{q(t+1)-q(t)|\mathbf{Q}(t)\right\} \\ & = \mathbb{E}\left\{q(t+1)-q(t)|Q_2(t) < Q_1(t) \le 2Q_2(t)\right\} Pr\left(Q_2(t) < Q_1(t) \le 2Q_2(t)\right) \\ & + \mathbb{E}\left\{q(t+1)-q(t)|Q_1(t) > 2Q_2(t)\right\} Pr\left(Q_1(t) > 2Q_2(t)\right) = -2\epsilon < 0 \end{aligned} \tag{3.41}$$

□

Capacity region of the network for $\frac{\rho_{12}}{1+\rho_{12}} < \rho_2 \le \frac{\rho_{12}}{1-\rho_{12}}$ can be seen in Figure 3.5.



λ_1
$\rho_1 + 0.5(1-\rho_1)[\rho_{12}(1-\rho_2)+\rho_2]$
$\rho_1 + (1-\rho_1)\rho_{12}(1-\rho_2)$
$\rho_1(1-\rho_2)$
$(1-\rho_1)[\rho_2-(1-\rho_2)\rho_{12}]$
ρ_2
λ_2

FIGURE 3.5: Cooperative Capacity region for $\frac{\rho_{12}}{1+\rho_{12}} < \rho_2 \le \frac{\rho_{12}}{1-\rho_{12}}$.

Theorem 4. For $\rho_2 > \frac{\rho_{12}}{1-\rho_{12}}$, the optimal capacity region is

$$\begin{aligned}\Lambda^c = \{ \boldsymbol{\lambda} | &\lambda_1 < \rho_1 + (1-\rho_1)\rho_{12}, \lambda_2 < (1-\rho_1)\left[\rho_2(1-\rho_{12}) - \rho_2\right];\\ &2\lambda_1 + \lambda_2 < 2\rho_1 + (1-\rho_1)\left[\rho_{12}(1-\rho_2) + \rho_2\right],\\ &(1-\rho_1)\left[\rho_2(1-\rho_{12}) - \rho_2\right] \leq \lambda_2 < (1-\rho_1)\left[\rho_2 - \rho_{12}(1-\rho_2)\right];\\ &\lambda_1 + \lambda_2 < \rho_1 + (1-\rho_1)\rho_2, (1-\rho_1)\left[\rho_2 - \rho_{12}(1-\rho_2)\right] \leq \lambda_2 < \rho_2 \ \} \end{aligned} \tag{3.42}$$

Proof. For $\lambda_2 < (1-\rho_1)\left[\rho_2(1-\rho_{12}) - \rho_{12}\right]$, we have $\lambda_2 + \mathbb{E}\left\{\mu_{12}(t)|Q_1(t) > 2Q_2(t)\right\} < \mathbb{E}\left\{\mu_2(t)|Q_1(t) > 2Q_2(t)\right\}$ which means that the Q_2, independent of λ_1 has a negative drift and if we choose $\lambda_1 + \epsilon = \rho_1 + (1-\rho_1)\rho_{12}$ for very small positive values of ϵ, the network will stay in $Q_1(t) > 2Q_2(t)$ states and also both queues have expected negative drift leading to stability of the network. The proof for $\lambda_2 \geq (1-p_1)\left[\rho_2(1-\rho_{12}) - \rho_2\right]$ follows exactly as in Theorem 2 and 3.

□

Capacity region of the network for $\rho_2 > \frac{\rho_{12}}{1-\rho_{12}}$ can be seen in Figure 3.6.

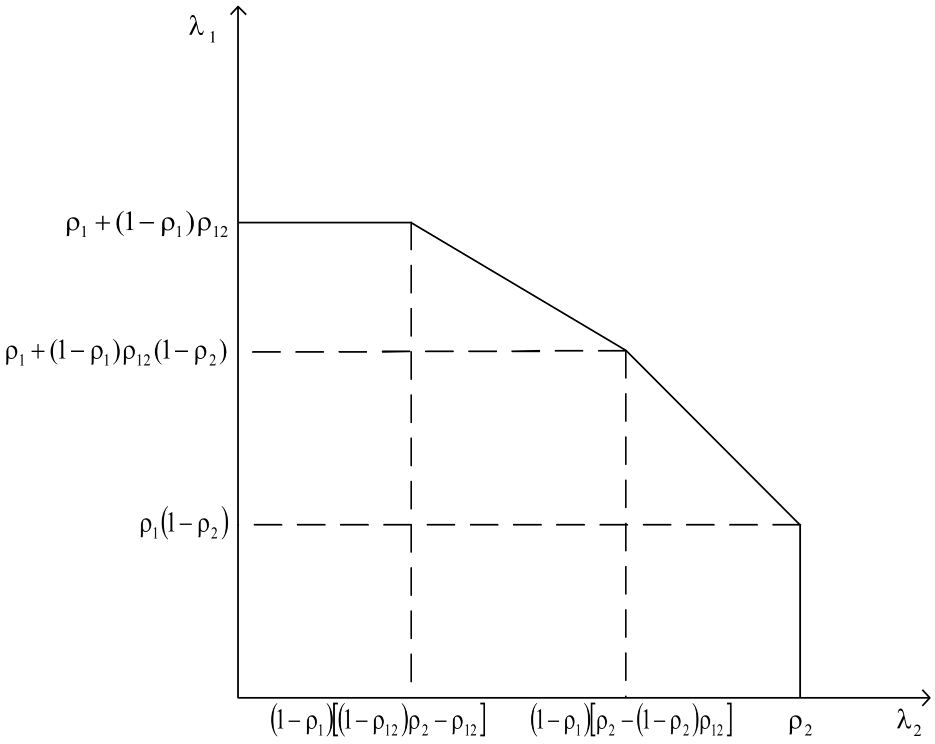


Figure 3.6: Cooperative capacity region for $\rho_2 > \frac{\rho_{12}}{1-\rho_{12}}$.

Chapter 4

Distributed Algorithm

In chapter 3 we described Maximum Weight Scheduling (MWS) for our cooperative network model. MWS algorithm is throughput optimal in the sense that, it can support all arrival vectors in the network capacity region. Implementing MWS has two main shortcomings. The first is high computational complexity in solving the optimization problem and the second one is the cost associated with the collection of network state information at a central location. Collecting the necessary information, introduces overhead which degrades efficiency of a time slot and compromises the throughput [11]. Recently, a class of schedulers based on CSMA is proposed to have optimal throughput properties [32, 33]. These algorithms only use information available at an individual link and by doing so, they circumvent problems associated with MWS. These algorithms use queue lengths to determine channel access probabilities, achieving the full capacity region in ad hoc wireless networks in a distributed manner. However, throughput optimality of these algorithms, are limited to non-fading wireless links. There is no guarantee on the performance of these algorithms, when the links suffer from fading. A simulation based study in [34, 35] suggests that, these algorithms (e.g., Q-CSMA [33]) are not throughput optimal.

Q-CSMA, fails to utilize opportunistic gain in fading channels. Recently, in [66] a modified version of Q-CSMA has been proven to be throughput optimal in a cognitive set up. Consider a cognitive network where a primary user and N secondary users are operating on a single channel. Secondary users are not allowed to be active whenever primary user is transmitting. Primary user's activity is modeled as a ON-OFF process. The work of [66], modifies Q-CSMA to account for variability of primary user activity. Primary user does not participate in the algorithm. Secondary users involved in the algorithm, have non-fading channels. In this work our aim is to design an algorithm in a fading environment where primary user is a part of the algorithm. Also, our work addresses the benefits of cooperation for primary user while the set up in [66] is merely cognitive and they aim to maximize capacity region of secodary users without interfering with primary user. We break non interfering operation of SUs by allowing secondary users in the same algorithm as primary user in order to maximize throughput of all users including primary users.

4.1 System Model for Distributed Algorithm

Here, we modify our system model as in Figure 4.1 to be suited for a distributed algorithm. We assume SU's have a separate queue to store PU packets. This modification does not change network capacity region [4]. Our motive in separating the SU queues is for PU to be able to calculate its weight without message passing. We re wright the weights as follow:

$$\begin{aligned} \omega_1^1(t) =& f(Q_1(t)) \\ \omega_1^0(t) =& f(\max_i (Q_1(t) - Q_{1i}(t))) \\ \omega_i^0(t) =& \omega_i^1(t) = f(\max (Q_i(t), Q_{1i}(t))) \end{aligned} \tag{4.1}$$

where, $i \in \{2, \cdots, N\}$, indicates $N-1$ SU indexes, ω^1 and ω^0 is the weight when the PU direct channel is ON or OFF respectively. Also $f(.)$ satisfies properties defined in Theorem 7.

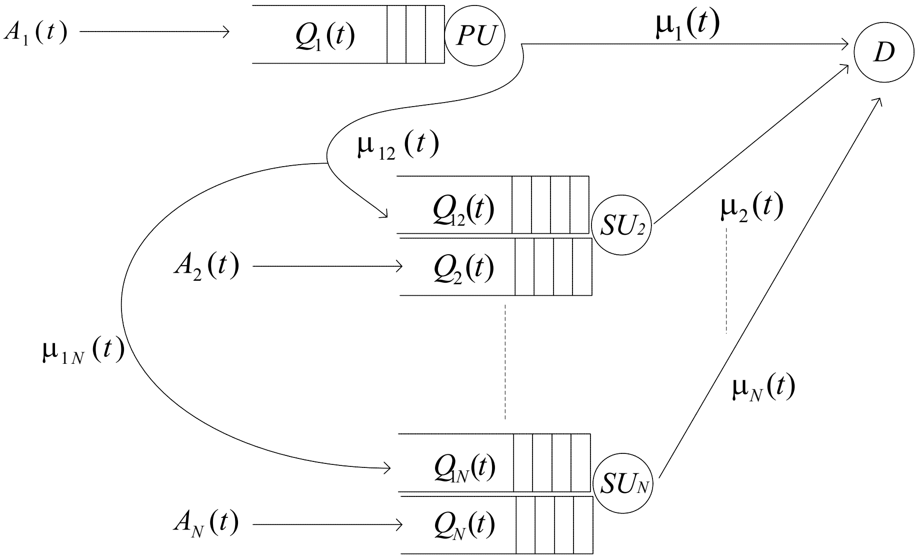


FIGURE 4.1: System Model

The neighborhood of a given user i ($i \in \{1, 2, \cdots, N\}$) which is all other users whose transmission will cause collision on the transmission of user i, is denoted by $\mathcal{C}(i)$. According to our interference model, only one user can transmit at any given time slot t, i.e.e, $\mathcal{C}(i) = \in \{1, 2, \cdots, N\} - \{i\}$.

A feasible schedule is denoted by a vector $\mathbf{x} \in \{0,\ 1\}^N$. If user i is scheduled, i^{th} element of $\mathbf{x}$ is equal to 1 (i.e., $x_i = 1$), otherwise $x_i = 0$. Note that we incorporated PU transmission to D or any other SU into a only single state (i.e., x_1) for the sake of simplicity. According to our model, PU can only transmit to a SU when its channel is OFF and transmits to D otherwise. So, PU upon transmitting, can decide to transmit to the desired destination. Similar to [33], with a little bit abuse of notation, we also treat $\mathbf{x}$ as a set and write $i \in \mathbf{x}$ if $x_i = 1$. According to our interference model, we have:

$$\sum_{i=1}^{N} x_i \leq 1 \tag{4.2}$$

which states that only one user can transmit at any given time slot t. Let $\mathcal{M}$ be the set of all feasible schedules of the network.

Cooperation is performed at the protocol level as follows: whenever a packet is transmitted from PU to SU_i, an ACK from SU_i is sent to PU informing deliverance of the packet and then SU_i, takes over the responsibility of delivering the packet to D by placing it in Q_{1i}. Whenever SU_i transmits a packet to D, the corresponding ACK from D also is sent to PU specifying if the packet was sent from Q_{1i}. Based on the ACKs received from SU's and D, PU can calculate Q_{1i} and consequently $\omega_1^0(t)$. Also, when the channel is OFF, PU given the right to transmit, by knowing all Q_{1i} can decide which SU_i maximizes $\omega_1^0(t)$ and based on that knowledge, transmits the packet to the corresponding SU.

4.2 Q-CSMA Review

As mentioned, throughput optimal CSMA-based distributed scheduling algorithms such as Q-CSMA [33] have been proposed in the recent past. Performance of Q-CSMA is limited to non-fading channels. In the following we will present brief review on Q-CSMA and state why it is not a good choice for our cooperative network model.

A discrete time distributed randomized algorithm is proposed in [33] to achieve the full capacity region in non-fading wireless ad-hoc networks. The algorithm of [33] is based on a generalization of Glauber dynamics in statistical physics. In Glauber dynamics, only one link has a state update within a time slot. In scheduling, a state update can be interpreted as a transition of a link from "transmitting" to "idle" or from "idle" to "transmitting". The incremental state update in every time slot leads to a scheduling policy sufficiently close to MWS, which guarantees the throughput optimality.

A more detailed description of the Q-CSMA algorithm is given in section 2.2.3. A brief review is as follow: Each time slot t is divided into a control slot and a data slot, where the control slot is much smaller than the data slot. In the control slot, a collision-free transmission schedule is generated and used for data transmission in the data slot. Let $\mathbf{m}(t)$ be a set of users that do not conflict with each other and selected randomly in the control slot. $\mathcal{M}_0 \subseteq \mathcal{M}$ denotes the set of all $\mathbf{m}(t)$ which is all possible schedules at slot t and $\mathcal{M}$, is the set of all feasible schedules. The network randomly selects a feasible schedule $\mathbf{m}(t)$, which is called the decision schedule in [33]. $\mathbf{m}(t)$ can be regarded as a candidate schedule. Note that $\mathbf{m}(t)$ and $\mathbf{m}(t-1)$ are independent for all $t > 0$ because $\mathbf{m}(t)$ is chosen independently in the subsequent control slot [33]. Each link within $\mathbf{m}(t)$ will be checked to decide whether it will be included in the transmission schedule $\mathbf{x}(t)$. Link $i \in m(t)$ may be included in $\mathbf{x}(t)$ if $\forall j \in \mathcal{C}(i)$, $j \notin x(t-1)$; otherwise, Link i is not included in $\mathbf{x}(t)$. Links in $\mathbf{m}(t)$ that had no neighbors active in the previous data slot are allowed to update their states with a certain probability which is a function of their queue lengths; those outside the decision schedule $\mathbf{m}(t)$ maintain their states. By explicitly taking into account collisions in the control slot, the algorithm generates collision-free transmission schedules $\mathbf{x}(t)$ for the data slot. More importantly, the Discrete Time Markov Chain (DTMC) with the transmission schedule chosen as the state is shown to be time-reversible and has product-form stationary distribution, which are used to prove throughput optimality of this algorithm.

Next, we illustrate why $\mathbf{x}$ is a poor choice to represent states in our model. Consider the network model when there is only one *PU* and one *SU*. To treat fading of *PU* channel, we further define $\mathbf{X} = (\mathbf{x}; s_1(t))$ to be system state at time slot t. $s_1(t) = 1$ if the channel is *ON* and $s_1(t) = 0$ if the channel is *OFF*. For example, $(0,\ 1; 1)$ indicates that channel is *ON* and *SU* is transmitting; $(1,\ 0; 0)$ means that the channel is *OFF* and *PU* is transmitting to *SU* and $(0,\ 1; 0)$ means that channel is *OFF* and *SU* is transmitting. α_0 is the probability that neither *PU* nor *SU* is selected in the decision schedule, α_1 is the probability that only *PU* is selected in the decision schedule and α_2 is the probability that only *SU* is selected in the decision schedule ($\alpha_0 + \alpha_1 + \alpha_2 = 1$). Note that in Q-CSMA, if user i has a right to transmit, it transmits with probability p_i and does not transmit with probability $1 - p_i$, where p_i is a function of weights defined in (4.1). We denote $p_i^s = \frac{e^{\omega_i^s}}{1+e^{\omega_i^s}}$ as the activation probability of user i when $s_1(t) = s,\ \ s \in \{0,\ 1\}$. Note that $p_1^1 \neq p_1^0$ (because $\omega_1^1 \neq \omega_1^0$). DTMC associated with $\mathbf{X}$ is depicted in Figure 4.2.

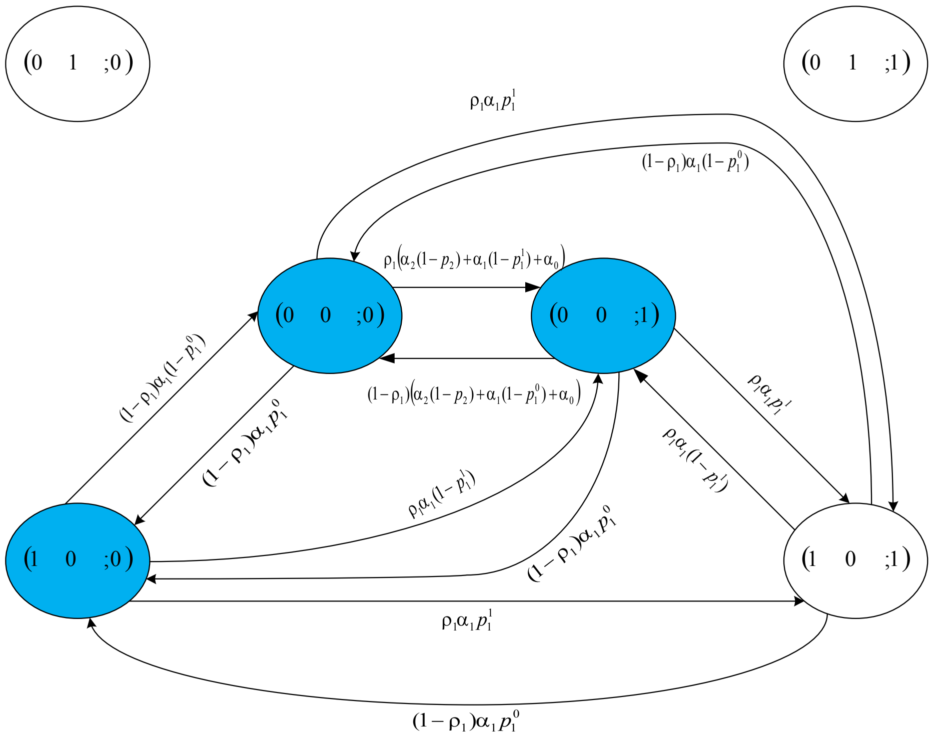
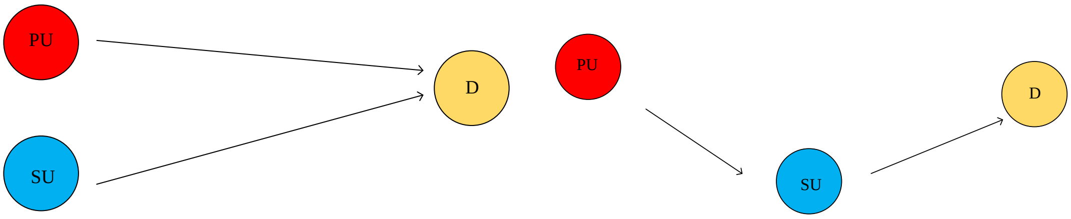

$(0 \quad 1 \quad ;0)$
$(0 \quad 1 \quad ;1)$
$\rho_1\alpha_1 p_1^1$
$(1-\rho_1)\alpha_1(1-p_1^0)$
$\rho_1\left(\alpha_2(1-p_2)+\alpha_1(1-p_1^1)+\alpha_0\right)$
$(0 \quad 0 \quad ;0)$
$(0 \quad 0 \quad ;1)$
$(1-\rho_1)\left(\alpha_2(1-p_2)+\alpha_1(1-p_1^0)+\alpha_0\right)$
$(1-\rho_1)\alpha_1(1-p_1^0)$
$(1-\rho_1)\alpha_1 p_1^0$
$\rho_1\alpha_1 p_1^1$
$\rho_1\alpha_1(1-p_1^1)$
$\rho_1\alpha_1(1-p_1^1)$
$(1-\rho_1)\alpha_1 p_1^0$
$(1 \quad 0 \quad ;0)$
$(1 \quad 0 \quad ;1)$
$\rho_1\alpha_1 p_1^1$
$(1-\rho_1)\alpha_1 p_1^0$

FIGURE 4.2: DTMC associated with $\mathbf{X}$

We now check to see whether the DTMC is time reversible by examining the transitions of highlighted states in Fig.4.2 clockwise and counter-clockwise. The product of clockwise transition probabilities is $\rho_1\alpha_1(1-\rho_1)^2 p_1^0(1-p_1^1)\left[\alpha_2(1-p_2^1)+\alpha_1(1-p_1^0+\alpha_0)\right]$ and the product of counter-clockwise transition probabilities is $\rho_1\alpha_1^2(1-\rho_1)^2 p_1^0(1-p_1^0)\left[\alpha_2(1-p_2^0)+\alpha_1(1-p_1^1+\alpha_0)\right]$. These two are not equal so the DTMC is not time reversible by Kolmogorov's criterion. The proof of throughput optimality for Q-CSMA presented [33] is based on the reversibility of the underlying DTMC. For this reason, direct application of Q-CSMA for our network model cannot be shown to be throughput optimal as in [33]

4.3 Distributed Algorithm

In our model, network topology changes whenever *PU* channel $s_1(t)$ changes its state. For instance, consider the network when there is a *PU* and *SU* as depicted in Figure 4.3. Whenever $s_1(t) = 1$, the network consists of two users transmitting to a common destination *D* as in Figure 4.3I. When $s_1(t) = 0$, network consists of a *PU* whose packets are transmitted through *SU*, while *SU* transmits packets to *D* as in Figure 4.3II. As we showed, direct application of Q-CSMA resulted in a non-reversible DTMC. To cope with this problem we want the DTMC's associated with different network topologies to



(I) Network topology when $s_1(t) = 1$ (II) Network topology when $s_1(t) = 0$

FIGURE 4.3: Different network topologies associated with $s_1(t)$

evolve in different dimensions. For example, when $s_1(t) = 1$, only Figure 4.3I evolves while DTMC associated with $s_1(t) = 0$, 4.3II, stops evolving.

Let's define:

$$\acute{\mathbf{y}}^0(t) = \{\mathbf{x}(\tau) : \text{ the largest } \tau \leq t \text{ with } s_1(t) = 0\} \tag{4.3}$$

$$\acute{\mathbf{y}}^1(t) = \{\mathbf{x}(\tau) : \text{ the largest } \tau \leq t \text{ with } s_1(t) = 1\} \tag{4.4}$$

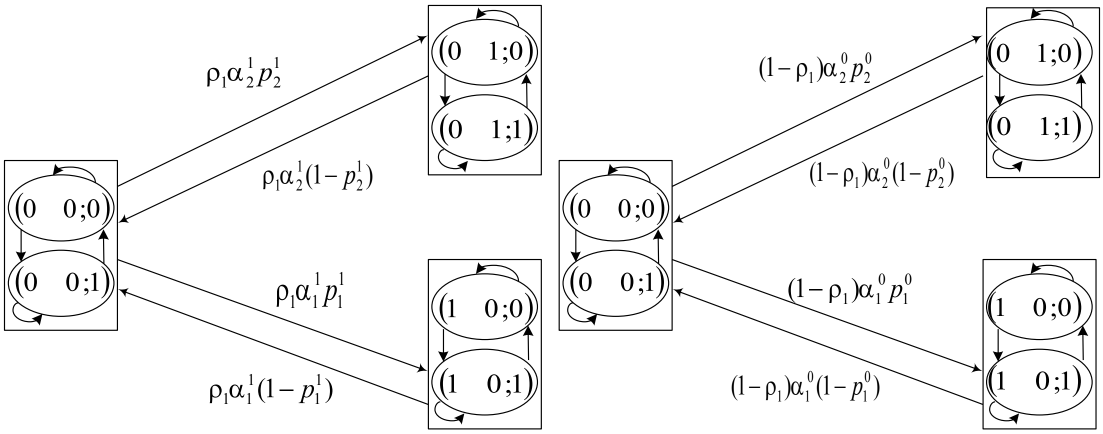
where $\acute{\mathbf{y}}^0(t)$ is the vector of transmission schedule in the most recent data slot (including time t) when the channel is OFF and $\acute{\mathbf{y}}^1(t)$ is the vector of transmission schedule in the most recent data slot when the channel is ON. According to (4.3), $\acute{\mathbf{y}}^0(t)$, associated with Figure 4.3II, is evolving when $s_1(t) = 0$, while $\acute{\mathbf{y}}^1(t)$, associated with Figure 4.3I, stops evolving. And similarly, according to (4.4), $\acute{\mathbf{y}}^1(t)$, associated with Figure 4.3I, is evolving when $s_1(t) = 1$, while $\acute{\mathbf{y}}^0(t)$, associated with Figure 4.3II, stops evolving. With a little bit abuse of notation, we treat $\acute{\mathbf{y}}^0$ ($\acute{\mathbf{y}}^1$) as a set and write $i \in \acute{\mathbf{y}}^0$ ($i \in \acute{\mathbf{y}}^1$) if $\acute{y}_i^0 = 1$ ($\acute{y}_i^1 = 1$). Corresponding transmission vector including the channel state is denoted by $\mathbf{y}^0 = \big(\acute{\mathbf{y}}^0(t); s_1(t)\big)$ and $\mathbf{y}^1 = \big(\acute{\mathbf{y}}^1(t); s_1(t)\big)$ for two chains respectively. For instance, in Figure 4.4, $\mathbf{y}^0 = (1\ 0; 0)$ indicates that the channel is OFF and at time t PU has sent a packet to SU and $\mathbf{y}^0 = (1\ 0; 1)$ means that at current time slot t the channel is ON and PU in the most recent slot when the channel was OFF had relayed a packet to SU.

It can be shown that $\mathbf{y}^0$ and $\mathbf{y}^1$ as the states are not time reversible. $\mathbf{y}^0$ and $\mathbf{y}^1$ are depicted in Figure 4.4II and 4.4I respectively. By Algorithm 2 it is clear that when the channel is OFF, outgoing probabilities from $\big(\acute{\mathbf{y}}^0(t); 0\big)$ and $\big(\acute{\mathbf{y}}^0(t); 1\big)$ to $\Big(\acute{\hat{\mathbf{y}}}^0(t); 0\Big)$ are the same but incoming probabilities from $\Big(\acute{\hat{\mathbf{y}}}^0(t); s_1(t)\Big)$ to $\big(\acute{\mathbf{y}}^0(t); 1\big)$ do not exist if $\acute{\mathbf{y}}^0(t) \neq \acute{\hat{\mathbf{y}}}^0(t)$. The same applies when the channel is ON.

Next let us define $DTMC^0$ and $DTMC^1$ as follow:

$$DTMC^0 \quad \bar{\mathbf{Y}}^0 = (\acute{\mathbf{y}}^0(t)) \tag{4.5}$$

$$DTMC^1 \quad \bar{\mathbf{Y}}^1 = (\acute{\mathbf{y}}^1(t)) \tag{4.6}$$


$\rho_1\alpha_2^1 p_2^1$
$\rho_1\alpha_2^1(1-p_2^1)$
$\rho_1\alpha_1^1 p_1^1$
$\rho_1\alpha_1^1(1-p_1^1)$
$(1-\rho_1)\alpha_2^0 p_2^0$
$(1-\rho_1)\alpha_2^0(1-p_2^0)$
$(1-\rho_1)\alpha_1^0 p_1^0$
$(1-\rho_1)\alpha_1^0(1-p_1^0)$
(0 0;0)
(0 0;1)
(0 1;0)
(0 1;1)
(1 0;0)
(1 0;1)

(I) DTMC evolution when $s_1(t) = 1$

(II) DTMC evolution when $s_1(t) = 0$

FIGURE 4.4: Different DTMC evolutions associated with $s_1(t)$

Note that $\bar{\mathbf{Y}}$ is aggregate state for each DTMC, which includes only $\bar{\mathbf{y}}$. The state evolution from $\acute{\mathbf{y}}^0(t)$ to another state $\hat{\mathbf{y}}^0(t)$ depends only on the current state $\acute{\mathbf{y}}^0(t)$ and the current input including the channel state $s_1(t)$ and the decision schedule. So both $\bar{\mathbf{Y}}^0$ and $\bar{\mathbf{Y}}^0$ are Markovian. Rectangles in Figure 4.4 are the associated states in $\bar{\mathbf{Y}}^1$ (Figure 4.4I) and $\bar{\mathbf{Y}}^0$ (Figure 4.4II). Next we investigate the evolution of queue lengths.

For evolution of queue length we have :

$$Q_1(t+1) = \big[Q_1(t) - \acute{y}_1^1(t)s_1(t) + (1-s_1(t))\acute{y}_1^0(t)\big] + A_1(t) \tag{4.7}$$

$$\begin{aligned} Q_{1i}(t+1) = \big[&Q_{1i}(t) - \acute{y}_i^1(t)s_1(t)1_{\{Q_{1i}(t)>Q_i(t)\}} + (1-s_1(t))\acute{y}_i^0(t)1_{\{Q_{1i}(t)>Q_i(t)\}}\big] \\ &+ (1-s_1(t))\acute{y}_1^0(t)1_{\{i=\arg\max_j \omega_j^0(t)\}} \end{aligned} \tag{4.8}$$

$$Q_i(t+1) = \big[Q_i(t) - \acute{y}_i^1(t)s_1(t)1_{\{Q_i(t)\geq Q_{12}(t)\}} + (1-s_1(t))\acute{y}_i^0(t)1_{\{Q_i(t)\geq Q_{1i}(t)\}}\big] + A_i(t) \tag{4.9}$$

$$i \in \{2, \cdots, N\}$$

It can be seen that the service rates only depends on $s_1(t)$, $\acute{\mathbf{y}}^1(t)$ and $\acute{\mathbf{y}}^0(t)$ and since $\acute{\mathbf{y}}^1(t)$ and $\acute{\mathbf{y}}^0(t)$ are Markovian, the queue lengths evolve as a Markov Chain with the transitions caused by arrivals, departures and channel state in the current time slot.

The candidate decision schedule for users when the channel is ON is denoted by $\mathbf{m}^1(t)$ and when the channel is OFF with $\mathbf{m}^0(t)$. The set of all $\mathbf{m}^1(t)$ and $\mathbf{m}^0(t)$ is denoted by $\mathcal{M}_0^1$ and $\mathcal{M}_0^0$, respectively. We define $\alpha^1\left(\mathbf{m}^1(t)\right)$ and $\alpha^0\left(\mathbf{m}^0(t)\right)$ as the probability that $\mathbf{m}^1(t)$ is chosen in the control slot when the channel is ON, and the probability that $\mathbf{m}^0(t)$ is chosen in the control slot when the channel is OFF, respectively.

Next, we describe MQ-CSMA1 in algorithm 2, which generates reversible DTMC's, $\bar{\mathbf{Y}}^1$ and $\bar{\mathbf{Y}}^0$ by characterizing the different network topologies when the channel is ON or

Algorithm 2 MQ-CSMA1

```
At each time slot, each node $i$ does the following procedure.
1: if $s_1(t) = 1$ then
2:     In the control slot, randomly select a decision schedule $\mathbf{m}^1(t) \in \mathcal{M}_0^1$ with probability $\alpha^1\left(\mathbf{m}^1(t)\right)$
3:     if $i \in \mathbf{m}^1(t)$ and $\acute{y}_j^1(t-1) = 0$ for all $j \neq i$ then
4:         $x_i(t) = 1$ with probability $p_i^1$
5:         $x_i(t) = 0$ with probability $1 - p_i^1$
6:     else if $i \in \mathbf{m}^1(t)$ and $\acute{y}_j^1(t-1) = 1$ for some $j \neq i$ then
7:         $x_i(t) = 0$
8:     else
9:         $x_i(t) = \acute{y}_i^1(t-1) \quad (i \notin \mathbf{m}^1(t))$
10:    In the Data slot, use $\mathbf{x}(t)$ as the transmission schedule
11: else
12:    Execute lines 2-10 by replacing all 1 with 0 in the superscript.
```

OFF. Algorithm 2 summarizes MQ-CSMA1 algorithm. At each slot t, all SU's and PU senses the channel to acquire state of $s_1(t)$ as shown in Figure 4.5. Next we elaborate on different behavior of users associated with channel state. When the channel is ON, users treat most recently ON slot as their previous slot ignoring the OFF period, and schedule packets in a way similar to Q-CSMA (Lines 2-10) where $\bar{p}_i = 1 - p_i$. When the channel is OFF, users treat most recently OFF slot as their previous slot ignoring the ON period, and schedule packets in a way similar to Q-CSMA (Lines 12 in Algorithm 2).

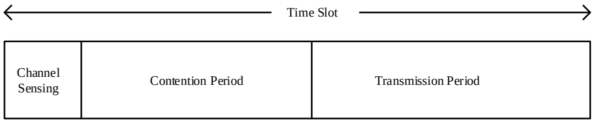


Figure 4.5: Time slot model.

To shed light on how the algorithm works, let us consider Fig.4.4. Rectangles in Figure 4.4I are states of aggregate Markov chain, $\bar{\mathbf{Y}}^1$, defined in (4.6) with $\acute{\mathbf{y}}^1 \in \{(0\ 0), (0\ 1), (1\ 0)\}$ and similarly rectangles Figure 4.4II are states of aggregate Markov chain, $\bar{\mathbf{Y}}^0$ defined in (4.5) with $\acute{\mathbf{y}}^0 \in \{(0\ 0), (0\ 1), (1\ 0)\}$. Ovals in Figure 4.4I are states, $\mathbf{y}^1$ with $\mathbf{y}^1 \in \{(0\ 0; 0), (0\ 0; 1), (0\ 1; 0), (0\ 1; 1), (1\ 0; 0), (1\ 0; 1)\}$ and also similarly, ovals in Figure 4.4II are states of $\mathbf{y}^0$ with $\mathbf{y}^0 \in \{(0\ 0; 0), (0\ 0; 1), (0\ 1; 0), (0\ 1; 1), (1\ 0; 0), (1\ 0; 1)\}$. For example, $\mathbf{y}^1 = (0\ 1; 1)$ indicates that the channel is ON and at time slot t, SU is transmitting and $\mathbf{y}^1 = (1\ 0; 0)$ indicates that the channel is OFF and in the most recent slot when the channel was ON, PU was transmitting. $\mathbf{y}^0 = (1\ 0; 0)$ indicates that channel is OFF and PU transmits a packet to SU and $\mathbf{y}^0 = (1\ 0; 1)$ indicates that in the most recent data slot when the channel was OFF PU was transmitting a packet to SU.

When the channel is ON, only the chain in Figure 4.4I, associated with $s_1(t) = 1$ evolves and the chain in Figure 4.4II associated with $s_1(t) = 0$ remains in the previous state (Lines 2-10 in Algorithm 2). When the channel is OFF, only the chain in Figure 4.4II associated with $s_1(t) = 0$ evolves and first the chain in Figure 4.4I, associated with $s_1(t) = 1$ remains in the previous state (Lines 12 in Algorithm 2).

Next, we show how to implement the algorithm in a distributed manner similar to [33]. We divide the time slot into three periods as shown in Fig. 4.5. In the channel sensing period, users acquire channel state of primary user, $s_1(t)$. Next to choose a randomize feasible schedule $\mathbf{m}^0(t)$ or $\mathbf{m}^1(t)$, (i.e. line 2 in the distributed algorithm) user i, randomly selects a number T_i uniformly distributed in $[1, W]$ and waits for T_i control mini slots, if user i hears a INTENT message from another user before $(T_i + 1)^{th}$ control mini-slot, i will not be included in $\mathbf{m}^0(t)$ or $\mathbf{m}^1(t)$ and consequently will not transmit a INTENT message in $(T_i + 1)^{th}$ control mini-slot. If user i does not hear a INTENT message from any other user before $(T_i + 1)^{th}$ mini slot, it will broadcast an INTENT message at the beginning of the $(T_i + 1)^{th}$ control mini-slot. If there is no collision in $(T_i + 1)^{th}$ control mini-slot, i will be included in $\mathbf{m}^0(t)$ or $\mathbf{m}^1(t)$ or else none of the users will transmit. After the end of transmission period, protocol level cooperation described in Section 4.1, performs.

In the following, we prove throughput optimality of Algorithm 2. Lemma 1 and Lemma 2 presents the transition probability of $\acute{\mathbf{y}}^0$ and $\acute{\mathbf{y}}^1$ respectively. Product-form of the stationary distribution of $\acute{\mathbf{y}}^0$ and $\acute{\mathbf{y}}^1$ are given in Theorem 5 and Theorem 6 respectively. And finally, Theorem 7 proves the throughput optimality of Algorithm 2.

Lemma 1. 1. A state $\acute{\mathbf{y}}^0$ can make a transition to a state $\hat{\mathbf{y}}^0$ $\left(\acute{\mathbf{y}}^0 \neq \hat{\mathbf{y}}^0\right)$ iff

$$\acute{\mathbf{y}}^0 \cup \hat{\mathbf{y}}^0 \in \mathcal{M}_0^0 \tag{4.10}$$

and there exists a decision schedule $\mathbf{m}^0 \in \mathcal{M}_0^0$ s.t.

$$\acute{\mathbf{y}}^0 \Delta \hat{\mathbf{y}}^0 := (\acute{\mathbf{y}}^0 \backslash \hat{\mathbf{y}}^0) \cup (\hat{\mathbf{y}}^0 / \acute{\mathbf{y}}^0) \subseteq \mathbf{m}^0 \tag{4.11}$$

2. The transition probability $P\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right)$ from $\acute{\mathbf{y}}^0$ to $\hat{\mathbf{y}}^0$ is

$$P\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right) = \sum_{\mathbf{m}^0 \in \mathcal{M}_0^0 : \acute{\mathbf{y}}^0 \Delta \hat{\mathbf{y}}^0 \subseteq \mathbf{m}^0} (1-\rho_1)\alpha^0(\mathbf{m}^0) \left(\prod_{l \in \acute{\mathbf{y}}^0 \setminus \hat{\mathbf{y}}^0} \bar{p^0}_l \right) \left(\prod_{k \in \hat{\mathbf{y}}^0 / \acute{\mathbf{y}}^0} p_k^0 \right) \left(\prod_{i \in \mathbf{m}^0 \cap (\acute{\mathbf{y}}^0 \cap) \hat{\mathbf{y}}^0} p_i^0 \right) \left(\prod_{j \in \mathbf{m}^0 \setminus (\acute{\mathbf{y}}^0 \cup \hat{\mathbf{y}}^0) \setminus C(\acute{\mathbf{y}}^0 \cup \hat{\mathbf{y}}^0)} \bar{p^0}_j \right) \tag{4.12}$$

where $C(\acute{\mathbf{y}}^0 \cup \hat{\mathbf{y}}^0)$ denotes the neighbors of nodes in $\acute{\mathbf{y}}^0 \cup \hat{\mathbf{y}}^0$.

Proof. Part one in can be proven as Lemma 2 in [33]. To prove Part 2. let us denote $P^{sch}\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right)$ as $P\left(\acute{\mathbf{y}}, \hat{\mathbf{y}}\right)$ in Lemma 2 of [33] which is the transition probability from $\acute{\mathbf{y}}$ to $\hat{\mathbf{y}}$ with always OFF channel. we need to show that

$$P\left((\acute{\mathbf{y}}^0; 0), \hat{\mathbf{y}}^0\right) = P\left((\acute{\mathbf{y}}^0; 1), \hat{\mathbf{y}}^0\right) = (1-\rho_1)P^{sch}\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right) \tag{4.13}$$

We have

$$\begin{aligned} P\left((\acute{\mathbf{y}}^0; 0), \hat{\mathbf{y}}^0\right) =& P\left((\acute{\mathbf{y}}^0; 0), (\hat{\mathbf{y}}^0; 0)\right) + P\left((\acute{\mathbf{y}}^0; 0), (\hat{\mathbf{y}}^0; 1)\right) \\ & (1-\rho_1)P^{sch}\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right) + 0 \end{aligned} \tag{4.14}$$

$$\begin{aligned} P\left((\acute{\mathbf{y}}^0; 1), \hat{\mathbf{y}}^0\right) =& P\left((\acute{\mathbf{y}}^0; 1), (\hat{\mathbf{y}}^0; 0)\right) + P\left((\acute{\mathbf{y}}^0; 1), (\hat{\mathbf{y}}^0; 1)\right) \\ & (1-\rho_1)P^{sch}\left(\acute{\mathbf{y}}^0, \hat{\mathbf{y}}^0\right) + 0 \end{aligned} \tag{4.15}$$

By Lemma 2 in [33] which states the transition probability with the always-available channel, we can prove Part 2. □

State transition probabilities shows that, $DTMC^0$ has product-form stationary distributions. Since there are two chains for two different channel state either one of them can be treated as the channel state unchanged.

Theorem 5. A necessary and sufficient condition for the $DTMC^0$ to be irreducible and aperiodic is $\bigcup_{\mathbf{m}^0 \in \mathcal{M}_0^0} \mathbf{m}^0 = \{1, \cdots, N\}$ and in this case the $DTMC^0$ is reversible and has the following product-form stationary distribution:

$$\pi\left(\bar{Y}^0\right) = \frac{1}{Z^0}\prod_{i\in\mathbf{y}^0}\frac{p_i^0}{\bar{p}_i^0} \tag{4.16}$$

$$Z^0 = \sum_{\mathbf{y}^0\in\mathcal{M}^0}\prod_{i\in\mathbf{y}^0}\frac{p_i^0}{\bar{p}_i^0} \tag{4.17}$$

Proof. Necessary and sufficient conditions can be proven as Proposition 1 in [33]. We can check that distribution in 4.16 satisfies the detailed balance equation:

$$\pi\left(\acute{\mathbf{y}}^0\right)P\left(\acute{\mathbf{y}}^0,\hat{\acute{\mathbf{y}}}^0\right) = \pi\left(\hat{\acute{\mathbf{y}}}^0\right)P\left(\hat{\acute{\mathbf{y}}}^0,\acute{\mathbf{y}}^0\right) \tag{4.18}$$

hence the $DTMC^0$ is reversible and 4.16 is indeed its stationary distribution. □

Similarly we need to show that $DTMC^1$ has a product-form stationary distribution.

Lemma 2. 1. A state $\acute{\mathbf{y}}^1$ can make a transition to a state $\hat{\acute{\mathbf{y}}}^1$ $\left(\acute{\mathbf{y}}^1 \neq \hat{\acute{\mathbf{y}}}^1\right)$ iff

$$\acute{\mathbf{y}}^1 \cup \hat{\acute{\mathbf{y}}}^1 \in \mathcal{M}_0^1 \tag{4.19}$$

and there exists a decision schedule $\mathbf{m}^1 \in \mathcal{M}_0^1$ s.t.

$$\acute{\mathbf{y}}^1\Delta\hat{\acute{\mathbf{y}}}^1 := (\acute{\mathbf{y}}^1\backslash\hat{\acute{\mathbf{y}}}^1)\cup(\hat{\acute{\mathbf{y}}}^1/\acute{\mathbf{y}}^1) \subseteq \mathbf{m}^1 \tag{4.20}$$

2. The transition probability $P\left(\acute{\mathbf{y}}^1,\hat{\acute{\mathbf{y}}}^1\right)$ from $\acute{\mathbf{y}}^1$ to $\hat{\acute{\mathbf{y}}}^1$ is

$$\begin{aligned} P\left(\acute{\mathbf{y}}^1,\hat{\acute{\mathbf{y}}}^1\right) = &\sum_{\mathbf{m}^1\in\mathcal{M}_0^1:\acute{\mathbf{y}}^1\Delta\hat{\acute{\mathbf{y}}}^1\subseteq\mathbf{m}^1}\rho_1\alpha^1(\mathbf{m}^1)\left(\prod_{l\in\acute{\mathbf{y}}^1\backslash\hat{\acute{\mathbf{y}}}^1}\bar{p^1}_l\right)\left(\prod_{k\in\hat{\acute{\mathbf{y}}}^1/\acute{\mathbf{y}}^1}p_k^1\right) \\ &\left(\prod_{i\in\mathbf{m}^1\cap(\acute{\mathbf{y}}^1\cap)\hat{\acute{\mathbf{y}}}^1}p_i^1\right)\left(\prod_{j\in\mathbf{m}^1\backslash(\acute{\mathbf{y}}^1\cup\hat{\acute{\mathbf{y}}}^1)\backslash C(\acute{\mathbf{y}}^1\cup\hat{\acute{\mathbf{y}}}^1)}\bar{p^1}_j\right) \end{aligned} \tag{4.21}$$

where $C(\acute{\mathbf{y}}^1\cup\hat{\acute{\mathbf{y}}}^1)$ denotes the neighbors of nodes in $\acute{\mathbf{y}}^1\cup\hat{\acute{\mathbf{y}}}^1$.

Proof. Part one in can be proven as Lemma 2 in [33]. To prove Part 2. let us denote $P^{sch}\left(\acute{\mathbf{y}}^1,\hat{\acute{\mathbf{y}}}^1\right)$ as $P\left(\acute{\mathbf{y}},\hat{\acute{\mathbf{y}}}\right)$ in Lemma 2 of [33] which is the transition probability from $\acute{\mathbf{y}}$ to $\hat{\acute{\mathbf{y}}}$ with always ON channel. we need to show that

$$P\left((\acute{\mathbf{y}}^1;0),\hat{\acute{\mathbf{y}}}^1\right) = P\left((\acute{\mathbf{y}}^1;1),\hat{\acute{\mathbf{y}}}^1\right) = \rho_1 P^{sch}\left(\acute{\mathbf{y}}^1,\hat{\acute{\mathbf{y}}}^1\right) \tag{4.22}$$

The rest of the proof is the same as in Lemma 1 □

Theorem 6. A necessary and sufficient condition for the $DTMC^1$ to be irreducible and aperiodic is $\bigcup_{\mathbf{m}^1 \in \mathcal{M}_0^1} \mathbf{m}^1 = \{1, \cdots, N\}$ and in this case the $DTMC^1$ is reversible and has the following product-form stationary distribution:

$$\pi\left(\bar{Y}^1\right) = \frac{1}{Z^1} \prod_{i \in \acute{\mathbf{y}}^1} \frac{p_i^1}{\bar{p}_i^1} \tag{4.23}$$

$$Z^1 = \sum_{\acute{\mathbf{y}}^1 \in \mathcal{M}^1} \prod_{i \in \mathbf{y}^1} \frac{p_i^1}{\bar{p}_i^1} \tag{4.24}$$

Proof. Necessary and sufficient conditions can be proven as Proposition 1 in [33]. We can check that distribution in 4.23 satisfies the detailed balance equation:

$$\pi\left(\acute{\mathbf{y}}^1\right) P\left(\acute{\mathbf{y}}^1, \hat{\mathbf{y}}^1\right) = \pi\left(\hat{\mathbf{y}}^1\right) P\left(\hat{\mathbf{y}}^1, \acute{\mathbf{y}}^1\right) \tag{4.25}$$

hence the $DTMC^1$ is reversible and 4.23 is indeed its stationary distribution. □

Product-form distribution of $\bar{Y}^0$ and $\bar{Y}^1$, suggests that we can use results established in [65] to prove throughput optimality of the algorithm.

Theorem 7. Let $\omega^*(t) := \max_{x \in \mathcal{M}(t)} \sum_{i \in x(t)} \omega_i(t)$, where $\mathcal{M}(t)$ is the set of all feasible schedules at time t. For a scheduling algorithm, if given any $0 < \epsilon, \delta < 1$, there exists $\beta > 0$ such that: if $\omega^*(t) > \beta$, the scheduling algorithm chooses a schedule $\mathbf{x}(t) \in \mathcal{M}(t)$ that satisfies

$$Pr\left\{\sum_{i \in \mathbf{x}(t)} \omega_i(t) \geq (1-\epsilon)\omega^*(t)\right\} \geq 1-\delta \tag{4.26}$$

where, $\omega_i(t)$ is a function of queue lengths defined in (4.1) with $f(.)$satisfying the following conditions:

1. $f_i(w)$ is a non decreasing, continuous function with $\lim_{Q_i \to \infty} f_i(w) = \infty$;
2. Given any $a \in \Re$, $\lim_{w \to \infty} \frac{f_i(w+a)}{f_i(w)} = 1$

Then the scheduling algorithm is throughput optimal.

Throughput optimality results established in Theorem 7 holds for any network topology with fading wireless channels as long as conditions are satisfied. We choose $p_i = \frac{e^{\omega_i(t)}}{1+e^{\omega_i(t)}}$ as long as conditions in Theorem 7 is satisfied. By choosing ω_i wisely, p_i changes slowly over time and we can assume that DTMC is in steady-state in every time slot (time scale separation) [33]. In the following we will show that MQ-CSMA1 is throughput optimal by showing that in each channel state it is close enough to another throughput optimal algorithm (i.e., MWS).

Theorem 8. Suppose $\bigcup_{m^0\in\mathcal{M}^0} m^0 = \{1,\cdots,N\}$ and $\bigcup_{m^1\in\mathcal{M}^1} m^1 = \{1,\cdots,N\}$. Let $p_i^0 = \frac{e^{\omega_i^0(t)}}{1+e^{\omega_i^0(t)}}$, $\forall i\in\{1,\cdots,N\}$ when $s(t)=0$ and $p_i^1 = \frac{e^{\omega_i^1(t)}}{1+e^{\omega_i^1(t)}}$, $\forall i\in\{1,\cdots,N\}$ when $s(t)=1$. Then the algorithm is throughput-optimal.

Proof. Theorems 5 and 6 states that both DTMC $\bar{Y}^0$ and $\bar{Y}^1$ have product-form stationary distributions. Given any $0<\epsilon,\delta<1$, we define $\omega^{0*}(t) = \max_{\mathbf{x}^0\in\mathcal{M}^0}\sum_{i\in\mathbf{x}^0}\omega_i^0(t)$. Based on this, four sets of states, when channel is OFF are defined as follows:

$$\chi_0^0 := \left\{(\acute{\mathbf{y}}^0;0)|\acute{\mathbf{y}}^0\in\mathcal{M}_0^0, \sum_{i\in\acute{\mathbf{y}}^0}\omega_i^0(t) < (1-\epsilon)\omega^{0*}(t)\right\} \tag{4.27}$$

$$\chi_1^0 := \left\{(\acute{\mathbf{y}}^0;1)|\acute{\mathbf{y}}^0\in\mathcal{M}_0^0, \sum_{i\in\acute{\mathbf{y}}^0}\omega_i^0(\tau) < (1-\epsilon)\omega^{0*}(\tau)\right\} \tag{4.28}$$

$$\varphi^0 := \chi_0^0\cup\chi_1^0 \tag{4.29}$$

$$\psi^0 := \left\{\bar{Y}^0 = (\acute{\mathbf{y}}^0)|\acute{\mathbf{y}}^0\in\mathcal{M}_0^0, \sum_{i\in\acute{\mathbf{y}}^0}\omega_i^0(t) < (1-\epsilon)\omega^{0*}(t)\right\} \tag{4.30}$$

where χ_0^0 includes all states with the channel being OFF and the sum of $\omega_i^0(t)$ from users chosen in the schedule is at least a fraction of ϵ away from $\omega^{0*}(t)$, χ_1^0 includes all states with the channel ON and the sum of $\omega_i^0(\tau)$ from users chosen in the schedule of the most recently OFF slot is at least a fraction of ϵ away from $\omega^{0*}(\tau)$. As a reminder, here τ is the most recent time slot where channel was OFF. It can be seen that if $(\acute{\mathbf{y}}^0;s(t))\in\varphi^0$, then $\bar{\mathbf{y}}^0 = (\acute{\mathbf{y}}^0)\in\psi^0$. We then calculate the probability of a state in set χ_0^0.

$$\begin{aligned}\pi(\chi_0^0) < \pi(\varphi^0) = \pi(\psi^0) = \sum_{\acute{\mathbf{y}}^0\in\psi^0}\pi(\bar{Y}^0) = \sum_{\acute{\mathbf{y}}^0\in\psi^0}\frac{e^{\sum_{i\in\mathbf{y}^0}\omega_i^0(t)}}{Z^0}\\ \leq \frac{(N+1)e^{(1-\epsilon)\omega^{0*}(t)}}{Z^0} < \frac{N+1}{e^{\epsilon\omega^{0*}(t)}}\end{aligned} \tag{4.31}$$

where,

$$Z^0 = \sum_{\acute{\mathbf{y}}^0\in\mathcal{M}_0^0} e^{\sum_{i\in\acute{\mathbf{y}}^0}\omega_i^0(t)} > e^{\max_{\acute{\mathbf{y}}^0\in\mathcal{M}_0^0}\sum_{i\in\acute{\mathbf{y}}^0}\omega_i^0(t)} = e^{\omega^{0*}(t)} \tag{4.32}$$

$\pi(\varphi^0) = \pi(\psi^0)$, because $1_{\{(\acute{\mathbf{y}}^0;0)\cup(\acute{\mathbf{y}}^0;1)\}} = 1_{\{(\acute{\mathbf{y}}^0)\}}$; The last equality is true because $|\psi^0| \leq |\mathcal{M}_0^0| = N+1$. Thus, $\exists\beta^0>0$, such that: $\omega^{0*}>\beta^0$ implies that $\pi(\chi_0^0)<$

$\delta \min(\rho_1, 1-\rho_1)$ with $\delta = \frac{N+1}{\min(\rho_1, 1-\rho_1)e^{\epsilon\beta^0}}$. Then we have the following result:

$$\begin{aligned} Pr\left\{\sum_{i\in\acute{\mathbf{y}}^0} \omega_i^0(t) \geq (1-\epsilon)\omega^{0*}(t)|s(t)=0\right\} &= 1 - Pr\left\{\sum_{i\in\acute{\mathbf{y}}^0} \omega_i^0(t) < (1-\epsilon)\omega^{0*}(t)|s(t)=0\right\} \\ &= 1 - \frac{\pi(\chi_0^0)}{1-\rho_1} > 1 - \frac{\delta\min(\rho_1, 1-\rho_1)}{1-\rho_1} \geq 1-\delta \end{aligned} \tag{4.33}$$

Similarly for $\bar{Y}^1$, let $\max_{\acute{\mathbf{y}}^1 \in \mathcal{M}_0^1} \sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t)$. Similar to $DTMC^0$ four sets are defined as follow:

$$\chi_0^1 := \left\{(\acute{\mathbf{y}}^1; 0)|\acute{\mathbf{y}}^1 \in \mathcal{M}_0^1, \sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)\right\} \tag{4.34}$$

$$\chi_1^1 := \left\{(\acute{\mathbf{y}}^1; 1)|\acute{\mathbf{y}}^1 \in \mathcal{M}_0^1, \sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)\right\} \tag{4.35}$$

$$\varphi^1 := \chi_0^1 \cup \chi_1^1 \tag{4.36}$$

$$\psi^1 := \left\{\bar{Y}^1 = (\acute{\mathbf{y}}^1)|\acute{\mathbf{y}}^1 \in \mathcal{M}_0^1, \sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)\right\} \tag{4.37}$$

We then calculate the probability of a state in set χ_1^1

$$\begin{aligned} \pi(\chi_1^1) < \pi(\varphi^1) = \pi(\psi^1) = \sum_{\acute{\mathbf{y}}^1\in\psi^1} \pi(\bar{Y}^1) &= \sum_{\acute{\mathbf{y}}^1\in\psi^1} \frac{e^{\sum_{i\in\acute{\mathbf{y}}^1}\omega_i^1(t)}}{Z^1} \\ &\leq \frac{(N+1)e^{(1-\epsilon)\omega^{1*}(t)}}{Z^1} < \frac{N+1}{e^{\epsilon\omega^{1*}(t)}} \end{aligned} \tag{4.38}$$

where,

$$Z^1 = \sum_{\acute{\mathbf{y}}^1\in\mathcal{M}_0^1} e^{\sum_{i\in\acute{\mathbf{y}}^1}\omega_i^1(t)} > e^{\max_{\acute{\mathbf{y}}^1\in\mathcal{M}_0^1}\sum_{i\in\acute{\mathbf{y}}^1}\omega_i^1(1)} = e^{\omega^{1*}(t)} \tag{4.39}$$

$\pi(\varphi^1) = \pi(\psi^1)$, because $1_{\{(\acute{\mathbf{y}}^1;0)\cup(\acute{\mathbf{y}}^1;1)\}} = 1_{\{(\acute{\mathbf{y}}^1)\}}$; The last equality is true because $|\psi^1| \leq |\mathcal{M}_0^1| = N+1$. Thus, $\exists\beta^1 > 0$, such that: $\omega^{1*} > \beta^1$ implies that $\pi(\chi_1^1) < \delta\min(\rho_1, 1-\rho_1)$ with $\delta = \frac{N+1}{\min(\rho_1,1-\rho_1)e^{\epsilon\beta^1}}$. Then we have the following result:

$$\begin{aligned} Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) \geq (1-\epsilon)\omega^{1*}(t)|s(t)=1\right\} &= 1 - Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)|s(t)=1\right\} \\ &= 1 - \frac{\pi(\chi_0^0)}{\rho_1} > 1 - \frac{\delta\min(\rho_1, 1-\rho_1)}{\rho_1} \geq 1-\delta \end{aligned} \tag{4.40}$$

We use the total probability formula to calculate the unconditional probability:

$$
\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{x}}(t)}\omega_i(t)\geq(1-\epsilon)\omega^*(t)\right\}\\
=&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^0(t)}\omega_i^0(t)\geq(1-\epsilon)\omega^{0*}(t)|s(t)=0\right\}Pr\left(s(t)=0\right)\\
+&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1(t)}\omega_i^1(t)\geq(1-\epsilon)\omega^{1*}(t)|s(t)=1\right\}Pr\left(s(t)=1\right)\\
\geq&(1-\delta)\rho_1+(1-\delta)(1-\rho_1)=1-\delta
\end{aligned}
\tag{4.41}
$$

Note that, $\mathbf{x}(t)=\acute{\mathbf{y}}^0(t)$, $\omega^*(t)=\omega^{0*}(t)$ when $s(t)=0$ and $\mathbf{x}(t)=\acute{\mathbf{y}}^1(t)$, $\omega^*(t)=\omega^{1*}(t)$ when $s(t)=1$. Hence by Theorem 7, MQ-CSMA1 is throughput optimal. □

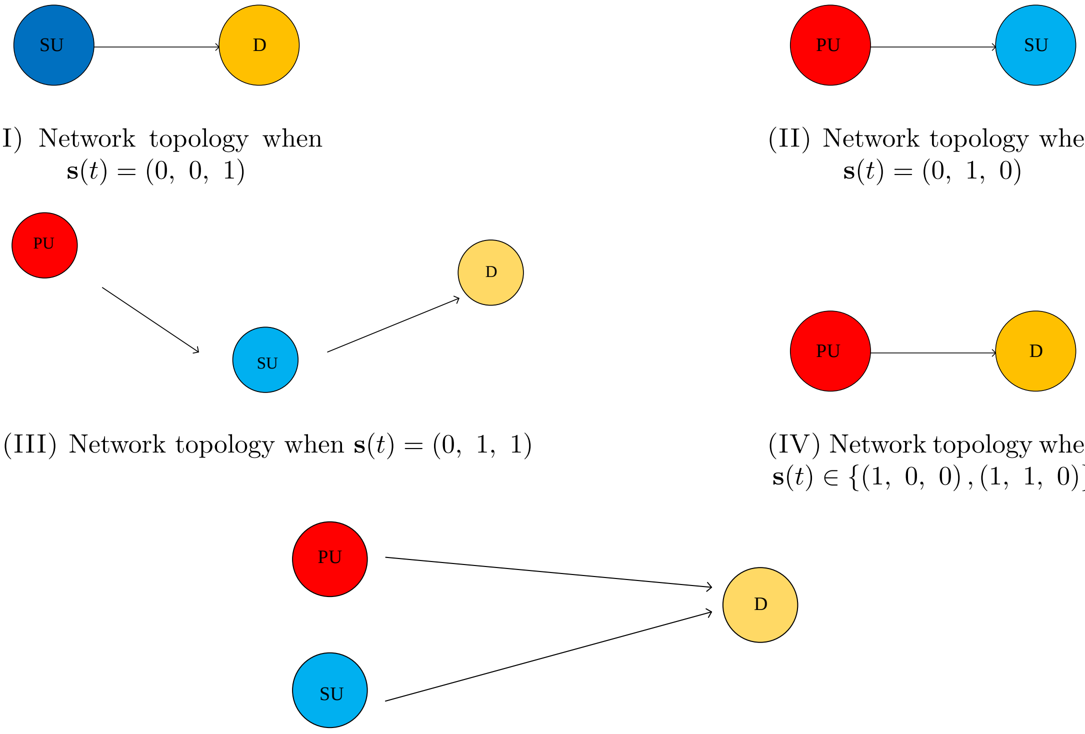
4.4 Extension to Multiple Fading channels (with N=2)

In section 3.5 we considered cooperative network when all the channels are fading. We assume separate queues in SU as in Figure 4.1 (with, $N=2$, i.e., one PU and one SU) so that PU can calculate its weight in a distributed manner as discussed in section 3.5. Our ultimate goal is to design a distributed algorithm to solve the scheduling problem as discussed in section 3.5. As a reminder, channels $PU-D$, $PU-SU$ and $SU-D$ are denoted by $s_1(t)$, $s_{12}(t)$ and $s_2(t)$ respectively, with $Pr\left(s_1(t)=1\right)=\rho_1$, $Pr\left(s_{12}(t)=1\right)=\rho_{12}$ and $Pr\left(s_2(t)=1\right)=\rho_2$. Channels state vector is i.i.d. over slots and is denoted by $\mathbf{s}(t)=(s_1(t),\ s_{12}(t),\ s_2(t))$.

4.4.1 Distributed Algorithm

Similar to section 4.3 we will associate a separate Markov chain with each channel state. Each channel state realization changes the topology of the network as depicted in Figure 4.6. Vector of transmission schedule is denoted by $\mathbf{x}(t)$.

Let us define the following:



(I) Network topology when $\mathbf{s}(t) = (0,\ 0,\ 1)$

(II) Network topology when $\mathbf{s}(t) = (0,\ 1,\ 0)$

(III) Network topology when $\mathbf{s}(t) = (0,\ 1,\ 1)$

(IV) Network topology when $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$

(V) Network topology when $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}$

FIGURE 4.6: Different network topologies associated with $\mathbf{s}(t)$

$$\acute{\mathbf{y}}^{1}(t) = \{\mathbf{x}(\tau) : \ \text{the largest } \tau \leq t \ \text{ with } \ \mathbf{s}(t) = (0,\ 0,\ 1)\} \tag{4.42}$$

$$\acute{\mathbf{y}}^{2}(t) = \{\mathbf{x}(\tau) : \ \text{the largest } \tau \leq t \ \text{ with } \ \mathbf{s}(t) = (0,\ 1,\ 0)\} \tag{4.43}$$

$$\acute{\mathbf{y}}^{3}(t) = \{\mathbf{x}(\tau) : \ \text{the largest } \tau \leq t \ \text{ with } \ \mathbf{s}(t) = (0,\ 1,\ 1)\} \tag{4.44}$$

$$\acute{\mathbf{y}}^{46}(t) = \{\mathbf{x}(\tau) : \ \text{the largest } \tau \leq t \ \text{ with } \ \mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}\} \tag{4.45}$$

$$\acute{\mathbf{y}}^{57}(t) = \{\mathbf{x}(\tau) : \ \text{the largest } \tau \leq t \ \text{ with } \ \mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}\} \tag{4.46}$$

where $\acute{\mathbf{y}}^{1}(t)$ is the vector of transmission schedule in the most recent data slot (including time t) when $\mathbf{s}(t) = (0,\ 0,\ 1)$, $\acute{\mathbf{y}}^{2}(t)$ is the vector of transmission schedule in the most recent data slot, when $\mathbf{s}(t) = (0,\ 1,\ 0)$, $\acute{\mathbf{y}}^{3}(t)$ is the vector of transmission schedule in the most recent data slot, when $\mathbf{s}(t) = (0,\ 1,\ 1)$, $\acute{\mathbf{y}}^{46}(t)$ is the vector of transmission schedule in the most recent data slot, when $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$ and $\acute{\mathbf{y}}^{57}(t)$ is the vector of transmission schedule in the most recent data slot, when $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}$. Note that, each $\acute{\mathbf{y}}(t)$ makes a transition into another state only in their own associated channel state and stop evolving otherwise. For instance,

$\acute{\mathbf{y}}^{57}(t)$ only makes transition if the channel state vector, $\mathbf{s}(t)$ is $(1,\ 0,\ 1)$ or $(1,\ 0,\ 1)$, otherwise it does not change its state. With a little abuse of notation, we treat $\acute{\mathbf{y}}$ as a set and write $i \in \acute{\mathbf{y}}$ if $\acute{y}_i = 1$. Further, we define $\mathbf{y}^s(t) = (\acute{\mathbf{y}}^s(t); \mathbf{s}(t))$ for all $s \in \{1, 2, 3, 46, 57\}$ which includes channels state vector $\mathbf{s}(t)$. The ovals in Figure 4.7 denote $\mathbf{y}(t)$'s for all $\mathbf{s}(t) \in \{(0,\ 0,\ 0), (0,\ 0,\ 1), (0,\ 1,\ 0), (0,\ 1,\ 1), (1,\ 0,\ 1), (1,\ 1,\ 0), (1,\ 1,\ 1)\}$.

Let us define:

$$\bar{\mathbf{Y}}^1 = (\acute{\mathbf{y}}^1(t)) \tag{4.47}$$

$$\bar{\mathbf{Y}}^2 = (\acute{\mathbf{y}}^2(t)) \tag{4.48}$$

$$\bar{\mathbf{Y}}^3 = (\acute{\mathbf{y}}^3(t)) \tag{4.49}$$

$$\bar{\mathbf{Y}}^{46} = (\acute{\mathbf{y}}^{46}(t)) \tag{4.50}$$

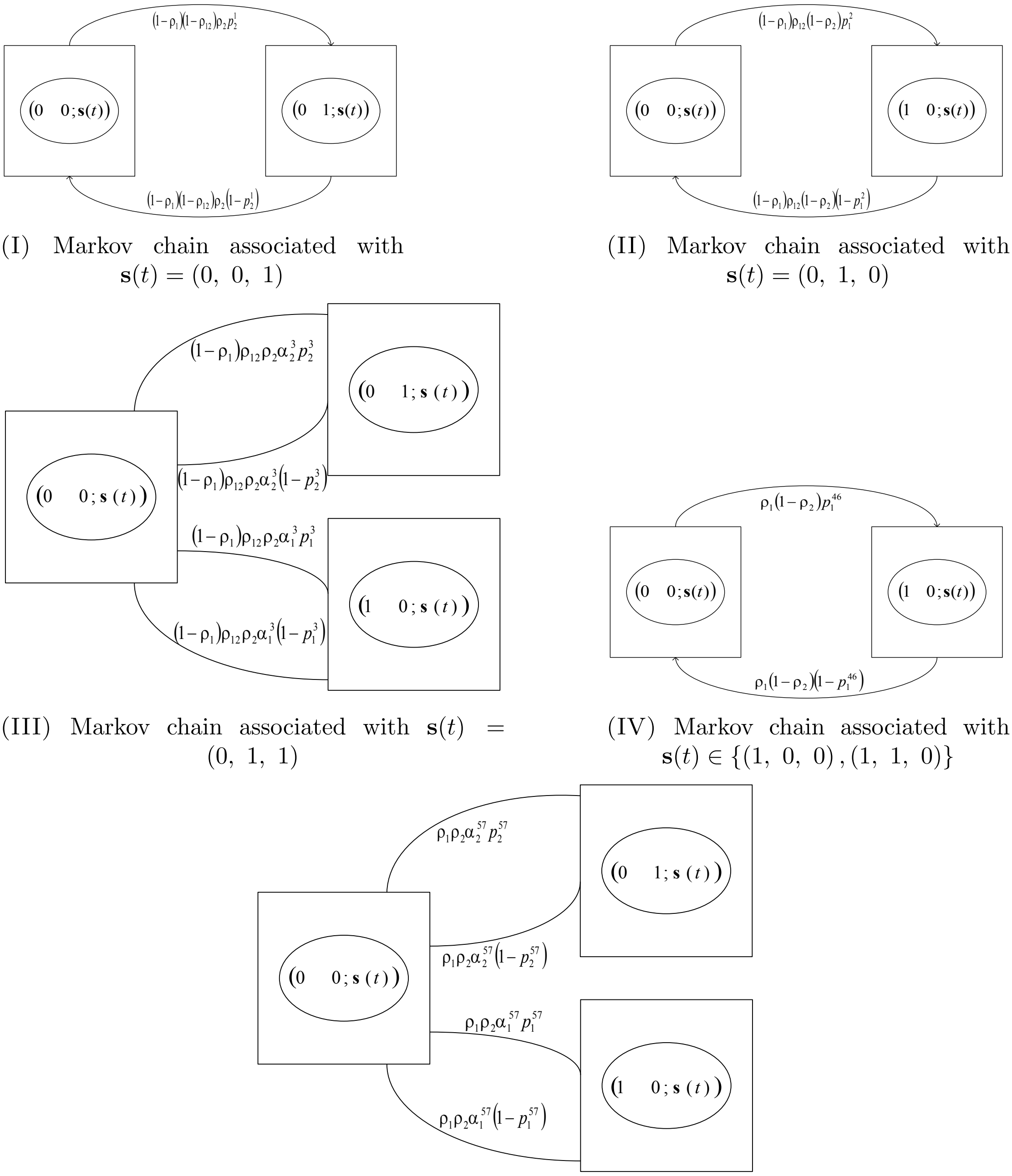
$$\bar{\mathbf{Y}}^{57} = (\acute{\mathbf{y}}^{57}(t)) \tag{4.51}$$

where, $\bar{\mathbf{Y}}$'s, are the aggregate state for each DTMC excluding channel state $\mathbf{s}(t)$.

Next we describe how the algorithm works. When the channel state vector is $\mathbf{s}(t) = (0,\ 0,\ 1)$, candid feasible schedule $\mathbf{m}^1(t)$ includes only *SU* transmitting to *D*. Accordingly $\alpha^1(\mathbf{m}^1(t))$, probability of choosing $\mathbf{m}^1(t)$, equals 1 (Lines 1-4 in Algorithm 3) and SU transmits with probability p_2^1. Also, when channel state vector is $\mathbf{s}(t) = (0,\ 1,\ 0)$, candid feasible schedule $\mathbf{m}^2(t)$ includes only *PU* transmitting to *SU*. Accordingly $\alpha^2(\mathbf{m}^2(t))$, probability of choosing $\mathbf{m}^2(t)$, equals 1 (Lines 5-8 in Algorithm 3) and PU transmits to SU with probability p_1^2. Similarly, given $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$, candid feasible schedule $\mathbf{m}^{46}(t)$ includes only *PU* transmitting to *D*. Accordingly $\alpha^{46}(\mathbf{m}^{46}(t))$, probability of choosing $\mathbf{m}^{46}(t)$, equals 1 (Lines 18-21 in Algorithm 3) and PU transmits with probability p_1^{46}. When $\mathbf{s}(t) = (0,\ 1,\ 1)$, candid feasible schedule $\mathbf{m}^3(t)$ includes transmission of *PU* to *SU*, or *SU* to *D* or none of them. In the control mini slot (Line 10 in Algorithm 3), $\mathbf{m}^3(t)$ is chosen with probability $\alpha^3(\mathbf{m}^3(t))$. Then if user i is selected in $\mathbf{m}^3(t)$, it transmits with probability p_i^3. And finally, given $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 0)\}$, candid feasible schedule $\mathbf{m}^{57}(t)$ includes transmission of *PU* to *D*, or *SU* to *D* or none of them. In the control mini slot (Line 23 in Algorithm 3), $\mathbf{m}^{57}(t)$ is chosen with probability $\alpha^{57}(\mathbf{m}^{57}(t))$. Then if user i is selected in $\mathbf{m}^{57}(t)$, it transmits with probability p_i^{57}.

4.4.2 Optimality

In this section we will formally prove throughput optimality of the Algorithm 3. First we derive stationary distribution of $\bar{Y}$.


$(1-\rho_1)(1-\rho_{12})\rho_2 p_2^1$
$(0\quad 0;\mathbf{s}(t))$
$(0\quad 1;\mathbf{s}(t))$
$(1-\rho_1)(1-\rho_{12})\rho_2(1-p_2^1)$
$(1-\rho_1)\rho_{12}(1-\rho_2)p_1^2$
$(0\quad 0;\mathbf{s}(t))$
$(1\quad 0;\mathbf{s}(t))$
$(1-\rho_1)\rho_{12}(1-\rho_2)(1-p_1^2)$
$(1-\rho_1)\rho_{12}\rho_2\alpha_2^3 p_2^3$
$(0\quad 1;\mathbf{s}(t))$
$(0\quad 0;\mathbf{s}(t))$
$(1-\rho_1)\rho_{12}\rho_2\alpha_2^3(1-p_2^3)$
$(1-\rho_1)\rho_{12}\rho_2\alpha_1^3 p_1^3$
$(1\quad 0;\mathbf{s}(t))$
$(1-\rho_1)\rho_{12}\rho_2\alpha_1^3(1-p_1^3)$
$\rho_1(1-\rho_2)p_1^{46}$
$(0\quad 0;\mathbf{s}(t))$
$(1\quad 0;\mathbf{s}(t))$
$\rho_1(1-\rho_2)(1-p_1^{46})$
$\rho_1\rho_2\alpha_2^{57}p_2^{57}$
$(0\quad 1;\mathbf{s}(t))$
$(0\quad 0;\mathbf{s}(t))$
$\rho_1\rho_2\alpha_2^{57}(1-p_2^{57})$
$\rho_1\rho_2\alpha_1^{57}p_1^{57}$
$(1\quad 0;\mathbf{s}(t))$
$\rho_1\rho_2\alpha_1^{57}(1-p_1^{57})$

(I) Markov chain associated with $\mathbf{s}(t) = (0,\ 0,\ 1)$

(II) Markov chain associated with $\mathbf{s}(t) = (0,\ 1,\ 0)$

(III) Markov chain associated with $\mathbf{s}(t) = (0,\ 1,\ 1)$

(IV) Markov chain associated with $\mathbf{s}(t) \in \{(1,\ 0,\ 0),(1,\ 1,\ 0)\}$

(V) Markov chain associated with $\mathbf{s}(t) \in \{(1,\ 0,\ 1),(1,\ 1,\ 1)\}$

FIGURE 4.7: Markov chains associated with $\mathbf{s}(t)$

Algorithm 3 MQ-CSMA2

```
At each time slot, each node $i$ does the following procedure.
if $\mathbf{s}(t) = (0,\ 0,\ 1)$ then
    $x_1(t) = 0$
    $x_2(t) = 1$ with probability $p_2^1$
    $x_2(t) = 0$ with probability $1 - p_2^1$
if $\mathbf{s}(t) = (0,\ 1,\ 0)$ then
    $x_2(t) = 0$
    $x_1(t) = 1$ with probability $p_1^2$
    $x_1(t) = 0$ with probability $1 - p_1^2$
if $\mathbf{s}(t) = (0,\ 1,\ 1)$ then
    In the control slot, randomly select a decision schedule $\mathbf{m}^3(t) \in \mathcal{M}_0^3$ with probability $\alpha^3\left(\mathbf{m}^3(t)\right)$
    if $i \in \mathbf{m}^3(t)$ and $\acute{y}_j^3(t-1) = 0$ for all $j \neq i$ then
        $x_i(t) = 1$ with probability $p_i^3$
        $x_i(t) = 0$ with probability $1 - p_i^3$
    else if $i \in \mathbf{m}^3(t)$ and $\acute{y}_j^3(t-1) = 1$ for some $j \neq i$ then
        $x_i(t) = 0$
    else
        $x_i(t) = \acute{y}_i^3(t-1) \quad (i \notin \mathbf{m}^1(t))$
if $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$ then
    $x_2(t) = 0$
    $x_1(t) = 1$ with probability $p_1^{46}$
    $x_1(t) = 0$ with probability $1 - p_1^{46}$
if $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 0)\}$ then
    In the control slot, randomly select a decision schedule $\mathbf{m}^{57}(t) \in \mathcal{M}_0^{57}$ with probability $\alpha^{57}\left(\mathbf{m}^{57}(t)\right)$
    if $i \in \mathbf{m}^{57}(t)$ and $\acute{y}_j^{57}(t-1) = 0$ for all $j \neq i$ then
        $x_i(t) = 1$ with probability $p_i^{57}$
        $x_i(t) = 0$ with probability $1 - p_i^{57}$
    else if $i \in \mathbf{m}^3(t)$ and $\acute{y}_j^3(t-1) = 1$ for some $j \neq i$ then
        $x_i(t) = 0$
    else
        $x_i(t) = \acute{y}_i^3(t-1) \quad (i \notin \mathbf{m}^1(t))$
if $\mathbf{s}(t) = (0,\ 0,\ 0)$ then
    $\mathbf{x}(t) = (0\ 0)$
Use $\mathbf{x}(t)$ and $\mathbf{s}(t)$ as the scheduling decision at time slot $t$.
```

Note that rectangles in Figure 4.7 are the states of their associated $\bar{Y}$.

- $\bar{Y}^1$ is depicted in Figure 4.7I. It is easy to verify that:

$$\pi(0,0) = 1 - p_2^1, \quad \pi(0,1) = p_2^1 \tag{4.52}$$

- $\bar{Y}^2$ is depicted in Figure 4.7II. We have:

$$\pi(0,0) = 1 - p_1^2, \quad \pi(1,0) = p_1^2 \tag{4.53}$$

- $\bar{Y}^3$ is depicted in Figure 4.7III. We have:

$$\pi(0,0) = \frac{1}{Z^3}, \quad \pi(1,0) = \frac{1}{Z^3}\frac{p_1^3}{1-p_1^3}, \quad \pi(0,1) = \frac{1}{Z^3}\frac{p_2^3}{1-p_2^3} \tag{4.54}$$
$$Z^3 = 1 + \frac{p_1^3}{1-p_1^3} + \frac{p_2^3}{1-p_2^3}$$

- $\bar{Y}^{46}$ is depicted in Figure 4.7IV. We have:

$$\pi(0,0) = 1 - p_1^{46}, \quad \pi(1,0) = p_1^{46} \tag{4.55}$$

- $\bar{Y}^{57}$ is depicted in Figure 4.7V. We have:

$$\pi(0,0) = \frac{1}{Z^{57}}, \quad \pi(1,0) = \frac{1}{Z^{57}}\frac{p_1^{57}}{1-p_1^{57}}, \quad \pi(0,1) = \frac{1}{Z^{57}}\frac{p_2^{57}}{1-p_2^{3}} \tag{4.56}$$
$$Z^{57} = 1 + \frac{p_1^{57}}{1-p_1^{57}} + \frac{p_2^{57}}{1-p_2^{57}}$$

Optimal weights for our network model are defined in section 3.5. Optimal weights as a function of their queues are as follow:

$$\begin{aligned}\omega_1(t) =& f\left(Q_1(t)s_1(t) + (1-s_1(t))s_{12}(t)\left(Q_1(t) - Q_{12}(t)\right)^+\right)\\ \omega_2(t) =& \max\left(Q_2(t), Q_{12}(t)\right)s_2(t)\end{aligned} \tag{4.57}$$

Next we associate weights to each channel states as follow:

- $\mathbf{s}(t) = (0,\ 0,\ 1)$

$$\omega_1^1(t) = 0, \quad \omega_2^1(t) = f\left(\max\left(Q_2(t), Q_{12}(t)\right)\right) \tag{4.58}$$

- $\mathbf{s}(t) = (0,\ 1,\ 0)$

$$\omega_1^2(t) = f\left((Q_1(t) - Q_{12}(t))^+\right), \quad \omega_2^2(t) = 0 \tag{4.59}$$

- $\mathbf{s}(t) = (0,\ 1,\ 1)$

$$\omega_1^3(t) = f\left((Q_1(t) - Q_{12}(t))^+\right), \quad \omega_2^3(t) = f\left(\max\left(Q_2(t), Q_{12}(t)\right)\right) \tag{4.60}$$

- $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$

$$\omega_1^{46}(t) = f\left(Q_1(t)\right), \quad \omega_2^{46}(t) = 0 \tag{4.61}$$

- $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}$

$$\omega_1^{57}(t) = f\left(Q_1(t)\right), \quad \omega_2^{57}(t) = f\left(\max\left(Q_2(t), Q_{12}(t)\right)\right) \tag{4.62}$$

Theorem 9. $\forall i \in \{1,\ 2\}$, let $p_i^1 = \frac{e^{\omega_i^1(t)}}{1+e^{\omega_i^1(t)}}$ when $\mathbf{s}(t) = (0,\ 0,\ 1)$, $p_i^2 = \frac{e^{\omega_i^2(t)}}{1+e^{\omega_i^2(t)}}$ when $\mathbf{s}(t) = (0,\ 1,\ 0)$, $p_i^3 = \frac{e^{\omega_i^3(t)}}{1+e^{\omega_i^1(t)}}$ when $\mathbf{s}(t) = (0,\ 1,\ 1)$, $p_i^{46} = \frac{e^{\omega_i^{46}(t)}}{1+e^{\omega_i^{46}(t)}}$ when $\mathbf{s}(t) \in \{(1,\ 0,\ 0), (1,\ 1,\ 0)\}$ and $p_i^{57} = \frac{e^{\omega_i^{57}(t)}}{1+e^{\omega_i^{57}(t)}}$ when $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}$, with ω_i's defined in (4.58, 4.59, 4.60, 4.61, 4.62). Then MQ-CSMA2 in algorithm 3 is throughput-optimal.

Proof.

- $\mathbf{s}(t) = (0,\ 0,\ 1)$

$$\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)\right\} \\
=&\pi\left(\acute{\mathbf{y}}^1 = (0\ 0); (0,\ 0,\ 1)\right) = \pi\left(\acute{\mathbf{y}}^1 = (0\ 0)\right)(1-\rho_1)(1-\rho_{12})\rho_2 \\
=&(1-\rho_1)(1-\rho_{12})\rho_2 \frac{1}{1+e^{w_2^1}(t)} < (1-\rho_1)(1-\rho_{12})\rho_2 \frac{1}{e^{w_2^1}(t)} \\
=&(1-\rho_1)(1-\rho_{12})\rho_2\delta
\end{aligned} \tag{4.63}$$

then we have:

$$\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) \geq (1-\epsilon)\omega^{1*}(t)|\mathbf{s}(t) = (0,\ 0,\ 1)\right\} \\
=&1 - Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1} \omega_i^1(t) < (1-\epsilon)\omega^{1*}(t)|\mathbf{s}(t) = (0,\ 0,\ 1)\right\} \\
\geq& 1 - \frac{(1-\rho_1)(1-\rho_{12})\rho_2\delta}{(1-\rho_1)(1-\rho_{12})\rho_2} = 1-\delta
\end{aligned} \tag{4.64}$$

- $\mathbf{s}(t) = (0,\ 1,\ 0)$

$$
\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^2}\omega_i^2(t) < (1-\epsilon)\omega^{2*}(t)\right\}\\
=&\pi\left(\acute{\mathbf{y}}^2 = (0\ 0); (0,\ 1,\ 0)\right) = \pi\left(\acute{\mathbf{y}}^2 = (0\ 0)\right)(1-\rho_1)\rho_{12}(1-\rho_2)\\
=&(1-\rho_1)\rho_{12}(1-\rho_2)\frac{1}{1+e^{w_1^2}(t)} < (1-\rho_1)\rho_{12}(1-\rho_2)\frac{1}{e^{w_1^2}(t)}\\
=&(1-\rho_1)\rho_{12}(1-\rho_2)\delta
\end{aligned}
\tag{4.65}
$$

then we have:

$$
\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^2}\omega_i^1(t) \geq (1-\epsilon)\omega^{2*}(t)|\mathbf{s}(t) = (0,\ 1,\ 0)\right\}\\
=&1 - Pr\left\{\sum_{i\in\acute{\mathbf{y}}^2}\omega_i^2(t) < (1-\epsilon)\omega^{2*}(t)|\mathbf{s}(t) = (0,\ 1,\ 0)\right\}\\
\geq& 1 - \frac{(1-\rho_1)\rho_{12}(1-\rho_2)\delta}{(1-\rho_1)\rho_{12}(1-\rho_2)} = 1-\delta
\end{aligned}
\tag{4.66}
$$

- $\mathbf{s}(t) = (0,\ 1,\ 1)$

$$
\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^3}\omega_i^3(t) < (1-\epsilon)\omega^{3*}(t)\right\}\\
=&Pr\left(\omega^{3*}(t) = \omega_1^3(t)\right)\left[\pi\left(\acute{\mathbf{y}}^3 = (0\ 0); (0,\ 1,\ 1)\right) + \pi\left(\acute{\mathbf{y}}^3 = (0\ 1); (0,\ 1,\ 1)\right)\right]\\
+&Pr\left(\omega^{3*}(t) = \omega_2^3(t)\right)\left[\pi\left(\acute{\mathbf{y}}^3 = (0\ 0); (0,\ 1,\ 1)\right) + \pi\left(\acute{\mathbf{y}}^3 = (1\ 0); (0,\ 1,\ 1)\right)\right]\\
=&Pr\left(\omega^{3*}(t) = \omega_1^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\left[\pi\left(\acute{\mathbf{y}}^3 = (0\ 0)\right) + \pi\left(\acute{\mathbf{y}}^3 = (0\ 1)\right)\right]\\
+&Pr\left(\omega^{3*}(t) = \omega_2^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\left[\pi\left(\acute{\mathbf{y}}^3 = (0\ 0)\right) + \pi\left(\acute{\mathbf{y}}^3 = (1\ 0)\right)\right]\\
=&Pr\left(\omega^{3*}(t) = \omega_1^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\left(\frac{1+e^{\omega_2^3(t)}}{Z^3}\right)\\
+&Pr\left(\omega^{3*}(t) = \omega_2^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\left(\frac{1+e^{\omega_1^3(t)}}{Z^3}\right)\\
<&Pr\left(\omega^{3*}(t) = \omega_1^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\frac{2}{e^{\epsilon\omega_2^3(t)}}\\
+&Pr\left(\omega^{3*}(t) = \omega_2^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\frac{2}{e^{\epsilon\omega_1^3(t)}}\\
<&Pr\left(\omega^{3*}(t) = \omega_1^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\delta + Pr\left(\omega^{3*}(t) = \omega_2^3(t)\right)(1-\rho_1)\rho_{12}\rho_2\delta\\
=&(1-\rho_1)\rho_{12}\rho_2\delta
\end{aligned}
\tag{4.67}
$$

where $\delta := \max\left(\frac{2}{e^{\epsilon\omega_1^3(t)}}, \frac{2}{e^{\epsilon\omega_2^3(t)}}\right)$.then we have:

$$\begin{aligned} &Pr\left\{\sum_{i\in\acute{\mathbf{y}}^3}\omega_i^3(t) \geq (1-\epsilon)\omega^{3*}(t)|\mathbf{s}(t)=(0,\ 1,\ 1)\right\} \\ =&1-Pr\left\{\sum_{i\in\acute{\mathbf{y}}^3}\omega_i^3(t) < (1-\epsilon)\omega^{3*}(t)|\mathbf{s}(t)=(0,\ 1,\ 1)\right\} \\ &\geq 1-\frac{(1-\rho_1)\rho_{12}\rho_2\delta}{(1-\rho_1)\rho_{12}\rho_2} = 1-\delta \end{aligned} \tag{4.68}$$

- $\mathbf{s}(t) \in \{(1,\ 0,\ 0),(1,\ 1,\ 0)\}$

$$\begin{aligned} &Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{46}}\omega_i^{46}(t) < (1-\epsilon)\omega^{46*}(t)\right\} \\ =&\pi\left(\acute{\mathbf{y}}^{46}=(0\ 0);(0,\ 0,\ 1)\ or\ (1,\ 1,\ 0)\right) = \pi\left(\acute{\mathbf{y}}^{1}=(0\ 0)\right)\rho_1(1-\rho_2) \\ =&\rho_1(1-\rho_2)\frac{1}{1+e^{w_2^{46}}(t)} < \rho_1(1-\rho_2)\frac{1}{e^{w_2^{46}}(t)} \\ =&\rho_1(1-\rho_2)\delta \end{aligned} \tag{4.69}$$

then we have:

$$\begin{aligned} &Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{46}}\omega_i^{46}(t) \geq (1-\epsilon)\omega^{46*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 0),(1,\ 1,\ 0)\}\right\} \\ =&1-Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{46}}\omega_i^{46}(t) < (1-\epsilon)\omega^{46*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 0),(1,\ 1,\ 0)\}\right\} \\ &\geq 1-\frac{\rho_1(1-\rho_2)\delta}{\rho_1(1-\rho_2)} = 1-\delta \end{aligned} \tag{4.70}$$

- $\mathbf{s}(t) \in \{(1,\ 0,\ 1), (1,\ 1,\ 1)\}$

$$\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{57}} \omega_i^{57}(t) < (1-\epsilon)\omega^{57*}(t)\right\} \\
=&Pr\left(\omega^{57*}(t)=\omega_1^{57}(t)\right)[\ \pi\left(\acute{\mathbf{y}}^{57}=(0\ 0);(1,\ 0,\ 1)\ or\ (1,\ 1,\ 1)\right) \\
&\ +\pi\left(\acute{\mathbf{y}}^{57}=(0\ 1);(1,\ 0,\ 1)\ or\ (1,\ 1,\ 1)\right)\ ] \\
+&Pr\left(\omega^{57*}(t)=\omega_2^{57}(t)\right)[\ \pi\left(\acute{\mathbf{y}}^{57}=(0\ 0);(1,\ 0,\ 1)\ or\ (1,\ 1,\ 1)\right) \\
&\ +\pi\left(\acute{\mathbf{y}}^{57}=(1\ 0);(1,\ 0,\ 1)\ or\ (1,\ 1,\ 1)\right)\ ] \\
=&Pr\left(\omega^{57*}(t)=\omega_1^{57}(t)\right)\rho_1\rho_2\left[\pi\left(\acute{\mathbf{y}}^{57}=(0\ 0)\right)+\pi\left(\acute{\mathbf{y}}^{57}=(0\ 1)\right)\right] \\
+&Pr\left(\omega^{57*}(t)=\omega_2^{57}(t)\right)\rho_1\rho_2\left[\pi\left(\acute{\mathbf{y}}^{57}=(0\ 0)\right)+\pi\left(\acute{\mathbf{y}}^{57}=(1\ 0)\right)\right] \\
=&Pr\left(\omega^{57*}(t)=\omega_1^{57}(t)\right)\rho_1\rho_2\left(\frac{1+e^{\omega_2^{57}(t)}}{Z^{57}}\right) \\
+&Pr\left(\omega^{57*}(t)=\omega_2^{57}(t)\right)\rho_1\rho_2\left(\frac{1+e^{\omega_1^{57}(t)}}{Z^{57}}\right) \\
<&Pr\left(\omega^{57*}(t)=\omega_1^{57}(t)\right)\rho_1\rho_2\frac{2}{e^{\epsilon\omega_2^{57}(t)}} \\
+&Pr\left(\omega^{57*}(t)=\omega_2^{57}(t)\right)\rho_1\rho_2\frac{2}{e^{\epsilon\omega_1^{57}(t)}} \\
<&Pr\left(\omega^{57*}(t)=\omega_1^{57}(t)\right)\rho_1\rho_2\delta+Pr\left(\omega^{57*}(t)=\omega_2^{57}(t)\right)\rho_1\rho_2\delta \\
=&\rho_1\rho_2\delta
\end{aligned} \tag{4.71}$$

where $\delta := \max\left(\frac{2}{e^{\epsilon\omega_1^{57}(t)}}, \frac{2}{e^{\epsilon\omega_2^{57}(t)}}\right)$. Then we have:

$$\begin{aligned}
&Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{57}} \omega_i^{57}(t) \geq (1-\epsilon)\omega^{57*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 1),(1,\ 1,\ 1)\}\right\} \\
=&1-Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{57}} \omega_i^{57}(t) < (1-\epsilon)\omega^{57*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 1),(1,\ 1,\ 1)\}\right\} \\
\geq&1-\frac{\rho_1\rho_2\delta}{\rho_1\rho_2}=1-\delta
\end{aligned} \tag{4.72}$$

- $\mathbf{s}(t) = (0,\ 0,\ 0)$. Note that when $\mathbf{s}(t) = (0,\ 0,\ 0)$, $\omega^*(t) = \omega_1(t) = \omega_2(t) = 0$ and $\mathbf{x}(t) = (0\ 0)$. So we can write:

$$Pr\left\{\sum_{i\in\mathbf{x}} \omega_i(t) \geq (1-\epsilon)\omega^*(t)|\mathbf{s}(t) = (0,\ 0,\ 0)\right\} = 1 \tag{4.73}$$

Next we evaluate following term using (4.64, 4.66, 4.68, 4.70, 4.72, 4.73):

$$\begin{aligned}
&Pr\left\{\sum_{i\in\mathbf{x}}\omega_i(t)\geq(1-\epsilon)\omega^*(t)\right\}\\
&=Pr\left\{\sum_{i\in\mathbf{x}}\omega_i(t)\geq(1-\epsilon)\omega^*(t)|\mathbf{s}(t)=(0,\ 0,\ 0)\right\}Pr\left\{\mathbf{s}(t)=(0,\ 0,\ 0)\right\}\\
&+Pr\left\{\sum_{i\in\acute{\mathbf{y}}^1}\omega_i^1(t)\geq(1-\epsilon)\omega^{1*}(t)|\mathbf{s}(t)=(0,\ 0,\ 1)\right\}Pr\left\{\mathbf{s}(t)=(0,\ 0,\ 1)\right\}\\
&+Pr\left\{\sum_{i\in\acute{\mathbf{y}}^2}\omega_i^1(t)\geq(1-\epsilon)\omega^{2*}(t)|\mathbf{s}(t)=(0,\ 1,\ 0)\right\}Pr\left\{\mathbf{s}(t)=(0,\ 1,\ 0)\right\}\\
&+Pr\left\{\sum_{i\in\acute{\mathbf{y}}^3}\omega_i^3(t)\geq(1-\epsilon)\omega^{3*}(t)|\mathbf{s}(t)=(0,\ 1,\ 1)\right\}Pr\left\{\mathbf{s}(t)=(0,\ 1,\ 1)\right\}\\
&+Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{46}}\omega_i^{46}(t)\geq(1-\epsilon)\omega^{46*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 0),(1,\ 1,\ 0)\}\right\}\\
&\times Pr\left\{\mathbf{s}(t)\in\{(1,\ 0,\ 0),(1,\ 1,\ 0)\}\right\}\\
&+Pr\left\{\sum_{i\in\acute{\mathbf{y}}^{57}}\omega_i^{57}(t)\geq(1-\epsilon)\omega^{57*}(t)|\mathbf{s}(t)\in\{(1,\ 0,\ 1),(1,\ 1,\ 1)\}\right\}\\
&\times Pr\left\{\mathbf{s}(t)\in\{(1,\ 0,\ 1),(1,\ 1,\ 1)\}\right\}\\
&=1-\delta\left(1-(1-\rho_1)(1-\rho_{12})(1-\rho_2)\right)=1-\acute{\delta}
\end{aligned}\tag{4.74}$$

Then by Theorem 7, Algorithm 3 is throughput optimal. □

Chapter 5

Numerical Results

In this section we evaluate performance of our algorithm against Q-CSMA, MWS and simple 802.11 algorithms. Q-CSMA and MWS algorithms were explained in chapter 2 in detail. For 802.11 algorithm we adapt a simple version of 802.11 defined in [33]. Simple 802.11 is described in Algorithm 4.

Algorithm 4 Simple 802.11

```
At each time slot, each node $i$ does the following procedure.
1: User $i$ selects a random backoff time $T_i = Uniform\,[1, W]$ and waits for $T_i$ control mini slots.
2: If user $i$ hears an RESV message from a link in $\mathcal{C}(i)$ before the $(T_i + 1)$-th control mini-slot, it will not be included in the transmission schedule $\mathbf{x}(t)$ and will not transmit an RESV message. User $i$ will set $x_i(t) = 0$
3: If user $i$ does not hear an RESV message from any user in $\mathcal{C}(i)$ before the $(T_i+1)$-th control mini-slot, it will send an RESV message to all links in $\mathcal{C}(i)$ at the beginning of $(T_i + 1)$-th control mini-slot.
     • If there is a collision, user $i$ will set $x_i(t) = 0$.
     • If there is no collision, user $i$ will set $x_i(t) = 1$.
4: If $x_i(t) = 1$, user $i$ will transmit a packet in the data slot. (Links with empty queues will keep silent in this time slot.)
```

We compare the algorithms in terms of average sum of queues in the network and also average queue size evolution of individual queues. Averages are with respect to sample path and are taken in 10 different sample paths. We divide this section to three. In section 5.1, we consider the network with one PU and one SU where only PU has fading channel. In section 5.2, We consider the network with one PU and $N - 1$ SUs where only PU has a fading channel. In section 5.3 we consider the network with one PU, one SU where all three channels are fading.

5.1 One PU with fading channel and one SU

Consider the network with one PU and one SU. PU's direct channel is fading with $Pr\left(s_1(t) = 1\right) = 0.4$. The capacity region for this network is calculated in Theorem 1. We pick a point $\left(\lambda_1^b,\ \lambda_2^b\right)$ on the boundary of the capacity region and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$. Sum of the queue sizes in the network is averaged over 10 sample paths. Queues are evolved over 10^5 time slots. In Figure 5.1, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.7,\ 0)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. In Figure 5.2, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.6,\ 0.2)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. In Figure 5.3, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.5,\ 0.4)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. It can be seen that MQ-CSMA1 has a better performance over Q-CSMA and simple 802.11.

We investigate the comparison further by plotting PU's queue evolution in time, averaged over 20 sample paths. In Figure 5.4 expected queue evolution of PU under MQ-CSMA1, Q-CSMA and 802.11 for $(\lambda_1,\ \lambda_2) = (0.45,\ 0.35)$, strictly inside and close to the capacity region is depicted. It shows that under MQ-CSMA1, average queue size of PU does not grow with time suggesting that the PU is stable, while under Q-CSMA and 802.11 average queue size of PU increases in time suggesting that PU queue is unstable.

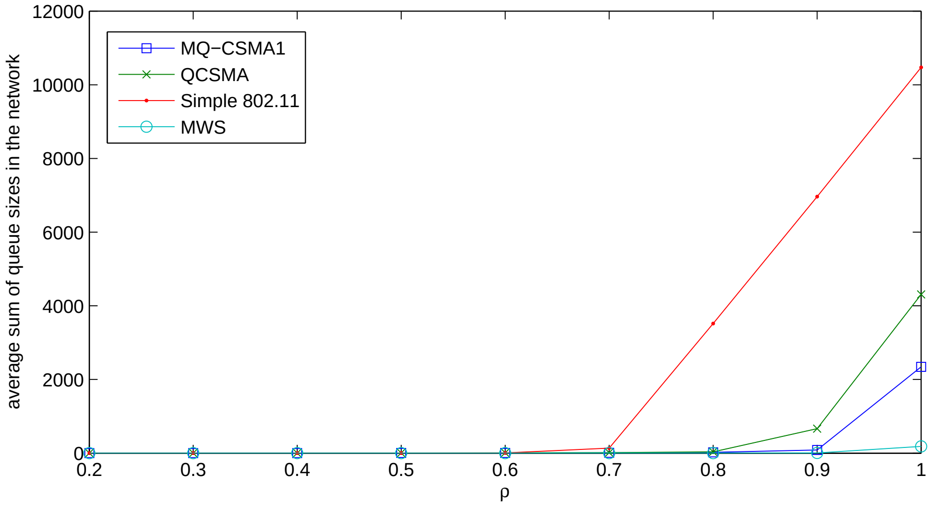


FIGURE 5.1: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.7,\ 0)$

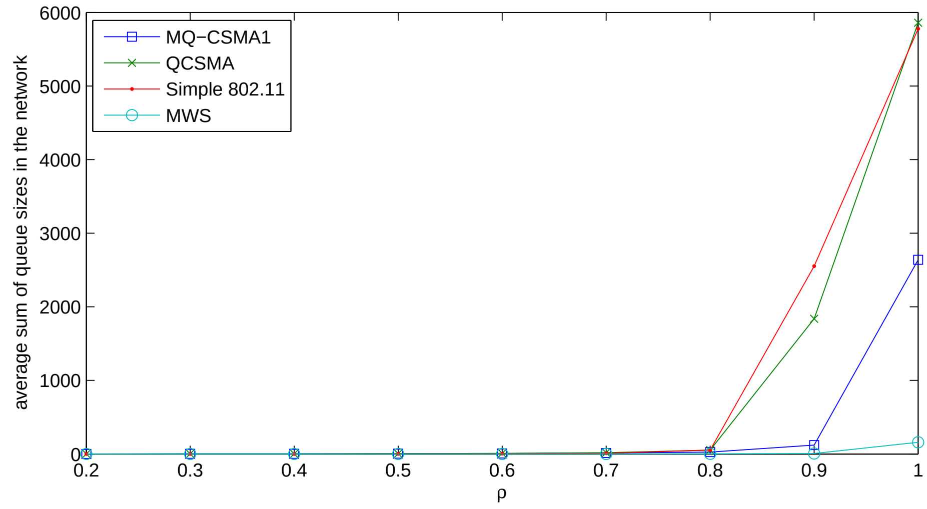


FIGURE 5.2: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.6,\ 0.2)$

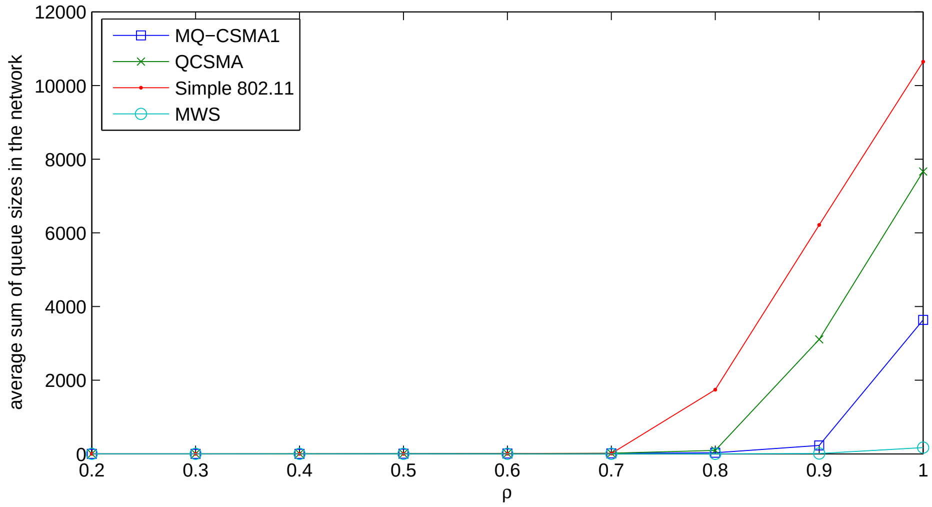


FIGURE 5.3: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.5,\ 0.4)$

5.2 One PU with fading channel and $N-1$ SUs

In this section we consider our cooperative network model with 5 users (i.e., $N = 5$), consisting of a PU and 4 secondary users. We assume that only PU has a fading channel with $Pr\left(s_1(t) = 1\right) = 0.4$. Although, the exact capacity region of the Network in unknown for this case, it can be seen that $\lambda_1 < 0.7,\ \ \lambda_i = 0$, for $i \in \{1, 2, 3, 4\}$ is inside the capacity region. Because given $\lambda_i = 0$, $\lambda_1 < 0.4+0.5(1-0.4) = 0.7$. We pick a point $\boldsymbol{\lambda}^b$ on the boundary of the capacity region and increase $\boldsymbol{\lambda}$ from $\rho_{min}\boldsymbol{\lambda}^b$ to $\rho_{max}\boldsymbol{\lambda}^b$. Sum of the queue sizes in the network is averaged over 10 sample paths. Queues are evolved over 10^5 time slots. In Figure 5.5, we consider $\boldsymbol{\lambda}^b = (0.7,\ 0,\ 0,\ 0,\ 0)$ and increase $\boldsymbol{\lambda}$

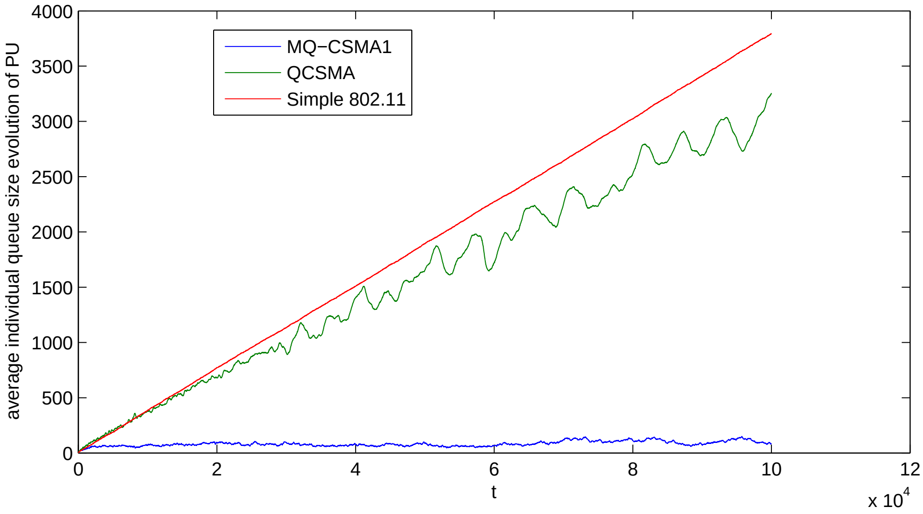


FIGURE 5.4: Average individual queue evolution of PU with $(\lambda_1,\ \lambda_2) = (0.45,\ 0.35)$

from $\rho_{min}\boldsymbol{\lambda}^b$ to $\rho_{max}\boldsymbol{\lambda}^b$ for $\rho_{min} = 0.5$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network.

We investigate the comparison further by plotting PU's queue evolution in time, averaged over 20 sample paths. In Figure 5.6 expected queue evolution of PU under MQ-CSMA1 and Q-CSMA for $\boldsymbol{\lambda} = (0.66,\ 0,\ 0,\ 0,\ 0)$, strictly inside and close to the capacity region is depicted. It shows that under MQ-CSMA1, average queue size of PU does not grow with time suggesting that the PU is stable, while under Q-CSMA average queue size of PU increases in time suggesting that PU queue is unstable.

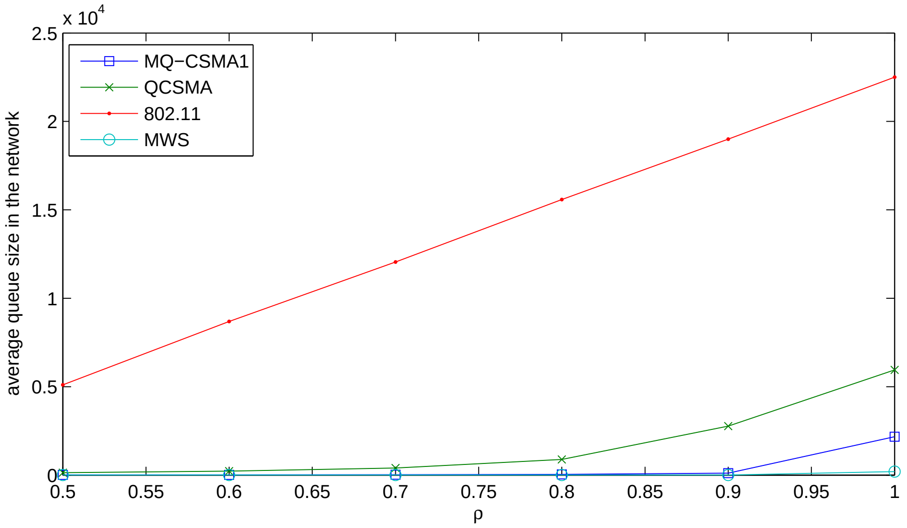


FIGURE 5.5: Average sum of queue sizes in the network with $\boldsymbol{\lambda}^b = (0.7,\ 0,\ 0,\ 0,\ 0)$

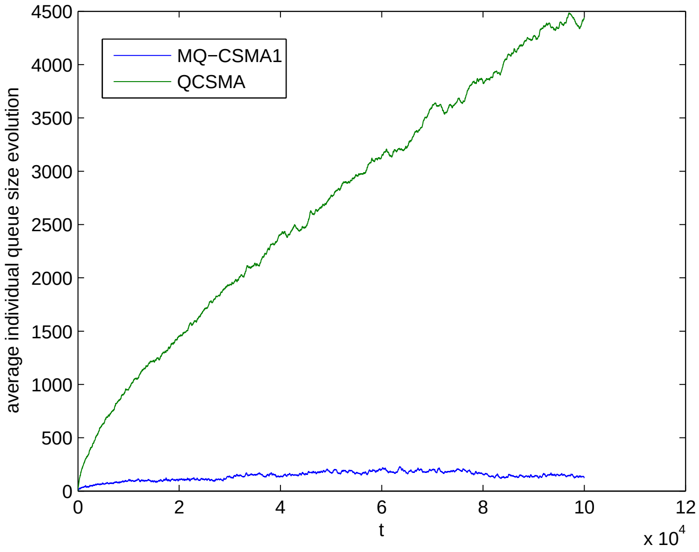


Figure 5.6: Average individual queue evolution of PU with $\boldsymbol{\lambda} = (0.66,\ 0,\ 0,\ 0,\ 0)$

5.3 One PU, One SU, multiple fading channels

Consider the network with one PU and one SU where all the links $PU - D$, $PU - SU$ and $SU - D$ has fading characteristics with $Pr\left(s_1(t) = 1\right) = \rho_1$, $Pr\left(s_{12}(t) = 1\right) = \rho_{12}$ and $Pr\left(s_2(t) = 1\right) = \rho_2$. Section 5.3.1, 5.3.2 and 5.3.3 compares performance of MQ-CSMA2, QCSMA, simple 802.11 and MWS for $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$, $\frac{\rho_{12}}{1-\rho_{12}} \leq \rho_2 < \frac{\rho_{12}}{1-\rho_{12}}$ and $\rho_2 \geq \frac{\rho_{12}}{1-\rho_{12}}$ respectively. Capacity of the network when $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$, $\frac{\rho_{12}}{1-\rho_{12}} \leq \rho_2 < \frac{\rho_{12}}{1-\rho_{12}}$ and $\rho_2 \geq \frac{\rho_{12}}{1-\rho_{12}}$ is calculated in Theorems 2, 3 and 4 respectively.

5.3.1 $\rho_2 < \frac{\rho_{12}}{1+\rho_{12}}$

We consider $\rho_1 = 0.4$, $\rho_{12} = 0.8$ and $\rho_1 = 0.4$. We pick a point $\left(\lambda_1^b,\ \lambda_2^b\right)$ on the boundary of the capacity region and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$. Sum of the queue sizes in the network is averaged over 10 sample paths. Queues are evolved over 10^5 time slots. In Figure 5.7, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.64,\ 0)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network . In Figure 5.8, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.54,\ 0.1)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. In Figure 5.9, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.44,\ 0.2)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. It can be seen that MQ-CSMA1 has a better performance over Q-CSMA and simple 802.11.

We investigate the comparison further by plotting PU's queue evolution in time, averaged over 20 sample paths. In Figure 5.10 expected queue evolution of PU under MQ-CSMA1, Q-CSMA and 802.11 for $(\lambda_1,\ \lambda_2) = (0.63,\ 0)$, strictly inside and close to the capacity region is depicted. It shows that under MQ-CSMA1, average queue size of PU does not grow with time suggesting that the PU is stable, while under Q-CSMA and 802.11 average queue size of PU increases in time suggesting that PU queue is unstable.

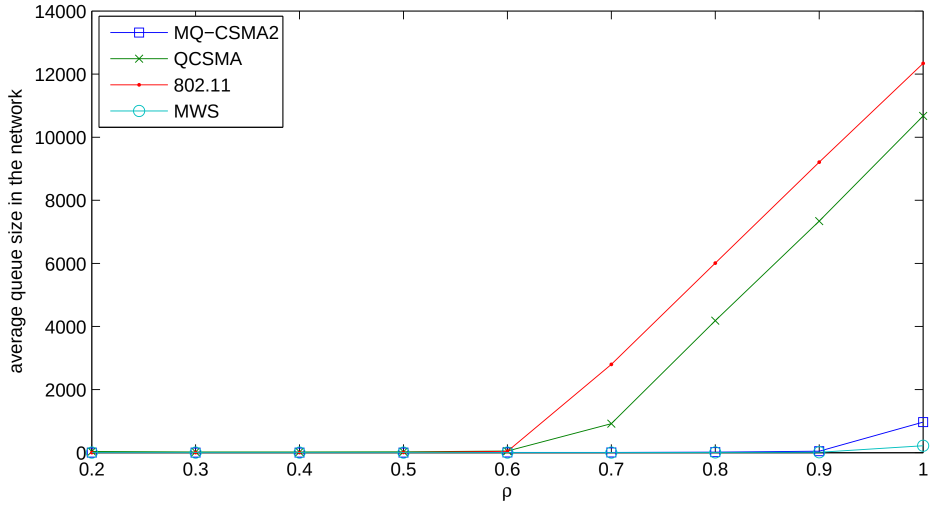


FIGURE 5.7: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.64,\ 0)$

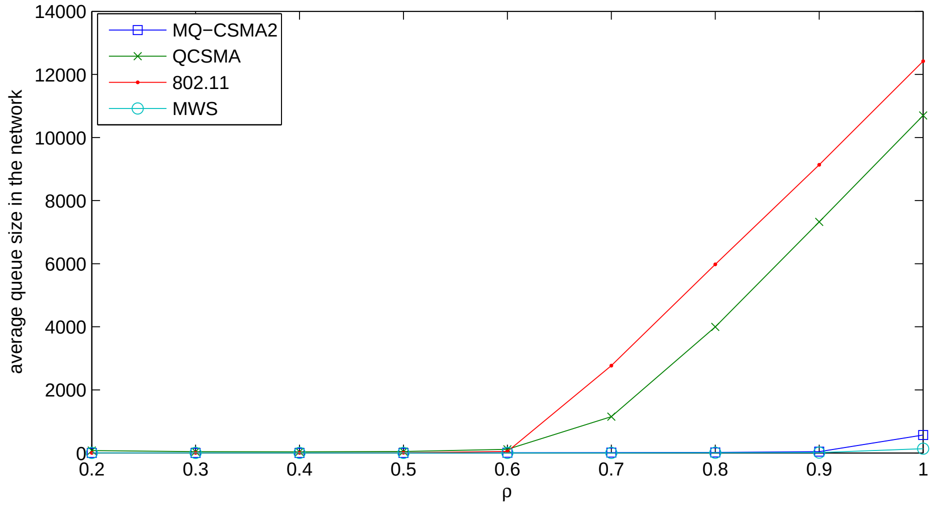


FIGURE 5.8: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.54,\ 0.1)$

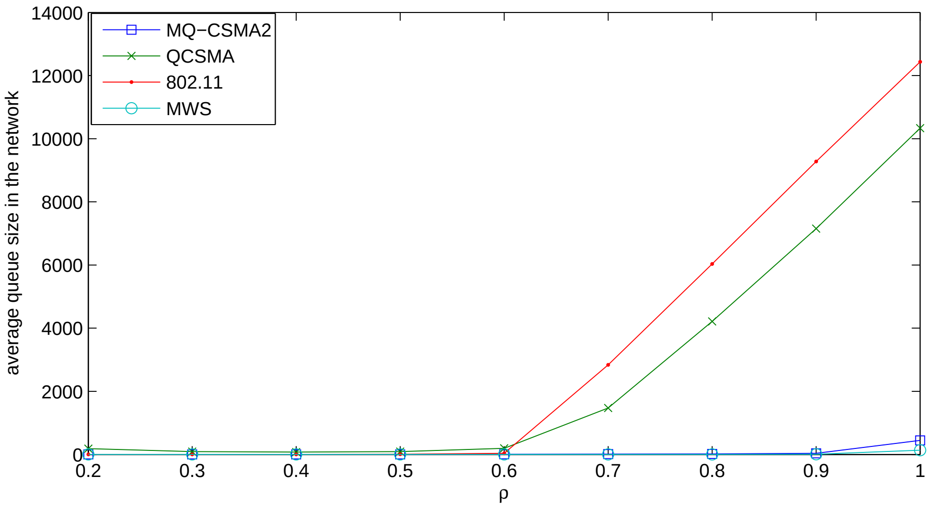


FIGURE 5.9: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.44,\ 0.2)$

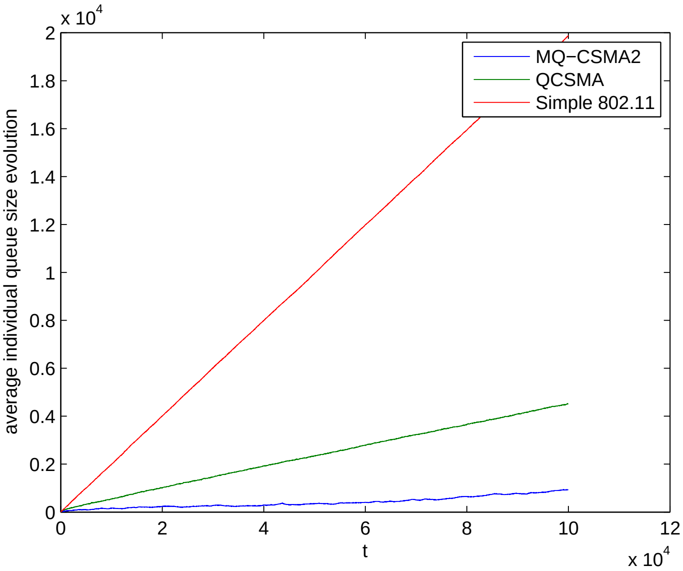


FIGURE 5.10: Average individual queue evolution of PU with $(\lambda_1,\ \lambda_2) = (0.63,\ 0)$

5.3.2 $\frac{\rho_{12}}{1-\rho_{12}} \leq \rho_2 < \frac{\rho_{12}}{1-\rho_{12}}$

We consider $\rho_1 = 0.4$, $\rho_{12} = 0.5$ and $\rho_1 = 0.8$. We pick a point $\left(\lambda_1^b,\ \lambda_2^b\right)$ on the boundary of the capacity region and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$. Sum of the queue sizes in the network is averaged over 10 sample paths. Queues are evolved over 10^5 time slots. In Figure 5.11, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.67,\ 0)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network . In Figure 5.12, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.62,\ 0.1)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$

and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. In Figure 5.13, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.5,\ 0.3)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. It can be seen that MQ-CSMA1 has a better performance over Q-CSMA and simple 802.11.

We investigate the comparison further by plotting PU's queue evolution in time, averaged over 20 sample paths. In Figure 5.14 expected queue evolution of PU under MQ-CSMA1, Q-CSMA and 802.11 for $(\lambda_1,\ \lambda_2) = (0.66,\ 0)$, strictly inside and close to the capacity region is depicted. It shows that under MQ-CSMA1, average queue size of PU does not grow with time suggesting that the PU is stable, while under Q-CSMA and 802.11 average queue size of PU increases in time suggesting that PU queue is unstable.

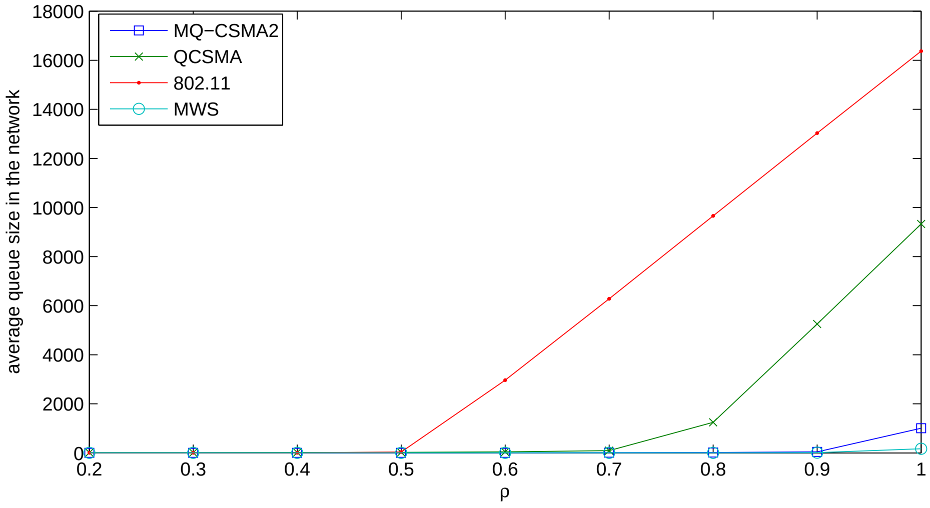


FIGURE 5.11: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.67,\ 0)$

5.3.3 $\rho_2 \geq \frac{\rho_{12}}{1-\rho_{12}}$

We consider $\rho_1 = 0.4$, $\rho_{12} = 0.5$ and $\rho_1 = 0.8$. We pick a point $\left(\lambda_1^b,\ \lambda_2^b\right)$ on the boundary of the capacity region and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$. Sum of the queue sizes in the network is averaged over 10 sample paths. Queues are evolved over 10^5 time slots. In Figure 5.15, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.64,\ 0)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network . In Figure 5.16, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.56,\ 0.2)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues in the network. In Figure 5.17, we consider $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.51,\ 0.3)$ and increase $(\lambda_1,\ \lambda_2)$ from $\rho_{min}\left(\lambda_1^b,\ \lambda_2^b\right)$ to $\rho_{max}\left(\lambda_1^b,\ \lambda_2^b\right)$ for $\rho_{min} = 0.2$ and $\rho_{max} = 1$ and plot the average queue size of all queues

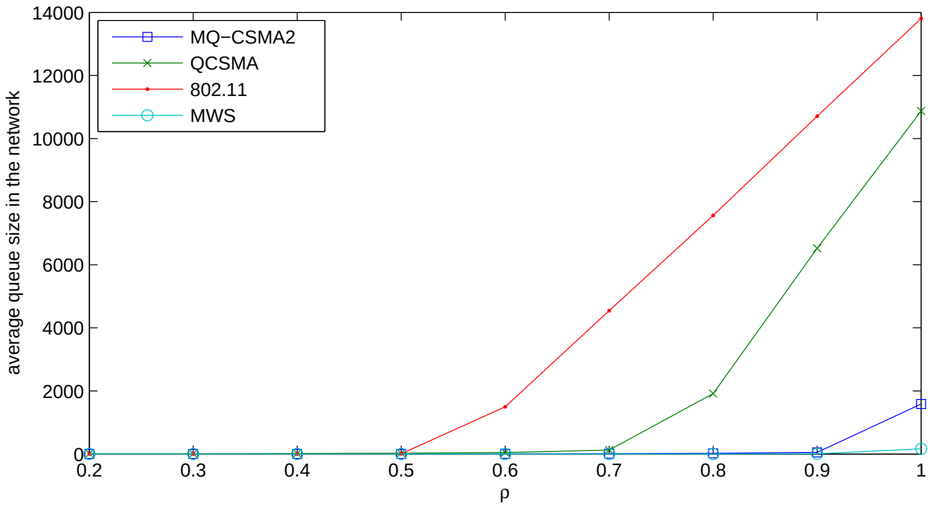


FIGURE 5.12: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.62,\ 0.1)$

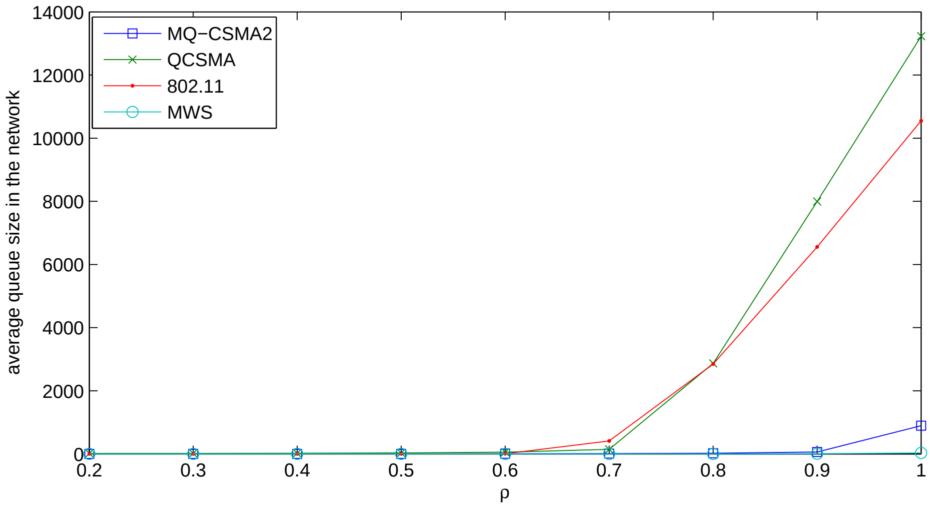


FIGURE 5.13: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.5,\ 0.3)$

in the network. It can be seen that MQ-CSMA1 has a better performance over Q-CSMA and simple 802.11.

We investigate the comparison further by plotting PU's queue evolution in time, averaged over 20 sample paths. In Figure 5.18 expected queue evolution of PU under MQ-CSMA1, Q-CSMA and 802.11 for $(\lambda_1,\ \lambda_2) = (0.63,\ 0)$, strictly inside and close to the capacity region is depicted. It shows that under MQ-CSMA1, average queue size of PU does not grow with time suggesting that the PU is stable, while under Q-CSMA and 802.11 average queue size of PU increases in time suggesting that PU queue is unstable.

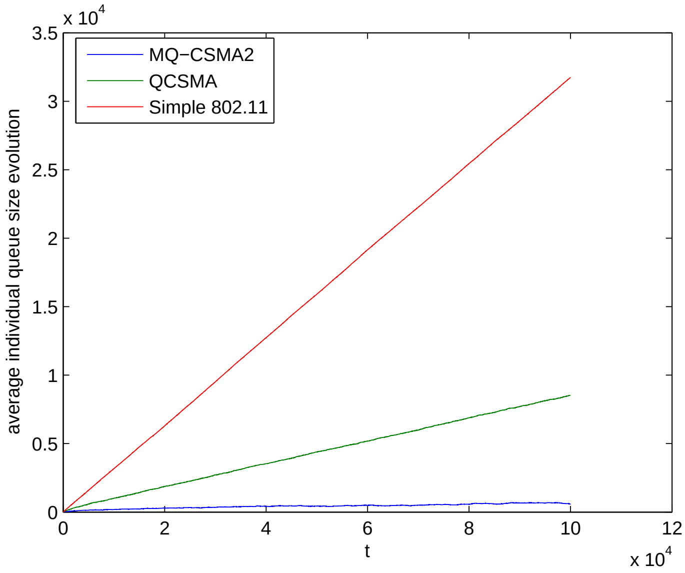


Figure 5.14: Average individual queue evolution of PU with $(\lambda_1,\ \lambda_2) = (0.66,\ 0)$

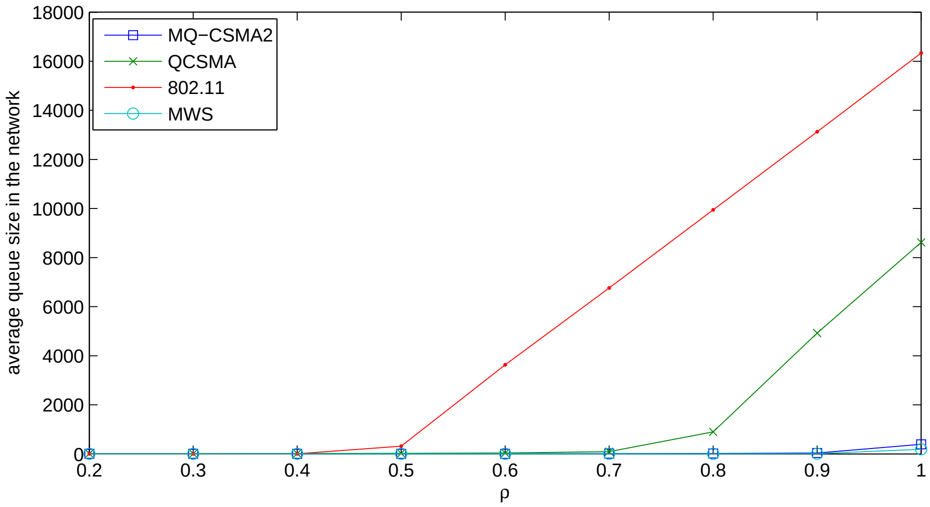


Figure 5.15: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.64,\ 0)$

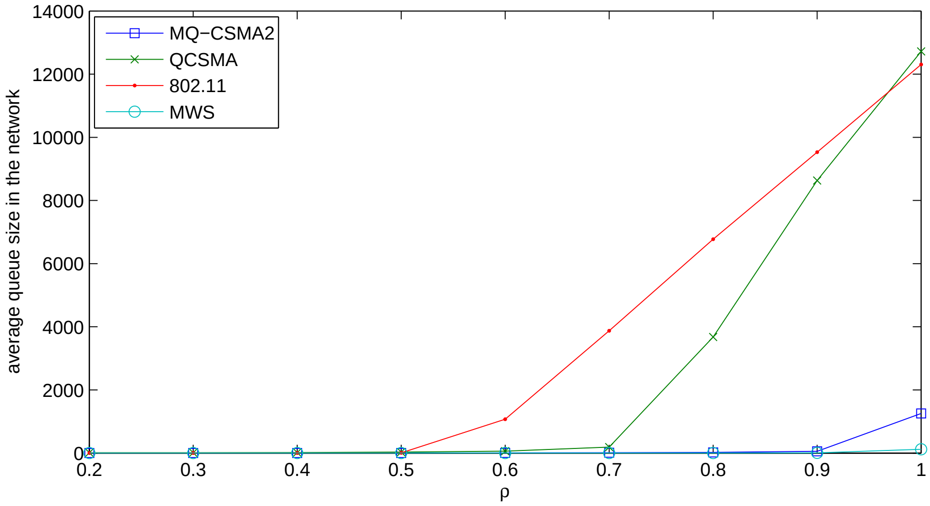


FIGURE 5.16: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.56,\ 0.2)$

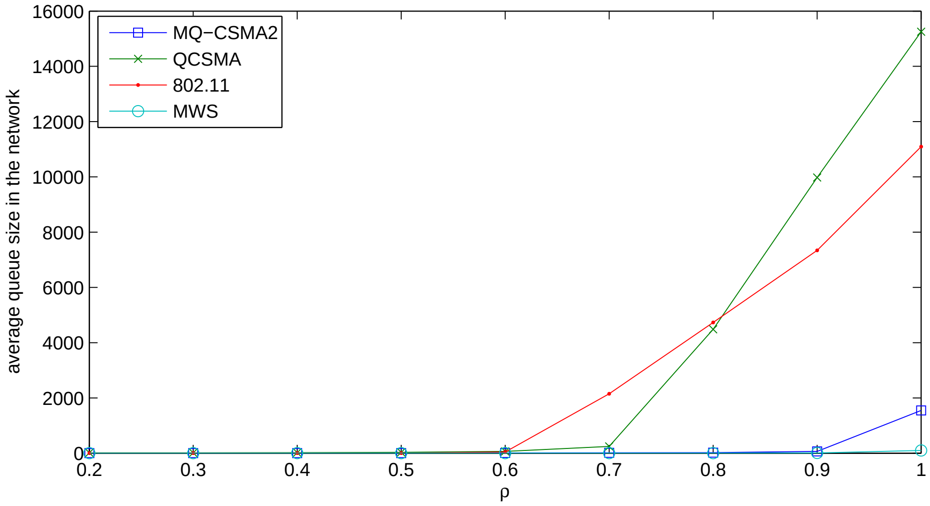


FIGURE 5.17: Average sum of queue sizes in the network with $\left(\lambda_1^b,\ \lambda_2^b\right) = (0.51,\ 0.3)$

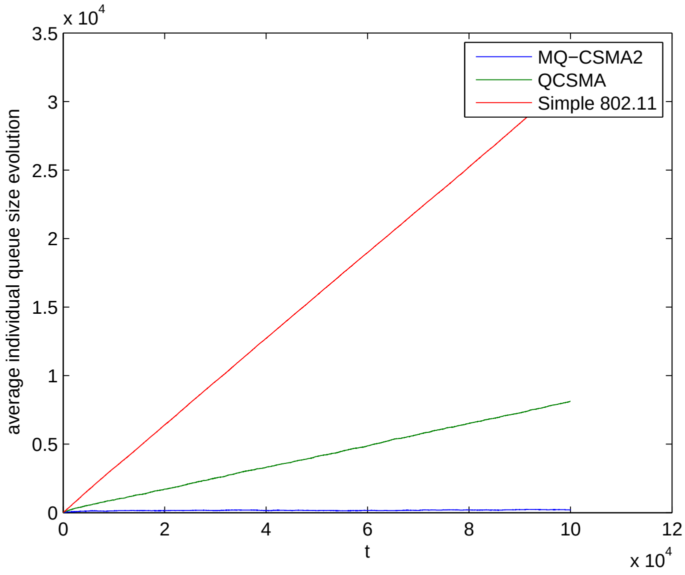


FIGURE 5.18: Average individual queue evolution of PU with $(\lambda_1,\ \lambda_2) = (0.63,\ 0)$

Chapter 6

Conclusions and Future Works

In this thesis we investigated problem of optimal resource allocation in a cooperative network, where SUs gain access to a licensed channel by cooperating with the channel owner by relaying its packets. We derived Maximum Weight Scheduling for our network model for two different scenarios. First scenario includes a PU and $(N-1)$ SUs in which only PU has a fading channel. We derive the exact capacity region of the network for $N = 2$ and establish the superiority of the cooperative network over the non-cooperative one in terms of capacity region. In the second scenario, we assume the cooperative network with one PU and one SU in which all the links between PU-D, PU-SU and SU-D suffer from fading. We also derive the MWS algorithm for this scenario and calculate the exact capacity region of the network. Capacity region of the network in case of multiple fading channels also shows that cooperative network is superior in terms of the capacity region. Our findings shows that SU does not need to have a better channel statistics in order to be able to improve PU performance. In fact for every possible set of $(\rho_1,\ \rho_{12},\ \rho_2)$, it is beneficial for PU to let the SU enter the cooperative network.

Maximum Weight Scheduling can achieve throughput optimality by exploiting opportunistic gain in general network topology with fading channels. Despite the advantage of opportunistic scheduling, this mechanism requires that the existing central scheduler is aware of network conditions such as channel state and queue length information of users. To solve this problem we propose a distributed algorithm based on CSMA that only requires individual queue size of each user. Existing throughput optimal algorithms in non-fading algorithms such as Q-CSMA are not suitable for fading environments, because they fail to exploit opportunistic gain in fading environments. By taking different network topologies associated with the fading states into the account, for each scenario we develop our algorithm and prove throughput optimality of the algorithm.

Numerical results show consistency with our analytical results in terms of stability of the network in points close to the boundary of the capacity region. Also, numerical results show that under Q-CSMA, individual queue size, grows in time, which suggests that Q-CSMA is not throughput optimal.

We conclude the thesis by briefly presenting some ideas which will motivate future studies on this topic. In chapter 4 we assume that in the case of multiple fading channels, users have a perfect knowledge of instantaneous channel states. However, in practice, acquiring channel states costs in terms of power and time slot efficiency. It would be interesting to design an algorithm which does not require all the channel states. Also in chapter 3 we give performance guarantees for the case of two users. However complete impact of cooperation for cases of more than two users are not investigated. Hence, investigating performance guarantees for general case would be of interest.

Bibliography

[1] E. C. Van Der Meulen. Three-terminal communication channels. 1971.

[2] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i. system description. *Communications, IEEE Transactions on*, 51(11):1927–1938, Nov 2003. ISSN 0090-6778.

[3] S. Mohajer, S.N. Diggavi, and D.N.C. Tse. Approximate capacity of a class of gaussian relay-interference networks. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 31–35, June 2009.

[4] Nikolaos Pappas, Jeongho Jeon, Anthony Ephremides, and Apostolos Traganitis. Wireless network-level partial relay cooperation. In *ISIT*, pages 1122–1126, 2012.

[5] S. Kompella, Gam D. Nguyen, J.E. Wieselthier, and Anthony Ephremides. Stable throughput tradeoffs in cognitive shared channels with cooperative relaying. In *INFOCOM, 2011 Proceedings IEEE*, pages 1961–1969, April 2011.

[6] Fatemeh Afghah, Abolfazl Razi, and Ali Abedi. Stochastic game theoretical model for packet forwarding in relay networks. *Telecommunication Systems*, 52(4):1877–1893, 2013. ISSN 1018-4864.

[7] F. Afghah, A. Razi, and A. Abedi. Throughput optimization in relay networks using markovian game theory. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 1080–1085, March 2011.

[8] L. Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on*, 37(12):1936–1948, Dec 1992. ISSN 0018-9286.

[9] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1990. ISBN 0716710455.

[10] G. Sharma, N.B. Shroff, and R.R. Mazumdar. Maximum weighted matching with interference constraints. In *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5 pp.–74, March 2006.

[11] M. Karaca, Y. Sarikaya, O. Ercetin, T. Alpcan, and H. Boche. Joint opportunistic scheduling and selective channel feedback. *Wireless Communications, IEEE Transactions on*, 12(6):3024–3034, June 2013. ISSN 1536-1276.

[12] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of Algorithms*, 7(4): 567 – 583, 1986. ISSN 0196-6774.

[13] M Luby. A simple parallel algorithm for the maximal independent set problem. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 1–10, New York, NY, USA, 1985. ACM. ISBN 0-89791-151-2.

[14] Prasanna Chaporkar, Koushik Kar, and Saswati Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *In Proceedings of 43d Annual Allerton Conference on Communication, Control and Computing*, pages 28–30, 2005.

[15] S. Sarkar and K. Kar. Queue length stability in trees under slowly convergent traffic using sequential maximal scheduling. *Automatic Control, IEEE Transactions on*, 53(10):2292–2306, Nov 2008. ISSN 0018-9286.

[16] Xinzhou Wu, R. Srikant, and J.R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *Mobile Computing, IEEE Transactions on*, 6(6):595–605, June 2007. ISSN 1536-1233.

[17] Antonis Dimakis and Jean Walrand. Sufficient conditions for stability of longest-queue-first scheduling: second-order properties using fluid limits. *Advances in Applied Probability*, 38(2):505–521, 06 2006.

[18] M.M. Halldórsson and J. Radhakrishnan. Greed is good: Approximating independent sets in sparse and bounded-degree graphs. *Algorithmica*, 18(1):145–163, 1997. ISSN 0178-4617.

[19] Nicholas William McKeown. Scheduling algorithms for input-queued cell switches. Technical report, 1995.

[20] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman, and Y. Zwols. Analyzing the performance of greedy maximal scheduling via local pooling and graph

theory. *Networking, IEEE/ACM Transactions on*, 20(1):163–176, Feb 2012. ISSN 1063-6692.

[21] Changhee Joo. A local greedy scheduling scheme with provable performance guarantee. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '08, pages 111–120, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-073-9.

[22] Xiaojun Lin and N.B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1804–1814 vol. 3, March 2005.

[23] Xiaojun Lin and S.B. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1258–1263, Dec 2006.

[24] Eytan Modiano, Devavrat Shah, and Gil Zussman. Maximizing throughput in wireless networks via gossiping. *SIGMETRICS Perform. Eval. Rev.*, 34(1):27–38, June 2006. ISSN 0163-5999.

[25] Yung Yi, G. de Veciana, and S. Shakkottai. Learning contention patterns and adapting to load/topology changes in a mac scheduling algorithm. In *Wireless Mesh Networks, 2006. WiMesh 2006. 2nd IEEE Workshop on*, pages 23–32, Sept 2006.

[26] Sujay Sanghavi, Loc Bui, and R. Srikant. Distributed link scheduling with constant overhead. *SIGMETRICS Perform. Eval. Rev.*, 35(1):313–324, June 2007. ISSN 0163-5999.

[27] A. Eryilmaz, A. Ozdaglar, and E. Modiano. Polynomial complexity algorithms for full utilization of multi-hop wireless networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 499–507, May 2007.

[28] L. Tassiulas. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 533–539 vol.2, Mar 1998.

[29] Xiaojun Lin and S.B. Rasool. Constant-time distributed scheduling policies for ad hoc wireless networks. In *Decision and Control, 2006 45th IEEE Conference on*, pages 1258–1263, Dec 2006.

[30] A. Gupta, Xiaojun Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. *Networking, IEEE/ACM Transactions on*, 17(6): 1846–1859, Dec 2009. ISSN 1063-6692.

[31] Changhee Joo and N.B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. In *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, pages 1937–1941, Oct 2006.

[32] Libin Jiang and Jean Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. Technical report, 2009.

[33] Jian Ni, Bo Tan, and R. Srikant. Q-csma: Queue-length-based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks. *Networking, IEEE/ACM Transactions on*, 20(3):825–836, June 2012. ISSN 1063-6692.

[34] Changhee Joo. On random access scheduling for multimedia traffic in multihop wireless networks with fading channels. *Mobile Computing, IEEE Transactions on*, 12(4):647–656, April 2013. ISSN 1536-1233.

[35] Changhee Joo, Xiaojun Lin, Jiho Ryu, and Ness B. Shroff. Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels. In *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '13, pages 89–98, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2193-8.

[36] T. Cover and A.E. Gamal. Capacity theorems for the relay channel. *Information Theory, IEEE Transactions on*, 25(5):572–584, Sep 1979. ISSN 0018-9448.

[37] A.E. Gamal and M. Aref. The capacity of the semideterministic relay channel (corresp.). *Information Theory, IEEE Transactions on*, 28(3):536–536, May 1982. ISSN 0018-9448.

[38] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part ii. implementation aspects and performance analysis. *Communications, IEEE Transactions on*, 51(11):1939–1948, Nov 2003. ISSN 0090-6778.

[39] J.N. Laneman, D.N.C. Tse, and Gregory W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *Information Theory, IEEE Transactions on*, 50(12):3062–3080, Dec 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.838089.

[40] G. Kramer, M. Gastpar, and P. Gupta. Cooperative strategies and capacity theorems for relay networks. *Information Theory, IEEE Transactions on*, 51(9):3037–3063, Sept 2005. ISSN 0018-9448.

[41] A. Zaidi, Shivaprasad Kotagiri, J.N. Laneman, and L. Vandendorpe. Cooperative relaying with state available at the relay. In *Information Theory Workshop, 2008. ITW '08. IEEE*, pages 139–143, May 2008.

[42] A.K. Sadek, K.J.R. Liu, and A. Ephremides. Cognitive multiple access via cooperation: Protocol design and performance analysis. *Information Theory, IEEE Transactions on*, 53(10):3677–3696, Oct 2007. ISSN 0018-9448.

[43] Beiyu Rong and Anthony Ephremides. Protocol-level cooperation in wireless networks: Stable throughput and delay analysis. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009. 7th International Symposium on*, pages 1–10, June 2009.

[44] Mahmoud Ashour, Amr A. El-Sherif, Tamer A. ElBatt, and Amr Mohamed. Cooperative access in cognitive radio networks: Stable throughput and delay tradeoffs. *CoRR*, abs/1309.1200, 2013.

[45] Beiyu Rong and Anthony Ephremides. Cooperative access in wireless networks: Stable throughput and delay. *Information Theory, IEEE Transactions on*, 58(9): 5890–5907, Sept 2012. ISSN 0018-9448.

[46] Leonidas Georgiadis, Michael J, and Ros Tassiulas. Resource allocation and cross-layer control in wireless networks. In *Foundations and Trends in Networking*, pages 1–149, 2006.

[47] L. Tassiulas. *Dynamic link activation scheduling in multihop radio networks with fixed or changing topology.* PhD thesis, University of Maryland, College Park, 1991.

[48] M.J. Neely, E. Modiano, and C.E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 745–755 vol.1, March 2003.

[49] M.J. Neely. *Dynamic power allocation and routing for satellite and wireless networks with time varying channels.* PhD thesis, Massachusetts Institute of Technology, 2003.

[50] Alexander L. Stolyar. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1–53, 02 2004.

[51] L. Tassiulas. Scheduling and performance limits of networks with constantly changing topology. *Information Theory, IEEE Transactions on*, 43(3):1067–1073, May 1997. ISSN 0018-9448.

[52] A. Eryilmaz and R. Srikant. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1794–1803 vol. 3, March 2005.

[53] M.J. Neely, E. Modiano, and Chih ping Li. Fairness and optimal stochastic control for heterogeneous networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1723–1734 vol. 3, March 2005.

[54] M.J. Neely. Energy optimal control for time-varying wireless networks. *Information Theory, IEEE Transactions on*, 52(7):2915–2934, July 2006. ISSN 0018-9448.

[55] AlexanderL. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005. ISSN 0257-0130.

[56] Alexander L. Stolyar. On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation. *Oper. Res.*, 53(1):12–25, January 2005. ISSN 0030-364X.

[57] M. Leconte, Jian Ni, and R. Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. *Networking, IEEE/ACM Transactions on*, 19(3):709–720, June 2011. ISSN 1063-6692.

[58] M. Leconte, Jian Ni, and R. Srikant. Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks. *Networking, IEEE/ACM Transactions on*, 19(3):709–720, June 2011. ISSN 1063-6692.

[59] B. Li and A. Eryilmaz. On the limitations of randomization for queue-length-based scheduling in wireless networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 2597–2605, April 2011.

[60] B. Li and A. Eryilmaz. On the boundaries of randomization for throughput-optimal scheduling in switches. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 691–698, Sept 2010.

[61] Xin Liu, Edwin K. P. Chong, and Ness B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 41:451–474, 2003.

[62] SeYoung Yun, Jinwoo Shin, and Yung Yi. Medium access over time-varying channels with limited sensing cost. *CoRR*, abs/1206.5054, 2012. URL http://arxiv.org/abs/1206.5054.

[63] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication.* Cambridge University Press, New York, NY, USA, 2005. ISBN 0-5218-4527-0.

[64] Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems.* Morgan and Claypool Publishers, 2010. ISBN 160845455X, 9781608454556.

[65] A. Eryilmaz, R. Srikant, and J.R. Perkins. Stable scheduling policies for fading wireless channels. *Networking, IEEE/ACM Transactions on*, 13(2):411–424, April 2005. ISSN 1063-6692.

[66] S. Li, E. Ekici, and N. Shroff. Throughput-optimal queue length based csma/ca algorithm for cognitive radio networks. *Mobile Computing, IEEE Transactions on*, PP(99):1–1, 2014. ISSN 1536-1233.