STABILITY-BASED TOPOLOGY CONTROL IN WIRELESS MULTIHOP NETWORKS WITH RESERVATION-BASED DISTRIBUTED-SCHEDULING POLICIES

By

GUSTAVO VEJARANO

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2011

ACKNOWLEDGMENTS

I would like to express my gratitude to my doctoral advisor Prof. Janise McNair. Her support and encouragement were essential for the successful completion of my doctorate and dissertation. Working under her direction was truly inspiring and enjoyable. I thank all the members of the Wireless and Mobile Systems (WAMS) Laboratory for their valuable feedback for my research and for all the great times we had in the laboratory. My gratitude also goes to Professors Yuguang “Michael” Fang, John Shea, and Shigang Chen for serving in my committee.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

STABILITY-BASED TOPOLOGY CONTROL IN WIRELESS MULTIHOP NETWORKS
WITH RESERVATION-BASED DISTRIBUTED-SCHEDULING POLICIES

By

Gustavo Vejarano

August 2011

Chair: Janise McNair
Major: Electrical and Computer Engineering

The topology of wireless multihop networks can be controlled by means of transmission power control, and this control can be performed with the objective of adapting the network topology to the data flows established by the end users. By adapting the network topology, the set of flow data rates that the network supports can be increased, or, if the flow data rates are fixed, the end-to-end delays experienced by the flows can be decreased. The research problem studied in this dissertation is the design of topology-control algorithms that maximize the size of the set of supported flow data rates.

The design of the algorithms is approached in three steps. In the first step, the supported set of flow data rates is characterized mathematically for fixed network topologies by means of a queueing-system stability analysis. Two main contributions are achieved in this step. These are (1) a novel stability-analysis technique for reservation-based distributed scheduling (RBDS) policies and (2) the greedy-maximal RBDS (GM-RBDS) policy that outperforms, in terms of throughput, the current policies available in the literature. In the second step, the mathematical characterization of the set of flow data rates supported by GM-RBDS networks is used for the design of a heuristic and centralized topology-control algorithm which outperforms the classic approach based on spatial reuse. The third step consists of the design of distributed topology-control algorithms that also use the mathematical characterization of the set

of flow data rates. These algorithms are designed using game theory and compared with the centralized topology control of the second step. The network scenarios in which each of these two approaches (i.e., centralized and distributed) outperforms the other are identified.

Finally, this dissertation also includes the design, implementation, and evaluation of a simulation framework for Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 wireless mesh networks using optimized network engineering tools (OPNET). The GM-RBDS policy and the topology-control algorithms are evaluated in this framework. To the best of our knowledge, this is the first OPNET simulation framework for this type of networks.

CHAPTER 1
INTRODUCTION

1.1 Research Problem

The main problem to be solved is to control the topology of wireless multihop networks (WMNs) in order to maximize the performance of end-to-end data flows established on the networks. This performance is given in terms of the set of end-to-end data-packet rates that the network supports while guaranteeing stability (i.e., the link queues are positive recurrent). The topology is controlled by means of transmission-power control.

1.1.1 Assumptions

It is assumed that the node transmissions are omnidirectional. The interference model is such that a packet reception is successful only if no other node that covers the node receiving the packet transmits during the packet reception. Also, the following parameters of the WMN are given.

- The paths of the end-to-end flows
- The link-scheduling policy
- The maximum transmission radius of the nodes

1.1.2 Technical Relevance

The philosophy behind the proposed research problem is to provide an independent autonomous WMN capable of organizing and maintaining itself, and adapting and integrating its topology with its environment by considering the clients' data flows and the available network resources. In this way, the WMN would provide final users with an easy deployment of the network and less maintenance needs, and would also increase the number of scenarios where the WMN can be used.

We propose the following guidelines for finding the solution of the topology-control problem.

- Autonomous formation and maintenance: It should be possible for final users with very basic knowledge of WMNs to seamlessly deploy a WMN, and for experienced engineers to achieve a good performance if the network is more carefully designed.
- Resilience: The dependence on any centralized controller should be avoided for the solution to be resilient. The WMN should not depend on a specific entity for its correct operation. It should rely on the capabilities enabled in different nodes in a distributed way.

1.1.3 Proposed Solution

The solution to the topology-control problem should consist of distributed algorithms, which are more feasible for implementation due to their lower complexity when compared with centralized algorithms.

In order to solve the topology-control problem, the following objectives have been proposed.

1. To design distributed link-scheduling policies that are more throughput efficient than the current policies available in the literature
2. To identify the dependence on the network topology of the performance of the link-scheduling policies of Objective 1
3. To design distributed topology-control algorithms using the dependence of the link-scheduling policies on the network topology
4. To develop a simulation framework that allows the evaluation of the link-scheduling policies of Objective 1 and the topology-control algorithms

Objectives 1 to 3 provide the roadmap for reaching the solution to the topology-control problem, and Objective 4 provides the framework for the performance evaluation of the solution.

The basic idea behind the topology-control approach of Objectives 1 to 3 is as follows. The maximum data-packet rate that a given flow supports depends on the throughput that each of the nodes along the flow's path supports, and the maximum data-packet rate that a node supports depends on its scheduling policy and the conflicts[1] with surrounding nodes which also need to schedule packet transmissions. Therefore, by means of transmission-power control, the nodes are able to reduce the number of conflicts either by decreasing the interference (i.e., reducing transmission power) and/or coordinating future packet-transmission times such that no conflicting transmissions are performed simultaneously. In the latter approach, i.e., coordinating future packet transmissions, the transmission power needs to be increased in some cases in order to enable the nodes to listen to each other's schedules.

The topology control lies on the stability region of the WMN. The maximum throughput that nodes can support is characterized by the physical-link capacity and the stability region of the link scheduling policy. This region is the set of input-packet rates supported by the links of the network that guarantee that their queues are stable (i.e., the link queues are positive recurrent). The physical-link capacity determines the maximum length in bits of the packets. In the transmission-power control approach, the network topology is modified such that the stability region is adapted to the given set of flows. The goal of this adaptation is to maximize the highest input-packet rate supported by the flows that guarantee stability.

1.2 Dissertation Organization

In Chapter 2, the mathematical framework for the analysis of reservation-based distributed scheduling (RBDS) policies was developed. This framework consists of a queuing-system model for the WMN. Using this model, the set of end-to-end packet rates (i.e., the flow packet rates) that guarantee the stability of the network is characterized mathematically. This is a novel stability-analysis technique that is an adaptation of the classic stability analysis done for non-reservation-based scheduling

[1] Scheduling conflicts arise between nodes when they attempt to transmit packets simultaneously and the interference they cause on each other is high enough to cause packet collisions.

policies. Also, in Chapter 2, a new RBDS policy is proposed. This policy is called greedy-maximal reservation-based distributed scheduling (GM-RBDS). It is shown that the GM-RBDS policy outperforms, in terms of throughput, the policies currently available in the literature. This result is achieved by means of the more scalable overhead of the GM-RBDS policy when compared with the other policies. In Chapter 3, the relation between the set of end-to-end packet rates supported by the GM-RBDS policy and the topology of the network is characterized mathematically. Based on this relation, a heuristic and centralized topology-control algorithm is proposed. This is a novel topology-control approach that outperforms the classic approaches based on spatial reuse and/or heuristics based on the hidden/exposed nodes. Our approach is able to achieve higher levels of end-to-end packet rates that guarantee the stability of the network. Chapter 4 consists of the design and evaluation of distributed topology-control algorithms that adapt the stability region of the WMN to the paths of the flows in the network. The algorithms are designed using a game-theoretical approach in which the flows are players that collaborate to adapt the stabiliyt region. The algorithms are evaluated by means of simulation. In Chapter 5, the simulation framework, used in all previous chapters, is developed for Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 WMNs using optimized network engineering tools (OPNET). To the best of our knowledge, this is the first OPNET simulation model for this type of networks. Finally, the results of the dissertation are summarized in Chapter 6.

CHAPTER 2
RESERVATION-BASED DISTRIBUTED SCHEDULING (RBDS)

A major challenge in wireless networks is the ability to achieve maximum throughput via link scheduling. Link scheduling refers to the selection of a subset of links for simultaneous transmission that have the following characteristic: When the links are activated simultaneously, the interference between them is low enough to allow successful reception for every activated link. A scheduling policy specifies how to determine the subset of links that fits this characteristic and calculates the subset of links for each frame. A scheduling policy's throughput performance is determined from its efficiency ratio, which is defined as the fraction of the optimal capacity region in which the policy guarantees the stability of the network, i.e., that guarantees that the links' queues are all positive recurrent [69]. An optimal scheduling policy has an efficiency ratio of unity. When the scheduling policy has an optimal efficiency ratio, the wireless network is able to support the largest set of input rates, and so it achieves maximum throughput.

The challenge in scheduling is that the policies are highly complex. The scheduling problem in general is nondeterministic polynomial time (NP) hard [66]. Therefore, the research literature has focused on policies with lower complexity that are more amenable to implementation [49].

Most distributed scheduling policies that achieve provable ratios calculate, at the onset of every frame, a subset of links that is allowed to transmit data in the immediately following frame only. In this chapter, we propose a distributed scheduling policy that selects links to transmit data in any future frames by means of frame reservations. Also, we propose a new theoretical framework for the stability analysis for reservation-based distributed scheduling (RBDS) policies. Since our framework considers reservations of any future frames, the current policies correspond to a special case within our framework, i.e., the case that links are allowed to reserve the next frame only.

The rest of this chapter is organized as follows. The related work and the major contributions are discussed in Section 2.1, followed by a description of our wireless network model in Section 2.2. In Section 2.3, we create a framework to analyze the stability of RBDS systems. To demonstrate its efficacy, we propose an RBDS policy, which we call greedy maximal RBDS (GM-RBDS), for Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 mesh networks and use the new framework to evaluate the stability of this policy. In Section 2.4, we validate our theoretical results with simulation results, and compare the capacity of the RBDS policy with non-reservation-based techniques. Finally, a summary of the chapter is presented in Section 2.5.

2.1 Related Work

The concept of optimal capacity region and a centralized scheduling policy with efficiency ratio of unity were introduced in [69]. The centralized scheduling policy attempts to solve a complex global optimization problem so that the entire network is stable for the largest possible set of input rates. Stability is defined as the positive recurrence of all of the link queues. Under the 1-hop interference model, the problem is shown to correspond to a maximum weighted matching (MWM), where the weights of the links are determined from the length of their queues. The solution to MWM has complexity $O(N^3)$ [29, 59], where N is the number of nodes. Under the k-hop interference model, the problem has been proven to be NP-Hard [66]. Therefore, the optimal scheduling policy is not convenient for implementation due to its high complexity. As a consequence, less complex scheduling policies that achieve only a fraction of the optimal capacity region for general network topologies have been developed [12, 14, 15, 17, 26, 35–37, 61, 63, 65, 66, 68, 83, 84], and their impact on higher layers has been studied in [49].

Proposed suboptimal scheduling policies can be characterized by the techniques they use for calculating the next schedule. The schedule calculation depends on the interference model assumed for the network and the links' weights at the onset of every frame. The scheduling can be based on centralized or distributed approaches. Thus, our related work discussion considers these two separate categories (Unless otherwise specified, the scheduling policies reviewed in this section consider 1-hop traffic only. See Section 2.2 for the definition of 1-hop traffic).

2.1.1 Centralized Policies

In [68], a centralized scheduling approach known as pick-and-compare [65] that achieves the optimal efficiency ratio is defined. The pick-and-compare scheduling policy selects the optimal schedule at every frame with some probability greater than zero. First, the scheduling algorithm randomly picks a new schedule such that the links can satisfy the interference model constraints. Then, the newly picked schedule is compared to the current schedule. If the picked schedule reduces the total weight of the network (i.e., queue lengths) by more than the current schedule, then the picked schedule is selected as the next schedule; otherwise the current schedule is used again. The pick-and-compare policy requires the calculation and comparison of the updated total weight for every frame. Therefore, the complexity of this technique grows linearly with N, which makes it difficult to implement in networks with a high number of nodes or in networks where nodes have low processing capabilities.

Greedy maximal scheduling (GMS) is a suboptimal, centralized scheduling policy. In GMS, the links of the network are ordered according to their weights, where the link with maximum weight is placed at the top of this globally ordered list. A valid schedule is found by selecting links from the list from top to bottom that do not interfere with each other. The complexity of GMS is $O(L\log(N))$, where L is the number of links [60]. GMS has efficiency ratio of $\frac{1}{2}$ under the 1-hop interference model [49], and under the k-hop interference model, GMS has efficiency ratio of 1, $\frac{1}{6}$, and $\frac{1}{49}$ for tree, geometric, and general network graphs respectively [37, 66].

2.1.2 Distributed Policies

A distributed version of the pick-and-compare scheduling policy was proposed in [61]. In this policy, a node is selected with some probability less than one to initiate the calculation of a schedule for the links in its neighborhood. The new schedule is selected for the next frame if the new schedule reduces the neighborhood's weight by more than the current schedule. The algorithm has constant complexity, so it does not depend on the number of nodes of the network. It does depend, however, on the diameter of the neighborhood. The efficiency ratio increases as the diameter of the neighborhood increases. The algorithm assumes the 1-hop interference model, so it can only be directly used on networks with physical layers such as frequency-hopping code-division-multiple-access (FH-CDMA) that allow that assumption to be made.

Greedy scheduling (GS) policies [60] have been developed that achieve the same efficiency ratio of GMS [17, 35, 65]. In the GS policies, nodes calculate locally the next schedule based on the links that have the maximum local weights.

In [14, 15, 63, 65, 84], a maximal scheduling (MS) approach is described. In this approach, maximum weight is not required to schedule a link. A link is eligible for the next schedule as long as it has enough packets in the queue to transmit during the entire duration of a frame. The efficiency ratio of MS scheduling policies is $\frac{1}{\kappa}$, where κ is the maximum number of non-interfering links in the interference set of any link in the network. MS policies have also been adapted to multi-hop flow scenarios in which a set of flows with their respective rates and routes are given [14, 15, 63, 83, 84].

Lastly, distributed scheduling policies of complexity $O(1)$ have been developed in [26, 36, 50]. These are known as constant time (CT) scheduling policies [65]. The CT approach differs from the MS approach in that when a link does not interfere with the links in a schedule, it is selected with probability less than one. Therefore, in CT scheduling policies, frames can be wasted with some probability greater than zero. In [50], CT policies are proposed for the 1-hop and 2-hop interference models. The efficiency ratios of these policies were improved in [26, 36]. In [35], the improved efficiency ratios are $\frac{1}{2} - \frac{1}{\sqrt{m}}$ and $\frac{2}{\hat{n}}\left(\frac{1}{2} - \frac{1}{\sqrt{m}}\right)$ for the 1-hop and 2-hop interference models respectively, where $\hat{n}$ is the maximum number of 1-hop neighboring links for any link of the network.

2.1.3 Contributions

The major contributions of this chapter are:

1. An RBDS policy is proposed for the scheduling of IEEE 802.16 wireless mesh networks to increase throughput.

2. To evaluate the new policy, a Markovian system model is developed for RBDS policies that enables the stability analysis of these policies.

3. This stability analysis of RBDS policies is a more general framework. In previous policies, the links compete for access to the next frame only. Since our approach considers reservations for all future frames, the previous techniques correspond to a special case within our framework.

4. The new RBDS policy is proposed and analyzed within the proposed framework. Specifically, sufficient conditions on the data-packet arrival rates that guarantee the stability of the network are found. From these conditions, the stability region of the RBDS policy is described and a lower-bound for the policy's efficiency ratio is calculated. It is shown that this bound depends on two characteristics of the network topology.

5. The results are validated through theoretical and simulation analysis, and performance comparisons are made to existing policies.

In the next section, we introduce the proposed framework by describing the network model.

2.2 Network Model

We consider a wireless network represented by the graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links respectively. The links are directional. The link directed from node i to node j is denoted by (i, j). Also, it is assumed that if $(i, j) \in \mathcal{L}$, then $(j, i) \in \mathcal{L}$. The interference set of (i, j) is denoted by $\mathcal{I}^{(i,j)}$. A transmission over (i, j) is successful if and only if no other transmission occurs simultaneously over any of the links that belong to $\mathcal{I}^{(i,j)}$. If this interference constraint is not met, there is a collision, and two links conflict with each other if they belong to each other's interference set.

The sets of node i's 1-hop and 2-hop neighbors are denoted by $\mathcal{S}_1^i$ and $\mathcal{S}_2^i$ respectively. The 1-hop neighborhood and 2-hop neighborhood of node i are denoted by $\mathcal{S}_{\leq 1}^i$ and $\mathcal{S}_{\leq 2}^i$ respectively. $\mathcal{S}_{\leq 1}^i$ is the set of nodes formed by node i and node i's 1-hop neighbors $\left(\text{i.e., } \mathcal{S}_{\leq 1}^i \triangleq i \cup \mathcal{S}_1^i\right)$. $\mathcal{S}_{\leq 2}^i$ is the set of nodes formed by node i and node i's 1-hop and 2-hop neighbors $\left(\text{i.e., } \mathcal{S}_{\leq 2}^i = i \cup \mathcal{S}_1^i \cup \mathcal{S}_2^i\right)$.

As in the IEEE 802.16 mesh mode standard [1], time is divided into frames, and each frame is divided into a control-subframe and a data-subframe. Control-subframes are divided into control-time-slots that are used for the exchange of scheduling packets, and data-subframes are divided into data-time-slots that are used for the transmission of data packets. This structure is shown in Figure 2-1. Frames, control-time-slots, and data-time-slots are numbered independently starting from 0. Links can transmit only one scheduling packet per control-time-slot and only one data packet per data-time-slot. There are m_{cs} and m_{ds} control-time-slots and data-time-slots per frame respectively.

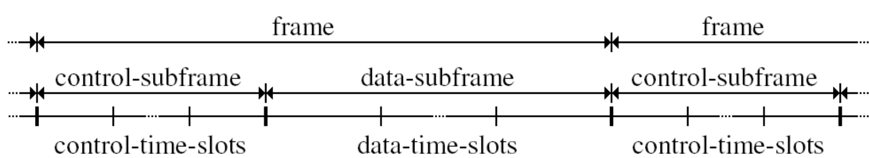


Figure 2-1. Frame structure

In this chapter, to introduce the reservation-based approach, we consider the 1-hop traffic model, as in [35–37, 61, 65, 66]. In future work, we will consider multi-hop traffic models. In the 1-hop traffic model, all data flows consist of only one hop. Further, we assume that there is only one flow per link. Therefore, data packets arrive at each link according to a random arrival process and leave the network once they reach their destination node which is one hop away from the source node. The random arrival process for the flow at link (i, j) is denoted by $A^{(i,j)}(k)$, where k is the current data-time-slot, and it has mean arrival rate $\lambda^{(i,j)}$. It is assumed that the arrival processes are independent and identically distributed (i.i.d.) sequences which are also independent across links.

In traditional network models, the data packets that arrive to a link are stored in one queue until they are transmitted. In our new reservation-based network model, we propose that each link has two queues (Figure 2-2). The input-queue stores data packets that arrive to the link and are waiting to be given a grant, which means they are waiting to be granted a data-time-slot. The output-queue stores the data packets that have received grants, i.e., have already been scheduled, and are waiting to be transmitted. When a link receives a grant, some of its unscheduled data packets are moved from its input-queue to its output-queue. The lengths of the input-queue and output-queue of link (i, j) at control-time-slot m are denoted by $Q_i^{(i,j)}(m)$ and $Q_o^{(i,j)}(m)$ respectively.

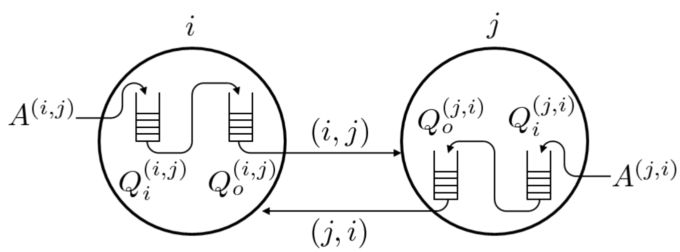


Figure 2-2. Data packet transmissions between nodes i and j

We define stability and the optimal capacity region of a wireless network as in [49, 69]. The network is stable when the system of queues across the network is positive recurrent [69]. The optimal capacity region, denoted by Λ, is the convex set $\text{Co}(\mathcal{R})$, where $\mathcal{R}$ is the set of all feasible schedules, and $\text{Co}(\mathcal{R})$ is its convex hull[1] . Λ is optimal in the sense that, when $[\lambda^{(i,j)}]$ is outside Λ, there is no scheduling policy that can

[1] Note that $\Lambda = \text{Co}(\mathcal{R})$ is true only for the traffic model considered in this chapter. Please refer to [69, 84] for the multi-hop traffic scenario.

stabilize the system. A feasible schedule is a binary vector of size $|\mathcal{L}|$ that specifies a subset of links that can be activated simultaneously without conflicts.

Finally, we define the efficiency ratio of a scheduling policy as in [84]. The efficiency ratio is the largest $\alpha \in (0, 1]$ such that the scheduling policy stabilizes the system if and only if $[\lambda^{(i,j)}] \in \alpha\Lambda$.

2.3 Reservation-Based Distributed Scheduling

In an RBDS wireless network, the nodes negotiate with their neighbors the reservation of future data-time-slots for their links. This negotiation is based on a three-way handshake that consists of a request, a grant, and a grant confirmation. Requests, grants, and grant confirmations are transmitted in scheduling packets. The nodes access the control-time-slots for transmitting scheduling packets using an election algorithm. Therefore, in an RBDS wireless network, the nodes access the wireless channel using two different algorithms: the election algorithm and the RBDS algorithm, whose roles are to avoid collisions and waisted time slots in the control and data subframes respectively.

In this chapter, we adopt the election algorithm of IEEE 802.16 mesh networks with coordinated distributed scheduling [1]. Also, it is assumed that the RBDS wireless network follows the 2-hop interference model, which is the model considered in the IEEE 802.16 mesh mode standard [1]. In the IEEE-802.16 election algorithm, the nodes in every 2-hop neighborhood take turns by competing between them to access the control-time-slots and transmit scheduling packets. We model the operation of this election algorithm as follows[2] .

- In order to avoid scheduling-packet collisions, no more than one node is selected in every $\mathcal{S}^i_{\leq 2}$ at any control-time-slot.

[2] The operation of the election algorithm for IEEE 802.16 mesh networks with coordinated distributed scheduling is described in detail in [13] and [78].

- The nodes in $\mathcal{S}^i_{\leq 2}$, where i can be any node in $\mathcal{N}$, are selected in cycles. We refer to these cycles as scheduling cycles.
- Within a scheduling cycle, the nodes in $\mathcal{S}^i_{\leq 2}$ are selected once and only once each. The order in which they are selected is uniformly distributed among all the possible orders of selection.
- The order that nodes in $\mathcal{S}^i_{\leq 2}$ are selected is independent across scheduling cycles.

2.3.1 RBDS Policies

When nodes i and j exchange scheduling messages to schedule data packets waiting to be transmitted on link (i, j), they follow a three-way handshake in order to multicast the negotiated grant to all links in $\mathcal{I}^{(i,j)}$. The handshake consists of the following steps[3] .

1. Node i sends a request to node j for a certain number of data-time-slots along with a set of data-time-slot numbers that are available for reservation at node i.
2. Node j sends a grant to node i for the requested number of data-time-slots according to its set of data-time-slots available for reservation and those of node i.
3. Node i confirms the successful reception of the grant by echoing the grant in its next scheduling-packet transmission.

The reservation of the data-time-slots takes place at steps 2 and 3. When node j transmits its scheduling packet, j's 1-hop neighbors receive the grant and mark the granted data-time-slots as unavailable. When node i confirms the grant, i's 1-hop neighbors receive the grant and mark the granted data-time-slots as unavailable too. Therefore, at the end of step 3, all links in $\mathcal{I}^{(i,j)}$ have made the granted data-time-slots unavailable $\big($i.e., the grant has been multicast to all links in $\mathcal{I}^{(i,j)}\big)$.

The requests and grants transmitted by the nodes are defined as follows.

Definition 1. Request $r_m^{(i,j)} \triangleq (f_\mathrm{s}, f_\mathrm{x}, z)$, where $(f_\mathrm{s}, f_\mathrm{x}, z) \in \mathbf{N}^3$, is the request transmitted by node i at control-time-slot m that requests for link (i, j) the data-time-slots of z

---

[3] It is assumed that in this handshake node j grants node i's request and that the data-packet-slot reservation is successful at both i and j.

consecutive data-subframes starting at frame f_s or any other frame after f_s. Request $r_m^{(i,j)}$ expires at the onset of frame f_x.

Definition 2. Grant $g_m^{(i,j)} \triangleq (f_s, f_e)$, where $(f_s, f_e) \in \mathbf{N}^2$, is the grant transmitted by node j at control-time-slot m that assigns to link (i, j) the data-time-slots of the series of frames that starts and ends with frames f_s and f_e respectively. Grant $g_m^{(i,j)}$ expires at the end of frame f_e.

Definition 3. The length of grant $g_m^{(i,j)}$, denoted by $|g_m^{(i,j)}|$, is the number of data-subframes assigned in the grant. Therefore,

$$|g_m^{(i,j)}| \triangleq f_e - f_s + 1.$$

In order to implement RBDS policies, each node maintains two tables per link that the node belongs to. These are the unavailable-data-time-slots table and the requested-data-time-slots table. The tables are updated with the grants and requests exchanged with the node's 1-hop neighbors. An unavailable-data-time-slots table contains the set of unexpired grants that interfere with the link that the table belongs to. This set is denoted by $\mathcal{T}_u^{(i,j)}(m)$ for link (i, j) and is given by Eq. 2–1, where $\left(g_m^{(x,y)}\right)_{f_e}$ is the f_e component of $g_m^{(x,y)}$, and f_m is the current frame number (i.e., the frame that control-time-slot m belongs to). The requested-data-time-slots table contains the set of unexpired requests made for the link the table belongs to. This set is denoted by $\mathcal{T}_r^{(i,j)}(m)$ for link (i, j) and is given by Eq. 2–2, where $\left(r_m^{(i,j)}\right)_{f_x}$ is the f_x component of $r_m^{(i,j)}$. $\mathcal{T}_u^{(i,j)}(m)$ and $\mathcal{T}_r^{(i,j)}(m)$ are functions of m given that the tables are updated with the grants and requests transmitted at every control-time-slot.

$$\mathcal{T}_u^{(i,j)}(m) \triangleq \left\{ g_l^{(x,y)} : \left(g_l^{(x,y)}\right)_{f_e} \geq f_m, (x, y) \in \mathcal{I}^{(i,j)},\ l \leq m \right\} \tag{2–1}$$

$$\mathcal{T}_r^{(i,j)}(m) \triangleq \left\{ r_l^{(i,j)} : \left(r_l^{(i,j)}\right)_{f_x} > f_m,\ l \leq m \right\} \tag{2–2}$$

In RBDS policies, two grants overlap with each other if the frame ranges given by their respective f_{s} and f_{e} frame numbers have one or more frame numbers in common.

2.3.2 Stability Analysis of RBDS Policies

2.3.2.1 RBDS Markovian system model

In an RBDS network, each link has an input-queue and an output-queue as defined in Section 2.2. The length of an input-queue $\left(\text{i.e., } Q_i^{(i,j)}(m)\right)$ is defined as the number of data packets in the queue. The length of an output-queue $\left(\text{i.e., } Q_o^{(i,j)}(m)\right)$ corresponds to the number of data-subframes in the following frame range: from the current frame to the last frame scheduled for the packets in the output-queue. Therefore, the length of output-queues does not depend on the number of scheduled packets waiting to be transmitted but on the schedules of such packets. The length of output-queues is formally defined by Eq. 2–3[4] , where $\mathcal{T}_g^{(i,j)}$ is the set of unexpired grants of link (i,j) $\Big(\text{i.e., }$ $\mathcal{T}_g^{(i,j)}(m) \triangleq \left\{g_l^{(i,j)} : \left(g_l^{(i,j)}\right)_{f_{\mathsf{e}}} \geq f_m,\ l \leq m\right\}\Big)$.

$$Q_o^{(i,j)}(m) \triangleq \left[\max\left(\left\{(g)_{f_{\mathsf{e}}} : g \in \mathcal{T}_g^{(i,j)}(m)\right\}\right) - f_m + 1\right]^+ \tag{2–3}$$

A node transmits scheduling packets by accessing control-time-slots according to the election algorithm. The next control-time-slot that node i is going to access is determined by the election algorithm. This control-time-slot is denoted by $M^i(m)$. That is, at control-time-slot m, the future control-time-slot that node i uses to transmit a scheduling packet is control-time-slot $M^i(m)$.

Based on the previous definitions, RBDS wireless network $\mathcal{G}$ can be represented as a Markovian system whose state $\mathcal{S}^{\mathcal{G}}$ is given by the lengths of the input and output queues of all the links and the scheduling control-time-slots of all the nodes. That is,

$$\mathcal{S}^{\mathcal{G}} \triangleq \left\{Q_i^{(i,j)}(m), Q_o^{(i,j)}(m), M^i(m) : (i,j) \in \mathcal{L}, i \in \mathcal{N}\right\}. \tag{2–4}$$

[4] $[\cdot]^+$ is the positive-part operator.

2.3.2.2 RBDS Markovian system state update

Given that $\mathcal{S}^{\mathcal{G}}$ is updated only during control-subframes (i.e., at control-time-slot m as given by Eq. 2–4), the data packet arrival and departure processes for link (i, j) can be modeled as shown in Figure 2-3[5] . There are data-packet arrivals and departures at the first control-time-slot of every control-subframe only. These correspond to the total number of arrivals and departures that occurred during the data-subframe previous to the control-subframe. In Figure 2-3, for control-time-slot m, the last data-time-slot of this data-subframe corresponds to k_m. The data-packet arrivals, which are denoted by $A'^{(i,j)}(m)$, correspond to the data packets that are input to $Q_i^{(i,j)}$ (i.e., data packets that need to be scheduled). The data-packet departures correspond to the data packets that are output from $Q_o^{(i,j)}$. The number of these departures does not affect the length of $Q_o^{(i,j)}$ according to Eq. 2–3. However, the data-subframe in which these departures occur affects $Q_o^{(i,j)}$. $Q_o^{(i,j)}$ is decreased by 1 data-subframe every time a data-subframe is over. This decrease is denoted by $D'(m)$. $A'^{(i,j)}(m)$ and $D'(m)$ are given as follows[6] .

$$A'^{(i,j)}(m) \triangleq \begin{cases} \sum_{l=0}^{m_{\text{ds}}-1} A^{(i,j)}(k_m - l) & m = \text{multiple of } m_{\text{cs}}, \\ 0 & \text{otherwise}. \end{cases} \tag{2–5}$$

$$D'(m) \triangleq \begin{cases} 1 & m = \text{multiple of } m_{\text{cs}}, \\ 0 & \text{otherwise}. \end{cases} \tag{2–6}$$

$\mathcal{S}^{\mathcal{G}}$ is also updated every time a grant is received by any of the links. When $M^j(m) = m$ for some j in $\mathcal{N}$, node j transmits a scheduling packet. The scheduling packet may

[5] Note that in Figure 2-3, it is assumed that $m_{\text{cs}} = m_{\text{ds}} = 3$.

[6] Note that in Figure 2-3, $A'^{(i,j)}(m)$ and $D'(m)$ are indicated only at the control-time-slots in which they are different from zero.

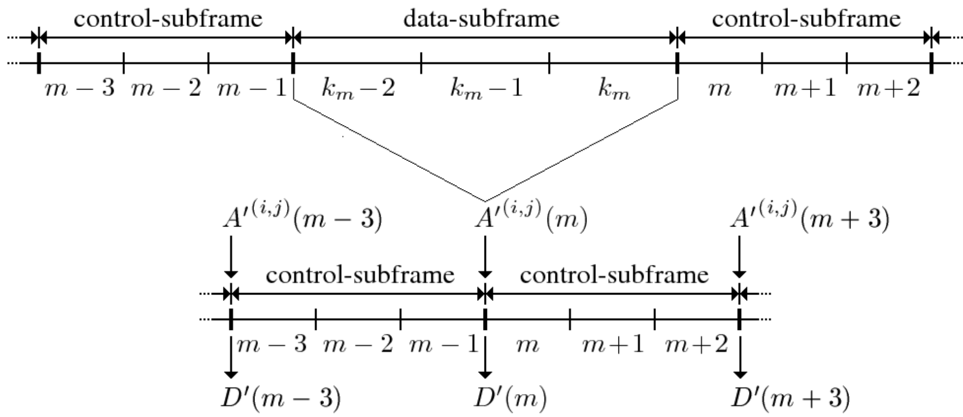


Figure 2-3. Link data packet arrival and departure processes

carry one or more grants for the node's incoming links (k, j), where node k can be any of node j's 1-hop neighbors. Each grant carries the schedules for a subset of data packets waiting in their corresponding input-queue $Q_i^{(k,j)}$ at node k. The grant received by link (i, j) at control-time-slot m and denoted by $G_m^{(i,j)}$ is given by Eq. 2–7. The condition $k = i$ in Eq. 2–7 represents the fact that the grant transmitted by node j (i.e., $g_m^{(k,j)}$) is directed to node i. Node j's grants can be directed to any of its 1-hop neighbors.

$$G_m^{(i,j)} \triangleq \begin{cases} g_m^{(k,j)} & \text{if } M^j(m) = m \text{ and } k = i, \\ \emptyset & \text{otherwise}. \end{cases} \tag{2–7}$$

Given $A'^{(i,j)}(m)$, $D'(m)$, and $G_m^{(i,j)}$, state $\mathcal{S}^{\mathcal{G}}$ is updated according to Eq. 2–8, Eq. 2–9, and Eq. 2–10. $Q_i^{(i,j)}(m)$ is updated when there are data-packet arrivals $\left(\text{i.e., } A'^{(i,j)}(m) \neq 0\right)$ or when packets are scheduled $\left(\text{i.e., } G_m^{(i,j)} \neq \emptyset\right)$. $Q_o^{(i,j)}(m)$ is updated when there are data-packet departures $\left(\text{i.e., } D'(m) \neq 0\right)$ or when data-subframes are granted $\left(G_m^{(i,j)} \neq \emptyset\right)$. When node i transmits a scheduling packet at control-time-slot $M^i(m) = m$, the next control-time-slot (i.e., $M^i(m+1)$) used for the next scheduling-packet transmission is ΔM_m^i control-time-slots away. Given that in the election algorithm (Section 2.3), the order that the nodes in $\mathcal{S}_{\leq 2}^i$ are selected in every scheduling cycle is independent across scheduling cycles, ΔM_m^j is an i.i.d. random sequence[7] , i.e., $\mathrm{P}[\Delta M_m^j = n] = \alpha_n \; \forall \; j \in \mathcal{N}$, where α_n is some pmf such that $\alpha_n = 0 \; \forall \; n \leq 0$.

$$Q_i^{(i,j)}(m+1) = \left[Q_i^{(i,j)}(m) + A'^{(i,j)}(m) - m_{\mathrm{ds}}\left|G_m^{(i,j)}\right|\right]^+ \tag{2–8}$$

$$Q_o^{(i,j)}(m+1) = \max\left(\left[Q_o^{(i,j)}(m) - D'(m)\right]^+, \left[\left(G_m^{(i,j)}\right)_{\mathrm{fe}} - f_m + 1\right]^+\right) \tag{2–9}$$

$$M^i(m+1) = \begin{cases} M^i(m) + \Delta M_m^i & \text{if } m = M^i(m), \\ M^i(m) & \text{if } m \neq M^i(m). \end{cases} \tag{2–10}$$

2.3.2.3 Scheduling in an RBDS wireless network

Let $\mathcal{Q}$ be the set of queues in the network $\big($i.e., $\mathcal{Q} \triangleq \left\{Q_i^{(i,j)}(m), Q_o^{(i,j)}(m) : (i,j) \in \mathcal{L}\right\}\big)$. According to Eq. 2–8 and Eq. 2–9, the updates of the queues in $\mathcal{Q}$ take place only when either of two events occur. These events are the beginning of a control-subframe and the transmission of a scheduling packet. According to Eq. 2–5 and Eq. 2–6, when a control-subframe starts (i.e., when m is multiple of m_{cs}), the number of data-packet arrivals and departures is different from zero. For the rest of the control-subframe, there are no arrivals nor departures. Therefore, queues $Q_i^{(i,j)}$ and $Q_o^{(i,j)}$ are updated with data-packet arrivals and departures respectively only at the first control-time-slot of every control-subframe. When scheduling packets are transmitted $\big($i.e., when $M^j(m) = m \; \exists \; j \in \mathcal{N}\big)$, queues $Q_i^{(i,j)}$ and $Q_o^{(i,j)}$ are updated at the nodes that received grants carried by the scheduling

[7] Note that the sequence ΔM_m^i is defined only for the control-time-slots when node i transmits a scheduling-packet.

packets. The update consists of scheduling data packets that are waiting in $Q_i^{(i,j)}$ and moving those packets to $Q_o^{(i,j)}$.

Let μ_1 be the event that a control-subframe starts, and let μ_2 be the event that at least one scheduling packet is transmitted.

The updates that take place when a μ_1-event occurs at control-time-slot m modify the queues in $\mathcal{Q}$ as follows.

- $A'^{(i,j)}(m)$ data packets are input to $Q_i^{(i,j)}$.
- $D'(m)$ data-subframes are removed from $Q_o^{(i,j)}$.

The updates that take place when a μ_2-event occurs at control-time-slot m modify the queues in $\mathcal{Q}$ as follows.

- $m_{\text{ds}}\left|G_m^{(i,j)}\right|$ data packets are removed from $Q_i^{(i,j)}$.
- $Q_o^{(i,j)}$ is increased with $\left(G_m^{(i,j)}\right)_{f_{\text{e}}}$ according to Eq. 2–9.

An RBDS policy has control only over the μ_2-type queue updates. The RBDS policy is responsible for determining the grants that are transmitted in scheduling packets. Specifically, the RBDS policy determines the grants for all node j's incoming links, where j is any node in $\mathcal{N}$, every time node j transmits a scheduling packet. Therefore, the RBDS policy determines the number of packets that are moved from every $Q_i^{(i,j)}$ to its corresponding $Q_o^{(i,j)}$, where (i,j) is any of node j's incoming links.

2.3.2.4 Stability analysis of an RBDS wireless network

Definition 4. Wireless network $\mathcal{G}$ is stable if the queue process $\mathcal{Q}$ in $\mathcal{S}^{\mathcal{G}}$ is positive recurrent.

In order to analyze the stability of network $\mathcal{G}$ under RBDS policies, a Markovian system denoted by $\mathcal{S}_{\text{s}}^{\mathcal{G}}$ and derived from Eq. 2–4 and the update equations Eq. 2–8, Eq. 2–9, and Eq. 2–10 is considered.

$\mathcal{S}_s^{\mathcal{G}}$ is updated only when there are scheduling-packet transmissions (i.e., the system's state is updated only when a μ_2 event occurs), which are scheduled by the election algorithm used for accessing the control-time-slots according to Eq. 2–10.

The election algorithm and the RBDS policy determine how the system's state is updated. Specifically, the election algorithm determines the subset of nodes that transmit scheduling packets at every μ_2 event and the number of data-time-slots between every two consecutive μ_2 events. The RBDS policy determines the grants that are carried by the scheduling packets.

In the following, the stability analysis of $\mathcal{S}_s^{\mathcal{G}}$ is performed in four steps. In the first two steps, the system updates due to the election algorithm and RBDS policies are characterized. The third step is an example that illustrates a system update. Finally, in the fourth step, the sufficient conditions that guarantee stability are calculated.

$\mathcal{S}_s^{\mathcal{G}}$ update and the election algorithm. Two queues are defined for each node j in $\mathcal{N}$ based on the queues of the node's incoming links. These are the input-queue $Q_i^j(n)$ and the output-queue $Q_o^j(n)$. Queue $Q_i^j(n)$ represents the total number of packets that are waiting at all of j's 1-hop neighbors' input-queues to be scheduled for transmission to node j (Eq. 2–11). Queue $Q_o^j(n)$ represents the number of data-subframes in the range of frames that starts at the current frame and ends at the latest frame granted by node j (Eq. 2–12).

$$Q_i^j(n) \triangleq \sum_{i \in \mathcal{S}_1^j} Q_i^{(i,j)}(m_n) \tag{2–11}$$

$$Q_o^j(n) \triangleq \max_{i \in \mathcal{S}_1^j} \left\{ Q_o^{(i,j)}(m_n) \right\} \tag{2–12}$$

The subset of nodes selected by the election algorithm for the n^{th} occurrence of event μ_2 is denoted by $\mathcal{N}_{\mu_2}(n)$ and is defined by Eq. 2–13, where m_n is the control-time-slot when the n^{th} μ_2 event occurs.

$$\mathcal{N}_{\mu_2}(n) \triangleq \{j : j \in \mathcal{N}, M^j(m_n) = m_n\} \tag{2–13}$$

The election algorithm does not guarantee scheduling-packet transmissions at every control-time-slot m. The index m_n denotes only the control-time-slots in which there is at least one scheduling-packet transmission in the network (i.e., the occurrence of a μ_2 event). Specifically, the n^{th} time that at least one scheduling packet is transmitted in the network takes place at control-time-slot m_n.

The election algorithm calculates the series of subsets of nodes $\mathcal{N}_{\mu_2}(n)$ based on the update equation given by Eq. 2–10. Each of the subsets in the series corresponds to the nodes that transmit scheduling packets simultaneously at a certain control-time-slot. Only the queues $Q_i^j(n)$ and $Q_o^j(n)$ of the nodes in $\mathcal{N}_{\mu_2}(n)$ are updated at control-time-slot m_n as follows: The input and output queues are decreased and increased respectively by the grants carried in the transmitted packets. The input queues are increased by the number of packets that arrived since the previous scheduling-packet transmissions. The output queues are decreased by the number of data-subframes between the previous and current scheduling-packet transmissions. In order to characterize this update process, the following definitions are considered.

The number of data-packet arrivals to the input-queue of link (i, j) between node j's n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions is denoted by $A_{\text{s}}^{(i,j)}(n)$ and is given by Eq. 2–14. The data-packet arrival rate in this process is given by Eq. 2–15[8] , where $N_{\mu_1}^j(n)$ is the number of μ_1 events between node j's n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions[9] .

[8] The data-packet arrival rate $\lambda_s^{(i,j)}$ is derived in Appendix A. In this chapter, $\overline{X}$ stands for the expected value of random variable X.

[9] $N_{\mu_1}^j(n)$ is determined by the number of control-time-slots between these transmissions, and this number is equal to ΔM_m^j (Eq. 2–10).

$$A_{\mathrm{s}}^{(i,j)}(n) \triangleq \sum_{m=m_{n+1}-\Delta M_{m_n}^{j}}^{m_{n+1}} A'^{(i,j)}(m) \tag{2–14}$$

$$\lambda_{\mathrm{s}}^{(i,j)} \triangleq \mathrm{E}\left[A_{s}^{(i,j)}(n)\right] = \overline{N_{\mu_1}^{j}} m_{\mathrm{ds}} \lambda^{(i,j)} \tag{2–15}$$

Let $\mathcal{M}$ be the set of grants carried by the scheduling packet transmitted by node j at control-time-slot m_n. The total number of data-subframes that link (i, j) is granted in $\mathcal{M}$ is given by Eq. 2–16.

$$\left|G^{(i,j)}(n)\right| \triangleq \sum_{G_{m_n}^{(i,j)} \in \mathcal{M}} \left|G_{m_n}^{(i,j)}\right| \tag{2–16}$$

The latest data-subframe that link (i, j) is granted in $\mathcal{M}$ is given by Eq. 2–17.

$$\left(G^{(i,j)}(n)\right)_{\mathrm{fe}} \triangleq \max\left(\left\{\left(G_{m_n}^{(i,j)}\right)_{\mathrm{fe}} : G_{m_n}^{(i,j)} \in \mathcal{M}\right\}\right) \tag{2–17}$$

Definition 5. The state $\mathcal{S}_{\mathrm{s}}^{\mathcal{G}}$ of RBDS wireless network $\mathcal{G}$ is given by

$$\mathcal{S}_{\mathrm{s}}^{\mathcal{G}} \triangleq \left\{Q_i^j(n), Q_o^j(n), M^j(m_n) : j \in \mathcal{N}\right\}$$

where $\mathcal{S}_{\mathrm{s}}^{\mathcal{G}}$ is updated according to Eq. 2–18, Eq. 2–19, and Eq. 2–10.

$$Q_i^j(n+1) = \begin{cases} \left[Q_i^j(n) + \sum_{i \in \mathcal{S}_1^j} A_{\mathrm{s}}^{(i,j)}(n) - m_{\mathrm{ds}} \sum_{i \in \mathcal{S}_1^j} |G^{(i,j)}(n)|\right]^+ & j \in \mathcal{N}_{\mu_2}(n), \\ Q_i^j(n) & j \notin \mathcal{N}_{\mu_2}(n). \end{cases} \tag{2–18}$$

$$Q_o^j(n+1) = \begin{cases} \max\left(\left[Q_o^j(n) - N_{\mu_1}^j(n)\right]^+, \left[\max_{i \in \mathcal{S}_1^j}\left\{\left(G^{(i,j)}(n)\right)_{\mathrm{fe}}\right\} - f_{m_n} + 1\right]^+\right) & j \in \mathcal{N}_{\mu_2}(n), \\ Q_o^j(n) & j \notin \mathcal{N}_{\mu_2}(n). \end{cases} \tag{2–19}$$

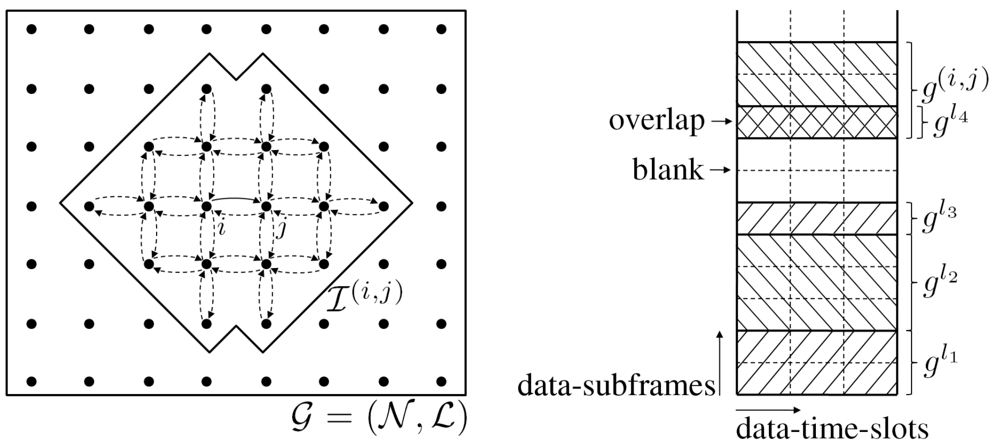
$\mathcal{S}_{\mathbf{s}}^{\mathcal{G}}$ **update and the RBDS policy**. According to Eq. 2–18 and Eq. 2–19, the updates of the queues consider the grants carried by each of the transmitted scheduling packets. Each grant assigns a set of data-time-slots to one of the incoming links of the node that transmitted the grant, and the incoming link schedules packets in its input-queue at the granted data-time-slots. This scheduling causes those packets to be moved from the link's input-queue to the link's output-queue.

The grants are calculated according to an RBDS policy. The task performed in this calculation is illustrated in Figure 2-4. In Figure 2-4A, a wireless network $\left(\text{i.e., } \mathcal{G} = (\mathcal{N}, \mathcal{L})\right)$ and one of its interfering sets $\left(\text{i.e., } \mathcal{I}^{(i,j)}\right)$ are shown[10] . Given that transmissions on any of the links in $\mathcal{I}^{(i,j)}$ interfere with transmissions on link (i,j), node j should generate grants for link (i,j) that consider the unexpired grants of all other links in $\mathcal{I}^{(i,j)}$. This is shown in Figure 2-4B. It is assumed that Figure 2-4B is a snapshot of all the unexpired grants of links in $\mathcal{I}^{(i,j)}$ right after node j transmits a scheduling packet with only one grant for link (i,j), which is denoted by $g^{(i,j)}$. That is, Figure 2-4B shows the data-subframes from the time of node j's scheduling-packet transmission until the last granted data-subframe that interferes with link (i,j). The data-time-slots in each of these frames are also included[11] . Assuming that grants g^{l_1}, g^{l_2}, g^{l_3}, g^{l_4} were transmitted by some interfering links in $\mathcal{I}^{(i,j)}$ before node j's scheduling-packet transmission, node j needs to select a set of available data-time-slots for grant $g^{(i,j)}$ that have not been included in any of the interfering grants. The selection made in Figure 2-4B shows two events that nodes need to avoid in order to improve the efficiency ratio. These are blanks and overlaps. Blanks correspond to data-time-slots which are available to a link, and the link does not include them in any of its grants. Overlaps correspond to data-time-slots which are unavailable to a link, and the link includes them in one or more

[10] In Figure 2-4A, not all the links in $\mathcal{G}$ are shown. Only the links in $\mathcal{I}^{(i,j)}$ are included. Within this set of links, link (i,j) is shown with a straight line, and all its interfering links are shown with dashed lines.

[11] It is assumed that there are 3 data-time-slots per data-subframe (i.e., $m_{\text{ds}} = 3$).

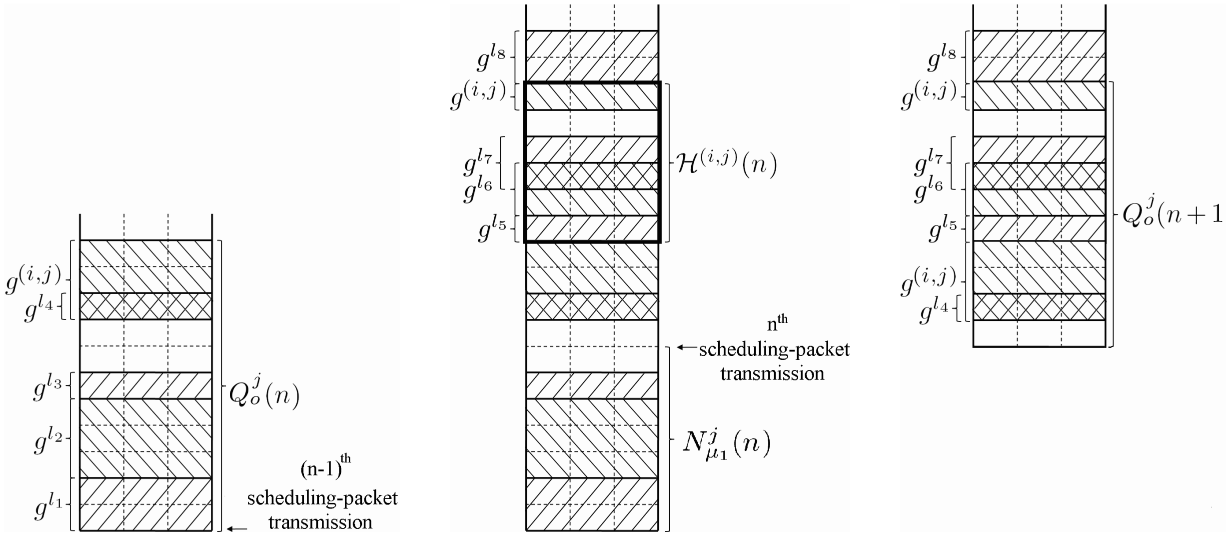
of its grants. Blanks and overlaps turn into wasted data-time-slots and data-time-slots with collisions respectively once they are the current data-time-slot in the RBDS wireless network. Therefore, the task of a node is to select available data-time-slots for grants so that the total number of blanks and overlaps is minimized across all the links in the network.



A Interfering-link set $\mathcal{I}^{(i,j)}$ B Grants in the interfering link set $\mathcal{I}^{(i,j)}$

Figure 2-4. Reservation-based distributed scheduling in the interfering-link set $\mathcal{I}^{(i,j)}$

$\mathcal{S}_{\mathbf{s}}^{\mathcal{G}}$ **update example**. The update process of $Q_o^j(n)$ is illustrated in Figure 2-5. After the n^{th} scheduling-packet transmission of node j, $Q_o^j(n)$ is as shown in Figure 2-5A. It has a length of 11 data-subframes, in which grants g^{l_1}, g^{l_2}, g^{l_3}, g^{l_4}, and $g^{(i,j)}$ have been placed. It is assumed that at the n^{th} scheduling-packet transmission of node j, node j transmits a scheduling packet with one grant for link (i, j) only. It is also assumed that grants g^{l_5}, g^{l_6}, g^{l_7}, and g^{l_8} were transmitted by links in $\mathcal{I}^{(i,j)}$ at some control-time-slots between node j's n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions, and that these grants and $g^{(i,j)}$ were assigned data-time-slots according to Figure 2-5B. Given that node j transmits a grant in its $(n+1)^{\text{th}}$ scheduling-packet transmission too, the length of $Q_o^j(n)$ is updated. This update consists of a decrease and an increase of data-subframes, which are explained next.



A $Q_o^j(n)$

B Transition from $Q_o^j(n)$ to $Q_o^j(n+1)$

C $Q_o^j(n+1)$

Figure 2-5. Queue $Q_o^j(n)$ update process

Sufficient conditions for the stability of $\mathcal{S}_{\mathbf{s}}^{\mathcal{G}}$. The decrease corresponds to the number of data-subframes between node j's n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions $N_{\mu_1}^j(n)$ (Eq. 2–19). The queue-length decrease size $N_{\mu_1}^j(n)$ is given by Eq. 2–20. In Figure 2-5B, the size of $N_{\mu_1}^j(n)$ is 7 data-subframes.

$$N_{\mu_1}^j(n) \triangleq \sum_{m=m_{n+1}-\Delta M_{m_n}^j}^{m_{n+1}} D'(m) \tag{2–20}$$

The queue-length increase corresponds to the total number of data-subframes included in the grants, blanks, and overlaps that were scheduled between the n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions. This set of data-subframes, which is denoted by $\mathcal{H}^j(n)$, is shown in Figure 2-5B. The specific RBDS policy implemented on the network is responsible for calculating the grants, blanks, and overlaps that $\mathcal{H}^j(n)$ includes. Therefore, the number of frames covered by $\mathcal{H}^j(n)$, which is denoted by $|\mathcal{H}^j(n)|$, depends on the number of frames covered by the grants, blanks, and overlaps determined by the RBDS policy. Given that the RBDS policy determines the number of frames covered by the grants, blanks, and overlaps from the number of unscheduled packets (i.e., the input-queue lengths), $|\mathcal{H}^j(n)|$ depends on the input-queues and not on the output queues[12] .

Given $\mathcal{H}^j(n)$, the update of $Q_o^j(n)$ defined in Eq. 2–19 can be expressed as in Eq. 2–21.

$$Q_o^j(n+1) = \begin{cases} \left[Q_o^j(n) + \left|\mathcal{H}^j(n)\right| - N_{\mu_1}^j(n)\right]^+ & j \in \mathcal{N}_{\mu_2}(n), \\ Q_o^j(n) & j \notin \mathcal{N}_{\mu_2}(n). \end{cases} \tag{2–21}$$

Definition 6. An RBDS wireless network is stationary if the random processes $N_{\mu_1}^j(n), |\mathcal{H}^j(n)|, |G^{(i,j)}(n)|$ are stationary for all j in $\mathcal{N}$ and all (i,j) in $\mathcal{L}$.

Theorem 2.1. *The output-queues in a stationary RBDS wireless network* $\big($*i.e.,* $\left\{Q_o^{(i,j)}(m) : (i,j) \in \mathcal{L}\right\}\big)$ *are stable if Eq. 2–22 holds, and the input-queues* $\big($*i.e.,* $\left\{Q_i^{(i,j)}(m) : (i,j) \in \mathcal{L}\right\}\big)$ *are stable if Eq. 2–23 holds. Therefore, an RBDS wireless network is stable if both Eq. 2–22 and Eq. 2–23 hold.*

$$\max_{j \in \mathcal{N}} \left(\overline{|\mathcal{H}^j|} - \overline{N_{\mu_1}^j}\right) < 0 \tag{2–22}$$

$$\max_{j \in \mathcal{N}} \left(\overline{N_{\mu_1}^j} \sum_{i \in \mathcal{S}_1^j} \lambda^{(i,j)} - \sum_{i \in \mathcal{S}_1^j} \overline{|G^{(i,j)}|}\right) < 0 \tag{2–23}$$

See Appendix B for the proof of Theorem 2.1[13] .

Remark. Intuitively, conditions given by Eq. 2–22 and Eq. 2–23 guarantee stability because they require that all the queues decrease their lengths at a rate lower than they increase. That is, the conditions select the node whose difference between increase and

[12] The independence of $|\mathcal{H}^j(n)|$ from the output-queues will be used in the proof of Theorem 2.1 (Appendix B).

[13] The proof of Theorem 2.1 is based on the proof for the stability of greedy scheduling policies under 1-hop traffic presented in [84].

decrease rates is the largest and makes sure that the increase rate is lower than the decrease rate.

Within the framework for RBDS wireless networks presented in this section (i.e., 2.3.2), different RBDS policies can be evaluated in terms of stability. In the following, we propose the Greedy-Maximal-RBDS (GM-RBDS) policy and evaluate it.

2.3.3 Stability and Complexity Analysis of the Greedy Maximal RBDS (GM-RBDS) Policy

The GM-RBDS policy is as follows. When any node i in $\mathcal{N}$ transmits a scheduling packet,

- It grants the longest request among all the unexpired requests made by its incoming links, and sets the grant's f_s component at the frame following the interfering grant that expires the latest.
- For every of its outgoing links, it requests as many consecutive data-subframes as unscheduled data packets cover entirely, sets every request's f_s component at the frame following the interfering grant that expires the latest, and sets every request's f_x component at the frame scheduled for its next scheduling-packet transmission.

And when any node i in $\mathcal{N}$ receives a scheduling packet, it checks whether there is a grant in the packet and whether the grant is directed to one of its outgoing links. If that is the case, it confirms the grant only if the grant does not overlap with any of the grants in the link's unavailable-data-time-slots table.

The GM-RBDS policy is greedy maximal in the sense that the requests that are granted are the longest requests and each request corresponds to the maximum integer number of data-subframes that are covered by a link's unscheduled data packets $\big($i.e., each request corresponds to $\lfloor \frac{Q_i^{(i,j)}}{m_\mathrm{ds}} \rfloor$, where $Q_i^{(i,j)}$ is the number of unscheduled data packets to be transmitted on link $(i, j)\big)$.

2.3.3.1 Complexity analysis

Let the complexity be defined in terms of the amount of control information that is transmitted during a control-subframe (i.e., overhead). In the GM-RBDS policy, whenever a node transmits in a control-time-slot, it sends at most 1 request per outgoing link, 1 grant, and 1 grant confirmation per outgoing link. Each grant and grant-confirmation corresponds to a range of frames that is specified with two integers: f_s and f_e (Definition 2). Each request corresponds to a range of frames and an expiration frame (Definition 1) which is a total of three integers (i.e., f_s, f_x, and z). Therefore, in the GM-RBDS policy, in the worst-case scenario, a node transmits at most $5\Delta + 2$ integers per control-time-slot, where Δ is the maximum node degree[14] , so a total of $m_{cs}(5\Delta + 2)$ integers are transmitted during a control-subframe. Then, in WMNs with randomly located nodes, the GM-RBDS policy has $O(\log |\mathcal{N}|)$ complexity[15] .

Remark. In the GM-RBDS policy, a node needs to know the unavailable-data-time-slot table of the links that interfere with its incoming links in order to calculate the grants it sends. This process is described in Sections 2.3.2.4 and 2.3.2.4. Note that the $m_{cs}(5\Delta + 2)$ integers are enough for specifying the unavailable-data-time-slots tables the nodes need to know. This is true because the unavailable-data-time-slots tables are implicitly specified by the end frames (i.e., f_e) of grants and grant-confirmations. That is, all the frames previous to those end frames have already been reserved because in the GM-RBDS policy, the nodes grant the closest frames in time that are available.

The stability analysis is done as follows. The sufficient conditions for the stability of the output-queues are found first in the next section (i.e., Section 2.3.3.2), then the sufficient conditions for the stability of input-queues are found in Section 2.3.3.3, and finally, the GM-RBDS stable region and efficiency ratio are calculated from the conditions in Section 2.3.3.4.

---

[14] The node degree is the number of bidirectional links connected to the node, where a bidirectional link is the pair of outgoing and incoming links that connect to and from the same 1-hop neighbor respectively.

[15] The number of control-time-slots per frame (i.e., m_{cs}) is a design parameter that is independent of the number of nodes.

2.3.3.2 Sufficient condition for the stability of output-queues

Given that ΔM_m^j is an i.i.d. random sequence (Section 2.3.2.2) and that the GM-RBDS policy has no dynamic parameters, GM-RBDS wireless networks are stationary. Therefore, the sufficient conditions for stability of GM-RBDS networks can be found using Theorem 2.1.

Every time a node is selected at a certain control-time-slot by the election algorithm, the node transmits a scheduling packet. The node writes only one grant for the longest of all incoming links' requests on the scheduling packet[16] . The transmission of this grant updates the output-queue of the node and the unavailable-data-time-slots tables of its 1-hop neighbors. The output-queue of the node is increased by the length of the grant. This is due to the fact that the grant's f_s component is set at the frame following the interfering grant that expires the latest (GM-RBDS policy in Section 2.3.3.). The unavailable-data-time-slots table of every 1-hop neighbor is updated by adding the grant to it (Eq. 2–1).

For example, if the nodes in any path in $\mathcal{G}$ have all the same output-queue length and transmit scheduling packets consecutively such that every node's transmission is followed by a 1-hop neighbor's transmission, the output-queue of every node is increased by the summation of all the previously transmitted grants. Also, if two paths connect at some node, the output-queue of this node is updated with the maximum summation of the two paths. This process is illustrated in Figure 2-6 for the case of three paths with three nodes each and all connected at one of their ends (i.e., node 7). Node i is the i^{th} node selected by the election algorithm, and G^i is the grant transmitted by node i. Q_o^i is the output-queue length of node i after the node has transmitted its scheduling packet.

[16] Only one grant is written on every scheduling packet due to the fact that only the longest request is granted.

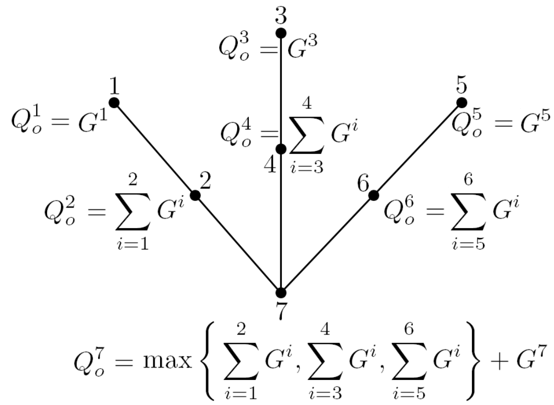


Figure 2-6. Output-queue updates during one scheduling cycle in a 2-hop neighborhood

Therefore, given that the paths that can be established between the nodes in $\mathcal{S}^i_{\leq 2}$ determine the grants that affect the output-queue length increase of node i, where i can be any node in $\mathcal{N}$, the stability of the output-queues of the nodes in $\mathcal{N}$ can be guaranteed by considering the length of such paths.

The following definitions are used for characterizing the sufficient conditions that guarantee the output-queues' stability.

Let $\mathcal{P}^i$ be the path of maximum length[17] in the undirected graph induced by $\mathcal{S}^i_{\leq 2}$ and that originates at node i. $\mathcal{P}_{\text{max}}$ is the longest $\mathcal{P}^i$ $\left(\text{i.e., } \mathcal{P}_{\text{max}} \triangleq \text{argmax}_{\mathcal{P}^i : i \in \mathcal{N}} |\mathcal{P}^i|\right)$. Let $\overline{N_{\mu_1}}$ be the expected number of μ_1 events between two consecutive scheduling-packet transmissions of any node[18]. Let n_{max} be the node with the highest output-queue increase rate $\left(\text{i.e., } n_{\text{max}} \triangleq \text{argmax}_{i \in \mathcal{N}} \left(\overline{|\mathcal{H}^i|} - \overline{N_{\mu_1}}\right)\right)$, and let $\mathcal{G}_{\text{max}} \triangleq \{\mathcal{N}_{\text{max}}, \mathcal{L}_{\text{max}}\}$ be the undirected graph induced by $\mathcal{S}^{n_{\text{max}}}_{\leq 2}$ $\Big(\text{i.e., } \mathcal{N}_{\text{max}} \triangleq \mathcal{S}^{n_{\text{max}}}_{\leq 2}$ and $\mathcal{L}_{\text{max}} \triangleq \big\{(i,j) : i,j \in$

[17] In the following, the length of path $\mathcal{P}$ is defined as the number of links in $\mathcal{P}$ and denoted by $|\mathcal{P}|$.

[18] Note that, given the assumption that the order nodes in every 2-hop neighborhood are selected in a scheduling cycle is uniformly distributed among all the possible orders of selection (Section 2.3.1), the nodes have the same pmf for ΔM^j_m. Therefore, the expected number of μ_1 events between two consecutive scheduling-packet transmissions is the same for all the nodes $\left(\text{i.e., } \overline{N^i_{\mu_1}} = \overline{N_{\mu_1}} \; \forall \; i \in \mathcal{N}\right)$.

$\mathcal{S}_{\leq 2}^{n_{\max}}; (i,j) \in \mathcal{L}\}\big)$. $G_m^{\max}$ is the grant transmitted in $\mathcal{G}_{\max}$ such that $\overline{|G_m^{\max}|} = \max\big(\{\overline{|G_m^{(i,j)}|} : j \in \mathcal{N}_{\max}; (i,j) \in \mathcal{L}\}\big)$.

Theorem 2.2. *The output-queues in $\mathcal{G}$ $\big($i.e., $\{Q_o^{(i,j)}(m) : (i,j) \in \mathcal{L}\}\big)$ are stable under the GM-RBDS policy if*

$$\overline{|G_m^{max}|} < \frac{\overline{N_{\mu_1}}}{2|\mathcal{P}_{max}| + 1}.$$

The proof of Theorem 2.2 is as follows. Let $\mathcal{T}_{\max}$ be some spanning tree of $\mathcal{G}_{\max}$ rooted at $n_{\max}$. Therefore, by definition, the height of $\mathcal{T}_{\max}$ is 2, and it has the structure shown in Figure 2-7. Let the grants transmitted by the nodes in $\mathcal{N}_{\max}$ during some scheduling cycle be denoted as shown in Figure 2-7. Grant G^0 is transmitted by the node in $\mathcal{T}_{\max}$ whose height is 0 (i.e., $n_{\max}$). Grants G^1, G^2, ..., $G^{|\mathcal{S}_1^0|}$ are transmitted by the nodes in $\mathcal{S}_1^0$, which is the set of nodes whose height is 1 (i.e., $\mathcal{S}_1^0 = \mathcal{S}_1^{n_{\max}}$). Grants $G^{i,1}$, $G^{i,2}$, ..., $G^{i,|\mathcal{S}_1^i|}$ are transmitted by the nodes in $\mathcal{S}_1^i$, which is the set of nodes whose height is 2 and are 1-hop neighbors of the i^{th} node in $\mathcal{S}_1^0$.

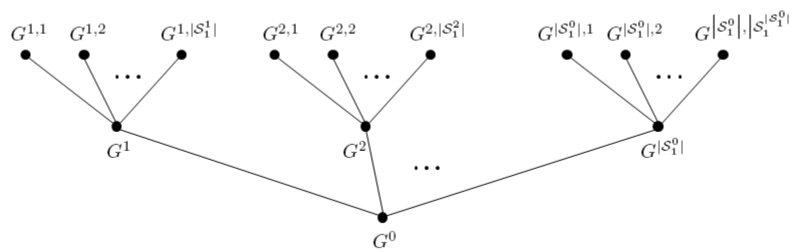
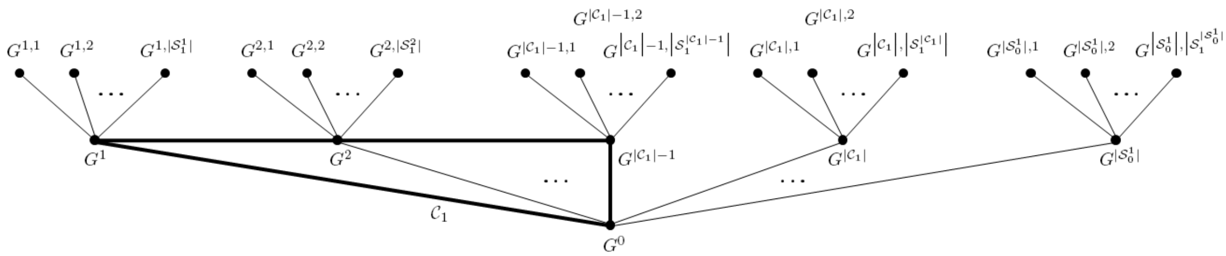


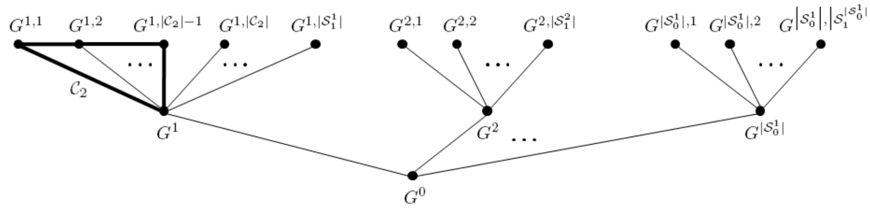
Figure 2-7. $\mathcal{G}_{\max}$'s spanning tree $\mathcal{T}_{\max}$

When adding one of the links in $\mathcal{G}_{\max} \backslash \mathcal{T}_{\max}$ to $\mathcal{T}_{\max}$, a cycle is created. This cycle can be of only one of the types shown in Figure 2-8A, Figure 2-8B, and Figure 2-8C. If another link is added to $\mathcal{T}_{\max}$, the existing cycle may be made longer or another cycle of only one of the specified types is created. This process repeats itself as all the links in $\mathcal{G}_{\max} \backslash \mathcal{T}_{\max}$ are added to $\mathcal{T}_{\max}$. Therefore, $\mathcal{G}_{\max}$ may contain one or more cycles which are of only one type each.

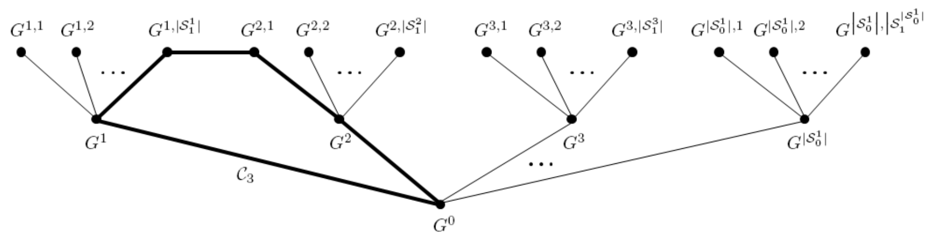
The cycle types in $\mathcal{G}_{\max}$ are defined as follows. A type-1 cycle (cycle $\mathcal{C}_1$ in Figure 2-8A) contains the root node and two or more nodes whose height is 1 only. A type-2



A Type-1 Cycle



B Type-2 Cycle



C Type-3 Cycle

Figure 2-8. $\mathcal{G}_{\text{max}}$'s cycle types

cycle (cycle $\mathcal{C}_2$ in Figure 2-8B) contains one node whose height is 1 and two or more nodes whose height is 2 only. A type-3 cycle (cycle $\mathcal{C}_3$ in Figure 2-8C) contains the root node, two or more nodes whose height is 1, and an even number of nodes whose height is 2 such that none of these nodes are 1-hop neighbors if they are connected to the same node of height 1.

Let i be some node in $\mathcal{N}$. Let $\mathcal{P}$ be some path in the undirected graph induced by $\mathcal{S}^i_{\leq 2}$ that originates at node i. $|G|_{\mathcal{P}}$ denotes the increase on $Q^i_o(n)$ when only the grants transmitted by the nodes in path $\mathcal{P}$ during a scheduling cycle are considered. Consider the difference between the output-queue lengths of every node of height 2 in $\mathcal{G}_{\text{max}}$ and n_{max} at the beginning of a scheduling cycle. Let this difference be denoted by $\Delta Q^{i,j}$ for the node of height 2 that transmits grant $G^{i,j}$.

Expected $|G|_{\mathcal{P}}$ when no cycles are considered. Consider the paths in $\mathcal{T}_{\max}$. The maximum $|G|_{\mathcal{P}}$ in $\mathcal{T}_{\max}$, denoted by $|G|_{\mathcal{T}_{\max}}$, is upper bounded as follows.

$$|G|_{\mathcal{T}_{\max}} \le \max_{\substack{i=1,2,\ldots,|\mathcal{S}_1^0| \\ j=1,2,\ldots,|\mathcal{S}_1^i|}} \left(|G^i| + |G^{i,j}| + [\Delta Q^{i,j}]^+\right) + |G^0|$$

Therefore, $|G|_{\mathcal{T}_{\max}}$ is upper bounded by the summation of three grant lengths and an output-queue difference that have maximum total length among all the paths from the leaf nodes to the root node. Let those grant lengths and output-queue difference be G^{i_m}, G^{i_m,j_m}, and $\Delta Q^{i_m,j_m}$.

Let n_2 be the node that transmits grant G^{i_m,j_m}, and let n_1 be the node that connects n_2 and $n_{\max}$. When no cycles are considered, $\Delta Q^{i_m,j_m}$ can be at most the summation of two grants. These two grants correspond to one of the paths in n_2's 2-hop neighborhood that has no nodes in $n_{\max}$'s 2-hop neighborhood. The grants cause an increase on the output-queues of n_2 that is not considered in $|G|_{\mathcal{T}_{\max}}$ until n_1 transmits its grant. Therefore, $\mathrm{E}\left[[\Delta Q^{i_m,j_m}]^+\right] \le 2\mathrm{E}[|G_m^{max}|]$, and the expected value of $|G|_{\mathcal{T}_{\max}}$ is upper bounded as follows.

$$\mathrm{E}\left[|G|_{\mathcal{T}_{\max}}\right] \le \mathrm{E}\left[|G^{i_m}| + |G^{i_m,j_m}| + |G^0| + [\Delta Q^{i_m,j_m}]^+\right] \le 5\mathrm{E}[|G_m^{max}|]$$

Expected $|G|_{\mathcal{P}}$ when only type-1 cycles are considered. Consider the paths in Figure 2-8A. The maximum $|G|_{\mathcal{P}}$, denoted by $|G|_{\mathcal{C}_1}$, is upper bounded as follows.

$$|G|_{\mathcal{C}_1} \le \max\left(|G|_{\mathcal{T}_{\max}}, \max_{\substack{i=1,2,\ldots,|\mathcal{C}_1|-1 \\ j=1,2,\ldots,|\mathcal{S}_1^i|}} \left(|G^{i,j}| + [\Delta Q^{i,j}]^+\right) + \sum_{G\in\mathcal{C}_1} |G|\right)$$

Therefore, $|G|_{\mathcal{C}_1}$ is upper bounded by either $|G|_{\mathcal{T}_{\max}}$ or by the summation of all the grant lengths along $\mathcal{C}_1$, a grant length of a leaf node, and an output-queue difference. Let T be the event that $|G|_{\mathcal{C}_1}$ is upper bounded by $|G|_{\mathcal{T}_{\max}}$, and let G^{i_m,j_m} and $\Delta Q^{i_m,j_m}$ be the grant and output-queue difference that satisfy $\max_{\substack{i=1,2,\ldots,|\mathcal{C}_1|-1 \\ j=1,2,\ldots,|\mathcal{S}_1^i|}} \left(|G^{i,j}| + [\Delta Q^{i,j}]^+\right)$.

Let n_2 be the node that transmits grant G^{i_m,j_m}, and let n_1 be the node that connects n_2 and $n_{\max}$. When only type-1 cycles are considered, $\Delta Q^{i_m,j_m}$ can be at most the summation of the grants across a type-1 cycle. These grants correspond to one of the cycles in n_2's 2-hop neighborhood that has no nodes in $n_{\max}$'s 2-hop neighborhood. Let $\mathcal{C}_{\text{aux}}$ be this cycle. The grants cause an increase on the output-queues of n_2 that is not considered in $|G|_{\mathcal{T}_{\max}}$ until n_1 transmits its grant. Therefore, $\mathsf{E}\big[[\Delta Q^{i_m,j_m}]^+\big] \leq |\mathcal{C}_{\text{aux}}|\mathsf{E}[|G_m^{max}|]$, and the expected value of $|G|_{\mathcal{C}_1}$ is upper bounded as follows[19] .

$$
\begin{aligned}
\mathsf{E}\big[|G|_{\mathcal{C}_1}\big] &\leq \mathsf{E}_T\big[|G|_{\mathcal{T}_{\max}}\big]\mathsf{P}\big[T\big] + \mathsf{E}_{\text{not}\,T}\Big[|G^{i_m,j_m}| + \Delta Q^{i_m,j_m} + \sum_{G\in\mathcal{C}_1}|G|\Big]\big(1-\mathsf{P}\big[T\big]\big) \\
&\leq 5\mathsf{E}[|G_m^{max}|]\mathsf{P}\big[T\big] + \big(1+|\mathcal{C}_{\text{aux}}|+|\mathcal{C}_1|\big)\mathsf{E}\big[|G_m^{max}|\big]\big(1-\mathsf{P}\big[T\big]\big) \\
&\leq \big(|\mathcal{C}_1|+|\mathcal{C}_{\text{aux}}|+1\big)\mathsf{E}\big[|G_m^{max}|\big]
\end{aligned}
$$

Expected $|G|_{\mathcal{P}}$ when only type-2 cycles are considered. Consider the paths in Figure 2-8B. The maximum $|G|_{\mathcal{P}}$, denoted by $|G|_{\mathcal{C}_2}$, is upper bounded as follows.

$$
|G|_{\mathcal{C}_2} \leq \max\Big\{|G|_{\mathcal{T}_{\max}}, \max_{i=1,2,\ldots,|\mathcal{C}_2|-1}\Big([\Delta Q^{i,j}]^+\Big) + \sum_{G\in\mathcal{C}_2}|G| + |G^0|\Big\}
$$

Therefore, $|G|_{\mathcal{C}_2}$ is upper bounded by either $|G|_{\mathcal{T}_{\max}}$ or by the summation of an output-queue difference, all the grant lengths along $\mathcal{C}_2$, and $|G^0|$. Let T be the event that $|G|_{\mathcal{C}_2}$ is upper bounded by $|G|_{\mathcal{T}_{\max}}$, and let $\Delta Q^{i_m,j_m}$ be the output-queue difference that satisfies $\max_{i=1,2,\ldots,|\mathcal{C}_2|-1}\big([\Delta Q^{i,j}]^+\big)$.

Let n_2 be the node that transmits grant G^{i_m,j_m}, and let n_1 be the node that connects the type-2 cycle n_2 belongs to and $n_{\max}$. When only type-2 cycles are considered, $\Delta Q^{i_m,j_m}$ can be at most the summation of the grants along a type-2 cycle. This cycle, denoted by $\mathcal{C}_{\text{aux}}$, corresponds to a type-2 cycle in n_2's 2-hop neighborhood that has no nodes in

[19] $\mathsf{E}_X[Y]$ denotes the expectation of Y given X.

$n_{\max}$'s 2-hop neighborhood. The grants cause an increase on the output-queue of n_2 that is not considered in $|G|_{\mathcal{T}_{\max}}$ until n_1 transmits its grant. Therefore, $\mathsf{E}\big[[\Delta Q^{i_m,j_m}]^+\big] \leq |\mathcal{C}_{\text{aux}}|\mathsf{E}[|G_m^{max}|]$, and the expected value of $|G|_{\mathcal{C}_2}$ is upper bounded as follows.

$$
\begin{aligned}
\mathsf{E}\big[|G|_{\mathcal{C}_2}\big] &\leq \mathsf{E}_{T}\big[|G|_{\mathcal{T}_{\max}}\big]\mathsf{P}\big[T\big] + \mathsf{E}_{\text{not}\,T}\Big[\Delta Q^{i_m,j_m} + \sum_{G\in\mathcal{C}_2}|G| + |G^0|\Big]\big(1-\mathsf{P}\big[T\big]\big) \\
&\leq 5\mathsf{E}[|G_m^{max}|]\mathsf{P}\big[T\big] + \big(|\mathcal{C}_{\text{aux}}| + |\mathcal{C}_2| + 1\big)\mathsf{E}\big[|G_m^{max}|\big]\big(1-\mathsf{P}\big[T\big]\big) \\
&\leq \big(|\mathcal{C}_2| + |\mathcal{C}_{\text{aux}}| + 1\big)\mathsf{E}\big[|G_m^{max}|\big]
\end{aligned}
$$

Expected $|G|_{\mathcal{P}}$ when only type-3 cycles are considered. Consider the paths in Figure 2-8C. The maximum $|G|_{\mathcal{P}}$, denoted by $|G|_{\mathcal{C}_3}$, is upper bounded as follows.

$$
|G|_{\mathcal{C}_3} \leq \max\Big\{|G|_{\mathcal{T}_{\max}},\ \max_{\substack{i=1,2\\ j=1,2,\ldots,|\mathcal{S}_1^i|}}\Big(|G^{i,j}| + [\Delta Q^{i,j}]^+\Big) + \sum_{G\in\mathcal{C}_3}|G|\Big\}
$$

By doing an analysis similar to the one performed for the upper bounds of $\mathsf{E}\big[|G|_{\mathcal{C}_1}\big]$ and $\mathsf{E}\big[|G|_{\mathcal{C}_2}\big]$, it can be shown that $\mathsf{E}\big[|G|_{\mathcal{C}_3}\big]$ is upper bounded as follows, where $\mathcal{C}_{\text{aux}}$ is a type-3 cycle.

$$
\mathsf{E}\big[|G|_{\mathcal{C}_3}\big] \leq \big(|\mathcal{C}_3| + |\mathcal{C}_{\text{aux}}| + 1\big)\mathsf{E}\big[|G_m^{max}|\big]
$$

Therefore, when there are no cycles or there is only one cycle $\mathcal{C}$ of any type in $\mathcal{G}_{\max}$, the expected value of the maximum $|G|_{\mathcal{P}}$ is upper-bounded by $\big(\max\big(5, 2|\mathcal{C}| + 1\big)\big)\overline{|G_m^{max}|}$, where $\mathcal{C}$ is the cycle of maximum length among all the cycles in the 2-hop neighborhoods of the network (i.e., if $\mathcal{C}^i$ is the maximum-length cycle in $S_{\leq 2}^i$, then $\mathcal{C} = \operatorname{argmax}_{\mathcal{C}^i : i\in\mathcal{N}}|\mathcal{C}^i|\big)$. This upper-bound can also be expressed in terms of $\mathcal{P}_{\max}$ as $(2|\mathcal{P}_{\max}| + 1)\overline{|G_m^{max}|}$ (i.e., $2|\mathcal{P}_{\max}| + 1 = \max\big(5, 2|\mathcal{C}| + 1\big)\big)$.

This result can be directly extended to the case in which there are two or more cycles of any type.

Expected $|G|_{\mathcal{P}}$ in the general scenario. Consider the set $\mathcal{G}_{\mathcal{P}}$ of all paths in the 2-hop neighborhoods of the network and which start at the corresponding neighborhoods' root nodes when all the cycles are present (i.e., if $\mathcal{G}^i$ is the set of all paths in the undirected graph induced by $\mathcal{S}^i_{\leq 2}$ and that originate at node i, then $\mathcal{G}_{\mathcal{P}} = \bigcup_{i\in\mathcal{N}} \mathcal{G}^i$). The $|G|_{\mathcal{P}}$ of each of these paths is the summation of the length of the grants transmitted by the nodes along the path plus an output-queue difference if the height of the path's end node is 2. The increase on the output-queue $Q_o^{n_{\max}}(n)$ (i.e., $|\mathcal{H}^{n_{\max}}(n)|$) is upper-bounded by the maximum $|G|_{\mathcal{P}}$, which is denoted by $|G|_{\mathcal{P}}^{\max}$. The expected value of $|G|_{\mathcal{P}}^{\max}$ is upper bounded as follows, where $I_{\mathcal{P}}$ is an indicator that $\mathcal{P}$ in $\mathcal{G}_{\mathcal{P}}$ has the maximum $|G|_{\mathcal{P}}$.

$$
\begin{aligned}
\mathrm{E}\big[|G|_{\mathcal{P}}^{\max}\big] &= \sum_{\mathcal{P}\in\mathcal{G}_{\mathcal{P}}} \mathrm{E}\big[|G|_{\mathcal{P}}\big]\mathrm{P}\big[I_{\mathcal{P}}=1\big] \\
&\leq \sum_{\mathcal{P}\in\mathcal{G}_{\mathcal{P}}} \mathrm{E}\big[|G_m^{\max}|\big](2|\mathcal{P}|+1)\mathrm{P}\big[I_{\mathcal{P}}=1\big] \\
&= \mathrm{E}\big[|G_m^{\max}|\big]\big(2\sum_{\mathcal{P}\in\mathcal{G}_{\mathcal{P}}} |\mathcal{P}|\mathrm{P}\big[I_{\mathcal{P}}=1\big]+1\big) \\
&\leq \mathrm{E}\big[|G_m^{\max}|\big]\big(2|\mathcal{P}_{\max}|+1\big)
\end{aligned}
$$

The proof follows from Theorem 2.1 and by observing that

$$
\max_{j\in\mathcal{N}}\left(\overline{|\mathcal{H}^j|}-\overline{N^j_{\mu_1}}\right) = \overline{|\mathcal{H}^{n_{\max}}|}-\overline{N^{n_{\max}}_{\mu_1}} \leq \mathrm{E}\big[|G|_{\mathcal{P}}^{\max}\big]-\overline{N_{\mu_1}} \leq \big(2|\mathcal{P}_{\max}|+1\big)\mathrm{E}\big[|G_m^{\max}|\big]-\overline{N_{\mu_1}}.
$$

2.3.3.3 Sufficient condition for the stability of input-queues

The stability of the input-queues is always guaranteed by the GM-RBDS policy due to the following. A node always requests as many data-subframes as can be entirely covered with the packets waiting to be scheduled in its input-queues (GM-RBDS policy in Section 2.3.3), and all the requests are successfully granted with some probability greater than zero according to Eq. 2–28. Therefore, the input-queues, with some probability greater than zero, may become empty independently of their current length, and the condition given by Eq. 2–23 always holds.

2.3.3.4 Stable region and efficiency ratio of GM-RBDS

It remains to find the expected value of $|G_m^{\max}|$ in terms of the mean data-packet arrival rates $\{\lambda^{(i,j)} : (i,j) \in \mathcal{L}\}$ in order to calculate a stable region and a lower-bound for the efficiency ratio of the GM-RBDS policy using Theorem 2.2.

For every outgoing link of every node, the node requests the maximum number of data-subframes that can be entirely covered with the current number of data packets in the link's input-queue. Also, the length of the grant for every request is always equal to the request's length. Therefore, the expected value of a grant length is upper-bounded by the expected value of the length of its corresponding input-queue as given by Eq. 2–24, where m_{r}^i is the control-time-slot when the request for link (i,j) is transmitted.

$$\mathsf{E}\left[|G_m^{(i,j)}|\right] = \mathsf{E}\left[\left\lfloor \frac{Q_i^{(i,j)}(m_{\mathrm{r}}^i)}{m_{\mathrm{ds}}} \right\rfloor\right] \leq \mathsf{E}\left[\frac{Q_i^{(i,j)}(m_{\mathrm{r}}^i)}{m_{\mathrm{ds}}}\right] \tag{2–24}$$

Let $Q_{\mathrm{s}}^{(i,j)}(n)$ be $\frac{Q_i^{(i,j)}(m_n^j)}{m_{\mathrm{ds}}}$, where m_n^j is the control-time-slot used for node j's n^{th} scheduling-packet transmission. $Q_{\mathrm{s}}^{(i,j)}(n)$ is increased by the number of data packet arrivals to link (i,j) between every two consecutive node j's scheduling-packet transmissions $\big($i.e., $A_{\mathrm{s}}^{(i,j)}(n)$ as given by Eq. 2–14$\big)$, and it is decreased by the grants transmitted by node j for link (i,j). According to the GM-RBDS policy (Section 2.3.3), for any of these grants to be confirmed by node i, two independent events must take place. First, when node j transmits a scheduling packet, link (i,j)'s request has maximum length among the unexpired requests transmitted by all of node j's incoming links. Second, when node i receives node j's scheduling packet, none of the unexpired grants received by node i overlap with the grant for link (i,j) carried by the scheduling packet. The occurrence of each of these two events at node j's n^{th} scheduling-packet transmission is denoted by $I_{Q_{\max}}^{(i,j)}(n)$ and $I_{\mathrm{s}}^{(i,j)}(n)$ respectively.

The grant length is equal to the number of data-subframes covered by the input-queue length of link (i,j) $\left(\text{i.e., } Q_i^{(i,j)}\right)$ at the control-time-slot that node i transmits link (i,j)'s request $\left(\text{i.e., } m_\text{r}^i\right)$. Therefore, the grant length can be expressed as $Q_\text{s}^{(i,j)}(n)$ minus the following: the number of data-subframes covered by the data packet arrivals to link (i,j) since the request is transmitted by node i until the grant is transmitted by node j $\left(\text{i.e., } \lfloor\frac{Q_i^{(i,j)}(m_n^j)-Q_i^{(i,j)}(m_\text{r}^i)}{m_\text{ds}}\rfloor\right)$. Let this number of data-subframes be denoted by $\Delta Q_\text{s}^{(i,j)}(n)$ for node j's n^th scheduling-packet transmission. $Q_\text{s}^{(i,j)}(n)$ is updated according to Eq. 2–8 and Eq. 2–14 as follows.

$$Q_\text{s}^{(i,j)}(n+1) = Q_\text{s}^{(i,j)}(n)+\frac{A_\text{s}^{(i,j)}(n+1)}{m_\text{ds}}-I_{Q_\text{max}}^{(i,j)}(n+1)I_\text{s}^{(i,j)}(n+1)\left(Q_\text{s}^{(i,j)}(n)-\Delta Q_\text{s}^{(i,j)}(n)\right) \quad (2\text{–}25)$$

In order to find the expected value of $Q_\text{s}^{(i,j)}(n)$, it is necessary to calculate the probability that $I_{Q_\text{max}}^{(i,j)}(n)$ and $I_\text{s}^{(i,j)}(n)$ are each equal to one.

Probability that $I_\text{s}^{(i,j)}(n)$ equals one. Let $\mathcal{T}_\text{u}^{(\text{i},j)}(m)$ and $\mathcal{T}_\text{u}^{(i,\text{j})}(m)$ (Eq. 2–1) be the unavailable-data-time-slots tables for link (i,j) maintained by nodes i and j respectively. These two tables are not necessarily equal due to the 1-hop neighbors of j hidden from i and vice versa. Let the length of an unavailable-data-time-slots table be the number of frames in the range that starts at the current frame and ends at the latest unavailable frame in the table. Let this length be denoted by $|\mathcal{T}_\text{u}^{(i,j)}(m)|$. The event that grants for link (i,j) transmitted by node j do not overlap any of the unexpired grants in node i when this node receives link (i,j)'s grants $\left(\text{i.e., } I_\text{s}^{(i,j)}(n) = 1\right)$ is equivalent to the following event: $|\mathcal{T}_\text{u}^{(i,\text{j})}(m_n^j)|$ is greater than or equal to $|\mathcal{T}_\text{u}^{(\text{i},j)}(m_n^j)|$. Let $\mathcal{T}_\text{s}^{i}(m_n^j)$ be the subset of $\mathcal{T}_\text{u}^{(\text{i},j)}(m_n^j)$ that contains all the grants received by node i from the moment it sends link (i,j)'s request to node j (i.e., m_r^i) to the moment node j replies with its n^th transmitted scheduling packet (i.e., m_n^j), which contains link (i,j)'s grant. $|\mathcal{T}_\text{u}^{(i,\text{j})}(m_n^j)|$ is greater than or equal to $|\mathcal{T}_\text{u}^{(\text{i},j)}(m_n^j)|$ only if the maximum expiration data-subframe number in

$\mathcal{T}_\mathrm{s}^i(m_n^j)$[20] is less than the maximum expiration data-subframe number in $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m_n^j)$. This is due to the minimum data-subframe number specified by node i that the request can be granted (GM-RBDS policy in Section 2.3.3). This minimum is no longer valid if any other grants that occupy data-subframes greater than or equal to it are received by node i and not by node j before node j sends the grant for link (i, j). That is, if such grants are received by node i and make $|\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m_n^j)|$ be greater than $|\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m_n^j)|$, node j grants data-subframes that were already included in the grants received by node i, so node i is not able to confirm node j's grant.

$\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m)$ and $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m)$ are updated with the grants transmitted by the nodes in $\mathcal{S}_{\leq 1}^i$ and $\mathcal{S}_{\leq 1}^j$ respectively. Consider the sets $\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j$, $\mathcal{S}_{\leq 1}^i \cap \mathcal{S}_{\leq 1}^j$, and $\mathcal{S}_1^j \backslash \mathcal{S}_{\leq 1}^i$. Given that the grants transmitted by the nodes in $\mathcal{S}_{\leq 1}^i \cap \mathcal{S}_{\leq 1}^j$ are received by both i and j, they do not make $\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m)$ and $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m)$ have different lengths. Therefore, $l_\mathrm{s}^{(i,j)}(n)$ is independent of such grants. The grants transmitted by the nodes in $\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j$ affect $\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m)$ and do not affect $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m)$, and the grants transmitted by the nodes in $\mathcal{S}_1^j \backslash \mathcal{S}_{\leq 1}^i$ affect $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m)$ and do not affect $\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m)$. Therefore, $l_\mathrm{s}^{(i,j)}(n)$ depends on such grants only. Let $\mathcal{T}_\mathrm{s}^{i\backslash j}(m_n^j)$ be the subset of $\mathcal{T}_\mathrm{s}^i(m_n^j)$ that contains all the grants transmitted by the nodes in $\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j$. When node j sends link (i, j)'s grant, the maximum expiration data-subframe number in $\mathcal{T}_\mathrm{s}^i(m_n^j)$ is less than the maximum expiration data-subframe number in $\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m_n^j)$ if it is not an f_e component of any of the grants in $\mathcal{T}_\mathrm{s}^{i\backslash j}(m_n^j)$. Therefore, the probability that $|\mathcal{T}_\mathrm{u}^{(i,\mathrm{j})}(m_n^j)|$ is greater than or equal to $|\mathcal{T}_\mathrm{u}^{(\mathrm{i},j)}(m_n^j)|$ is lower-bounded by the probability that $\mathcal{T}_\mathrm{s}^{i\backslash j}(m_n^j)$ is empty. Considering that the nodes in $S_{\leq 2}^i$ transmit only one scheduling packet per

[20] The expiration data-subframe numbers in $\mathcal{T}_\mathrm{s}^i(m_n^j)$ correspond to the f_e components of the grants they contain (Definition 2).

scheduling cycle and that the order of these transmissions is independent across the scheduling cycles, the lower-bound is given as follows[21] .

$$\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n)=1\right] \geq \mathsf{P}\left[\mathcal{T}_{\mathrm{s}}^{i\setminus j}(m_n^j)=\emptyset\right] = \frac{1}{1+\left|\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j\right|} \tag{2–26}$$

In order to find an upper-bound for the expected value of all the input-queue lengths, in the following, it is assumed that $\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n)=1\right]$ equals the lower-bound in Eq. 2–26. This is true because the input-queues decrease at a lower rate under this assumption.

Probability that $I_{Q_{\mathbf{max}}}^{(i,j)}(n)$ equals one. The event that link (i,j)'s request is granted by node j $\left(\text{i.e., } I_{Q_{\max}}^{(i,j)}(n)=1\right)$ depends on the lengths of the requests in $\bigcup_{i\in\mathcal{S}_1^j}\mathcal{T}_{\mathrm{r}}^{(i,j)}(m_n^j)$ only. When node j transmits a scheduling packet, there can be only one unexpired request per incoming link $\left(\text{i.e., } \{(i,j) : i \in \mathcal{S}_1^j\}\right)$. This is due to the following. Every node in $\mathcal{S}_{\leq 2}^j$ transmits only one scheduling packet per scheduling cycle, there is only one request to node j in every scheduling packet, and this request always expires at the next scheduling-packet transmission of node i, where i is any node in $\mathcal{S}_1^j$ (GM-RBDS policy in Section 2.3.3).

Let m_n^j and m_{n-1}^j be the control-time-slots used by node j for its current and previous scheduling-packet transmissions respectively, and let m_{r}^i be the control-time-slot used by node i in $\mathcal{S}_1^j$ for its scheduling-packet transmission previous to m_n^j. At m_n^j, node j grants the longest request among the last requests sent by each of its 1-hop neighbors (i.e., requests sent to node j before m_n^j). Let i_{s} be the 1-hop neighbor that sent the longest request. Node i_{s} is given as follows.

[21] The lower-bound is calculated based on the fact that the event that $\mathcal{T}_{\mathrm{s}}^{i\setminus j}(m_n^j)$ is empty is equivalent to the following event: within a scheduling cycle, node j's scheduling-packet transmission precedes the scheduling-packet transmissions of the nodes in $\left|\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j\right|$.

$$i_{\mathrm{s}} = \operatorname*{argmax}_{i \in \mathcal{S}_1^j} \left\{ Q_{\mathrm{s}}^{(i,j)}(n) - \Delta Q_{\mathrm{s}}^{(i,j)}(n) \right\} = \operatorname*{argmax}_{i \in \mathcal{S}_1^j} \left\{ \frac{Q_{\mathrm{i}}^{(i,j)}(m_{\mathrm{r}}^i)}{m_{\mathrm{ds}}} \right\}$$

Therefore, given that the m_{r}^i's are i.i.d. across the scheduling cycles and that grants decrease input-queue lengths with probability $\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n) = 1\right]$, the probability that node i's request is granted depends on the rate that the input-queues of node j's incoming links increase as follows.

$$\mathsf{P}\left[I_{Q_{\max}}^{(i,j)}(n) = 1\right] = \mathsf{P}[i = i_{\mathrm{s}}] = \frac{\lambda^{(i,j)}\left(\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n) = 1\right]\right)^{-1}}{\sum_{k \in \mathcal{S}_1^j} \lambda^{(k,j)}\left(\mathsf{P}\left[I_{\mathrm{s}}^{(k,j)}(n) = 1\right]\right)^{-1}} \tag{2–27}$$

Expected value of $|G_m^{\mathbf{max}}|$. From Eq. 2–26 and Eq. 2–27 and given the independence of $I_{\mathrm{s}}^{(i,j)}(n)$ and $I_{Q_{\max}}^{(i,j)}(n)$[22] and the assumption that $\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n) = 1\right]$ equals the lower-bound in Eq. 2–26, the probability that node i's request for link (i,j) is granted by node j is given by Eq. 2–28, where $s^j \triangleq \max\left\{1 + \left|\mathcal{S}_1^k \backslash \mathcal{S}_{\leq 1}^j\right| : k \in \mathcal{S}_1^j\right\}$, and λ^j is the total packet-arrival rate at node j $\left(\text{i.e., } \lambda^j = \sum_{i \in \mathcal{S}_1^j} \lambda^{(i,j)}\right)$.

$$\begin{aligned} \mathsf{P}\left[I_{Q_{\max}}^{(i,j)}(n) = 1, I_{\mathrm{s}}^{(i,j)}(n) = 1\right] &= \mathsf{P}\left[I_{Q_{\max}}^{(i,j)}(n) = 1\right]\mathsf{P}\left[I_{\mathrm{s}}^{(i,j)}(n) = 1\right] \\ &\geq \frac{\lambda^{(i,j)}}{\sum_{k \in \mathcal{S}_1^j} \lambda^{(k,j)}\left(1 + \left|\mathcal{S}_1^k \backslash \mathcal{S}_{\leq 1}^j\right|\right)} \geq \frac{\lambda^{(i,j)}}{s^j \lambda^j} \end{aligned} \tag{2–28}$$

[22] The independence of $I_{\mathrm{s}}^{(i,j)}$ and $I_{Q_{\max}}^{(i,j)}$ comes from the following. The GM-RBDS policy calculates the grants transmitted by the nodes in $\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j$ and $\mathcal{S}_1^j \backslash \mathcal{S}_{\leq 1}^i$ without considering the number of unscheduled packets of link (i,j) $\left(\text{i.e., } Q_i^{(i,j)}\right)$. The probability that a grant for link (i,j) overlaps the unexpired grants received by node i depends on the grants transmitted by the nodes in $\mathcal{S}_1^i \backslash \mathcal{S}_{\leq 1}^j$ and $\mathcal{S}_1^j \backslash \mathcal{S}_{\leq 1}^i$ only (Section 2.3.3.4). Therefore, this probability does not depend on the number of unscheduled packets of link (i,j), i.e., $I_{\mathrm{s}}^{(i,j)}$ is independent of $Q_i^{(i,j)}$. The same analysis can be done for any incoming link of node j, so $I_{\mathrm{s}}^{(i,j)}$ is independent of $Q_i^{(k,j)}$, where node k is any 1-hop neighbor of node j. Therefore, $I_{\mathrm{s}}^{(i,j)}$ is independent of the event that $Q_i^{(i,j)}$ has maximum length among the input queues of the incoming links of node j, i.e., $I_{\mathrm{s}}^{(i,j)}$ is independent of $I_{Q_{\max}}^{(i,j)}$.

The expected value of the length of any grant can upper-bounded using Eq. 2–24, Eq. 2–25, and Eq. 2–28 as follows.

Lemma 1. *The expected value of grant* $\left|G_m^{(i,j)}\right|$ *is upper-bounded as follows.*

$$\overline{\left|G_m^{(i,j)}\right|} = E\left[Q_s^{(i,j)}(n) - \Delta Q_s^{(i,j)}(n)\right] \leq s^j \lambda^j \overline{N_{\mu_1}}$$

See Appendix C for the proof of Lemma 1.

A sufficient condition for the stability of a GM-RBDS network can be found using Lemma 1.

GM-RBDS stable region and efficiency ratio.

Theorem 2.3. *Let* ν *be given by* $\nu = (2|\mathcal{P}_{max}| + 1) \max\left\{s^j : j \in \mathcal{N}\right\}$*. GM-RBDS wireless network* $\mathcal{G}$ *is stable if*

$$\sum_{i \in \mathcal{S}_1^j} \lambda^{(i,j)} < \frac{1}{\nu} \quad \forall \ j \in \mathcal{N}. \tag{2–29}$$

The proof of Theorem 2–29 is as follows. The sufficient condition in Theorem 2.2 for the stability of the output-queues in $\mathcal{G}$ is equivalent to

$$\overline{\left|G_m^{(i,j)}\right|} < \frac{\overline{N_{\mu_1}}}{2|\mathcal{P}_{\max}| + 1} \quad \forall \ (i,j) \in \mathcal{L}.$$

According to Lemma 1, this condition is satisfied if

$$\sum_{k \in \mathcal{S}_1^j} \lambda^{(k,j)} < \frac{1}{s^j(2|\mathcal{P}_{\max}| + 1)} \quad \forall \ j \in \mathcal{N}.$$

This concludes the proof of Theorem 2–29.

Remark. Notice that the sufficient condition for stability given by Eq. 2–29 is of the same form of the condition for the non-reservation-based greedy policies analyzed in [84] (Eq. (4) in [84]). That is, the total packet-arrival rate of a set of interfering links needs to be lower than some constant in order to guarantee stability, and the constant depends on some characteristic of the network topology (i.e., $|\mathcal{P}_{\text{max}}|$ and s^j for the GM-RBDS policy, and κ for the greedy policies in [84]).

Finally, a lower-bound for the efficiency ratio of the GM-RBDS policy can be found based on the region in which the data-packet arrival rates satisfy the stability condition given by Eq. 2–29.

Theorem 2.4. *Consider RBDS wireless network $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ with 1-hop traffic, its optimal capacity region Λ, and its maximum number κ of non-interfering links in the interference set $\mathcal{I}^{(i,j)}$ of any link (i, j) in $\mathcal{L}$. $\mathcal{G}$ is stable under the GM-RBDS policy for any set of data-packet arrival rates $\{\lambda^{(i,j)} : (i, j) \in \mathcal{L}\}$ that lies inside the region $\frac{\Lambda}{\nu\kappa}$.*

The proof of Theorem 2.4 is as follows. Let $\mathcal{C}_1$ be the set of data-packet arrival rates that, for all (i, j) in $\mathcal{L}$, satisfy Eq. 2–29, and let $\mathcal{C}_2$ be the set of data-packet arrival rates that, for all (i, j) in $\mathcal{L}$, satisfy

$$\sum_{(i,j)\in\mathcal{I}^{(i,j)}} \lambda^{(i,j)} < \frac{1}{\nu} \quad \forall \;\; (i, j) \in \mathcal{L}.$$

Therefore, $\mathcal{C}_2 \subseteq \mathcal{C}_1$, and network $\mathcal{G}$ is stable for any set of data-packet arrival rates that lies in $\mathcal{C}_2$.

In [84], it is shown that a necessary condition for network stability under any scheduling policy is

$$\sum_{(i,j)\in\mathcal{I}^{(i,j)}} \lambda^{(i,j)} < \kappa \quad \forall \;\; (i, j) \in \mathcal{L}$$

Therefore, network $\mathcal{G}$ is stable when the set of data-arrival rates lies inside the region $\frac{\Lambda}{\nu\kappa}$. This concludes the proof of Theorem 2.4.

Therefore, the stability of IEEE 802.16 mesh networks under the GM-RBDS policy (Theorem 2.3) depends on ν only, which is a local characteristic of the topology of the network. That is, ν depends on the 2-hop neighborhoods of the network but not on the network as a whole.

2.4 Simulation Results

The stability of an IEEE 802.16 mesh network under the GM-RBDS policy with 1-hop traffic and Poisson data-packet arrivals is evaluated by means of simulation using the simulator proposed in Chapter 5. The network topology is a grid of 16 nodes (i.e., 4×4) and 48 links. The frame-structure parameters m_{cs} and m_{ds} are set at 9 and 256 respectively[23] . The frame length is 10 ms, and a total of 3000 frames is simulated.

Figure 2-9 shows the average input and output queue lengths for increasing traffic loads. The traffic loads are $\frac{4}{24}$, $\frac{8}{24}$, $\frac{12}{24}$, $\frac{16}{24}$, $\frac{17}{24}$, $\frac{18}{24}$, $\frac{20}{24}$, and 1 of the optimal capacity. According to Figure 2-9A, the input-queues are always stable (i.e., they do not grow indefinitely) as expected (Section 2.3.3.3). On the other hand, according to Figure 2-9B, the output queues become unstable for traffic loads greater than $\frac{18}{24}$. Therefore, the GM-RBDS policy reaches an efficiency ratio of approximately $\frac{18}{24}$ in the simulation. This is in agreement with the theoretical lower-bound. The efficiency ratio is lower-bounded by $\frac{1}{153}$ according to Theorem 2.4 (i.e., $\mathcal{P}_{\text{max}} = 8$, $\max_{j \in \mathcal{N}} s^j = 3$, and $\kappa = 3$).

2.4.1 GM-RBDS Throughput Evaluation

The performance of the GM-RBDS policy is compared in Figure 2-10A[24] with the W and Enhanced-Local-Greedy-Scheduling (ELGS) policies proposed in [36] and [35]. From the policies reviewed in Section 2.1, W and ELGS were selected for the comparison because they consider the same interference model of IEEE 802.16 mesh networks (i.e., 2-hop interference model), which is the model of GM-RBDS too. Also, the GM-RBDS, W, and ELGS policies require the same control information for making scheduling decisions. Each node requires the queue lengths of the nodes in

[23] In the simulation, the NextXmtMx and XmtHoldoffExponent parameters of the election algorithm [1] are set at 3 and 4 respectively.

[24] In Figure 2-10, the queue lengths have been normalized to their respective maximum values in order to make the comparison.

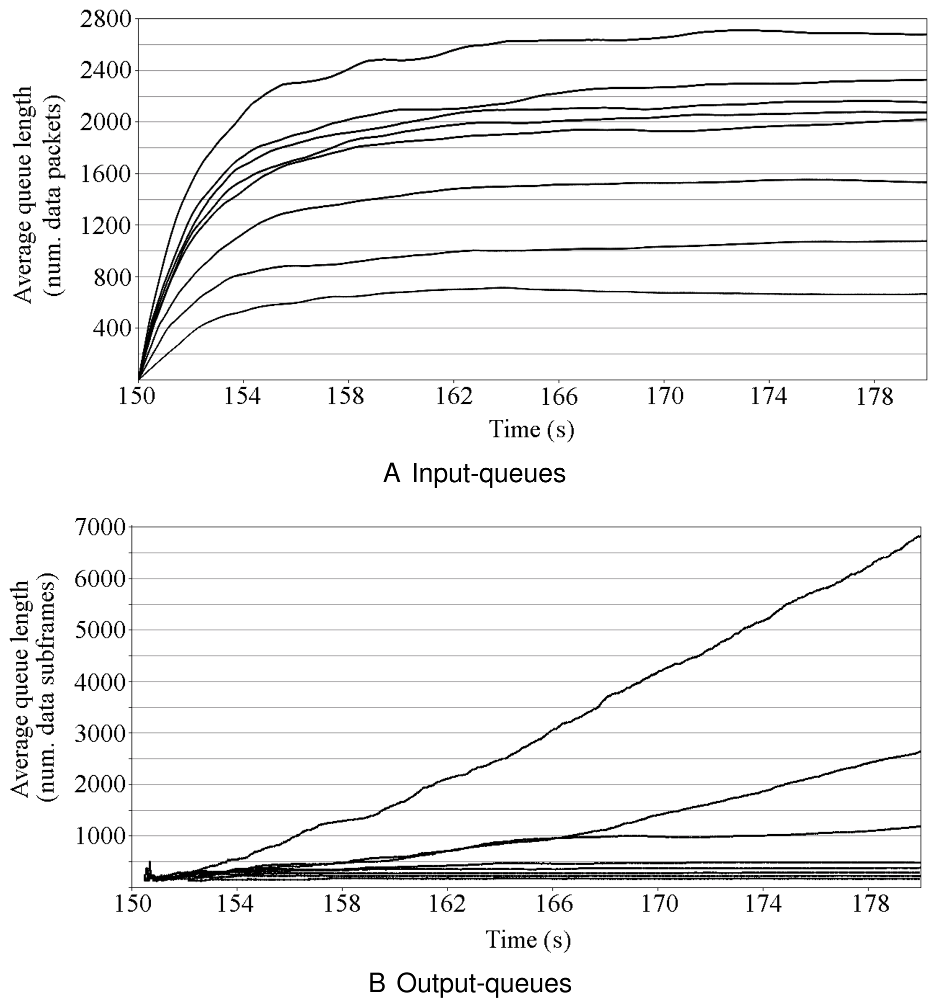


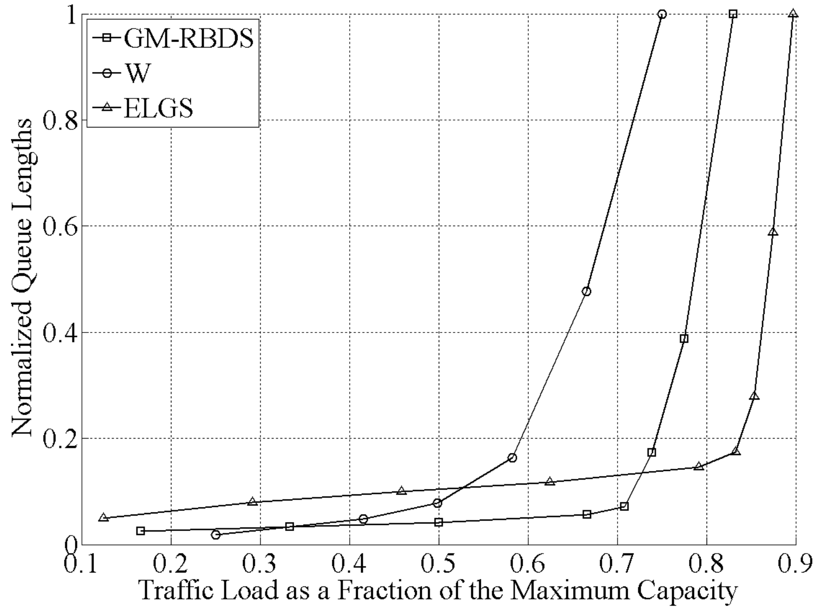
Figure 2-9. Average input and output queue lengths for increasing traffic loads

its two-hop neighborhood for making any link scheduling. Figure 2-10A includes the best performance curves of the W and ELGS policies reported in [36] and [35] when the overhead (i.e., the fraction of a frame that is dedicated to the control subframe) is not considered. It shows that GM-RBDS outperforms W. That is, as the traffic load approaches the optimal capacity of the network, the queues under policy W grow faster than the queues under policy GM-RBDS. Therefore, the GM-RBDS policy is able to maintain the network stable under higher throughput levels when compared with W. On the other hand, ELGS outperforms GM-RBDS in Figure 2-10A. However, the capacity achieved by the ELGS policy can be reduced considerably by its overhead when compared with GM-RBDS. The GM-RBDS policy does not require the amount of overhead required by the ELGS policy. This is due to its ability for scheduling any future data-subframes. Figure 2-10B shows that if the maximum node degree is 8 (i.e., $\Delta = 8$) and the control packets are formatted according to the IEEE-802.16-mesh-networks standard [1], the GM-RBDS and ELGS policies perform equally. For higher values of Δ (i.e., $\Delta \geq 8$), the GM-RBDS policy outperforms the ELGS policy[25] . This is shown in Figure 2-11. Figure 2-11A shows the traffic load necessary to reach the maximum average queue length reported in Figure 2-10 as a function of Δ, and Figure 2-11B shows the overhead as a function of Δ. The overhead is specified as the portion of the total bandwidth it takes. According to Figure 2-11, the GM-RBDS policy achieves greater throughput than the ELGS policy when $\Delta \geq 8$. The reason is that the GM-RBDS overhead grows more slowly with Δ than the ELGS overhead does. This is shown in Figure 2-11B.

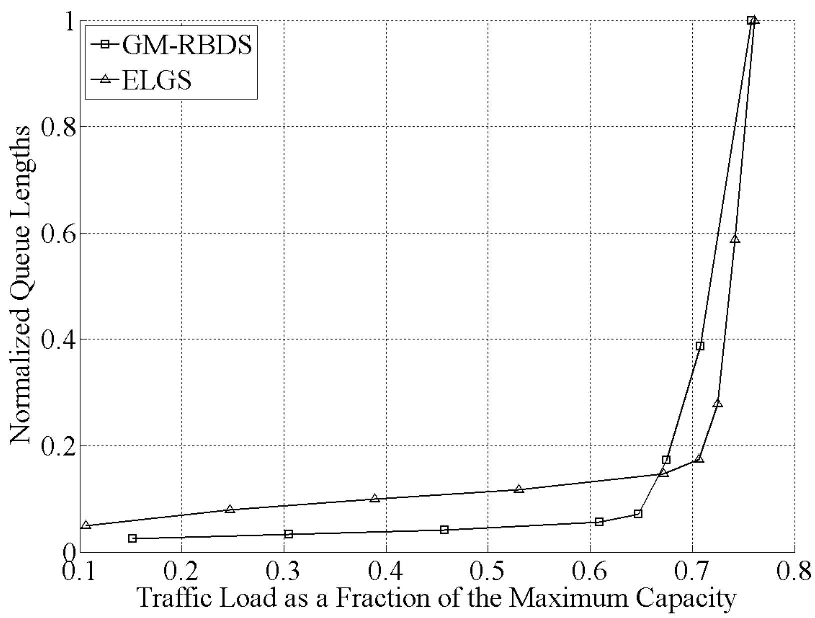
2.4.2 Overhead Comparison of GM-RBDS and Enhanced Local Greedy Scheduling

In the ELGS policy, the links are scheduled on a largest-queue-first basis. This is done locally in order to make the policy distributed. Therefore, the information required by the nodes is the queue length of the interfering links which correspond to one integer each. Also, the nodes need to know link identifications (ID) in order to solve the problem of two or more links with the same queue length [35]. This ID is another integer per link that the nodes need to know. In the worst-case scenario (i.e., when the highest number of links of nodes in a 2-hop neighborhood are activated), there are $\Delta(\Delta - 1)$ link activations. This case occurs as follows. Consider the node with maximum degree and activate one of the outgoing links of every 2-hop neighbor. This is shown

[25] This also implies that the GM-RBDS policy outperforms the W policy because the overhead of the W policy is greater than the overhead of the ELGS policy [35].

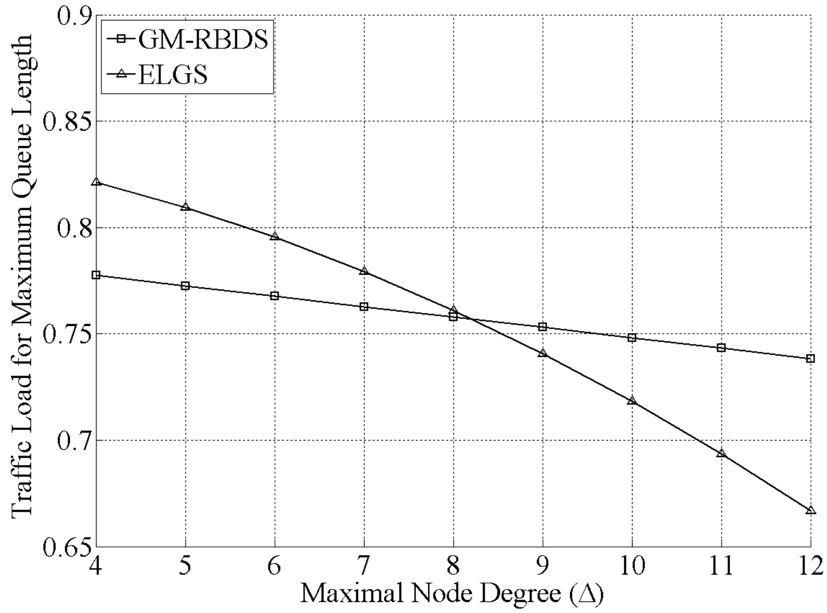


A With no overhead

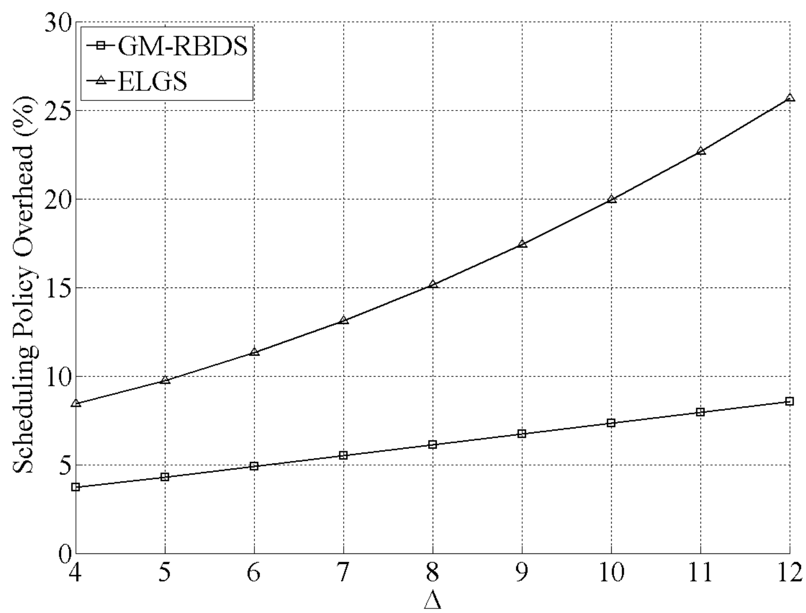


B With overhead ($\Delta = 8$)

Figure 2-10. Performance comparison of the greedy-maximal reservation-based-distributed-scheduling (GM-RBDS) policy, policy W [36], and the enhanced-local-greedy-scheduling (ELGS) policy [35]



A Maximum traffic load supported by the policies



B Policies' overhead

Figure 2-11. Effect of the overhead of the GM-RBDS and ELGS [35] policies in Figure 2-12, where it is assumed that $\Delta = 3$. Therefore, in the worst case scenario, the total amount of control information (i.e., overhead) that needs to be communicated to calculate a schedule consists of $2\Delta(\Delta - 1)$ integers. In addition to this, the ELGS overhead also includes a series of control-time-slots for calculating link eligibilities which are of $O(\log^2 |\mathcal{N}|)$ complexity [35]. This makes the ELGS overhead have length of at most $2\Delta(\Delta - 1)$ integers plus the series of control-time-slots used for ELGS link eligibility. Therefore, in WMNs with nodes located randomly, the ELGS overhead is of $O(\log^2 |\mathcal{N}|)$ complexity. This result shows that the GM-RBDS overhead is more scalable. The GM-RBDS overhead is of $O(\log |\mathcal{N}|)$ complexity (the complexity analysis of the GM-RBDS policy is in Section 2.3.3.1).

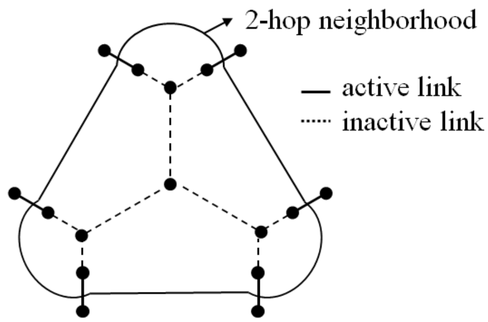


Figure 2-12. ELGS worst-case scenario

Remark. Even though the GM-RBDS policy has good performance in terms of throughput according to the previous results, the policy has a fairness problem. The links are able to reserve any number of future data-subframes, and therefore, a link is able to monopolize the channel causing starvation on other links. We believe that further research is necessary in this direction in order to design throughput-efficient policies that are fair too.

Remark. Note that the purpose of the election algorithm is two-fold. It allows the nodes to exchange requests, grants, and grant-confirmations, and it allows them to propagate within every 2-hop neighborhood, the state of the output queues (i.e., the list of reserved frames are implicitly specified in grants and grant-confirmations). To the best of our knowledge, the mechanisms for performing the previous tasks (i.e., propagating the current state of the system) in the policies revised in Section 2.1 have not been specified. In the analysis of those policies, it is assumed that the nodes know the queue lengths of the interfering links. Therefore, given that the counterpart of the election algorithm for the policies of Section 2.1 has not been specified, the overhead of the election algorithm was not considered in the previous analysis. The overhead of the election algorithm accounted for 0.09% of the total bandwidth in the scenario simulated for the GM-RBDS policy[26] .

2.5 Summary

A new framework for the stability analysis of scheduling policies for wireless networks that allow the reservation of future data-subframes has been proposed. The concepts of input-queue and output-queue were introduced into the framework in order to account for the packets waiting to be scheduled and the schedules assigned to these packets. Based on these concepts, sufficient conditions for the stability of RBDS wireless networks were found.

Within the proposed framework, an RBDS policy which uses the concept of greedy-maximal scheduling was analyzed. The nodes implement this policy by exchanging scheduling packets using the IEEE 802.16 election algorithm. A region in which the proposed reservation-based scheduling policy is stable was found using the framework. It was shown that the size of this region depends on the factor ν which is determined by two characteristics of the network topology only (i.e., s^j and $|\mathcal{P}_{\text{max}}|$). An IEEE 802.16 mesh network with the proposed scheduling policy (i.e., GM-RBDS)

[26] The overhead of the election algorithm was calculated as the bandwidth consumed by the MSH-NCFG control packets specified in the IEEE-802.16 standard [1]. These packets carry the information necessary for the nodes to execute the election algorithm. In the simulated scenario, they are transmitted once every 32 frames, which is one of the allowed settings in the IEEE-802.16 standard [1].

was simulated. It was shown that the policy always guaranteed the stability of the input-queues and that the output-queues were stable when the load was within $\frac{18}{24}$ of the optimal region. Finally, the performance of the GM-RBDS policy was compared with the W and ELGS policies. It was shown that the GM-RBDS policy has an advantage over the W and ELGS policies in terms of the required overhead.

CHAPTER 3
HEURISTIC CENTRALIZED TOPOLOGY CONTROL

Transmission power (TP) control in wireless multihop networks (WMNs) is an important problem due to the effects it has on the different layers of the protocol stack [39]. For example, the network connectivity, energy consumption, total physical-link throughput, spatial reuse, and total end-to-end throughput as a function of the TP have been investigated in [19, 20, 41, 62, 67] respectively. In this chapter, we look at the problem of TP control for adapting the stability region[1] of the WMN to a given set of flows such that the total throughput and end-to-end delay are improved. Specifically, we ask the question of what are the nodes' TPs that adapt the stability region to the flows in the network when a set of source-destination pairs, the routing algorithm, and the link-scheduling policy are given.

By adapting the stability region of the WMN, the queue lengths across the network are decreased in average for a given set of input-packet rates. In this way, the flows among the source-destination pairs are able to maintain higher levels of end-to-end throughput and lower levels of end-to-end delay while guaranteeing queue stability. Therefore, the problem considered in this chapter is of particular interest for applications that establish non-bursty sessions between source-destination pairs such as audio/video calls.

In order to adapt the stability region, we propose an algorithm that is executed by the flows established between the source-destination pairs. The idea behind the algorithm is to adapt a lower-bound region of the stability region (i.e., a region covered by the stability region) by modifying the TPs. The lower-bound region is a widely accepted theoretical performance metric used for comparing different link-scheduling

[1] The stability region is defined for link-scheduling policies as the set of input-packet rates under which the queues in the network are stable (i.e., positive recurrent).

policies [84]. In the algorithm, once the flows' paths are determined by the routing algorithm, the flows calculate the maximum input-packet rate they can support within the lower-bound region; then, each flow tries to stretch the lower-bound region by modifying the TP of nodes surrounding it. The effect that the stretch of the lower-bound region has on the stability region is another stretch on this region. Therefore, the result is a stability region adapted to the flows that allows them to support higher input-packet rates while guaranteeing the stability of the network.

In this chapter, we consider the minimum hop (min-hop) routing algorithm and the greedy-maximal reservation-based-distributed-scheduling (GM-RBDS) policy (Chapter 2) in Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 mesh networks. However, our results can be readily extended to other WMNs, routing algorithms, and link-scheduling policies.

The rest of this chapter is organized as follows. The related work and contributions are discussed in Section 3.1. The network model is presented in Section 3.2. In Section 3.3, our TP control algorithm is explained. The performance of the algorithm is evaluated in Section 3.4 by means of simulation. Finally, a summary of the chapter is presented Section 3.5.

3.1 Related Work

3.1.1 Link-Scheduling Policies and the Stability Region

The stability region for WMNs was first defined in [69] as the set of input-packet rates under which the queues in the WMN are stable. The stability region is defined for link-scheduling policies. Different link-scheduling policies achieve different stability regions, and it is said that a link-scheduling policy outperforms another policy in terms of throughput when it has a larger stability region. The optimal link-scheduling policy is the one whose stability region is a superset of the stability region of any other policy [69]. In terms of complexity, it is usually the case that the less complex the link-scheduling policy is, the smaller its stability region is. Therefore, based on the tradeoff between the size of the stability region and the complexity, different link-scheduling policies have been proposed in the literature [26, 35, 36, 50, 60, 61, 69, 71, 74, 83, 84]. These policies are characterized with a provable performance guarantee which is a region within the policy's stability region. That is, a set of input-packet rates is calculated for which the policy is guaranteed to be stable. The WMN may be stable under input-packet rates outside that set, but this is not guaranteed. Therefore, the stability region of the link-scheduling policy is at least as large as the calculated set of input-packet rates. We call this set the lower-bound region.

The lower-bound region depends on certain characteristics of the physical topology of the network. For example, the stability properties of the greedy maximal scheduling (GMS) [37] and the bipartite simulation (BP-SIM) [26] policies depend on the local-pooling factor and the maximum node degree of the network respectively. The local-pooling factor is a topological property of the network whose definition can be consulted in [37], and the node degree is defined as the number of links that the node belongs to.

3.1.2 Stability-Region Expansion Algorithms

The main idea presented in this chapter (i.e., adapting the stability region of a given link-scheduling policy by means of TP control) is based on the results obtained in [11, 89]. In [11], the network is partitioned based on the notion of local pooling, and each partition is assigned to a channel of the network. In this way, the GMS policy is guaranteed to achieve the optimal stability region in each channel. In [89], network topologies are identified for which distributed link-scheduling policies achieve the optimal stability region. However, these network topologies are not suitable for real scenarios [37] because of their sufficient conditions that guarantee the optimal stability region. These conditions include [89], 1-hop interference, 1-hop traffic, and a topology that is a graph that belongs to one of the following perfect-graph classes: chordal graphs, chordal bipartite graphs, cographs, and a subgroup of co-comparability graphs. In real scenarios, these conditions limit the suitability of WMNs. For example, only a few physical-layer technologies such as code-division-multiple-access (CDMA) can be approximated with the 1-hop interference model, and the traffic in WMNs is multihop by definition. Also, making the topology fall within the previous graph families imposes constraints on the locations and TPs of the nodes and the available routes. The multihop traffic case was considered in [89], and it was shown that only a subset of the previous graph families guarantee the optimal stability region in the multihop-traffic scenario. These were identified as forest of stars, where every connected component of the network graph is a star graph. Also, the results in [11, 89] are valid only for GMS policies under 1-hop traffic or backpressure routing-scheduling policies under multihop traffic[2] .

Our approach is built upon the idea of [11, 89] that under certain topologies a link scheduling policy performs better. We modify realistically the network topology using TP control to adapt the policy's stability region to the flows. The algorithm receives any set of end-to-end paths, node locations, and scheduling policy, and adapts the policy's stability region to the paths. Such an approach is beneficial because it improves the end-to-end throughput and delay without the restrictions previously discussed. In this chapter, we consider the case of min-hop routing, GM-RBDS scheduling, and randomly chosen source-destination pairs of nodes in IEEE 802.16 mesh networks.

Other heuristic algorithms have been proposed in the literature that improve the performance of the link-scheduling policy in terms of throughput by means of TP control. These algorithms include the ones reported in [54, 56, 57] whose basic idea is to increase the total throughput in the network by means of spatial reuse. The spatial reuse is increased by reducing the TP of the nodes. The algorithms differ between them in the

[2] It should be noted that the objective in [11, 89] was mainly to identify the topologies that enable the optimality of the GMS policy, and not to design topology-control algorithms.

way they are adapted to request-to-send (RTS)/clear-to-send (CTS) based protocols. In [30, 33], it is shown that better throughput improvements can be achieved not only by decreasing the TP to increase the spatial reuse but also by considering the hidden and exposed nodes. The algorithms proposed in [30] perform TP control with the objective of avoiding hidden nodes. In this way, the links in the network are able to sustain higher input-packet rates. In [81], a TP control algorithm for RTS/CTS-based protocols is proposed that decreases the area occupied by links during their transmissions, which is defined as the area in which other nodes must remain silent during the time the link is active. Then, it is shown that with this scheme, routing algorithms that favor short hops achieve higher levels of throughput. The goal of our algorithm is similar to the goal of the previous algorithms [30, 33, 54, 56, 57, 81], i.e., to increase the input-packet rates that a given link-scheduling policy can support by means of TP control. However, our approach differs in that it is directly based on a quantitative metric which is the stability region. It is not based on qualitative observations of the operation of the link-scheduling policy such as the hidden and exposed nodes in RTS/CTS-based policies. Therefore, it can be readily adapted to any link-scheduling policy whose stability region has been characterized such as the ones discussed in Section 3.1.1.

A different type of TP control algorithms, which are based on optimization techniques, are discussed in [7, 38]. In [7], the problem of integrated link scheduling and TP control for throughput optimization is shown to be nondeterministic polynomial time (NP) complete. Therefore, a heuristic algorithm is developed. The goal of the algorithm is to minimize the schedule length necessary to satisfy all the link loads determined by a given routing algorithm. By minimizing the schedule length, the total throughput of the network is increased because more scheduling cycles can be performed per time unit. In [38], the problem of jointly optimizing the flow routes, link schedules, TP, modulation and coding schemes is addressed. This is a more general problem than the one considered in [7] given that it does not only include the calculation of TPs and link schedules but also includes the routing and physical layers (i.e., flow routes, modulation, and coding schemes). In our algorithm, we are only concerned in the TP control problem when the flows and link-scheduling policy are given. That is, for a given set of flows, we determine TPs that improve the performance of the link-scheduling policy in terms of throughput and end-to-end delay.

3.1.3 Contributions

The contributions of this chapter are as follows.

A new TP control algorithm is proposed which increases the input-packet rates that flows can support and decreases the end-to-end delays while guaranteeing queue stability across the network. Also, the algorithm does not make any assumption on the paths followed by the flows which is not the case for the algorithms proposed in [30, 33, 54, 56, 57, 81], and it is not limited to RTS/CTS-based policies either. Our algorithm can be readily adapted to link-scheduling policies whose stability region has been characterized.

Our algorithm is based on the adaptation of the stability region of the given link-scheduling policy when only the links that belong to the given flows are considered. To the best of our knowledge, this technique has not been used before, and it is inspired on the results reported in [11, 89].

The improvement on throughput achieved by our algorithm is evaluated by means of simulation using the simulator proposed in Chapter 5 for the min-hop routing algorithm and the GM-RBDS policy (Chapter 2) in IEEE 802.16 mesh networks [1].

3.2 Network Model

We consider a WMN whose communication graph is denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links respectively. The links are directional. The link directed from node i to node j is denoted by (i, j). The node transmissions are omnidirectional. Link (i, j) belongs to $\mathcal{L}$ if and only if node j is within node i's transmission range. The 2-hop interference model is adopted. Therefore, two links interfere with each other only if they are at least two hops away from each other. We adopt this interference model because it is the model that the link scheduling in IEEE 802.16 WMNs is based on. In this type of networks, a node grants access to an incoming link only if none of the incoming links of the nodes that are at most 2 hops away has already been granted access.

Time is divided into frames, and each frame is divided into a control-subframe and a data-subframe. Control-subframes are divided into control-time-slots that are used for the exchange of scheduling packets, and data-subframes are divided into data-time-slots that are used for the transmission of data packets[3] . Links are allowed to transmit only one scheduling packet per control-time-slot and only one data packet per data-time-slot. A packet transmission over link (i,j) is successful if and only if no other link that interferes with link (i,j) is activated at any time during the transmission. If two or more interfering links are activated simultaneously at any time, there is a collision and none of the links' packet transmissions affected by the collision are successful.

The data traffic consists of a set of flows. This set is denoted by $\mathcal{F}$. The flows in $\mathcal{F}$ are enumerated. The n^{th} flow is denoted by f_n. It consists of a path and a mean input-packet rate which are denoted by p_n and λ_n respectively (i.e., $f_n \triangleq (p_n, \lambda_n)$). Path p_n is the sequence of nodes on which flow f_n is established. The m^{th} node in p_n is denoted by $p_n(m)$. It is assumed that the paths are calculated using min-hop routing. The first and last nodes of p_n are the source and destination nodes of f_n respectively. The intermediate nodes of p_n are the forwarding nodes of flow f_n. At the source node of flow f_n, the data-packet arrival process is Poisson distributed with mean λ_n. Every node that is an intermediate or destination node of at least one flow has a maximum packet rate that it can assign to its incoming traffic while guaranteeing the stability of its queues. Each of these nodes equally divides its maximum packet rate among all the flows for

[3] This is the frame structure adopted for IEEE 802.16 mesh networks [1].

which it is an intermediate or destination node. The maximum packet rate that node i can support for each of these flows is denoted by λ_{f}^{i}.

The number of flows that traverse a link is the degree of the link. The link degree of link (i, j) is denoted by $d^{(i,j)}$. If a flow is forwarded from node i to node j, then such a flow counts for the degree of link (i, j) and not for the degree of link (j, i), i.e., both the direction of the flow and its path determine which links have their degree affected by the flow.

We follow the reservation-based network model described in Chapter 2 in which there are two queues involved in the scheduling process of every link. These are the input and output queues which are denoted by $Q_{\mathrm{i},f_n}^{(i,j)}$ and $Q_{\mathrm{o},f_n}^{(i,j)}$ for flow f_n on link (i, j) respectively (Figure 3-1). The input-queue stores data packets that are waiting to be scheduled, which means they are waiting to be granted a future data-time-slot. The output-queue stores the data packets that have been scheduled, i.e., have already been granted a future data-time-slot, and are waiting to be transmitted. When a link schedules data packets, some of its unscheduled data packets are moved from its input-queue to its output-queue. The nodes implement the concept of regulators[4] introduced in [84]. The role of regulators is to regulate the burstiness of the node's incoming traffic. This task is performed for flow f_n on link (i, j) using queue $Q_{\mathrm{r},f_n}^{(i,j)}$.

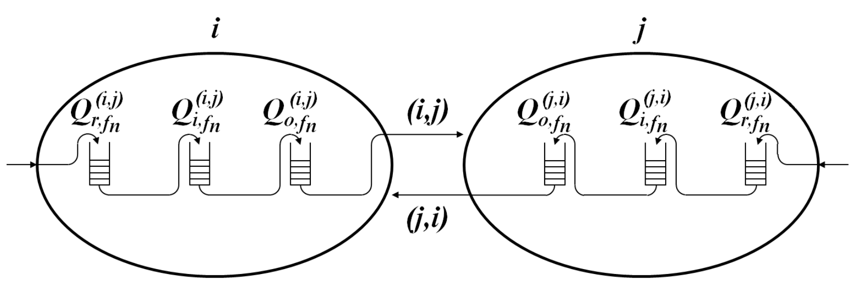


Figure 3-1. Data packet transmissions between nodes i and j

[4] A detailed description of the operation of regulators is provided in D.

Node j is a 1-hop neighbor of node i if node i is within node j's transmission range. The 1-hop neighborhood of a node is the set of all the node's 1-hop neighbors. It is denoted by $\mathcal{S}_1^i$ for node i. The 2-hop neighborhood of a node is defined as the set of nodes that are at least two hops away from the node and the node itself. It is denoted by $\mathcal{S}_{\leq 2}^i$ for node i. The active 1-hop neighborhood of a node is defined as the set of 1-hop neighbors that are intermediate or destination nodes of at least one flow. It is denoted by $\mathcal{S}_{\mathrm{a}}^i$ for node i. The direct 1-hop neighborhood of a node is the set of 1-hop neighbors that send data packets to the node. Therefore, the direct 1-hop neighbors of a node always precede the node in at least one path. Node i's direct 1-hop neighborhood is denoted by $\mathcal{S}_{\mathrm{d}}^i$.

The transmission range of node i is denoted by r^i. The maximum transmission range of any node is denoted by $r^{\max}$, and the Euclidean distance between nodes i and j is denoted by $||i, j||$.

3.3 Queue-Stability-Based Transmission Power (TP) Control Algorithm

3.3.1 GM-RBDS and its Stability Region

The link-scheduling adopted for the network is the GM-RBDS policy described in Chapter 2, which is based on the scheduling framework specified for IEEE 802.16 mesh networks with distributed scheduling. This framework can be summarized as follows. Nodes take turns to transmit scheduling packets on control-time-slots using the election algorithm specified for IEEE 802.16 mesh networks [1, 13, 78]. The objective of the election algorithm is to avoid scheduling-packet collisions. The links follow a three-way handshake by means of scheduling-packet exchanges. The goal of this handshake is to schedule data-packet transmissions in future data-time-slots and to reserve those data-time-slots among the interfering links such that data-packet collisions are avoided. The three-way handshake consists of (1) a request of future data-time-slots, (2) a grant for assigning the requested data-time-slots, and (3) a grant confirmation for acknowledging the grant.

In GM-RBDS, whenever a node is selected by the election algorithm, it transmits a scheduling packet that contains the following requests, grants, and grant confirmations. The node requests for every outgoing link as many data subframes as can be covered completely with unscheduled data packets $\Big($i.e., if there are m_{ds} data-time-slots per data-subframe, the number of data-subframes requested for link (i, j) is $\left\lfloor \frac{\sum_{f_n \in \mathcal{F}} Q^{(i,j)}_{\mathrm{i}, f_n}}{m_{\mathrm{ds}}} \right\rfloor\Big)$. The node grants the longest unexpired request it has received so far and which has not already been granted. The node confirms any grants it has received which have not been confirmed before and that do not collide with each other. In this way, by following GM-RBDS, the nodes attempt to schedule as many data-packet transmissions as they need, and these transmissions are scheduled on an as-soon-as-possible manner.

The size of the stability region of the GM-RBDS policy depends on the ability of the links to perform the three-way handshakes successfully (Chapter 2). If the probability that a link finishes successfully a three-way handshake is low, the link's queue will decrease at a lower rate. Therefore, the link's ability to forward data packets within some time range is going to be lower (i.e., the highest packet rate supported by the link is lowered), and this reduces the size of the stability region. The probability that a three-way handshake of link (i, j) is successful depends on the nodes that i can listen to but j can not (i.e., the active nodes hidden from j, which are $\mathcal{S}^i_{\mathrm{a}} \backslash \mathcal{S}^j_{\mathrm{a}}$) and the degree of link (i, j).

Based on the probability of successful three-way handshakes, sufficient conditions that guarantee queue stability under the GM-RBDS policy in the multihop scenario are given as follows.

Theorem 3.1. *Network $\mathcal{G}$ under GM-RBDS, min-hop routing, and the 2-hop interference model is stable if the packet rate λ^j_f supported by every node j in $\mathcal{N}$ satisfy Eq. 3–1*[5] *.*

[5] The set operator $\backslash$ refers to the relative complement, i.e., $\mathcal{S}^i_{\mathrm{a}} \backslash \mathcal{S}^j_{\mathrm{a}} \triangleq \left\{ k \in \mathcal{S}^i_{\mathrm{a}} : k \notin \mathcal{S}^j_{\mathrm{a}} \right\}$.

$$\lambda_f^j < \frac{1}{5\sum_{i\in\mathcal{S}_d^j} d^{(i,j)}\left|\mathcal{S}_a^i \backslash \mathcal{S}_a^j\right|} \quad \forall\ j \in \mathcal{N} \tag{3–1}$$

See Appendix D for the proof of Theorem 3.1.

Therefore, in order to guarantee stability in the multihop scenario with min-hop routing and GM-RBDS, the input-packet rate of flow f_n (i.e., λ_n) must be less than the following rate: the minimum packet rate among all the packet rates that nodes along the flow's path can assign to the flow. This is shown in Eq. 3–2, where λ_{max}^j is the upper-bound for node j's rate λ_{f}^j according to Eq. 3–1 $\left(\text{i.e., } \lambda_{\mathrm{max}}^j \triangleq \left(5\sum_{i\in\mathcal{S}_{\mathrm{d}}^j} d^{(i,j)}\left|\mathcal{S}_{\mathrm{a}}^i \backslash \mathcal{S}_{\mathrm{a}}^j\right|\right)^{-1}\right)$.

$$\lambda_n < \min\left\{\lambda_{\mathrm{max}}^j : j \in p_n\right\} \quad \forall\ f_n \in \mathcal{F} \tag{3–2}$$

3.3.2 TP Control Algorithm

The goal of our TP control algorithm is to expand the lower-bound region given by Eq. 3–2. By expanding this region, the flow rates λ_n can take higher values while guaranteeing stability, and therefore, the maximum total throughput the network can support for the given flows is increased. Let the maximum total throughput be denoted by λ_{T} and defined in terms of the lower-bound region for the flows' input-packet rates given by Eq. 3–2 as follows.

$$\lambda_{\mathrm{T}} \triangleq \sum_{p_n\in\mathcal{F}} \min\left\{\lambda_{\mathrm{max}}^j : j \in p_n\right\} \tag{3–3}$$

According to Eq. 3–3, Eq. 3–2, and Eq. 3–1, λ_{T} depends on the direct 1-hop neighborhoods $\left(\text{i.e., } \left\{\mathcal{S}_{\mathrm{d}}^j : j \in \mathcal{N}\right\}\right)$, the link degrees $\left(\text{i.e., } \left\{d^{(i,j)} : (i,j) \in \mathcal{L}\right\}\right)$, and the active 1-hop neighborhoods $\left(\text{i.e., } \left\{\mathcal{S}_{\mathrm{a}}^j : j \in \mathcal{N}\right\}\right)$ as follows.

$$\lambda_{\mathrm{T}} = \sum_{p_n\in\mathcal{F}} \left(\min_{j\in p_n} \frac{1}{5\sum_{i\in\mathcal{S}_{\mathrm{d}}^j} d^{(i,j)}\left|\mathcal{S}_{\mathrm{a}}^i \backslash \mathcal{S}_{\mathrm{a}}^j\right|}\right) \tag{3–4}$$

Given that the flows are determined by the min-hop routing algorithm, the following parameters in Eq. 3–4 are fixed: $\{p_n \in \mathcal{F}\}$, $\left\{\mathcal{S}_{\mathrm{d}}^{j} : j \in \mathcal{N}\right\}$, and $\left\{d^{(i,j)} : (i,j) \in \mathcal{L}\right\}$. Therefore, in order to increase λ_{T}, the only parameters that can be modified are the active 1-hop neighborhoods $\left(\text{i.e., } \left\{\mathcal{S}_{\mathrm{a}}^{j} : j \in \mathcal{N}\right\}\right)$. They can be modified by means of TP control such that λ_{T} is maximized. This optimization problem, which we call stability region adaptation for throughput maximization (SRA-TM), is given as follows.

Definition 7. Given a set of flows $\mathcal{F}$ calculated by the min-hop routing algorithm, the SRA-TM problem consists of the maximization of λ_{T} by means of TP control such that none of the nodes exceed the maximum TP and none of the paths are broken. That is,

$$\begin{aligned} \text{maximize} \quad & \sum_{p_n \in \mathcal{F}} \left(\min_{j \in p_n} \frac{1}{5 \sum_{i \in \mathcal{S}_d^j} d^{(i,j)} \left| \mathcal{S}_a^i \backslash \mathcal{S}_a^j \right|} \right) \\ \text{subject to} \quad & 0 \leq r^i \leq r^{\max} \quad \forall \; i \in \mathcal{N} \\ & r^i, r^j \geq \|i, j\| \quad \forall \; i \in \mathcal{S}_d^j, j \in \mathcal{N} \end{aligned} \tag{3–5}$$

Remark. In the SRA-TM problem, the flow paths are given and left unmodified. Higher values for λ_{T} could be achieved if the flow paths were modified by including them as decision variables. For example, a routing scheme can uniformly distribute the traffic loads across the links of the network so that links with high levels of congestion are avoided. This problem corresponds to a joint optimization of the topology and flow paths based on the stability region. This problem can be further studied due to its potential benefits on λ_{T}. However, this chapter deals only with the stability-region-based topology control as a first step towards the problem of stability-region-based joint topology and routing control.

Remark. If the data traffic in the network changes dynamically, the flow paths may change as well. In this scenario, the SRA-TM problem needs to be solved for every flow-path change. Therefore, the speed of convergence of algorithms that solve the SRA-TM problem is an important metric for such a scenario. The algorithms should be able to keep up with the rate of change of the flow paths. On the other hand, if the data-traffic levels of a set of flows change but the flow paths do not change, the SRA-TM problem does not need to be solved again. The reason is that the solution of the SRA-TM problem is the topology that allows those flows to support the maximum level of data traffic while guaranteeing stability. This means that the data-traffic levels in the flows may vary as long as they do not exceed such maximum levels (i.e., $\min\left\{\lambda^{j}_{\mathrm{max}} : j \in p_n\right\} \quad \forall \quad f_n \in \mathcal{F}$), and this can be guaranteed by means of call-admission-control algorithms.

In order to solve the SRA-TM problem, the TP algorithm shown in Figure 1 is proposed[6] . This algorithm is called heuristic stability region adaptation (HSRA).

The following definitions are necessary for the operation of the HSRA algorithm.

Definition 8. The bottleneck node of flow f_n is the node with the lowest maximum rate among all the intermediate and destination nodes of the flow, i.e., let j be the bottleneck node of f_n, then $j = \mathrm{argmin}_{i \in \{p_n(m) : 2 \le m \le |p_n|\}} \lambda^{i}_{\mathrm{f}}$.

Definition 9. Node h is hidden from node j if and only if $h \in \mathcal{S}^{i}_{\mathrm{a}} \backslash \mathcal{S}^{j}_{\mathrm{a}}$ for some $i \in \mathcal{S}^{j}_{\mathrm{d}}$.

Definition 10. The MinPower setup is the set of minimum TPs whose transmission ranges guarantee that none of the links of the flows in $\mathcal{F}$ is broken.

The operation of the algorithm is as follows. First, the nodes' TPs $\left(\text{i.e., } \{r^i : i \in \mathcal{N}\}\right)$ are set according to the MinPower setup (line 2 in Algorithm 1). By reducing the TPs (Definition 10), the spatial reuse in the network is increased, and as a consequence, the total throughput is increased as well[7] . Then, the maximum throughput that intermediate and destination nodes can support for the flows they belong to is calculated (line 3

[6] The SRA-TM problem is formulated as a mixed integer program with non-linear constraints in E. This formulation is used in Section 3.4 for calculating the optimal solution of the simulated instances of the SRA-TM problem.

[7] This spatial-reuse-based TP control is the basis of the algorithms proposed in [54, 56, 57, 81].

Algorithm 1 HSRA Algorithm

procedure HSRA($\mathcal{N}$, $\mathcal{F}$, M)
    $\{r^i\} \leftarrow$ MINPOWERSETUP($\mathcal{N}$)
    $\{\lambda^i_{\text{max}}\} \leftarrow$ NODEMAXRATES($\mathcal{N}$)
    $T \leftarrow$ TOTALTHROUGHPUT($\mathcal{F}$, $\{\lambda^i_{\text{max}}\}$)
    for $m \leftarrow 0,\ m < M,\ m \leftarrow m + 1$ **do**
        $f_n \leftarrow$ PICKFLOWRANDOMLY($\mathcal{F}$)
        $j \leftarrow$ BOTTLENECKNODE(f_n, $\{\lambda^i_{\text{max}}\}$)
        $\mathcal{S}^j_{\text{d}} \leftarrow$ DIRECT1HOPNEIGH(j)
        for every i in $\mathcal{S}^j_{\text{d}}$ **do**
            $\mathcal{S}^i_{\text{a}} \backslash \mathcal{S}^j_{\text{a}} \leftarrow$ HIDDENACTIVENODES((i, j))
        end for
        $i \leftarrow$ NODEHIDDENTHEMOST($\{\mathcal{S}^i_{\text{a}} \backslash \mathcal{S}^j_{\text{a}}\}$)
        if $||i, j|| < r^{\text{max}}$ **then**
            $r_{\text{aux}} \leftarrow r^i$
            $r^i \leftarrow ||i, j||$
            $\{\lambda^i_{\text{aux}}\} \leftarrow$ NODEMAXRATES($\mathcal{N}$)
            $T_{\text{aux}} \leftarrow$ TOTALTHROUGHPUT($\mathcal{F}$, $\{\lambda^i_{\text{aux}}\}$)
            if $T_{\text{aux}} > T$ **then**
                $\{\lambda^i_{\text{max}}\} \leftarrow \{\lambda^i_{\text{aux}}\}$
                $T \leftarrow T_{\text{aux}}$
            else
                $r^i \leftarrow r_{\text{aux}}$
            end if
        end if
    end for
end procedure

in Algorithm 1). This is done using Eq. 3–1, which defines the nodes' maximum throughput. Based on these maximums, the total throughput the network can support is calculated (line 4 in Algorithm 1).

Once the total throughput under the MinPower setup is known, flows are selected randomly one-by-one for a number of M times (line 5 in Algorithm 1). Every time a flow is selected, the maximum throughput the flow can support is increased if this causes that the total throughput be increased as well. Otherwise, the flow is left unmodified. The throughput of the selected flow is increased as follows.

Let the selected flow be denoted by f_n (line 6 in Algorithm 1). The bottleneck node of f_n is found first by tracking the node of the flow with the lowest maximum throughput

(Eq. 3–2). Let this node be denoted by j (line 7 in Algorithm 1). The maximum rate of j $\left(\text{i.e., } \lambda_{\text{f}}^{j}\right)$ is increased by increasing the TP of one of j's 2-hop neighbors (lines 8 to 15 in Algorithm 1). However, this TP increase is confirmed only if the total throughput (i.e., λ_{T}) is increased as well (lines 16 to 20 in Algorithm 1). Otherwise, the TP of j's 2-hop neighbor is left unmodified (line 22 in Algorithm 1). The total throughput may be decreased given that the TP increase of j's 2-hop neighbor may decrease the maximum rate of other bottleneck nodes in the network, and this maximum-rate decrease may be higher than the increase on j's maximum rate.

The 2-hop neighbor of node j whose TP is increased is selected so that the factor $|\mathcal{S}_{\text{a}}^{i} \backslash \mathcal{S}_{\text{a}}^{j}|$ on the denominator of the upper-bound for λ_{f}^{j} is decreased (Eq. 3–1). Qualitatively, this TP increase can be explained as follows. Node j (i.e., the bottleneck node) has a set of 1-hop neighbors that are sending data packets to it $\left(\text{i.e., } \mathcal{S}_{\text{d}}^{j}\right)$. Let i be one of these nodes, and consider the link (i, j) and the input and output queues $Q_{i}^{(i,j)}$ and $Q_{o}^{(i,j)}$ of node i as shown in Figure 3-1. In order for i to transmit packets to j, a reservation of future data-time-slots is required. When nodes i and j finish this reservation successfully, data packets in node i's input-queue $\left(\text{i.e., } Q_{i}^{(i,j)}\right)$ are moved to node i's output-queue $\left(\text{i.e., } Q_{o}^{(i,j)}\right)$, and these packets are later pulled from $Q_{o}^{(i,j)}$ for their transmission. Therefore, for the queues $Q_{i}^{(i,j)}$ and $Q_{o}^{(i,j)}$ to have their lengths decreased, the reservation performed by nodes i and j needs to be successful, i.e., the three-way handshake for scheduling data-packet transmissions on link (i, j) needs to be successful. The probability that the handshake is successful and that the queues decrease their length depends on the grants received by node i and not received by node j. In the following, we refer to these grants as $hiddengrants$. If i requests future data-time-slots to j and a hidden grant is received by i before j transmits its grant to i, j's grant may not be confirmed by i. This is because the hidden grant may interfere with j's grant. On the other hand, if j is able to listen to the hidden grant, j is able to generate its grant such that it does not interfere with the hidden grant, and i will be able confirm j's grant[8] . Therefore, in order to increase the probability of handshake success and queue decrease, the TP of the node that transmits the hidden grant (i.e., the node hidden from j) can be increased such that node j is able to listen to the hidden node's transmissions.

Node j may have more than 1 hidden nodes in every incoming link from the nodes in its direct 1-hop neighborhood. The HSRA algorithm chooses only one of those hidden nodes for increasing its TP. The node that is chosen is the node that is hidden from the highest number of nodes (i.e., node j and all the other intermediate or destination nodes unable to listen to the hidden node). This is performed in lines 8 to 12 in Algorithm 1. In this way, the maximum rate is increased for all those nodes so that, if one or more of those nodes are bottleneck nodes, higher improvements on the total throughput can be achieved.

The role that the objective function of the SRA-TM problem (Eq. 3–5) plays in the HSRA algorithm is the quantification of the throughput improvement by the TP increase on hidden nodes. By increasing the TP of a node hidden from a bottleneck node, the factor $\mathcal{S}_{\mathrm{a}}^{i} \backslash \mathcal{S}_{\mathrm{a}}^{j}$ in the denominator of Eq. 3–5 is decreased for the bottleneck node, and as a consequence the bottleneck node's maximum rate is increased. However, the TP increase on the hidden node may also cause an increase on the $\mathcal{S}_{\mathrm{a}}^{i} \backslash \mathcal{S}_{\mathrm{a}}^{j}$ factor of other bottleneck nodes. Therefore, the objective function allows the algorithm to trade off between decreasing the number of hidden nodes by increasing TP and maintaining spatial reuse by not increasing TP. In the algorithm, this tradeoff is achieved by testing the improvement on the total throughput (lines 14 to 23 in Algorithm 1).

[8] The problem of node j not being able to listen hidden grants is the hidden-node problem version for reservation-based distributed scheduling policies. This problem is studied in detail in Chapter 2.

3.4 Simulation Results

The performance evaluation of the HSRA algorithm was performed by means of simulation using the simulator proposed in Chapter 5. The simulated network is an IEEE 802.16 mesh network with distributed scheduling under the configuration shown in Table 3-1. The number of nodes was specified as a simulation parameter. The nodes were uniformly distributed in a square area such that the node density was always kept at 15 nodes per unit area. The maximum transmission range of the nodes was set at 0.3 (i.e., $r^{\max} = 0.3$). The connectivity of the network under bidirectional links and with the nodes' transmission ranges set at $r^{\max}$ was confirmed before executing the min-hop routing algorithm. The number of flows was specified as a simulation parameter. The source and destination nodes of every flow were uniformly distributed among all the nodes in the network. The min-hop routing algorithm calculated the flow paths under the MaxPower setup which is the power assignment when all the nodes' transmission ranges are set at the maximum (i.e., $r^{\max}$). Once the paths were calculated, the transmission ranges of the nodes were found using the HSRA algorithm. Also, the optimal transmission ranges (i.e., the solution to the SRA-TM problem (Eq. 3–5)), which we call OptPower, were found using the formulation of the SRA-TM problem as a mixed integer program with non-linear constraints (MIP-NLC) (Appendix E). The MIP-NLC was solved using the Branch And Reduce Optimization Navigator (BARON) Solver [4], which is a system for solving non-convex optimization problems to global optimality. Finally, the network was simulated under the MaxPower, MinPower, OptPower, and HSRA setups.

Figure 3-2 shows the average output-queue length for three networks with 20 nodes each and increasing number of flows (i.e., 10 flows in Figure 3-2A, 15 flows in Figure 3-2B, and 20 flows in Figure 3-2C). The input-queues have been omitted because they are guaranteed to always be stable (Chapter 2). The flow rates in each network were all set at the same value. These are 8, 6, and 5 packets per frame for the networks in Figures 3-2A, 3-2B, and 3-2C respectively. These values were set so that their

Table 3-1. Institute of Electrical and Electronics Engineers (IEEE) 808.16 mesh network configuration

Parameter	Value
Frame length	10 ms
Control-time-slot length	63 μs
Number of control-time-slots per frame	4
Number of data-time-slots per frame	256
NextXmtMx†	7
XmtHoldoffExponent†	6
Link scheduling	GM-RBDS
Routing	min-hop

† This is a parameter of the election algorithm used for specifying the frequency that nodes transmit scheduling packets.

corresponding HSRA network became unstable if they were increased by at least one point. In this way, the network operates at a point inside the stability region and close to its boundary. Therefore, when any of the rates is increased by at least one point, the network operates outside the stability region, and therefore, it is unstable.

The stability of the HSRA network was determined by a visual test on Figure 3-2A, Figure 3-2A, and Figure 3-2A. However, a more rigorous procedure for testing stability is by means of a constant false-alarm rate (CFAR) test. The CFAR test proposed in [82] tests the hypothesis that a network is stable from samples of the queue lengths. According to this test, the false-alarm rate (i.e., the probability that a stable network is regarded as unstable) of the HSRA networks of Figure 3-2A, Figure 3-2A, and Figure 3-2A are 0.17, 1.9×10^{-7}, and 3.1×10^{-30} respectively[9] . This means that according to the test, the HSRA networks are likely to be unstable. In other words, if the networks are regarded as unstable, the probability that this judgment is incorrect

[9] These results were obtained by sampling the queue lengths at 25 points equally spaced during the last 25% of the simulation time. This is done in order to approximate the assumptions of the CFAR test in [82]. The assumptions are that the samples are statistically independent and that the initial transient is over.

Table 3-2. False-alarm rate comparison for the heuristic-stability-region-adaptation (HSRA), MinPower, MaxPower, and OptPower configurations

	MaxPower	MinPower	HSRA	OptPower
10 flows	1.2×10^{-131}	3.2×10^{-5}	0.17	0.57
15 flows	0	0.83	1.9×10^{-7}	0.90
20 flows	6.6×10^{-136}	0.75	3.1×10^{-30}	1.3×10^{-6}

(i.e., the probability that the networks are stable) is 0.17, 1.9×10^{-7}, and 3.1×10^{-30} respectively. For the purposes of the analysis of the results, this fact is ignored[10] . The average queue lengths may increase indefinetly (i.e., they are unstable) according to the CFAR test but at a low rate. Thus, the HSRA networks may be operating at a point outside the stable region but close to its boundary, which is the important fact when comparing the algorithms (i.e., MaxPower, MinPower, OptPower, and HSRA).

Table 3-2 shows the false-alarm rates of all the cases in Figure 3-2A, Figure 3-2A, and Figure 3-2A. These results are useful for the cases that are difficult to compare visually. For example, in Figure 3-2A, the MinPower, HSRA, and OptPower networks all have similar average queue lengths. However, Table 3-2 shows that among these three networks, the networks that are most and less likely to be unstable are MinPower and OptPower respectively.

Close to the end of the simulation time, when the transient behavior of the queues is over, the average queue lengths of the different power setups (i.e., MaxPower, MinPower, OptPower, and HSRA) can be compared. In Figure 3-2A, the MaxPower setup has the worst performance (i.e., the largest average queue length), and the MinPower, OptPower, and HSRA have similar performance. Therefore, when the number of flows is low (i.e., 10 flows), the MinPower and the HSRA algorithms are able to

[10] Otherwise, the simulations would have to be run several times to fine tune the packet rates so that stability is guaranteed according to the CFAR test. This fine tuning is not practical due to the length of the simulations. These are long due to the fact that the networks operate near to the stable-region boundary. On average, one network simulation takes 72 hours to complete.

achieve queue lengths that are close to the lengths achieved by the optimal solution (i.e., OptPower). On the other hand, when the number of flows increases, the MinPower algorithm does not achieve a performance close to the optimal one while the HSRA algorithm does. This is shown in Figures 3-2B and 3-2C. The MaxPower and MinPower algorithms have similar performance which is worse when compared with the HSRA algorithm. The HSRA algorithm achieves average queue lengths that are close to the lengths achieved by the optimal solution. Therefore, the HSRA algorithm enables the flows to carry more traffic while guaranteeing stability than the MaxPower and MinPower algorithms do. Also, it is confirmed that the technique of only maximizing the spatial reuse by reducing the transmission ranges (i.e., MinPower) does not perform well when the flow density increases (i.e., when the number of flows increases and the number of nodes is kept constant.). On the other hand, the technique of adapting the stability region to the given set of flows by means of TP control (i.e., HSRA) does perform well when the flow density increases.

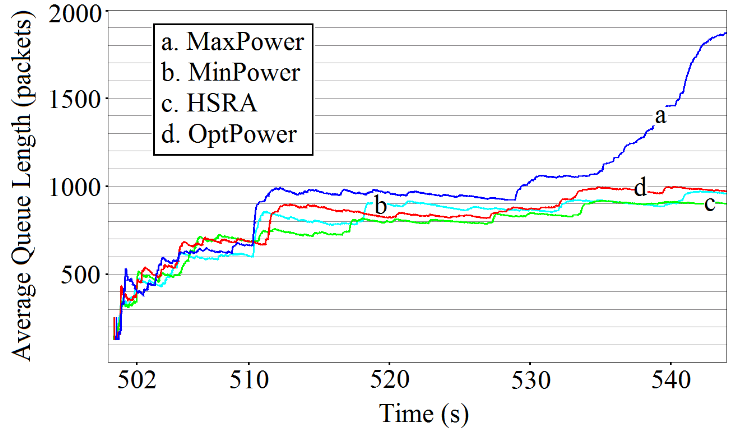
The HSRA algorithm achieves average queue lengths that are slightly lower than those of the OptPower setup in some cases (Figure 3-2A and Figure 3-2C at the end of the simulation time). This is due to the fact that the SRA-TM problem is based on the lower-bound region (Section 3.3.2). In the SRA-TM problem, the lower-bound region, and not the stability region itself, is adapted to the set of flows, i.e., the stability region is adapted through the lower-bound region. The reason for this is that the exact formulation of the stability region is not available. The stability region is usually characterized with the lower-bound region because its exact characterization is not feasible due to its complexity[11] . Therefore, the accuracy of the solutions based on the

[11] See [26, 35, 36, 50, 60, 61, 69, 71, 74, 83, 84] for the literature on the problem of characterizing the stability region of link-scheduling policies.

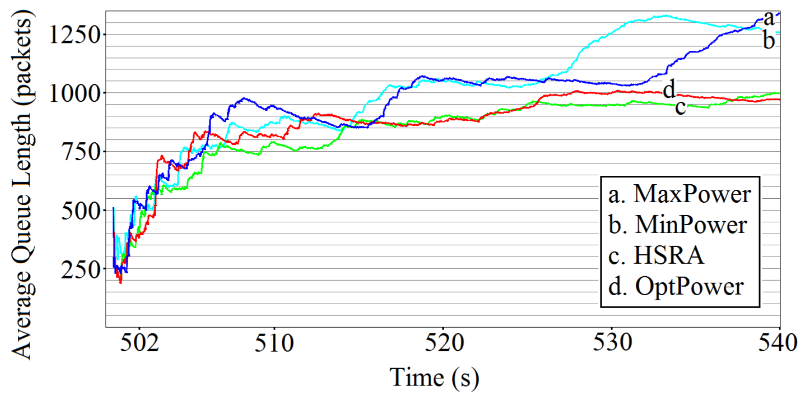
lower-bound region such as HSRA and OptPower are affected by the tightness of this region.

Figure 3-3 shows a comparison of the performance improvement achieved by the HSRA and MinPower algorithms. This comparison is based on the average-queue-length reduction achieved by the HSRA and MinPower algorithms with respect to the average queue length of the MaxPower algorithm. For example, in Figure 3-3A, for the case of 10 nodes, the MinPower and HSRA queue-length-reduction bars indicate that the MinPower and HSRA algorithms achieve average queue lengths that are 16% and 23% smaller than the MaxPower average queue length respectively. Figure 3-3A shows the performance improvement when the number of nodes increases and the number of flows is fixed at 10. Figure 3-3B shows the performance improvement when the number of flows increases and the number of nodes is fixed at 20. The flow rates were set at 10, 10, 8, and 10 packets per frame in Figure 3-3A, and they were set at 8, 6, 5, and 3, in Figure 3-3B. These frame rates were also set such that if they are increased by at least one point, their corresponding networks become unstable.

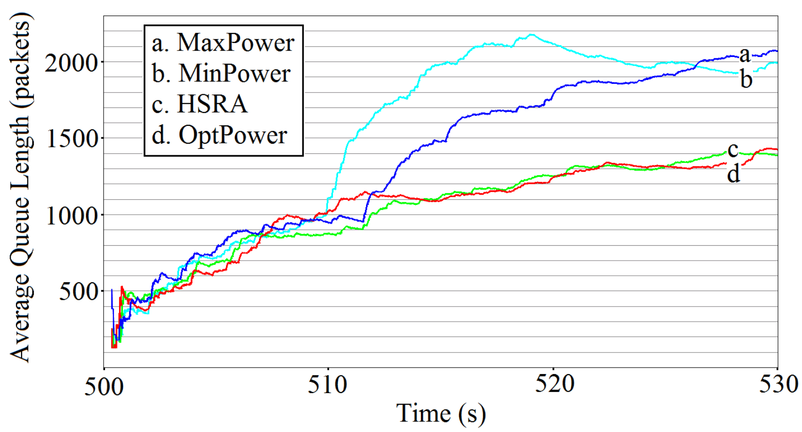
In Figure 3-3A, when the number of nodes increases, the HSRA algorithm always achieves an improvement higher than the one achieved by the MinPower algorithm. The average-queue-length reduction achieved by the HSRA algorithm is on average 6.8 points above the reduction achieved by the MinPower algorithm. This difference of improvements does not increase as the number of nodes in the network increases. The HSRA improvement is above the MinPower improvement by 7.02, 9.68, 3.07, and 7.37 points for the cases of 10, 15, 20, and 25 nodes respectively. Therefore, both algorithms, MinPower and HSRA, respond similarly as the number of nodes increases. That is, the MinPower and HSRA algorithms are able to maintain reductions of the average queue



A 10 flows

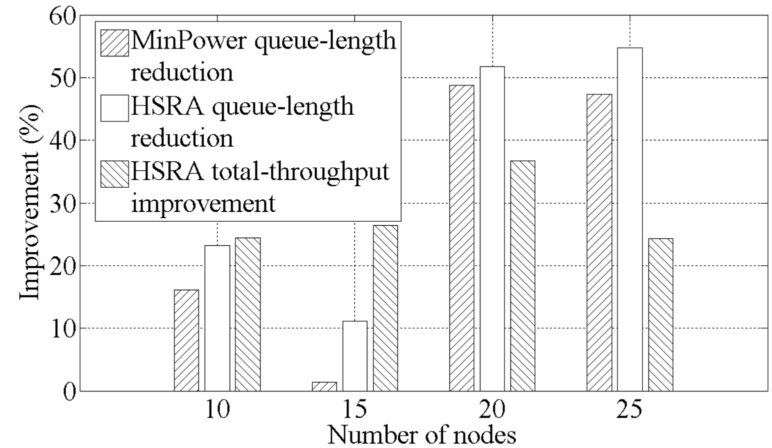


B 15 flows

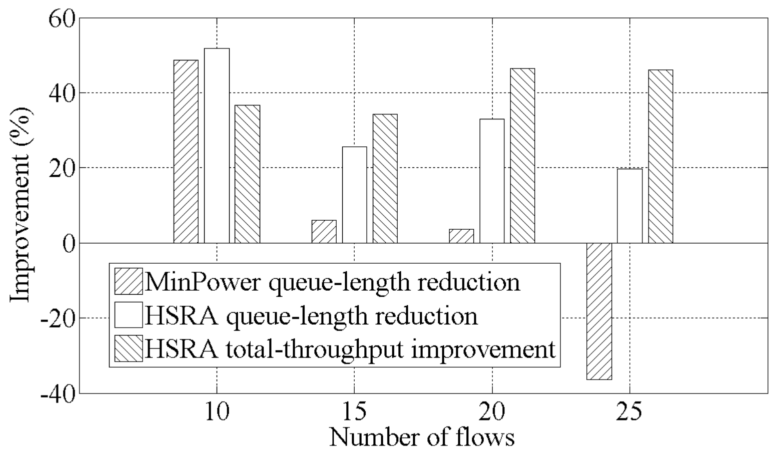


C 20 flows

Figure 3-2. Average output-queue length comparison for the heuristic-stability-region-adaptation (HSRA), MinPower, MaxPower, and OptPower configurations



A Increasing number of nodes



B Increasing number of flows

Figure 3-3. Performance comparison of the HSRA and MinPower algorithms

lengths as the length of the paths increase[12] . This results in higher transport capacities when the MinPower and HSRA algorithms are used[13] . Between the two algorithms, HSRA achieves the highest transport capacity.

[12] Note that given that the node density and number of flows are kept constant as the number of nodes increases, the average length of the paths increases with the number of nodes.

[13] The transport capacity of a flow is the product of the maximum rate supported by the flow and the length of its path. The transport capacity of the network is summation of the flows' transport capacities.

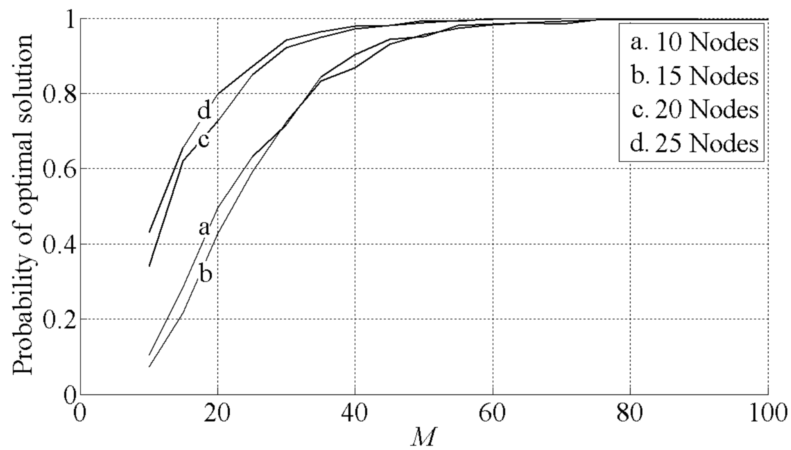
In Figure 3-3B, when the number of flows increases, the HSRA algorithm always achieves an improvement that is higher than the one achieved by the MinPower algorithm too. The average-queue-length reduction achieved by the HSRA algorithm is on average 27.0 points above the reduction achieved by the MinPower algorithm. This difference of improvements is greater than the one achieved when the number of nodes increases (Figure 3-3A), which is 6.8 points. Also, the difference of improvements increases with the number of flows in the network, which was not the case when the number of nodes increases. The HSRA improvement is above the MinPower improvement by 3.1, 19.5, 29.3, and 56.1 points for the cases of 10, 15, 20, and 25 flows respectively. Therefore, when the flow density, which is defined as the ratio among the number of flows and nodes in the network, increases, the HSRA algorithm achieves average-queue-length reductions that increase too while the MinPower algorithm achieves average-queue-length reductions that decrease. This behavior shows that the HSRA algorithm outperforms the MinPower algorithm when the flow density increases. For example, when the number of flows is 25, the MinPower algorithm has an average-queue-length improvement of -36.4% (i.e., it increases the average-queue-length by 36.4%) while the HSRA algorithm has an average-queue-length improvement of 19.7% (Figure 3-3B).

According to Figure 3-3, the general tendency of the HSRA queue-length-reduction is that it increases when the flow density decreases (Figure 3-3A), and it decreases when the flow density increases (Figure 3-3B). Therefore, the HSRA algorithm performs better than MaxPower, and the HSRA reaches higher performance levels when the flow density is low. In Figure 3-3A and Figure 3-3B there is an exception to the general tendency when the number of nodes increases from 10 to 15 and when the number of flows increases from 15 to 20 respectively. The reason is that for the particular simulated scenarios with 15 nodes and 20 flows respectively, the topology induced by the MaxPower setup happened to be similar to the topology induced by the OptPower setup due to the random node locations and flows' sources and destinations. Therefore, in these scenarios, the HSRA algorithm improved the initial MaxPower topology by smaller amounts.

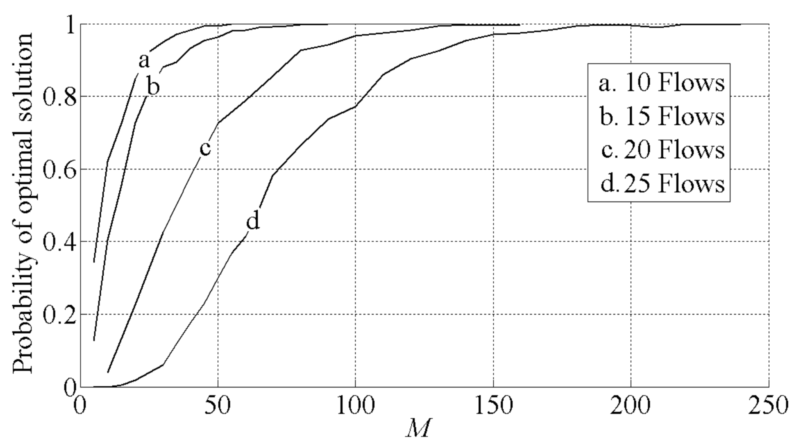
Figure 3-3 also shows the increase that the HSRA algorithm causes on the total throughput (i.e., the objective function λ_T of the SRA-TM problem (Eq. 3–5)). This increase corresponds to the stretch that the algorithm does to the lower-bound region of the link-scheduling policy. Initially, the network stability region is determined by the MaxPower setup. Once the algorithm is executed, the stability region is determined by the HSRA power setup. Therefore, the increase that the HSRA algorithm causes on the total throughput corresponds to the increase achieved by modifying the MaxPower stability region according to the HSRA algorithm. For example, in Figure 3-3A, for the case of 10 nodes, the HSRA algorithm increases the total throughput by 24.4%. The total-throughput increase does not show a direct-proportional nor an inverse-proportional relation with the number nodes or flows. In both Figures 3-3A and 3-3B, the total-throughput improvement increases and decreases with both the number of nodes and flows. Also, when the total-throughput improvements achieved when the number of nodes increases (Figure 3-3A) are compared with the ones achieved when the number of flows increases (Figure 3-3B), they are similar on average. The average total-throughput improvements in Figures 3-3A and 3-3B are 40.5% and 40.9% respectively. The total-throughput improvements in Figure 3-3A are 24.4%, 76.5%, 36.7%, and 24.4% for the cases of 10, 15, 20, and 25 nodes respectively, and the total-throughput improvements in Figure 3-3B are 36.7%, 34.3%, 46.4%, and 46.1% for the cases of 10, 15, 20, and 25 flows respectively. Therefore, the HSRA algorithm performs similarly when either the average flow length or the average flow density increases. This result extends to the average-queue-length reductions achieved by the HSRA algorithm when the number of nodes and flows increase. These averages are similar to each other; they are 35.2% and 32.5% respectively. However, there is a difference between the average-queue-length and total-throughput improvements achieved by the HSRA algorithm. For example, in Figure 3-3A, for the case of 25 nodes, the average-queue-length improvement is 30.4 points above the total-throughput improvement, and in Figure 3-3B, for the case of 25 flows, the average-queue-length improvement is 26.4 points below the total-throughput improvement. Therefore, although the improvement on the maximum total-throughput achieved by the HSRA algorithm improves the average queue length in the network, the amount of the total-throughput improvement differs from the amount of average-queue-length improvement. This difference is due to the fact that the HSRA algorithm is based on the lower-bound region (Eq. 3–1) and not on the stability region itself. Therefore, the tightness of the lower-bound region (i.e., how close it is to the stability region) affects the performance of the technique the HSRA algorithm is based on. The algorithm can predict more accurately the size of the expansion it performs on the stability region with a tighter lower-bound region.

Figure 3-4 shows the performance of the HSRA Algorithm as a function of M. This performance is evaluated in terms of the probability that the HSRA Algorithm finds the optimal solution of the SRA-TM problem when M increases. Figure 3-4A shows the performance when the number of flows is fixed at 10 and the number of nodes is 10, 15, 20, and 25. Figure 3-4B shows the performance when the number of nodes is fixed at 20 and the number of flows is 10, 15, 20, and 25. According to Figure 3-4A, the performance is similar when there are 10 and 15 nodes, and it is also similar when there are 20 and 25 nodes. However, the general trend is that the performance improves as the number of nodes increases (i.e., as the flow density decreases). When the flow density decreases, the probability that a flow interferes with another flow decreases. Therefore, the number of bottleneck nodes whose maximum throughput can be increased by increasing the TP of nodes in adjacent flows (i.e., hidden nodes) is lower, so the HSRA algorithm is able to find those hidden nodes with a lower M.

According to Figure 3-4B, the performance decreases with the number of flows. This result is also due to the relation between the HSRA Algorithm and the flow density. As the number of flows increases in Figure 3-4B, the flow density increases, and therefore, the HSRA needs a higher M to find the hidden nodes whose TP needs to be increased.



A Increasing number of nodes



B Increasing number of flows

Figure 3-4. Probability that HSRA calculates the optimal solution as a function of M

3.5 Summary

The Heuristic-Stability-Region-Adaptation algorithm has been proposed for transmission power control. This algorithm increases the input-packet rates that flows can support and decreases the end-to-end delays. It is is based on the adaptation of the stability region of a given link-scheduling policy when only the links that belong to a given set of flows are considered. The algorithm can be readily adapted to any link-scheduling policy whose stability region has been characterized, so it is not limited to any specific scheduling approach such as RTS/CTS-based policies. The improvement on throughput achieved by our algorithm was evaluated by means of simulation for the min-hop routing algorithm and the GM-RBDS link-scheduling policy in IEEE 802.16 mesh networks. It was shown that it outperforms the classical solution of reducing transmission powers to increase spatial reuse. Also, its performance was evaluated. It was found that it depends on the flow density. The performance increases as the flow density decreases.

CHAPTER 4
DISTRIBUTED TOPOLOGY CONTROL USING POTENTIAL GAMES

In this chapter, the problem of maximizing the total end-to-end throughput when a set of flows is given is addressed. The set of flows represents the traffic between end users that is generated by their applications such as end-to-end video/audio sessions. The maximization of the total end-to-end throughput is performed by means of transmission power (TP) control that the flows perform collaboratively.

Intuitively, our TP control can be described as follows. The maximum throughput that a given flow supports depends on the throughput that each of the nodes along the flow's path supports, and the maximum throughput that a node supports depends on its scheduling policy and the conflicts[1] with surrounding nodes which also need to schedule packet transmissions [49]. Therefore, by means of TP control, the nodes are able to reduce the number of conflicts either by decreasing the interference (i.e., reducing TP) and/or coordinating future packet-transmission times such that no conflicting transmissions are performed simultaneously. The latter approach is used in this chapter, i.e., the flows control the TPs so that nodes are able to listen to each other's schedules in order to coordinate future packet transmissions more successfully.

Mathematically, our TP control can be characterized using the stability region of the WMN. The maximum throughput that nodes support is characterized by the physical-link capacity and the stability region of the link scheduling policy. This region is the set of input-packet rates supported by the links of the network that guarantee that their queues are stable (i.e., the link queues are positive recurrent) [69]. The physical-link capacity determines the maximum number of bits that the packets can carry. In the TP control approach, the network topology is modified such that the stability region is

[1] Scheduling conflicts arise between nodes when they attempt to transmit packets simultaneously and the interference they cause on each other is high enough to cause packet collisions.

adapted to the given set of flows. The goal of this adaptation is to maximize the highest input-packet rate supported by the flows that guarantee stability. This approach was originally proposed in Chapter 3, in which the TP-control algorithm was heuristic and centralized. In this chapter, we propose and analyze a distributed TP-control algorithm by means of potential games [53]. Specifically, we formulate the TP-control optimization problem as a potential game in which the flows (i.e., players) collaborate to maximize the packet rates they can support by adapting the stability region. The game is formulated such that the flows compete for helping each other. They help each other by making sure that none of their individual actions affect other flows negatively. Therefore, in the game, the flows are rewarded when they benefit not only themselves but other flows as well, i.e., they are rewarded when they collaborate.

The TP-control optimization problem is a mixed integer program with non-linear constraints (Chapter 3 and Appendix E). Therefore, in order to analyze the potential game, the problem is modified heuristically such that it can be formulated as an integer linear program. Based on this modification, the Nash equilibrium of the game is characterized. However, due to the modification, the Nash equilibrium is suboptimal with respect to the objective function of the original problem (i.e., the problem with no modifications). Therefore, a performance bound for the equilibrium is found. In this way, the equilibrium can be compared with the optimal value of the objective function of the original problem.

The chapter is organized as follows. The related work and contributions are discussed in Section 4.1. Section 4.2 describes the network model. In Section 4.3, the potential game for maximizing the total end-to-end throughput is proposed and analyzed. In order to demonstrate the how the potential game can be used for a specific network, a distributed topology-control algorithm for Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 WMNs is developed in Section 4.4 based on the game. A performance bound for the algorithm is also calculated. In Section 4.5, the bound is compared with simulation results. Also, the proposed algorithm is compared with the centralized and heuristic algorithm of Chapter 3 and the algorithm based on spatial-reuse maximization (i.e., nodes transmitting at minimum power levels that guarantee connectivity). Finally, a summary of the chapter is presented in Section 4.6.

4.1 Related Work

Our stability-based topology-control approach is primarily based on the ideas presented in [10, 11, 89] and Chapter 3. In [11], the network is partitioned based on the notion of local pooling[2] , and each partition is assigned to a channel of the network. In this way, greedy maximal scheduling (GMS) is guaranteed to achieve the optimal stability region in each channel. In [10, 89], network topologies are identified for which GMS policies achieve the optimal stability region. Although [10, 89] provide insightful results for the understanding of GMS policies, the network topologies they identify are not suitable for real scenarios [37]. This is due to the conditions that guarantee optimality for such network topologies. These conditions include 1-hop interference, 1-hop traffic, and a topology that corresponds to an $\mathcal{F}$-*free* graph[3] . Real WMNs hardly meet these restrictive conditions. In our algorithm, the flows modify realistically the network topology using TP control to adapt the stability region. The algorithm is distributed, and it is based on the heuristic and centralized algorithm proposed in Chapter 3.

In [58], the dynamic routing and power control (DRPC) algorithm was proposed. DRPC finds the optimal solution. At every time slot, DRPC is given the set of end-to-end nodes with their corresponding end-to-end data rates, the channel states, and the queue lengths, and finds the optimal set of routes and TPs that maximize throughput. However, this is a constrained optimization problem that requires global information of the network

[2] See Section 4.3.1 for the definition of local pooling.

[3] See [10] for the definition of $\mathcal{F}$-*free* graphs.

(i.e., channel states and queue lenghts) at every time slot, so, in [58], a suboptimal distributed algorithm was also proposed for random-access scheduling policies. In [25], another random-power-selection algorithm for random-access scheduling policies was proposed. It is shown that it achieves maximal throughput in the following sense: the throughput achieved by any fixed power selection is at most equal to the throughput achieved by the random-power-selection algorithm.

Other topology-control algorithms for throughput maximization were proposed in [7, 30, 33, 54, 56, 57, 81]. In [25, 54, 56, 57], the total throughput is increased by increasing the spatial reuse. This is achieved by reducing the nodes' TPs. In [30, 33], the total throughput is increased further by not only considering the spatial reuse but also the exposed and hidden nodes. In [81], a TP control algorithm for request-to-send (RTS)/clear-to-send (CTS) based protocols is proposed that decreases the area occupied by links during their transmissions, and it is shown that with this scheme, routing algorithms that favor short hops achieve higher levels of throughput. In [7], the problem of integrated link scheduling and TP control for throughput optimization is shown to be nondeterministic polynomial time (NP) complete. Therefore, a heuristic algorithm is developed. Its goal is to minimize the schedule length necessary to satisfy all the link loads determined by a given routing algorithm. In this way, the total throughput of the network is increased because more scheduling cycles can be performed per time unit.

Our approach differs from the previous algorithms in that it is based on the stability region of the link-scheduling policy. Therefore, our approach can be used for any policy whose stability region has been characterized. Also, our approach differs in that it is based on a distributed collaboration of the flows to increase their corresponding throughput.

Contributions.

The contributions of this chapter are as follows.

- A new framework for the development of distributed algorithms that maximize the total end-to-end throughput in WMNs is proposed. This framework is based on the stability region of the WMN's link-scheduling policy. It consists of a potential game in which a given set of flows act as players that collaborate to maximize the packet rates they can support while guaranteeing stability.
- Based on the proposed framework, a new distributed TP-control algorithm is developed for IEEE 802.16 WMNs. The Nash equilibrium of this game is characterized by means of integer-linear-programming techniques.
- A performance bound for the new TP-control algorithm is found and compared with simulation results. Also, it is shown that in more than 30% of the simulated networks, our algorithm finds a topology that reaches a throughput that is 96% to 100% of the maximum throughput. This performance is superior when compared with the one achieved by the classic TP-control approach based spatial-reuse maximization.

4.2 Network Model

A WMN whose communication graph is denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ is considered. $\mathcal{N}$ is the set of nodes in the network, and $\mathcal{L}$ is the set of links. Links are directional. The link directed from node i to node j is denoted by (i, j). The nodes' transmissions are omnidirectional, and a link belongs to $\mathcal{L}$ if and only if the transmission range of the source node covers the destination node. The interference set of link (i, j) is denoted by $\mathcal{E}^{(i,j)}$ and contains all the links in $\mathcal{L}$ that interfere with link (i, j). The transmission packet rate of link (i, j) is denoted by $\lambda^{(i,j)}$, and the vector of link packet rates is denoted by $\boldsymbol{\lambda} = [\lambda^{(i,j)}]_{(i,j)\in\mathcal{L}}$.

Time is divided into frames, and each frame is divided into a control-subframe and a data-subframe. Control-subframes are divided into control-time-slots that are used for the transmission of scheduling packets, and data-subframes are divided into data-time-slots that are used for the transmission of data packets[4] . A packet reception

[4] This is the frame structure for the WMNs in [1, 71, 74, 84] and references therein.

over link (i,j) is successful if and only if no other link in $\mathcal{E}^{(i,j)}$ is activated while the packet is being transmitted. If such link is activated, there is a collision at node j (i.e, the destination node of link (i,j)) and the packet reception is unsuccessful.

A feasible schedule $\mathbf{h}$ on $\mathcal{H}$, where $\mathcal{H}$ is some subset of $\mathcal{L}$, is a link-activation vector $[0,1]^{|\mathcal{H}|}$ that when all the links in it are activated simultaneously, all the packet receptions are successful. A feasible schedule $\mathbf{h}$ on $\mathcal{H}$ is maximal if, when all the links in $\mathbf{h}$ are activated, no more links can be activated without violating the interference constraints. The set of all possible maximal schedules on $\mathcal{H}$ is denoted by $\mathcal{M}_{\mathcal{H}}$, and its convex hull is denoted by $Co(\mathcal{M}_{\mathcal{H}})$.

The data traffic consists of a set of flows denoted by $\mathcal{F}$. The flows in $\mathcal{F}$ are enumerated. The n-th flow is denoted by f_n. It consists of a path and a mean input-packet rate which are denoted by $\mathcal{P}^{f_n}$ and λ^{f_n} respectively (i.e., $f_n \triangleq (\mathcal{P}^{f_n}, \lambda^{f_n})$). Path $\mathcal{P}^{f_n}$ is the set of nodes on which flow f_n is established. The data traffic on flow f_n is generated at its source node by a data-packet-arrival process that is Poisson distributed with mean λ^{f_n}. The packets leave the network once they arrive at the flow's destination node. The nodes in the path that forward the data packets from source to destination are the intermediate nodes. Every node that is an intermediate or destination node of at least one flow has a maximum packet rate that it can assign to its incoming traffic while guaranteeing the stability of its incoming links' queues. Each of these nodes equally divides its maximum packet rate among all the flows for which it is an intermediate or destination node. The maximum packet rate that node j supports for each of these flows while guaranteeing stability is denoted by $\lambda^j_{\max}$.

The degree of a link is defined as the number of flows it belongs to. It is denoted by $d^{(i,j)}$ for link (i,j). The set of links that belong to at least on flow is denoted by $\mathcal{L}^*$. The vector of link packet rates of the links in $\mathcal{L}^*$ is denoted by $\boldsymbol{\lambda}^* = [\lambda^{(i,j)}]_{(i,j)\in\mathcal{L}^*}$. The set of nodes that belong to at least one flow is denoted by $\mathcal{N}^*$.

Node j is a 1-hop neighbor of node i if node i is within node j's transmission range. Node j is a 2-hop neighbor of node i if node j is a 1-hop neighbor of any of node i's 1-hop neighbors. The active 1-hop neighborhood of a node is defined as the set of 1-hop neighbors that are intermediate or destination nodes of at least one flow. It is denoted by $\mathcal{S}_{\mathrm{a}}^{i}$ for node i. The direct 1-hop neighborhood of a node is the set of 1-hop neighbors that send data packets to the node. Therefore, the direct 1-hop neighbors of a node always precede the node in at least one flow's path. Node i's direct 1-hop neighborhood is denoted by $\mathcal{S}_{\mathrm{d}}^{i}$.

The transmission range of node i is denoted by r^i. The vector of transmission powers of the nodes in $\mathcal{N}^*$ is denoted by $\mathbf{r}$, and the vector of their maximum transmission ranges is denoted by $\mathbf{r}_{\mathrm{max}}$. The Euclidean distance between nodes i and j is denoted by $||i, j||$.

Table F-1 in APPENDIX F summarizes the previous notation of the network model.

4.3 Stability-Region Adaptation

We define the normalized transport capacity of a WMN based on the transport capacity defined in [28] and the queueing-system stability region defined in [69].

Definition 11. For a given set of flows, the normalized transport capacity (NTC) is the maximum total number of packets transported in the WMN from the flows' sources to destinations per distance unit per time unit that guarantees the stability of all link queues.

For example, if there are 2 flows in the network, the distance between the flows' source and destination is 5 meters, the links can transmit up to 10^3 bits per packet, and each flow can transport up to 10^4 bits per second while guaranteeing stability, the NTC is $2 \times 5 \times \frac{10^4}{10^3} = 10^2$ packets-meter per second.

Note that the NTC definition requires a pre-specified set of flows. Therefore, the NTC is not an absolute performance metric of the WMN. It is a performance metric of the WMN for the given set of flows, so a WMN may have different NTC values for different sets of flows.

The highest packet rate that a flow can transport while guaranteeing stability is determined by the packet rates that the intermediate and destination nodes can forward and receive respectively. Let $\mathcal{P}^{f_n}_{\text{int}}$ be the set of intermediate and destination nodes of f_n (i.e., f_n's path without the source node), and let $\{\lambda^j_{\text{max}} : j \in \mathcal{P}^{f_n}_{\text{int}} \text{ for some } f_n \in \mathcal{F}\}$ be the set of upper-bounds for the packet rates the nodes can forward/receive that guarantee stability. These upper-bounds are determined from the link-scheduling policy's stability region as explained in Section 4.3.1. In order to guarantee stability of the WMN, the source node of every flow cannot generate data packets at any rate higher than the minimum upper-bound among the upper-bounds of the intermediate and destination nodes of the flow. That is, the WMN is stable if

$$\lambda^{f_n} < \min\{\lambda^j_{\text{max}} : j \in \mathcal{P}^{f_n}_{\text{int}}\} \quad \forall \quad \lambda^{f_n} \in \mathcal{F}. \tag{4–1}$$

The NTC is then determined by the distance between flows' source and destination nodes and the upper-bounds for the flows' packet rates given by Eq. 4–1. The source and destination locations are given and cannot be modified to increase the NTC. However, the stability region and the paths can be controlled in order to increase the NTC. In this chapter, we consider the problem of increasing the NTC by controlling only the stability region distributively as follows.

4.3.1 Access Schemes and the Stability Region

The stability region is defined for WMNs in which the time is slotted and interfering links take turns to access the slots such that collisions are avoided [69]. The link-scheduling policy coordinates the access of the links to the slots. The stability region of a link-scheduling is the set of packet-arrival rates for which the policy stabilizes the system. In terms of capacity, the optimal link-scheduling policy is the one whose stability region is a superset of the stability region of any other link-scheduling policy. It is given by $\Lambda = \{\boldsymbol{\lambda} : \boldsymbol{\lambda} \preceq \phi \text{ for some } \phi \in Co(\mathcal{M}_{\mathcal{L}})\}$ [69]. In terms of complexity, it is usually the case that the link-scheduling policies with larger stability regions are more complex. For example, the optimal scheduling policy [69] requires the solution of an NP-hard problem (i.e., maximal weighted matching) [66].

The stability region of different link-scheduling policies is characterized by a set of conditions that if satisfied, the WMN is guaranteed to be stable[5] . The basic idea behind our NTC-adaptation approach is based on these sufficient conditions for stability. We claim that these can be controlled by manipulating the network topology such that the highest packet rates the flows can support while guaranteeing stability are increased.

In order to illustrate this idea, we classify the sufficient conditions that guarantee stability into two main categories as follows[6] .

The link-scheduling policies proposed in [26, 44, 51, 84] and Chapter 2 (i.e., greedy and constant-time policies) belong to the first category. These policies are characterized by the sets of interfering links $\{\mathcal{E}^{(i,j)} : (i,j) \in \mathcal{L}\}$ and an increasing function $f^{(i,j)} : \lambda^{\mathcal{E}^{(i,j)}} \to \mathbb{R}_+$, where $\lambda^{\mathcal{E}^{(i,j)}}$ is the set of transmission packet rates of the links in $\mathcal{E}^{(i,j)}$ (i.e., $\lambda^{\mathcal{E}^{(i,j)}} \triangleq \{\lambda^{(k,l)} : (k,l) \in \mathcal{E}^{(i,j)}\}$). Under these policies, the WMNs are stable if $f^{(i,j)} < 0$ for every (i,j) in $\mathcal{L}$. The intuition behind this is that the combination (i.e., $f^{(i,j)}$) of the transmission packet rates of interfering links (i.e., $\lambda^{\mathcal{E}^{(i,j)}}$) cannot exceed certain limit in order to guarantee the stability of the WMN. Otherwise, the queues of the interfering links may build up and never return to the empty state making the network unstable. For example, greedy scheduling policies [84] require that the conditions in Eq. 4–2 be met,

[5] See [27, 65] for a review and comparison of the different link-scheduling policies, and see [34] and Chapter 2 for link-scheduling policies for IEEE 802.11 and 802.16 WMNs.

[6] The link-scheduling policies proposed in [65, 68] (i.e., pick-and-compare policies [65]), are not discussed here since they already reach the optimal stability region, but at a high complexity cost [65].

where $c^{(i,j)}$ is the capacity of link (i,j) in units of packets-per-slot, and κ is the maximum number of non-interfering links in the interference set of any link in the network.

$$\sum_{(k,l)\in\mathcal{E}^{(i,j)}} \frac{\lambda^{(k,l)}}{c^{(k,l)}} < \kappa \quad \forall \quad (i,j)\in\mathcal{L} \tag{4–2}$$

By considering only the links that belong to at least one flow f_n in $\mathcal{F}$ (i.e., $\mathcal{L}^*$), the sets of interfering links $\{\mathcal{E}^{(i,j)} : (i,j) \in \mathcal{L}^*\}$ can be modified by means of topology control in order to minimize the rate at which the functions $f^{(i,j)}$ increase with the transmission packet rates in $\lambda^{\mathcal{E}^{(i,j)}}$. In this way, the NTC of the network can be increased by means of topology control.

The link-scheduling policies discussed in [35, 37, 45, 46] (i.e., greedy maximal scheduling) belong to the second category. These policies are characterized by the concept of local-pooling factor, which is an indicator of how different the effectiveness of the different maximal link schedules are from each other [45]. This effectiveness makes reference to their ability to reduce the queue lengths. The higher the local-pooling factor, the closer the effectiveness of the different maximal schedules. When the different maximal link schedules are similarly effective, GMS policies are able to support packet rates that are closer to the boundaries of the optimal stability region. The local-pooling factor is defined as follows [37]. Let $\mathcal{H}$ be some subset of links (i.e., $\mathcal{H} \subset \mathcal{L}$). The local-pooling σ^* factor of a WMN is then given by

$$\sigma^* = \text{sup}\{\sigma : \sigma\boldsymbol{\mu} \not\leq \boldsymbol{\nu} \;\; \forall \;\; \boldsymbol{\mu},\boldsymbol{\nu} \in Co(\mathcal{M}_{\mathcal{H}}), \;\; \forall \;\; \mathcal{H} \in \mathcal{L}\}.$$

Therefore, σ^* depends on the sets of maximal schedules on $\mathcal{H}$. The sets of maximal schedules are determined by the interference sets $\{\mathcal{E}^{(i,j)} : (i,j) \in \mathcal{L}^*\}$, so σ^* can be modified by considering only the interference sets of links that belong to at least one flow f_n in $\mathcal{F}$ (i.e., $\mathcal{L}^*$) and controlling the topology of the network. In this way, the NTC of the WMN can be increased by means of topology control.

All the previous link-scheduling polices depend on the links in the interfering sets $\{\mathcal{E}^{(i,j)} : (i,j) \in \mathcal{L}^*\}$, and these sets depend on the nodes transmission ranges $\mathbf{r}$. Therefore, the stability regions can be controlled by means of $\mathbf{r}$ in order to adapt them to the given set of flows. The goal of this adaptation can be formulated as the improvement of the NTC by solving the NTC-adapation problem, which is defined next.

The vector of transmission ranges of the nodes that belong to at least one flow, i.e., $\mathbf{r}$, is feasible if the nodes do not exceed their maximum TP (i.e., $\mathbf{r} \preceq \mathbf{r}_{\text{max}}$) and none of the links in $\mathcal{L}^*$ is broken (i.e., no flow is broken). Let $\mathbf{r}_{\text{min}}$ be the vector of minimum transmission ranges that guarantee that none of the links in $\mathcal{L}^*$ is broken (i.e., $r^i_{\text{min}}, r^j_{\text{min}} \geq ||i,j|| \; \forall \; i \in \mathcal{S}^j_d, j \in \mathcal{N}$). Then, $\mathbf{r}$ is feasible if $\mathbf{r}_{\text{min}} \preceq \mathbf{r} \preceq \mathbf{r}_{\text{max}}$.

Let $\lambda^{f_n}_{\text{max}}$ be the highest packet rate that flow f_n supports while guaranteeing stability. According to Eq. 4–1, $\lambda^{f_n}_{\text{max}}$ is given by the minimum of the highest packet rates supported by the nodes in $\mathcal{P}^{f_n}_{\text{int}}$ (i.e., $\min\{\lambda^j_{\text{max}} : j \in \mathcal{P}^{f_n}_{\text{int}}\}$). According to the previous discussion on stability regions, these packet rates are a function of the link-interference sets $\{\mathcal{E}^{(i,j)} : (i,j) \in \mathcal{L}^*\}$, which in turn, are a function of the transmission ranges $\mathbf{r}$. This dependence of $\mathcal{E}^{(i,j)}$ on $\mathbf{r}$ is denoted by $\mathcal{E}^{(i,j)}_{\text{r}}$. Therefore, the highest packet rates supported by the nodes are a function of the transmission ranges (i.e., $\lambda^j_{\text{max}}(\mathcal{E}^j_{\text{r}}) : \mathbb{R}^{|\mathcal{L}^*|}_+ \to \mathbb{R}_+$, where $\mathcal{E}^j_{\text{r}} \triangleq \{\mathcal{E}^{(i,j)}_{\text{r}} : i \in \mathcal{S}^j_{\text{d}}\}$), and $\lambda^{f_n}_{\text{max}}$ depends on $\mathbf{r}$ as shown in Eq. 4–3.

$$\lambda^{f_n}_{\text{max}}(\mathbf{r}) \triangleq \min\{\lambda^j_{\text{max}}(\mathcal{E}^j_{\text{r}}) : j \in \mathcal{P}^{f_n}_{\text{int}}\}. \tag{4–3}$$

Definition 12. The NTC-adaptation problem (NTC-AP) is the problem of finding a feasible vector of transmission ranges $\mathbf{r}^*$ such that there is no feasible $\mathbf{r}$ that meets the following condition: $\lambda^{f_n}_{\text{max}}(\mathbf{r}) > \lambda^{f_n}_{\text{max}}(\mathbf{r}^*)$ for at least one f_n in $\mathcal{F}$ and $\lambda^{f_n}_{\text{max}}(\mathbf{r}) \geq \lambda^{f_n}_{\text{max}}(\mathbf{r}^*)$ for all other f_n in $\mathcal{F}$, i.e., there is no vector $[\lambda^{f_n}_{\text{max}}(\mathbf{r})]_{f_n \in \mathcal{F}}$ that dominates vector $[\lambda^{f_n}_{\text{max}}(\mathbf{r}^*)]_{f_n \in \mathcal{F}}$.

Therefore, the goal of the NTC-AP is to find a vector $\mathbf{r}^*$ of feasible node-transmission ranges that guarantees Pareto optimality. This can be done, according to Eq. 4–3, by controlling the values of the highest packet rates supported by the nodes $\{\lambda^j_{\max}(\mathcal{E}^j_{\mathrm{r}}) : j \in \mathcal{P}^{f_n}_{\mathrm{int}}, f_n \in \mathcal{F}\}$ which are defined by the particular link-scheduling policy adopted in the network.

In the following section, the framework for solving the NTC-AP is proposed. It is based on potential-game theory. In this framework, the flows are the players of the potential game, and they adapt the network topology by controlling the TPs of the nodes that affect the highest packet rates that guarantee stability. Using the framework, a distributed algorithm for solving the NTC-AP in IEEE 802.16 WMNs under the greedy policy proposed in Chapter 2 is proposed and analyzed in Section 4.4.

4.3.2 Distributed TP-Control Algorithms using Potential Games

The following definitions are necessary to formulate the NTC-AP as a normal-form game. Table F-2 in Appendix F summarizes the notation introduced in this formulation.

The interference set of link (i, j) under the nodes' transmission ranges given by $\mathbf{r}$ is denoted by $\mathcal{E}^{(i,j)}_{\mathrm{r}}$. When all the nodes transmit at their maximum TP, this set is denoted by $\mathcal{E}^{(i,j)}_{\max}$.

Note that the interference set $\mathcal{E}^{(i,j)}_{\mathrm{r}}$ depends on the TP of the nodes that belong to at least one flow (i.e., $\mathbf{r}$) only. It does not depend on the TPs of nodes that do not belong to any flow because these nodes do not transmit nor receive any packets.

The NTC-AP is formulated as a normal-form game as follows. Individual flows form the player set $\mathcal{F} \triangleq \{f_1, f_2, \ldots, f_n, \ldots, f_N\}$. Each flow f_n can autonomously set the TP of the nodes in the set $\mathcal{S}^{f_n}$ which is defined as follows. $\mathcal{S}^{f_n}$ is the set of nodes that are able to affect f_n's highest packet rate that guarantees stability when all the allowed TP levels are considered. Therefore, these are the nodes that are within some maximum distance from the intermediate and destination nodes of the flow (i.e., $\mathcal{P}^{f_n}_{\mathrm{int}}$). This distance depends on the maximum transmission range of the nodes and the interference model. Specifically, $\mathcal{S}^{f_n}$ is defined in terms of the sets $\mathcal{S}^j$. $\mathcal{S}^j$ is the set of nodes that are able to affect the highest packet rate that node j supports for any of its incoming flows (i.e., λ^j_{max}) when all the allowed TP levels are considered. Therefore, $\mathcal{S}^j$ is determined from the sets of interfering links $\mathcal{E}^{(i,j)}_{\text{max}}$, i.e., link (i,j)'s interfering links when all the nodes transmit at the maximum TP. $\mathcal{S}^j$ is given by Eq. 4–4[7] . According to Eq. 4–4, in $\mathcal{S}^j$, only the incoming links of node j formed with its direct 1-hop neighbors (i.e., $\mathcal{S}^j_{\text{d}}$) are considered. The interference sets of these links are calculated when all the nodes transmit at the maximum TP so that all the nodes that are able to affect λ^j_{max} are considered.

$$\mathcal{S}^j \triangleq \left\{ k : k \in \bigcup_{i \in \mathcal{S}^j_{\text{d}}} \mathcal{E}^{(i,j)}_{\text{max}} \right\} \tag{4–4}$$

$\mathcal{S}^{f_n}$ is defined by Eq. 4–5. It is the union of the $\mathcal{S}^j$ of the nodes along f_n's path with the exception of the source node (i.e., $j \in \mathcal{P}^{f_n}_{\text{int}}$). The source node is excluded because this node does not limit the flow's highest packet rate that guarantees stability. It is the ability of the intermediate and destination nodes to forward and receive the data packets generated by the source node what limits the highest packet rate supported by the flow.

$$\mathcal{S}^{f_n} \triangleq \bigcup_{j \in \mathcal{P}^{f_n}_{\text{int}}} \mathcal{S}^j \tag{4–5}$$

The action space of flow f_n is the set of feasible transmission ranges of the nodes in $\mathcal{S}^{f_n}$. This set is denoted by $\mathcal{R}_n$. An action of f_n is denoted by the vector $\mathbf{r}_n$. Therefore, action $\mathbf{r}_n$ belongs to $\mathcal{R}_n$, and it specifies the transmission ranges controlled by f_n (i.e., the transmission ranges of the nodes in $\mathcal{S}^{f_n}$). The game's action space $\mathcal{R}$ is the set of feasible transmission ranges of the nodes controlled by the flows (i.e., the nodes in $\bigcup_{f_n \in \mathcal{F}} \mathcal{S}^{f_n}$). Action $\mathbf{r}_{-n}$ specifies the transmission ranges not controlled by f_n (i.e.,

[7] In Eq. 4–4 and for the rest of the analysis, it is said that a node k belongs to a set of links $\mathcal{H}$ if k is present in any of the links in $\mathcal{H}$.

the transmission ranges of the nodes in $(\bigcup_{f_i \in \mathcal{F}} \mathcal{S}^{f_i}) \backslash \mathcal{S}^{f_n})$. Therefore, $\mathbf{r}_{-n}$ belongs to $\mathcal{R}_{-n} = \mathcal{R} \backslash \mathcal{R}_n$.

When flow f_n makes a move (i.e., updates its action vector $\mathbf{r}_n$), the highest packet rates that it and other flows support may be affected. This relation between f_n and other flows is reflected on its utility function $\mu_n : \mathcal{R} \to \mathbb{R}$ in order to enable collaboration among flows. Let $\mathcal{F}^n$ be the set of flows whose highest supported packet rates are affected by any of the moves of f_n (Eq. 4–6). Flow f_n's utility function μ_n is defined from the highest packet rates supported by the flows in $\mathcal{F}^n$ as given by Eq. 4–7. Therefore, according to Eq. 4–7, flow f_n's utility increases not only when its highest supported packet rate $\lambda_{\max}^{f_n}$ increases, but also when the highest supported packet rates of the flows in $\mathcal{F}^n$ increase as well. In this way, f_n is encouraged to collaborate with the flows that are affected or potentially affected by its moves, i.e., when any flow makes a move, the flow tends to benefit other flows as well because in this way its utility function can be increased more effectively.

$$\mathcal{F}^n \triangleq \{f_i : \mathcal{S}^{f_n} \cap \mathcal{S}^{f_i} \neq \emptyset, f_i \in \mathcal{F}\} \tag{4–6}$$

$$\mu_n(\mathbf{r}) \triangleq \sum_{f_i \in \mathcal{F}^n} \lambda_{\max}^{f_i}(\mathbf{r}) \tag{4–7}$$

The vector of utility functions is $\mathbf{u} = [\mu_1, \mu_2, ..., \mu_N] : \mathcal{R} \to \mathbb{R}^N$.

Definition 13. The game[8] NTC-AP $= \langle \mathcal{F}, \mathcal{R}, \mathbf{u}(\mathbf{r}) \rangle$ is an ordinal potential game (OPG) if there exists a function $V : \mathcal{R} \to \mathbb{R}$ such that $\forall f_n \in \mathcal{F}$, $\forall \mathbf{r}_{-n} \in \mathcal{R}_{-n}$, and $\mathbf{x}_n, \mathbf{y}_n \in \mathcal{R}_n$

$$V(\mathbf{x}_n, \mathbf{r}_{-n}) - V(\mathbf{y}_n, \mathbf{r}_{-n}) > 0 \Leftrightarrow \mu_n(\mathbf{x}_n, \mathbf{r}_{-n}) - \mu_n(\mathbf{y}_n, \mathbf{r}_{-n}) > 0$$

V is called the ordinal potential function (OPF) of the NTC-AP game.

[8] See [53] for the theory of potential games.

Theorem 4.1. *The game NTC-AP* $= \langle \mathcal{F}, \mathcal{R}, \mathbf{u}(\mathbf{r}) \rangle$ *is an OPG. An OPF is given by* $\lambda_T(\mathbf{r})$*, which is the total highest packet rate supported by the network for the given set of flows* $\mathcal{F}$ *(Eq. 4–8)*[9] *.*

$$\lambda_T(\mathbf{r}) \triangleq \sum_{f_n \in \mathcal{F}} \min\{\lambda^j_{max}(\mathbf{r}) : j \in \mathcal{P}^{f_n}_{int}\} \tag{4–8}$$

The proof of Theorem 4.1 is as follows. From Eq. 4–3, Eq. 4–8, and Eq. 4–7, $\lambda_\mathrm{T}(\mathbf{r})$ can be rewritten as follows.

$$\lambda_\mathrm{T}(\mathbf{r}) = \sum_{f_i \in \mathcal{F}} \lambda^{f_i}_{\mathrm{max}}(\mathbf{r}) = \sum_{f_i \in \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{r}) + \sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{r}) = \mu_n(\mathbf{r}) + \sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{r})$$

Therefore,

$$\lambda_\mathrm{T}(\mathbf{x}_n, \mathbf{r}_{-n}) - \lambda_\mathrm{T}(\mathbf{y}_n, \mathbf{r}_{-n}) = \mu_n(\mathbf{x}_n, \mathbf{r}_{-n}) - \mu_n(\mathbf{y}_n, \mathbf{r}_{-n}) + \sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{x}_n, \mathbf{r}_{-n}) - \sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{y}_n, \mathbf{r}_{-n})$$

The highest packet rates supported by the flows in $\mathcal{F} \setminus \mathcal{F}^n$ (i.e., $\{\lambda^{f_i}_{\mathrm{max}} : f_i \in \mathcal{F} \setminus \mathcal{F}^n\}$) are independent of the actions of f_n (i.e., $\lambda^{f_i}_{\mathrm{max}}(\mathbf{x}_n, \mathbf{r}_{-n}) = \lambda^{f_i}_{\mathrm{max}}(\mathbf{y}_n, \mathbf{r}_{-n})\ \ \forall f_i \in \mathcal{F} \setminus \mathcal{F}^n, \forall \mathbf{r}_{-n} \in \mathcal{R}_{-n}, \forall \mathbf{x}_n, \mathbf{y}_n \in \mathcal{R}_n$) because, by definition (Eq. 4–6), $\mathcal{F}^n$ is the set of flows whose highest packet rates are affected by f_n's actions $\mathbf{r}_n \in \mathcal{R}_n$. Therefore,

$$\sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{x}_n, \mathbf{x}_{-n}) = \sum_{f_i \in \mathcal{F} \setminus \mathcal{F}^n} \lambda^{f_i}_{\mathrm{max}}(\mathbf{y}_n, \mathbf{x}_{-n})$$

This concludes the proof of Theorem 4.1.

In the following section, the NTC-AP game is analyzed for IEEE 802.16 WMNs. A distributed algorithm is proposed for flows to determine their individual actions.

[9] The direct dependence of λ^j_{max} on $\mathcal{E}^{(i,j)}_\mathrm{r}$, i.e., $\lambda^j_{\mathrm{max}}(\mathcal{E}^{(i,j)}_\mathrm{r})$ (Eq. 4–3), has been omitted in Eq. 4–8, i.e., $\lambda^j_{\mathrm{max}}(\mathbf{r})$, to simplify the notation.

The convergence and performance of the algorithm are characterized with the Nash equilibria of the game.

4.4 Stability-Region Adaptation in Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 Wireless Multihop Networks (WMN)

In IEEE 802.16 WMNs [1], time is divided into frames according to the description given in Section 4.2. In the control-time-slots, the nodes are selected by an election algorithm [1, 13, 78] such that when a node is selected, none of its 1-hop and 2-hop neighbors are selected. The election algorithm selects nodes in every control-time-slot, and a node transmits a scheduling packet every time it is selected. A scheduling packet carries three types of messages. These are $\mathit{request}$, grant, and $\mathit{confirmation}$. There can be more than one message per type in a scheduling packet (e.g., a scheduling packet may carry 3 requests, 1 grant, and 2 confirmations). The nodes make reservations of future data-time-slots for transmitting data packets stored in their link queues. The reservations are done on a per-link basis by means of a three-way handshake. First, when the link's source node is selected by the election algorithm, it sends a request to the link's destination node and waits for a reply. Then, when the link's destination node is selected, it replies by sending a grant to the link's source node. Finally, when the link's source node is selected again, it sends a confirmation, which is a copy of the grant. The nodes that have either the link's source or destination node as 1-hop neighbors (i.e., the nodes within the transmission range of either the link's source or destination node) listen to the transmitted grant and confirmation. Every node in the WMN keeps track of the reserved future data-time-slots. Based on this information, the nodes determine the future data-time-slots they include in their requests and grants such that collisions are avoided. The nodes are able to use any link-scheduling policy[10] that adapts to the IEEE-802.16 standard [1] for determining such sets of data-time-slots (i.e., requested and granted data-time-slots). In the following, the 2-hop interference model and the link-scheduling policy proposed in Chapter 2 will be adopted. We give a brief description of this policy and its stability region in order to illustrate how the NTC-AP game can be used in IEEE 802.16 WMNs.

The adopted policy is the greedy-maximal reservation-based-distributed-scheduling (GM-RBDS) policy (Chapter 2), which is summarized as follows.

Whenever a node is selected,

- for every outgoing link, confirm any non-interfering grants received since the previous time the node was selected and schedule data packets for transmission according to these grants
- for every incoming link, grant as many data-time-slots as requested at the future data-time-slots that have not been reserved yet and that are the closest in time
- for every outgoing link, request as many data-time-slots as unscheduled data packets stored in the link's queue

The size of the stability region of the GM-RBDS policy depends on the ability of the links to perform the three-way handshakes successfully[11]. If the probability that a link finishes successfully a three-way handshake is low, the link's queue will decrease at a lower rate. Therefore, the link's ability to forward data packets within some time range is going to be lower (i.e., the highest packet rate supported by the link is lowered), and this reduces the size of the stability region. The probability that a three-way handshake is successful depends on the grants received by the link's source node from the time instant it sends the request until the time instant it receives the grant from the link's

[10] The IEEE 802.16 standard [1] does not specify any particular link scheduling policy. It only provides the framework for implementing them (i.e., election algorithm, control and data subframes, and control messages).

[11] See Chapter 2 for a detailed stability analysis of the GM-RBDS policy.

destination node. If any of these grants is not heard by the link's destination node and the link's destination node's grant overlaps them, the three-way handshake is unsuccessful. That is, the link's source node will not be able to confirm the grant sent by the link's destination node because other grants previously received already reserved the future data-time-slots granted by the link's destination node. Therefore, the highest packet rate that a node supports for the flows it forwards or serves as a sink node depends on the following nodes: the nodes that transmit grants that are not received by it and are received by its 1-hop neighbors (i.e., hidden nodes). For example, consider some link (i, j). The nodes that transmit the grants received by i and j are the active 1-hop neighbors $\mathcal{S}_{\mathrm{a}}^{i}$ and $\mathcal{S}_{\mathrm{a}}^{j}$ of i and j. The nodes in $\mathcal{S}_{\mathrm{a}}^{i} \backslash \mathcal{S}_{\mathrm{a}}^{j}$ transmit grants that are received by i but not by j. They are hidden from j. Therefore, if the grants transmitted by the nodes in $\mathcal{S}_{\mathrm{a}}^{i} \backslash \mathcal{S}_{\mathrm{a}}^{j}$ overlap the grants for link (i, j), which are transmitted by j, link (i, j)'s three-way handshakes are unsuccessful. When all the incoming links of node j that belong to at least one flow are considered, the maximum packet rate $\lambda_{\mathrm{max}}^{j}$ that node j is able to grant for each of its incoming flows is given by Eq. 4–9[12] .

Theorem 4.2. *Network $\mathcal{G}$ under the GM-RBDS policy and the 2-hop interference model is stable if the packet rate of every incoming flow of node j is not greater than λ_{max}^{j} for every j in $\mathcal{N}$, where λ_{max}^{j} is given by Eq. 4–9.*

$$\lambda_{max}^{j} = \frac{1}{5 \sum_{i \in \mathcal{S}_{d}^{j}} d^{(i,j)} |\mathcal{S}_{a}^{i} \backslash \mathcal{S}_{a}^{j}|} \tag{4–9}$$

Corollary 1. *Network $\mathcal{G}$ under the GM-RBDS and the 2-hop interference model is stable if the packet rate λ^{f_n} of every flow f_n in $\mathcal{F}$ satisfies Eq. 4–10.*

$$\lambda^{f_n} < \min \left\{ \frac{1}{5 \sum_{i \in \mathcal{S}_{d}^{j}} d^{(i,j)} |\mathcal{S}_{a}^{i} \backslash \mathcal{S}_{a}^{j}|} : j \in \mathcal{P}_{int}^{f_n} \right\} \tag{4–10}$$

[12] See Chapter 2 for the proof of Theorem 4.2.

The only factor in Eq. 4–10 that depends on the transmission ranges $\mathbf{r}$ is $|\mathcal{S}_{\mathrm{a}}^{i} \backslash \mathcal{S}_{\mathrm{a}}^{j}|$, which will be denoted by $a_{\mathrm{r}}^{(i,j)}$. The expression for $\lambda_{\mathrm{max}}^{f_n}(\mathbf{r})$ that defines $\mathbf{u}(\mathbf{r})$ (Eq. 4–7) and λ_{T} (Eq. 4–8) for the NTC-AP game in IEEE 802.16 WMNs under the GM-RBDS policy is then given as follows.

$$\lambda_{\mathrm{max}}^{f_n}(\mathbf{r}) = \min \left\{ \frac{1}{5 \sum_{i \in \mathcal{S}_{\mathrm{d}}^{j}} d^{(i,j)} a_{\mathrm{r}}^{(i,j)}} : j \in \mathcal{P}_{\mathrm{int}}^{f_n} \right\} \tag{4–11}$$

Definition 14. The bottleneck node of flow f_n is the node among the flow's intermediate and destination nodes that has the minimum highest packet rate these nodes support, i.e., let j be the bottleneck node of f_n, then $j = \mathrm{argmin}_{i \in \mathcal{P}_{\mathrm{int}}^{f_n}} \lambda_{\mathrm{max}}^{i}$.

In the game, the goal of the flows is to increase the value of their utility functions (i.e., $\mathbf{u}(\mathbf{r})$) by adjusting the values of the factors $\{a_{\mathrm{r}}^{(i,j)} : (i,j) \in \mathcal{L}^*\}$. However, given that $\lambda_{\mathrm{max}}^{f_n}(\mathbf{r})$ is a nonlinear function of $a_{\mathrm{r}}^{(i,j)}$, the utilities $\mathbf{u}(\mathbf{r})$ and the OPF $\lambda_{\mathrm{T}}(\mathbf{r})$ are also nonlinear functions of $a_{\mathrm{r}}^{(i,j)}$.

In the following, the NTC-AP game in IEEE 802.16 WMNs is approximated with a game whose utilities and OPF are linear functions of $a_{\mathrm{r}}^{(i,j)}$. This linear version of the NTC-AP game will be called Lin-NTC-AP. Based on the linearity, the Nash equilibria of the Lin-NTC-AP game is analyzed in Section 4.4.1, and the difference between the solutions reached by the NTC-AP and Lin-NTC-AP games is upper-bounded in Section 4.4.2.

Proposition 4.1. *The solution set of the optimization problem given by Eq. 4–12 is also a solution to the NTC-AP (Defintion 12).*

$$\begin{aligned} \text{minimize} \quad & \sum_{f_n \in \mathcal{F}} \frac{1}{5\lambda_{max}^{f_n}(\mathbf{r})} \\ \text{subject to} \quad & \mathbf{r}_{min} \preceq \mathbf{r} \preceq \mathbf{r}_{max}. \end{aligned} \tag{4–12}$$

The proof is as follows. First, it is noted that a solution to Eq. 4–13 is also a solution to the NTC-AP, and second, it is proved that Eq. 4–13 and Eq. 4–12 are equivalent.

$$
\begin{aligned}
&\textit{find some } \mathbf{r}^* \\
&\textit{such that i) } \lambda_{\max}^{f_n}(\mathbf{r}^*) \geq \lambda_{\max}^{f_n}(\mathbf{r}) \;\; \forall\, f_n \in \mathcal{F} \\
&\textit{ii) } \mathbf{r}_{\min} \preceq \mathbf{r}^*, \mathbf{r} \preceq \mathbf{r}_{\max}
\end{aligned}
\tag{4–13}
$$

A solution to Eq. 4–13 is also a solution to the NTC-AP due to its first constraint, i.e., the constraint that $\lambda_{\max}^{f_n}(\mathbf{r}^*) \geq \lambda_{\max}^{f_n}(\mathbf{r}) \;\; \forall\, f_n \in \mathcal{F}$ guarantees that vector $[\lambda_{\max}^{f_n}(\mathbf{r}^*)]_{f_n \in \mathcal{F}}$ is not dominated by any other vector $[\lambda_{\max}^{f_n}(\mathbf{r})]_{f_n \in \mathcal{F}}$.

The equivalence of Eq. 4–13 and Eq. 4–12 is due to the fact that $\mathbf{r}$ solves Eq. 4–13 iff $\mathbf{r}$ minimizes Eq. 4–12.

Let $\mathbf{r}_1$ be a solution to Eq. 4–13. Therefore, $\lambda_{\max}^{f_n}(\mathbf{r}_1) \geq \lambda_{\max}^{f_n}(\mathbf{r})$ for all $\mathbf{r} : \mathbf{r}_{\min} \preceq \mathbf{r} \preceq \mathbf{r}_{\max}$ and for all $f_n \in \mathcal{F}$. Assume that $\mathbf{r}_1$ is not a solution to Eq. 4–12. Therefore, there must be some $\mathbf{r}_2 : \mathbf{r}_{\min} \preceq \mathbf{r}_2 \preceq \mathbf{r}_{\max}$ and some $f_n \in \mathcal{F}$ such that $\lambda_{\max}^{f_n}(\mathbf{r}_2) > \lambda_{\max}^{f_n}(\mathbf{r}_1)$. However, this is impossible since $\mathbf{r}_1$ is an optimal solution to Eq. 4–13.

The same argument can be made in the opposite direction. That is, if $\mathbf{r}_1$ is a solution to Eq. 4–12, then it is also a solution to Eq. 4–13. This concludes the proof of the proposition.

In the following, the concept of contention level is introduced in order to define the Lin-NTC-AP game in Eq. 4–16. The contention level of node j is defined according to Eq. 4–14, i.e., the summation of the number of hidden active nodes (i.e., $a_{\mathrm{r}}^{(i,j)}$) in every incoming link (i.e., $\{(i,j) : i \in \mathcal{S}_{\mathrm{d}}^{j}\}$) weighted by the link degree (i.e., $d^{(i,j)}$). Therefore, when the contention experienced by a node is high, the amount of traffic it can forward and/or receive is low because its grants may overlap with higher probability any of the grants transmitted by the hidden active nodes.

$$
c^{j}(\mathbf{r}) \triangleq \sum_{i \in \mathcal{S}_{\mathrm{d}}^{j}} d^{(i,j)} a_{\mathrm{r}}^{(i,j)}
\tag{4–14}
$$

From Eq. 4–11, Eq. 4–12, and Eq. 4–14, the problem given by Eq. 4–13 (i.e., NTC-AP) can be rewritten in terms of the contention level as follows.

$$\begin{aligned} &\textit{minimize} \quad \sum_{f_n \in \mathcal{F}} \max\{c^j(\mathbf{r}) : j \in \mathcal{P}_{\text{int}}^{f_n}\} \\ &\textit{subject to} \quad \mathbf{r}_{\min} \preceq \mathbf{r} \preceq \mathbf{r}_{\max} \end{aligned} \tag{4–15}$$

Therefore, the NTC-AP reduces to finding a set of feasible transmission ranges that minimizes the highest contention level experienced by every flow.

The solution to Eq. 4–13 (i.e., NTC-AP) is approximated with the solution to the Lin-NTC-AP. The Lin-NTC-AP is formulated as given by Eq. 4–16. The terms $c_{\text{T}}^{f_n}(\mathbf{r})$ and $c_{\text{V}}^{f_n}(\mathbf{r})$ in Eq. 4–16 are defined as follows. The total contention experienced by f_n is $c_{\text{T}}^{f_n}(\mathbf{r}) \triangleq \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} c^j(\mathbf{r})$. The mean contention experienced by f_n is $\bar{c}^{f_n}(\mathbf{r}) \triangleq |\mathcal{P}_{\text{int}}^{f_n}|^{-1} \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} c^j(\mathbf{r})$. The contention variation experienced by f_n is $c_{\text{V}}^{f_n}(\mathbf{r}) \triangleq \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} |c^j(\mathbf{r}) - \bar{c}^{f_n}(\mathbf{r})|$.

$$\begin{aligned} &\textit{minimize} \quad \sum_{f_n \in \mathcal{F}} (c_{\text{T}}^{f_n}(\mathbf{r}) + c_{\text{V}}^{f_n}(\mathbf{r})) \\ &\textit{subject to} \quad \mathbf{r}_{\min} \preceq \mathbf{r} \preceq \mathbf{r}_{\max}. \end{aligned} \tag{4–16}$$

Therefore, in the Lin-NTC-AP, the goal is to minimize the total contention and the contention variation experienced by the flows. Intuitively, the Lin-NTC-AP approximates the NTC-AP based on the following two observations. First, by making sure that the maximum contention experienced by a flow at a certain node is not too different from the contention at the other nodes of the flow, the contention along the flow's path is more uniform. This is achieved when the contention variation is reduced. Second, given that the contention is more uniform, reducing the total contention experienced by the flow also reduces the maximum contention along the flow's path, which is the goal in the NTC-AP (Eq. 4–15).

In the Lin-NTC-AP game, which is defined next, each flow competes for minimizing the total contention and contention variation of itself and of any other flows it affects with its actions. Therefore, the player set $\mathcal{F}$, the players' action sets $\{\mathcal{R}_n : f_n \in \mathcal{F}\}$, and the game's action space $\mathcal{R}$ are the same for both the NTC-AP and the Lin-NTC-AP games[13] . The only difference in the formulation of these two games is the utility functions. In the Lin-NTC-AP game, the utility function $\mu_n^*(\mathbf{r}) : \mathcal{R} \to \mathbb{R}$ of f_n is defined as follows.

$$\mu_n^*(\mathbf{r}) \triangleq \sum_{f_i \in \mathcal{F}^n} (c_\mathrm{T}^{f_i}(\mathbf{r}) + c_\mathrm{V}^{f_i}(\mathbf{r}))$$

Note that the Lin-NTC-AP game is also an OPG[14] . An OPF for this game is given by Eq. 4–17. This OPF represents the total contention and contention variation experienced by all the flows.

$$c_{\mathrm{Lin}}(\mathbf{r}) \triangleq \sum_{f_n \in \mathcal{F}} (c_\mathrm{T}^{f_n}(\mathbf{r}) + c_\mathrm{V}^{f_n}(\mathbf{r})) \tag{4–17}$$

Algorithm 2, called WiMAX-Mesh-NTC, is proposed for the flows to decrease the value of their utility functions[15] . The algorithm requires that the transmission ranges be initialized with their minimum values that do not disconnect any of the flows. The algorithm also requires the following information which is constant for the whole duration of the game: the set $\mathcal{P}^n$ of paths of the flows in $\mathcal{F}^n$ (i.e, $\mathcal{P}^n \triangleq \{\mathcal{P}^{f_n} : f_n \in \mathcal{F}^n\}$), the degrees $\mathcal{D}^n$ of the incoming links of the nodes that belong to the paths in $\mathcal{P}^n$ and that

[13] See Section 4.3.2 for their definitions.

[14] The proof that the Lin-NTC-AP game is an OPG follows the same argument of the proof that the NTC-AP game is an OPG (i.e., proof of Theorem 4.1). It has been omitted for the sake of brevity.

[15] The WiMAX-Mesh-NTC algorithm is based on the HSRA algorithm proposed in Chapter 3.

have degree greater than zero (i.e., $\mathcal{D}^n \triangleq \{d^{(i,j)} : j \in \mathcal{P}^n, i \in \mathcal{S}_{\mathrm{d}}^j\}$). Finally, the algorithm requires the following information that changes with the actions taken by the flows: the set $\mathcal{H}_{\mathrm{r}}^n$ of hidden active nodes of the links whose degrees were included in $\mathcal{D}^n$ (i.e., $\mathcal{H}_{\mathrm{r}}^n \triangleq \{\mathcal{S}_{\mathrm{a}}^i(\mathbf{r}) \backslash \mathcal{S}_{\mathrm{a}}^j(\mathbf{r}) : j \in \mathcal{P}^n, i \in \mathcal{S}_{\mathrm{d}}^j\}$).

Algorithm 2 Player f_n's Algorithm

Require: TPs are set to their minimum values that do not disconnect any of the flows

procedure WiMAX-Mesh-NTC(f_n,$\mathcal{P}^n$, $\mathcal{D}^n$, $\mathcal{H}_{\mathrm{r}}^n$)

    $\mu_n^* \leftarrow$ CalculateMyUtility(f_n, $\mathcal{P}^n$, $\mathcal{D}^n$, $\mathcal{H}_{\mathrm{r}}^n$)

    $b \leftarrow$ GetMyBottleNeckNode(f_n, $\mathcal{P}^n$, $\mathcal{D}^n$, $\mathcal{H}_{\mathrm{r}}^n$)

    $\mathcal{C} \leftarrow$ GetHiddenActiveNodes(b, $\mathcal{H}_{\mathrm{r}}^n$)

    $c \leftarrow$ SelectTheBestHiddenNode(μ_n^*, $\mathcal{C}$, $\mathcal{P}^n$)

    if $c \neq \emptyset$ **then**

        $\mathbf{r}(c) \leftarrow ||c, b||$

    end if

end procedure

The WiMAX-Mesh-NTC algorithm works as follows. In the game, the flows take turns to take action (e.g., by passing a token from flow to flow). For every of its turns, a flow calculates its action according to WiMAX-Mesh-NTC as follows. The flow first calculates its utility function $\mu_n^*(\mathbf{r})$ for the current set of TPs $\mathbf{r}$ (line 2 in Algorithm 2). Then, it finds its bottleneck node (line 3 in Algorithm 2) and the active nodes hidden from the bottleneck node (line 4 in Algorithm 2). If b denotes the bottleneck node (line 3 in Algorithm 2), the set of active nodes hidden from the bottleneck node is $\mathcal{C} = \{\mathcal{S}_{\mathrm{a}}^i(\mathbf{r}) \backslash \mathcal{S}_{\mathrm{a}}^b(\mathbf{r}) : i \in \mathcal{S}_{\mathrm{d}}^b\}$ (line 4 in Algorithm 2). The flow then selects one of the nodes in this set (line 5 in Algorithm 2) as follows. For every node in the set, the flow checks whether the node's TP can be increased by the minimum amount of power necessary to cover the bottleneck node. If such a TP increase is possible, the flow calculates its utility function with the new TP. The node that was able to decrease the flow's utility function the most is selected. When none of the nodes is able to decrease the utility function, none of the them is selected. Therefore, at the end (lines 6 to 8), the flow checks whether any of the active nodes hidden from the bottleneck node was selected, and if it was, it sets the TP of the selected node to the minimum TP that covers the bottleneck node (line 7 in Algorithm 2).

Therefore, the goal of the WiMAX-Mesh-NTC algorithm is to decrease the flow's utility function by increasing the TP of a node so that the flow's bottleneck node is able to listen to it. This node is the active node hidden from the bottleneck node that decreases the flow's utility function the most.

4.4.1 Nash Equilibria and Linear Integer Programming

The reason for approximating the NTC-AP game with the Lin-NTC-AP game is that due to the linear-integer-programming nature of Lin-NTC-AP (Theorem 4.3), its set of optimal solutions (i.e., $\mathrm{argmin}_{\mathrm{r}_{\min} \preceq \mathrm{r} \preceq \mathrm{r}_{\max}} \sum_{f_n \in \mathcal{F}} (c_{\mathrm{T}}^{f_n}(\mathbf{r}) + c_{\mathrm{V}}^{f_n}(\mathbf{r})))$ can be characterized with the Nash-equilibria (Theorem 4.4). Therefore, given that the Lin-NTC-AP game is potential, it is guaranteed to converge to the optimal solution of the Lin-NTC-AP (Corollary 2).

Theorem 4.3. *The Lin-NTC-AP is a linear integer program.*

The proof of Theorem 4.3 is as follows. The Lin-NTC-AP can be formulated in terms of the vector $\mathbf{a} \triangleq [a_{\mathrm{r}}^{(i,j)}]_{(i,j) \in \mathcal{L}^*}$, i.e., the variables are the elements of vector $\mathbf{a}$ and not the elements of vector $\mathbf{r}$. Any set of TPs $\mathbf{r}$ that satisfies the solution $\mathbf{a}$ achieves the same objective-function value. The objective function c_{Lin} is formulated in terms of $\mathbf{a}$ as follows[16] .

[16] Given that the variables are $[a_{\mathrm{r}}^{(i,j)}]_{(i,j) \in \mathcal{L}^*}$, the $\mathbf{r}$ subscript is dropped.

$$
\begin{aligned}
c_{\text{Lin}}(\mathbf{a}) &= \sum_{f_n \in \mathcal{F}} (c_{\text{T}}^{f_n}(\mathbf{a}) + c_{\text{V}}^{f_n}(\mathbf{a})) \\
&= \sum_{f_n \in \mathcal{F}} \left(\sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} c^j(\mathbf{a}) + \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} |c^j(\mathbf{a}) - \bar{c}^{f_n}(\mathbf{a})| \right) \\
&= \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} c^j(\mathbf{a}) + \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} \left| c^j(\mathbf{a}) - \frac{1}{|\mathcal{P}_{\text{int}}^{f_n}|} \sum_{k \in \mathcal{P}_{\text{int}}^{f^n}} c^k(\mathbf{a}) \right| \\
&= \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} \sum_{i \in \mathcal{S}_{\text{d}}^j} d^{(i,j)} a^{(i,j)} + \\
&\qquad \sum_{f_n \in \mathcal{F}} \sum_{j \in \mathcal{P}_{\text{int}}^{f_n}} \left| \sum_{i \in \mathcal{S}_{\text{d}}^j} d^{(i,j)} a^{(i,j)} - \frac{1}{|\mathcal{P}_{\text{int}}^{f_n}|} \sum_{k \in \mathcal{P}_{\text{int}}^{f^n}} \sum_{i \in \mathcal{S}_{\text{d}}^k} d^{(i,k)} a^{(i,k)} \right|
\end{aligned}
$$

Let d^j be the number of flows for which node j is an intermediate or destination node. Let $m_k^{(i,j)}$ be an indicator that link (i,j) points to node k (i.e., $m_k^{(i,j)} = 1$ if $j = k$, otherwise $m_k^{(i,j)} = 0$). Let $m_{f_n}^{(i,j)}$ be an indicator that link (i,j) points to an intermediate or destination node of flow f_n (i.e., $m_{f_n}^{(i,j)} = 1$ if $j \in \mathcal{P}_{\text{int}}^{f_n}$, otherwise $m_{f_n}^{(i,j)} = 0$). The objective function c_{Lin} can be formulated in terms of d^j, $m_k^{(i,j)}$, and $m_{f_n}^{(i,j)}$ as follows.

$$
c_{\text{Lin}}(\mathbf{a}) = \sum_{(i,j) \in \mathcal{L}} d^j d^{(i,j)} a^{(i,j)} + \sum_{f_n \in \mathcal{F}} \sum_{k \in f_n} \left| \sum_{(i,j) \in \mathcal{L}} m_k^{(i,j)} d^{(i,j)} a^{(i,j)} - \frac{1}{|\mathcal{P}_{\text{int}}^{f_n}|} \sum_{(i,j) \in \mathcal{L}} m_{f_n}^{(i,j)} d^{(i,j)} a^{(i,j)} \right|
$$

The objective function c_{Lin} can be formulated as a linear function of $\mathbf{a}$ using the vectors $\mathbf{d} \triangleq [d^j d^{(i,j)}]_{(i,j) \in \mathcal{L}^*}$, $\mathbf{m}_k \triangleq [m_k^{(i,j)}]_{(i,j) \in \mathcal{L}^*}$, $\mathbf{m}_{f_n} \triangleq [m_{f_n}^{(i,j)}]_{(i,j) \in \mathcal{L}^*}$, and $\mathbf{f} \triangleq \mathbf{d} + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}_{\text{int}}^{f_n}} |\mathbf{m}_k - \mathbf{m}_{f_n}|$ as follows.

$$
c_{\text{Lin}}(\mathbf{a}) = \mathbf{d} \cdot \mathbf{a} + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}_{\text{int}}^{f_n}} |\mathbf{m}_k \cdot \mathbf{a} - \mathbf{m}_{f_n} \cdot \mathbf{a}| =
$$

$$
\mathbf{d} \cdot \mathbf{a} + \sum_{f_n \in \mathcal{F}} \sum_{k \in \mathcal{P}_{\text{int}}^{f_n}} |\mathbf{m}_k - \mathbf{m}_{f_n}| \cdot \mathbf{a} = \mathbf{f} \cdot \mathbf{a} \quad (4\text{–}18)
$$

In the following, the feasible region of $\mathbf{a}$ is characterized based on the feasible region of $\mathbf{r}$, i.e., $\mathbf{r}_{\text{min}} \preceq \mathbf{r} \preceq \mathbf{r}_{\text{max}}$.

Let $\mathcal{S}_{\text{a}}^{i\setminus j}$ be the set of nodes in $\mathcal{S}_{\text{a}}^{i}$ that are closer to node i than to node j. Note that the value of $\mathcal{S}_{\text{a}}^{i\setminus j}$ varies with the network topology. Therefore, $\mathcal{S}_{\text{a}}^{i\setminus j}$ depends on $\mathbf{r}$. This is denoted by $\mathcal{S}_{\text{a}}^{i\setminus j}(\mathbf{r})$. The nodes in $\mathcal{S}_{\text{a}}^{i\setminus j}(\mathbf{r}_{\text{max}})$ are the nodes that for some feasible $\mathbf{r}$ (i.e., $\mathbf{r}_{\text{min}} \preceq \mathbf{r} \preceq \mathbf{r}_{\text{max}}$) are hidden nodes of link (i,j), i.e., the nodes that do not belong to $\mathcal{S}_{\text{a}}^{i\setminus j}(\mathbf{r}_{\text{max}})$ are not hidden nodes of link (i,j) for any feasible value of $\mathbf{r}$. We refer to the nodes in $\mathcal{S}_{\text{a}}^{i\setminus j}(\mathbf{r}_{\text{max}})$ as the potential-hidden nodes of link (i,j).

The feasible region of the Lin-NTC-AP can be formulated in terms of the following three constraints, which are defined in terms of $|\mathcal{S}_{\text{a}}^{i\setminus j}|_{\text{r}} \triangleq |\mathcal{S}_{\text{a}}^{i\setminus j}(\mathbf{r})|$.

Constraint 1.

$$a^{(i,j)} \geq \left|\mathcal{S}_a^{i\setminus j}\right|_{\text{r}_{min}} \ \forall \ (i,j) \in \mathcal{L}$$

Constraint 2.

$$a^{(i,j)} \leq \left|\mathcal{S}_a^{i\setminus j}\right|_{\text{r}_{max}} \ \forall \ (i,j) \in \mathcal{L}$$

Constraint 3.

$$a^{(i,j)} > a^{(h,k)} - \left|\mathcal{S}_a^{h\setminus k} \setminus \mathcal{S}_a^{i\setminus j}\right|_{\text{r}_{max}} \textit{ if}$$
$$\left|\mathcal{S}_a^{i\setminus h} \cap \mathcal{S}_a^{i\setminus j} \cap \mathcal{S}_a^{h\setminus k}\right|_{\text{r}_{max}} > \left|\mathcal{S}_a^{h\setminus i} \cap \mathcal{S}_a^{i\setminus j} \cap \mathcal{S}_a^{h\setminus k}\right|_{\text{r}_{max}}$$
$$\forall \ (i,j),(h,k) \in \mathcal{L}, (i,j) \neq (h,k)$$

Constraints 1 and 2 guarantee that $a^{(i,j)}$ is not lower and greater than its minimum and maximum possible values respectively. The value of $a^{(i,j)}$ represents the number of active nodes that cover i and do not cover j (i.e., active nodes that i is able to listen to and that are hidden from j). This number cannot be lower/greater than the number of active nodes that are closer to i than to j, and that cover i at minimum/maximum TP (i.e., $|\mathcal{S}_{\text{a}}^{i\setminus j}|_{\text{r}_{\text{min}}}$ and $|\mathcal{S}_{\text{a}}^{i\setminus j}|_{\text{r}_{\text{max}}}$ respectively).

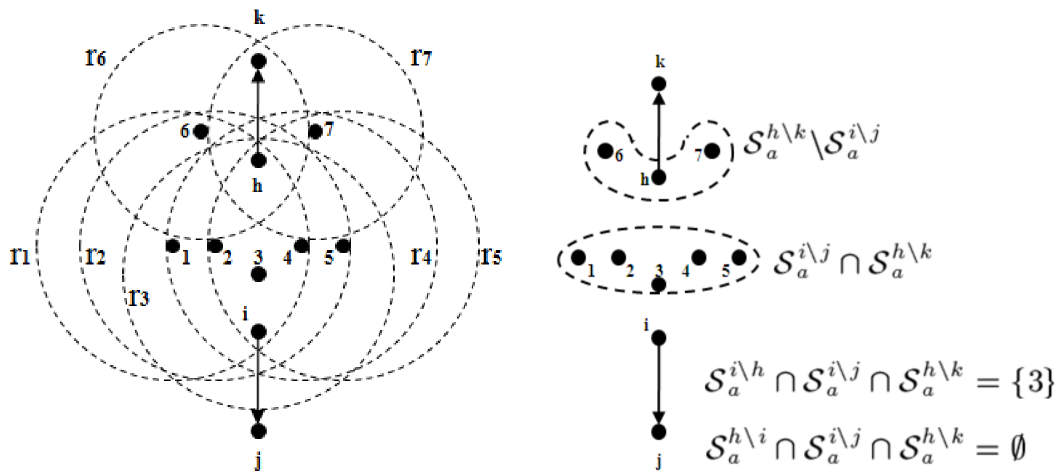
Constraint 3 guarantees that when two different links (e.g., (i, j) and (h, k)) share potential-hidden nodes (i.e., $\left|\mathcal{S}_{\text{a}}^{i \setminus j} \cap \mathcal{S}_{\text{a}}^{h \setminus k}\right|_{\text{r}_{\text{max}}} \neq \emptyset$), the link that has, at maximum TP, the highest number of shared potential-hidden nodes closest to its source node always has a higher number of hidden nodes[17] . When the TPs of the shared potential-hidden nodes are being increased, they cover first the link's source node that is closest to them. Therefore, the link with the highest number of shared potential-hidden nodes closer to its source node always has higher number of hidden nodes. This is shown in the example of Figure 4-1. In Figure 4-1A, two links (i.e., (i, j) and (h, k)) and their potential-hidden nodes are shown with their corresponding maximum TPs[18] . The potential-hidden nodes of link (i, j) are nodes 1, 2, ..., 5 (i.e., $\mathcal{S}_{\text{a}}^{i \setminus j}(\mathbf{r}_{\text{max}}) = \{1, 2, \ldots, 5\}$). The potential-hidden nodes of (h, k) are nodes 1, 2, ..., 7 (i.e., $\mathcal{S}_{\text{a}}^{h \setminus k}(\mathbf{r}_{\text{max}}) = \{1, 2, \ldots, 7\}$). The potential-hidden nodes shared by (i, j) and (h, k) are 1, 2, ..., 5 (i.e., $\left|\mathcal{S}_{\text{a}}^{i \setminus j} \cap \mathcal{S}_{\text{a}}^{h \setminus k}\right|_{\text{r}_{\text{max}}} = \{1, 2, \ldots, 5\}$). Whenever the TP of any of these nodes is being increased, the source node of link (h, k) (i.e., node h) is never covered after the source node of link (i, j) (i.e., node i) has been covered. Therefore, link (i, j) always has as many hidden nodes as link (h, k) at least[19] . The potential-hidden nodes of link (h, k) which are not potential-hidden nodes of link (i, j) (i.e., the nodes in $\mathcal{S}_{\text{a}}^{h \setminus k}(\mathbf{r}_{\text{max}}) \setminus \mathcal{S}_{\text{a}}^{i \setminus j}(\mathbf{r}_{\text{max}}) = \{6, 7\}$) are able to increase the number of hidden nodes of (h, k) (i.e., $\left|\mathcal{S}_{\text{a}}^{h} \setminus \mathcal{S}_{\text{a}}^{k}\right|$) without increasing the number of hidden nodes of (i, j) (i.e., $\left|\mathcal{S}_{\text{a}}^{i} \setminus \mathcal{S}_{\text{a}}^{j}\right|$). Therefore, in order to account for these nodes, the factor $\left|\mathcal{S}_{\text{a}}^{h \setminus k} \setminus \mathcal{S}_{\text{a}}^{i \setminus j}\right|_{\text{r}_{\text{max}}}$ is introduced in Constraint 3 by subtracting it from the number of hidden nodes of (h, k). For example, nodes 6 and 7 in Figure 4-1 are potential-hidden nodes of

[17] It is being assumed that $\left|\mathcal{S}_{\text{a}}^{h \setminus k} \setminus \mathcal{S}_{\text{a}}^{i \setminus j}\right|_{\text{r}_{\text{max}}} = 0$. The general case of $\left|\mathcal{S}_{\text{a}}^{h \setminus k} \setminus \mathcal{S}_{\text{a}}^{i \setminus j}\right|_{\text{r}_{\text{max}}} \geq 0$ is considered next.

[18] It is assumed that the nodes 1, 2, ..., 7 in Figure 4-1 are active.

[19] Nodes 6 and 7 have not been considered yet, i.e., it is being assumed that $\left|\mathcal{S}_{\text{a}}^{h \setminus k} \setminus \mathcal{S}_{\text{a}}^{i \setminus j}\right|_{\text{r}_{\text{max}}} = 0$.

(h, k) but not of (i, j) (i.e., $\mathcal{S}_{\mathrm{a}}^{h\backslash k}(\mathbf{r}_{\max})\backslash\mathcal{S}_{\mathrm{a}}^{i\backslash j}(\mathbf{r}_{\max}) = \{6, 7\}$), so they are able to increase the number of hidden nodes of (h, k) without increasing the number of hidden nodes of (i, j).



A Maximum TPs (i.e., $\mathbf{r}_{\max}$)

B Potential-hidden nodes of links (i, j) and (h, k) at $\mathbf{r} = \mathbf{r}_{\max}$

Figure 4-1. An example of potential hidden nodes

According to Eq. 4–18 and Constraints 1, 2, and 3, the Lin-NTC-AP can be formulated as a linear integer program as follows, where $\mathcal{A}$ is the feasible region determined by Constraints 1, 2, and 3.

$$\begin{aligned} &\textit{minimize} \;\; \mathbf{f}\cdot\mathbf{a} \\ &\textit{subject to} \;\; \mathbf{a}\in\mathcal{A} \end{aligned} \tag{4–19}$$

This concludes the proof of Theorem 4.3.

Theorem 4.4. *The optimal solution set of the Lin-NTC-AP and the Nash equilibria set of the Lin-NTC-AP game are equivalent.*

The proof of Theorem 4.4 is as follows. Let $\mathbf{a}^{\mathrm{opt}}$ be an optimal solution. Assume that $\mathbf{a}^{\mathrm{opt}}$ is not a Nash equilibrium. Therefore, there exists some player f_n and a strategy $\mathbf{a}_n$ such that $\mu_n^*(\mathbf{a}_n, \mathbf{a}_{-n}) < \mu_n^*(\mathbf{a}_n^{\mathrm{opt}}, \mathbf{a}_{-n}^{\mathrm{opt}})$. Given that the Lin-NTC-AP game is an OPG, and $c_{\text{Lin}}(\mathbf{a})$ is an OPF, the previous result implies that $c_{\text{Lin}}(\mathbf{a}_n, \mathbf{a}_{-n}) < c_{\text{Lin}}(\mathbf{a}_n^{\text{opt}}, \mathbf{a}_{-n}^{\text{opt}})$. This contradicts the assumption that $\mathbf{a}^{\text{opt}}$ is optimal.

Let $\mathbf{a}^{\text{eq}}$ be a Nash equilibrium. Assume that $\mathbf{a}^{\text{eq}}$ is not optimal. Therefore, there exists some player f_n and a strategy $\mathbf{a}_n$ such that $c_{\text{Lin}}(\mathbf{a}_n, \mathbf{a}_{-n}) < c_{\text{Lin}}(\mathbf{a}_n^{\text{eq}}, \mathbf{a}_{-n}^{\text{eq}})$. Given that the Lin-NTC-AP game is an OPG, and $c_{\text{Lin}}(\mathbf{a})$ is an OPF, the previous result implies that $\mu_n^*(\mathbf{a}_n, \mathbf{a}_{-n}) < \mu_n^*(\mathbf{a}_n^{\text{eq}}, \mathbf{a}_{-n}^{\text{eq}})$. This contradicts the assumption that $\mathbf{a}^{\text{eq}}$ is a Nash equilibrium. This result concludes the proof of Theorem 4.4.

From Theorems 4.3 and 4.4, the following result is immediate.

Corollary 2. *The WiMAX-Mesh-NTC always converges to an optimal solution of the Lin-NTC-AP.*

Remark. According to Eq. 4–19, the Lin-NTC-AP consists of finding a vector $\mathbf{a}$ that belongs to $\mathcal{A}$ whose scalar projection onto vector $\mathbf{f}$ is minimum. Minimizing this scalar projection is equivalent to minimizing c_{Lin}, i.e., minimizing the total contention and contention variation (Eq. 4–16). Therefore, with the WiMAX-Mesh-NTC algorithm, the flows adjust iteratively vector $\mathbf{a}$ until they minimize the scalar projection. According to Corollary 2, this minimization is always achieved.

Remark. Vector $\mathbf{f}$ is determined by the paths of the flows only (Eq. 4–18), and vector $\mathbf{a}$ is determined by the hidden nodes of the links of the network only. Therefore, if the problem of joint routing and topology control was considered, both $\mathbf{f}$ and $\mathbf{a}$ would be controlled in order to minimize their scalar projections onto each other. Both $\mathbf{f}$ and $\mathbf{a}$ are vectors of nonnegative integers by definition (i.e., $\mathbf{f}, \mathbf{a} \in \mathbb{Z}_+$), and the feasible region $\mathcal{A}$ is a convex set if $\mathbf{a}$ is relaxed by allowing it to have real values.

4.4.2 Performance Bound

The optimal solutions of the Lin-NTC-AP are not guaranteed to be optimal for the NTC-AP. However, the difference between the values of the NTC-AP objective function evaluated at its optimal solution and evaluated at the optimal Lin-NTC-AP solution is upper-bounded (Theorem 4.5). Therefore, the optimal Lin-NTC-AP solutions are able to reach the maximum NTC within a bounded error. The error's bound can be characterized based on the following observation.

An active node that is a hidden node of one or more links of a flow contributes to the contention experienced by the flow. When the TP of an active node partially covers a flow, the node becomes an active hidden node of the flow's links that originate within the TP and terminate outside the TP. This is shown in Figure 4-2 in which link (i, j) originates within h's TP (i.e., i is covered by h's TP) and terminates outside h's TP (i.e., j is not covered by h's TP). Given that h is a hidden node of one f_n's links (i.e., (i, j)), h contributes to the contention experienced by the link, and as consequence, it contributes to the contention experienced by f_n. According to Eq. 4–14 and Eq. 4–15, h contributes an additive factor of $d^{(i,j)}$ and of $\frac{d^{(i,j)}}{|\mathcal{P}_{\text{int}}^{f_n}|}$ to f_n's total contention and mean contention respectively. The contribution of h to the contention variation in f_1, denoted by $\Delta_{\text{V}}^{f_n}$, can be positive or negative depending on the network topology. For example, in Figure 4-2, h increases f_n's contention variation because without h, all the nodes in $\mathcal{P}_{\text{int}}^{f_n}$ experience the same contention, while with h, node j's contention is increased by $d^{(i,j)}$ while the other nodes' contentions remain the same. Therefore, without h, f_n's contention variation is zero (i.e., $\Delta_{\text{V}}^{f_n} = 0$), while with h, f_n's contention is greater than zero (i.e., $\Delta_{\text{V}}^{f_n} > 0$).

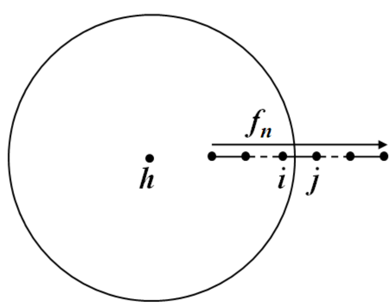


Figure 4-2. Hidden-node example: node h is a hidden node of link (i, j), i.e., j cannot listen to h while i can listen to it

Let $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ and $\mathbf{a}_{\text{NTC}}^{\text{opt}}$ be optimal solutions to the Lin-NTC-AP and NTC-AP respectively. Let $c_{\text{NTC}}(\mathbf{a})$ be the objective function of the NTC-AP as formulated by Eq. 4–15 (i.e.,

$c_{\mathsf{NTC}}(\mathbf{a}) \triangleq \sum_{f_n \in \mathcal{F}} \max\{c^j(\mathbf{r}) : j \in \mathcal{P}_{\text{int}}^{f_n}\})$. Let $d_{\max}$ be the maximum link degree (i.e., $d_{\max} \triangleq \max\{d^{(i,j)} : (i,j) \in \mathcal{L}\})$.

Theorem 4.5. *The difference between the values of the NTC-AP objective function evaluated at* $\mathbf{a}_{Lin}^{opt}$ *and* $\mathbf{a}_{NTC}^{opt}$ *is upper-bounded as follows.*

$$c_{NTC}(\mathbf{a}_{Lin}^{opt}) - c_{NTC}(\mathbf{a}_{NTC}^{opt}) \leq d_{max}\frac{|\mathcal{F}| - 2}{2}$$

See APPENDIX G for the proof of Theorem 4.5.

4.5 Simulation Results

The performance evaluation of the WiMAX-Mesh-NTC algorithm was performed by means of simulation using the simulator proposed in Chapter 5. This evaluation is given in terms of the throughput error denoted by δ_{T} and defined as follows. Let $\lambda_{\mathsf{T}}(\mathbf{r})$ be the highest total throughput supported by the flows in $\mathcal{F}$ under power setting $\mathbf{r}$ (i.e., $\lambda_{\mathsf{T}}(\mathbf{r}) \triangleq \sum_{f_n \in \mathcal{F}} \lambda_{\max}^{f_n}(\mathbf{r})$). Let $\mathbf{r}_{\text{opt}}$ be the feasible set of transmission ranges that maximizes λ_{T}. The throughput error of the topology induced by $\mathbf{r}$, where $\mathbf{r}$ can be any feasible set of transmission ranges, is given by Eq. 4–20.

$$\delta_{\mathsf{T}}(\mathbf{r}) = \frac{\lambda_{\mathsf{T}}(\mathbf{r}_{\text{opt}}) - \lambda_{\mathsf{T}}(\mathbf{r})}{\lambda_{\mathsf{T}}(\mathbf{r}_{\text{opt}})} \tag{4–20}$$

The WiMAX-Mesh-NTC algorithm was compared with the HSRA, MinPower, and MaxPower algorithms. The HSRA algorithm (Chapter 3) is a heuristic and centralized algorithm that aims to find a set of transmission ranges that maximizes λ_{T}. The MinPower algorithm aims to maximize the spatial reuse by setting the nodes' transmission ranges at the minimum values that do not disconnect any of the flows in $\mathcal{F}$. The MaxPower algorithm sets all the nodes' transmission ranges at their maximum values.

The simulation was configured as follows[20] . The link scheduling policy was GM-RBDS (Chapter 2). The flow paths were obtained using the min-hop routing algorithm. The nodes were uniformly distributed in a square such that the node density was always kept at 15 nodes per area unit. There were a total of 20 nodes. The maximum transmission range for all the nodes was set at 0.3 (i.e., $\mathbf{r}_{\mathrm{max}} = [0.3]_{i \in \mathcal{N}}$). The connectivity of the flows with the nodes' transmission ranges set at $\mathbf{r}_{\mathrm{max}}$ was verified before executing the min-hop routing, WiMAX-Mesh-NTC, HSRA, and MinPower algorithms. The source and destination of every flow were uniformly distributed across all the nodes in the network. The min-hop algorithm calculated the flow paths when the transmission ranges were set at $\mathbf{r}_{\mathrm{max}}$.

Each of the topology-control algorithms (i.e., WiMAX-Mesh-NTC, HSRA, MinPower, and MaxPower) calculated the set of transmission ranges for 300 different networks configured as explained previously. These networks were divided into three groups of 100 networks each depending on the number of flows in the networks. In the first, second, and third groups the networks had 10, 15, and 20 flows respectively. The algorithms calculated a set of transmission ranges for each of the 300 networks, and the throughput error δ_{T} was calculated for each of the set of transmission ranges[21] . Figure 4-3 shows the ratio of topologies calculated by the algorithms that had a δ_{T} less than or equal to 4%. Therefore, Figure 4-3 shows the ability of the algorithms to reach the maximum possible total throughput (i.e., $\lambda_{\mathrm{T}}(\mathbf{r}_{\mathrm{opt}})$).

According to Figure 4-3, the WiMAX-Mesh-NTC algorithm always outperforms the MinPower and MaxPower algorithms. When there are 10 flows (Figure 4-3A), 44.3% of

[20] This is the same configuration used in Chapter 3.

[21] To find $\mathbf{r}_{\mathrm{opt}}$, which is necessary for calculating δ_{T} (Eq. 4–20), the Branch And Reduce Optimization (BARON) Solver [4] was used. BARON is a system for solving non-convex optimization problems to global optimality.

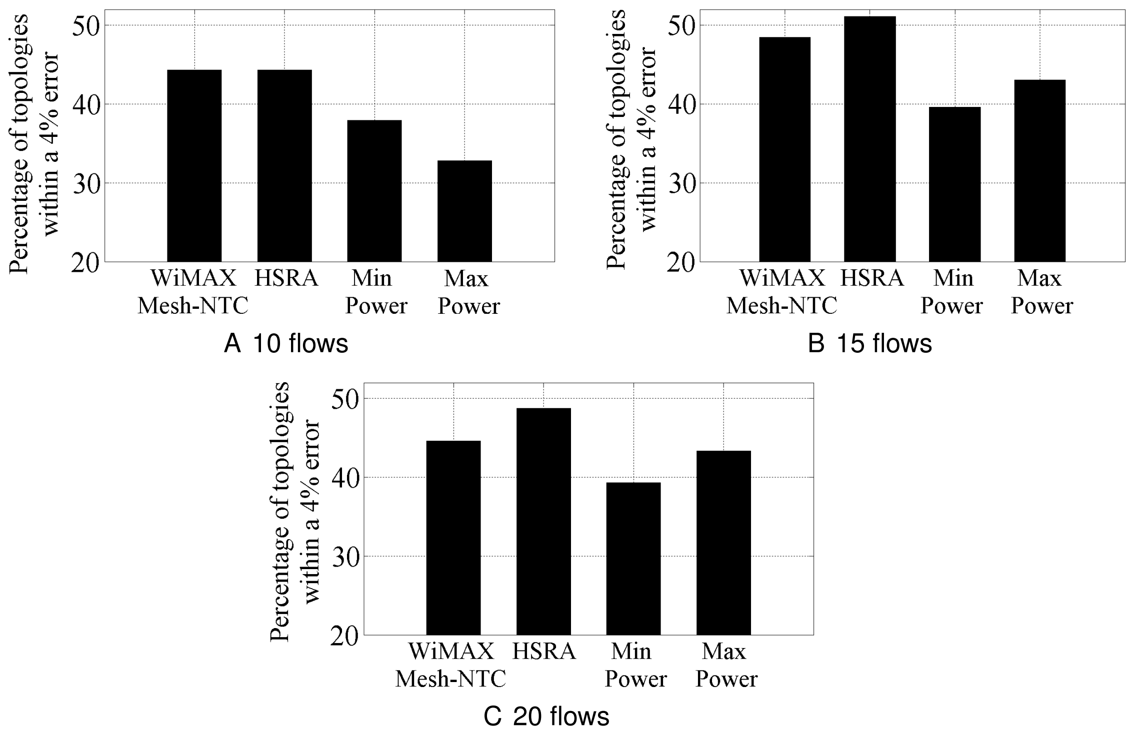


Figure 4-3. Percentage of topologies whose δ_T is within 4%

the topologies calculated by WiMAX-Mesh-NTC have a δ_T of at most 4% while 37.9% and 32.8% of the topologies calculated by MinPower and MaxPower have such δ_T. When there are 15 flows (Figure 4-3B), these percentages are 48.5, 39.6, and 43.1 respectively, and when there are 20 flows (Figure 4-3C), they are 44.6, 39.3, and 43.3. Therefore, in all the cases the WiMAX-Mesh-NTC algorithm was able to find more effectively a topology that maximizes the total throughput λ_T.

When the WiMAX-Mesh-NTC and HSRA algorithms are compared, the flow density needs to be considered. Let the flow density be the ratio among the number of flows and the number of nodes. Given that in Figure 4-3, the number of nodes does not change while the number flows increases from 10 to 20, the flow density increases. Figure 4-3A shows that the algorithms have the same performance when the flow density is low. Figure 4-3B/Figure 4-3C shows that WiMAX-Mesh-NTC has worse performance than HSRA when the flow density is medium/high. The intuition behind this behavior is that the probability that any of the cases shown in Figure G-1 takes place increases with the flow density, and such cases affect the performance of WiMAX-Mesh-NTC (Theorem 4.5). This is not the case for HSRA because HSRA is a centralized algorithm, so it has a global view of the network. However, the advantage of WiMAX-Mesh-NTC is that it is distributed while HSRA is not. WiMAX-Mesh-NTC requires knowledge about the flows in $\mathcal{F}^n$ while HSRA requires knowledge about all the flows in $\mathcal{F}$. Therefore, WiMAX-Mesh-NTC is more amenable for implementation.

4.6 Summary

A new framework for the development of distributed algorithms that maximize the total end-to-end throughput in WMNs was proposed. It is based on the link-scheduling policy's stability region, and it consists of a potential game in which a given set of flows act as players that collaborate to maximize the packet rates they can support while guaranteeing stability. Based on the proposed framework, the WiMAX-Mesh-NTC algorithm was developed. WiMAX-Mesh-NTC aims to reduce the maximum contention experienced by every flow. The convergence of WiMAX-Mesh-NTC was characterized by means of the Nash equilibrium, and a performance bound was calculated by considering all the possible worst-case scenarios. Finally, the WiMAX-Mesh-NTC performance was compared by means of simulation with the performance of other topology-control algorithms (i.e., HSRA, MinPower, MaxPower). It was shown that WiMAX-Mesh-NTC always outperforms MinPower and MaxPower, and that it outperforms HSRA when the flow density is medium.

CHAPTER 5
WORLDWIDE-INTEROPERABILITY-FOR-MICROWAVE-ACCESS RBDS SIMULATOR (WIMAX-RBDS-SIM): AN OPTIMIZED-NETWORK-ENGINEERING-TOOLS (OPNET) SIMULATION FRAMEWORK FOR WIRELESS MESH NETWORKS

The Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 standard [1] defines the physical and medium-access-control (MAC) layers for wireless mesh networks (WMN). Although the standard provides the architecture necessary for the implementation of any scheduling policy, no policy is specified by the standard. The MAC layer is based on time-division multiple access (TDMA), where time is divided into frames that are simultaneously used by non-interfering links. Each frame is divided into a control subframe and a data subframe. Scheduling messages are exchanged in control subframes containing scheduling information such as requests for use of data subframes and grants of data subframes. An election algorithm determines the subset of nodes that access the control subframe such that the transmissions of scheduling messages do not interfere with each other. The scheduling messages, control subframes, and the election algorithm enable the implementation of any given scheduling policy.

The standard also defines the procedures that nodes need to follow for joining an IEEE 802.16 WMN and for establishing physical links with other nodes. These procedures need to be finished before a node begins scheduling data packets.

In this paper, an optimized-network-engineering-tools (OPNET) [3] simulation framework is proposed. It is called worldwide-interoperability-for-microwave-access reservation-based-distributed-scheduling simulator (WiMAX-RBDS-Sim). Its purpose is to enable the evaluation of distributed scheduling policies and the link-establishment process in IEEE 802.16 WMNs. It implements the scheduling messages, election algorithm, and link-establishment and provides an interface for the integration of different distributed scheduling algorithms. To the best of our knowledge, WiMAX-RBDS-Sim is the first OPNET-based simulator for IEEE 802.16

WMNs with distributed scheduling. In Section 5.1, the related work and contributions are discussed. An overview of IEEE 802.16 WMNs is presented in Section 5.2. In Section 5.3, the architecture of WiMAX-RBDS-Sim is explained in detail using the proposed algorithm Sliced-GM-RBDS as an example. In Section 5.4, the OPNET implementation of the WiMAX-RBDS-Sim architecture is described. In Section 5.5, the simulation results obtained with WiMAX-RBDS-Sim for the proposed algorithm are provided and discussed. The performance of WiMAX-RBDS-Sim is evaluated in Section 5.6. Finally, a summary of the chapter is presented in Section 5.7.

5.1 Related Work

The research community has recently investigated the challenges of centralized and distributed scheduling in 802.16 WMNs. In this paper we focus on the distributed scheduling problem[1] .

Distributed-scheduling research can be classified into two groups: election-algorithm-based policies and RBDS policies. In [13, 23], the performance of the election algorithm is evaluated theoretically and by means of simulation. In [6, 40, 78, 87], the performance is improved in terms of number of collisions and control-subframe utilization by dynamically adjusting the parameters of the algorithm. In [31, 85, 86], a quality-of-service (QoS) differentiation scheme is proposed based on the adjustment of the election-algorithm parameters so that a node transmits scheduling messages more often when its data transmissions have higher priority. In [47], a congestion control mechanism is proposed in which one of the election-algorithm parameters is adjusted by each node according to the congestion measured locally. The effect of this adjustment is that a node waits longer for sending scheduling messages when the local congestion is high.

[1] For more information on centralized scheduling, please refer to [16, 52].

The second group for distributed-scheduling research considers RBDS policies. These policies are based on the reservation of future data subframes so that no two interfering links are assigned the same data subframes. In [21], the nodes prioritize the traffic flows going through them according to the weights of each of the flows, and the number of data subframes reserved for each flow is proportional to its weight. The data subframes eligible for reservation are finite in the sense that only the frames that are a certain number of frames away from the current one can be included in the reservations. This set of data subframes is known as the schedule horizon. In [79], data-subframe utilization is improved by allowing every reservation to contain non-contiguous data subframes that are all assigned to the same link. In this way, the number of wasted data subframes (i.e., data subframes that are never assigned to any link) is reduced. In [43], the reservations for each link are calculated based on the statistical characteristics of the data traffic generated at the source node in order to reduce the scheduling overhead. In [24], an end-to-end reservation scheme is proposed for constant-bit-rate flows such as voice-over-IP. Other simpler reservation-based distributed scheduling schemes are proposed in [18, 42, 48], and a broadcasting algorithm that aims to minimize the number of reservations per broadcast is proposed in [88].

5.1.1 Distributed-Scheduling Simulators

The IEEE 802.16 standard defines two operating modes for metropolitan access networks. These are the point-to-multipoint (PMP) and mesh modes. In the literature, research has been mostly focused on simulators for the PMP mode [8, 55, 64]. To the best of our knowledge, there are two simulators for IEEE 802.16 WMNs with distributed scheduling that have been discussed in the literature. These are WiMsh [22], which is a patch for the Network Simulator 2 (ns-2) [2], and the National Chiao Tung University network simulator (NCTUns) [32], which is a network simulator and emulator that includes a simulation tool for IEEE 802.16 WMNs with distributed scheduling.

WiMAX-RBDS-Sim is a packet-level simulation model for the OPNET discrete event simulator [3]. It is an implementation of the MAC Common Part Sublayer given in the standard [1]. The architecture of WiMAX-RBDS-Sim was conceived for the simulation of the RBDS policies defined in Chapter 2 which are based on the input/output queue concepts.

Although, WiMAX-RBDS-Sim and WiMsh have similar functionalities, there are differences between these two simulators in terms of architecture and network topology. These differences are as follows.

The architecture of WiMAX-RBDS-Sim is based on a theoretical framework that focuses on a stability analysis for wireless multi-hop networks under any RBDS policy. WiMAX-RBDS-Sim provides the means for obtaining simulation results for such policies in order to compare them with the theoretical results obtained for the same policies under the theoretical framework.

WiMAX-RBDS-Sim implements the link establishment algorithm, which is not implemented in WiMsh, and provides the framework for the implementation and evaluation of improvements to the link establishment process.

In WiMAX-RBDS-Sim, the links established between the nodes depend on the location of the nodes, the physical channel model selected for the simulation, and the nodes' antennas. Therefore, the topology of the network is calculated based on those parameters. In WiMsh, the links are given as a simulation parameter by specifying the network topology from a set of predefined topologies.

5.1.2 Contributions

In this paper, the design and implementation of WiMAX-RBDS-Sim is presented. WiMAX-RBDS-Sim has the following characteristics:

- It provides a framework for the implementation and evaluation of algorithms that adjust the parameters of the election algorithm for each node dynamically. WiMAX-RBDS-Sim allows the evaluation of these algorithms in terms of scheduling delay, control-subframe utilization, and number collisions in control subframes.

- It provides a framework in which scheduling policies can be implemented and evaluated in terms of capacity (i.e., the set of maximum data input rates that the policy can handle while guaranteeing the stability of all the queues in the network), throughput, and delay.
- It provides a framework for the implementation and evaluation of link-establishment algorithms. This evaluation is performed in terms of link-establishment delay (i.e., the time required by two nodes to establish links between them).

5.2 IEEE 802.16 WMN Overview

In an 802.16 WMN there are two types of nodes: base stations (BS) and subscriber stations (SS). The 802.16 WMN is connected to external networks such as the Internet through BSs. Unidirectional links can be established between any of these nodes, and the information is transmitted on a hop-by-hop basis. Each link is uniquely identified in the network with both the node identification (ID) and the link ID. The link ID identifies the link among the set of outgoing links of a node, and the node ID identifies the node from which the link originates. The system access follows a frame-based approach which is shown in Figure 5-1. Each channel is divided in time into series of frames. In Figure 5-1, each of these series consists of eight frames (i.e., from frame n to frame $n+7$). A frame is further divided into two subframes: a control subframe and a data subframe. The control subframes are used for establishing links and scheduling data. The data subframes are used for data transmission. Control subframes are divided into slots, which are used for transmitting scheduling packets and link-establishment packets. The control subframe of the first frame of a series (e.g., frame n in Figure 5-1) is used for transmitting link-establishment packets[2] . The control subframes of the following frames of the series (e.g., frames $n+1$, $n+2$, ..., $n+7$ in Figure 5-1) are used for transmitting scheduling packets only. Data subframes are divided into slots which are used for transmitting data packets.

[2] These packets carry network-configuration information too, but this is out of the scope of this paper.

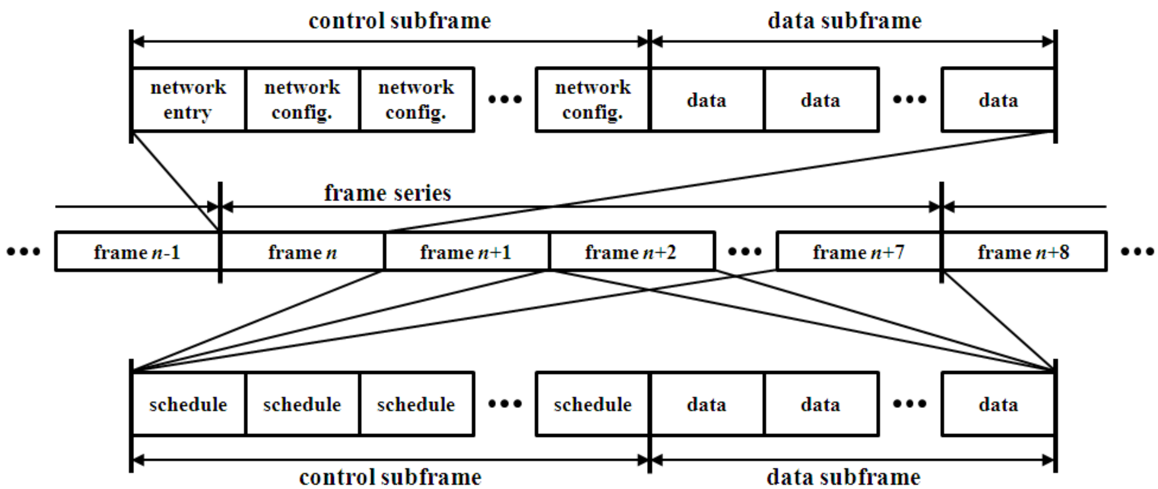


Figure 5-1. Frame structure of the Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 mesh mode

Nodes access the control slots (i.e., the slots in control subframes) using the IEEE 802.16 election algorithm [13]. The election algorithm guarantees that whenever a node transmits, all the other nodes in its 2-hop neighborhood stay quiet, where the 2-hop neighborhood of a node consists of all the nodes that are at most 2 hops away from it. In this way, collisions in control slots are avoided and link-establishment and scheduling packets are successfully received.

Nodes access the data slots (i.e., slots in data subframes) using a scheduling policy that is implemented by exchanging scheduling packets in the control subframe. The goal of the scheduling policy is to avoid unused data slots and collisions of data packets while optimizing one or more metrics of the network such as its capacity.

5.2.1 Data-Slot Scheduling

The scheduling policy for accessing data slots is not specified in the IEEE 802.16 standard. The standard only defines the mesh-distributed-scheduling (MSH-DSCH) message, which contains scheduling information and is carried by scheduling packets, so that different distributed scheduling policies can be implemented. The information contained in an MSH-DSCH message is organized in information elements (IE) as follows.

- Request IE: A node can make several requests simultaneously on a one-request-per-link basis. The information included in a request is the link ID, number of requested data slots per data subframe, and number of requested data subframes. The number of data subframes may be infinite so that streams of information can be transmitted in the link.
- Availability IE: A node notifies its 1-hop neighbors[3] of the data slots it has available for reservation. This IE specifies a set of available data slots with a start frame number, number of frames, start data slot number, number of data slots, direction (i.e., the minislots may be available for transmission, reception, or both of them), and the channels the available data slots belong to.
- Grant IE: This IE includes the same parameters specified for the availability IE. However, these are used for specifying a set of data slots that have been assigned to a link.

The scheduling procedure follows a three-way handshake. First, a node sends an MSH-DSCH message to one of its 1-hop neighbors requesting a set of data slots. In the message, the node also includes the set of data slots that it has available for reservation. The 1-hop neighbor grants the request by replying with another MSH-DSCH message that specifies a set of data slots that satisfies the availability of data slots at both nodes. Finally, the first node confirms the reservation of such set of data slots by echoing the grant in another MSH-DSCH message. By following this three-way handshake, the 1-hop neighbors of the two nodes become aware of the data slot reservation so that the data slots in the grant become unavailable for them.

5.2.2 Link Establishment

In order to exchange data packets in both directions, two 1-hop neighbors need to establish two links (i.e., one link in each direction). This is achieved by means of a three-way handshake performed with mesh network configuration (MSH-NCFG) messages which are transmitted in link-establishment packets. The handshake is initiated by the node with lowest ID. First, the lowest-ID node sends a challenge to its

[3] Two nodes are 1-hop neighbors if and only if they both can receive packets from each other directly.

new 1-hop neighbor. Second, the 1-hop neighbor replies with a challenge-response and the ID for its outgoing link (i.e., the incoming link of the lowest-ID node). Finally, the lowest-ID node replies with an accept and the ID for its outgoing link (i.e., the incoming link of the 1-hop neighbor). The link-establishment information that the nodes exchange during the handshake (i.e., challenge, challenge-response, accept, and link IDs) is written on link-establishment IEs, and these are attached to MSH-NCFG messages.

5.3 WiMAX-RBDS-Sim Architecture

The WiMAX-RBDS-Sim architecture is shown in Figure 5-2. The architecture consists of the physical, MAC, and logical-link layers. The nodes all share the same architecture. There are 10 radio channels in the physical layer. The MAC layer implements the option for distributed scheduling in wireless mesh networks of the IEEE 802.16 standard [1][4] . It consists of four main functional modules. These are the main-processing module, the packet-classifier module, the output-queue and channel-classifier module, and the time-synchronization module. The logical-link layer is the node's source and sink of data packets. It does not implement any logical-link protocol.

Each of the radio channels can be set at the configuration modes specified in the standard [1]. These modes determine the frame length, channel bandwidths, and forward-error-correction and modulation schemes. They are specified as simulation parameters.

In order to make comparisons between theoretical and simulation results, the WiMAX-RBDS-Sim architecture follows the 1-hop traffic model (i.e., the destination node of every packet is always 1 hope away from the source node). The logical-link layer

[4] Specifically, the MAC layer implements the MAC Common Part Sublayer for IEEE 802.16 wireless mesh networks with distributed scheduling.

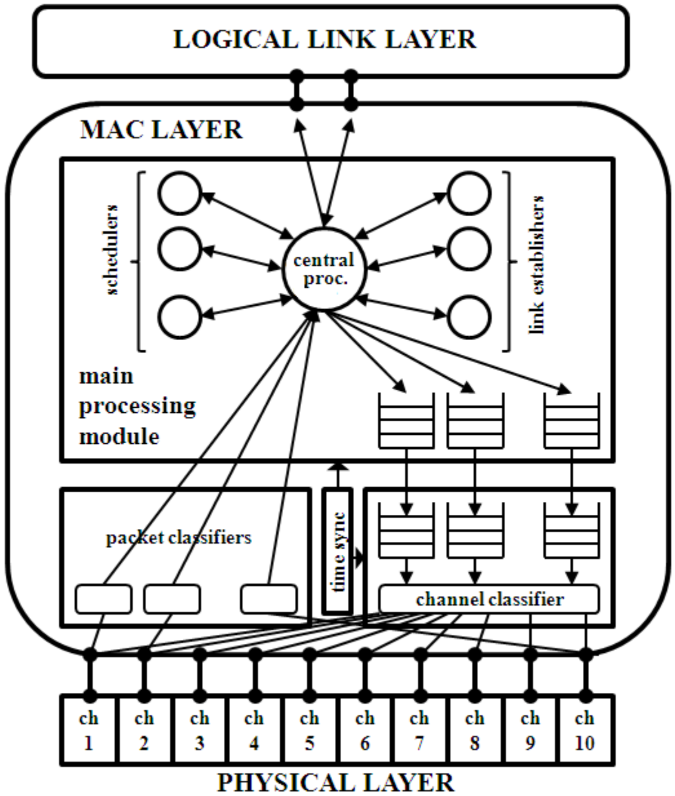


Figure 5-2. Architecture of the worldwide-interoperability-for-microwave-access reservation-based-distributed-scheduling simulator (WiMAX-RBDS-Sim)

can be set at some data-packet generation rate, and the data-packet source generates data-packets at such rate for each of the node's 1-hop neighbors.

In the MAC layer, the packet-classifier module classifies all the received packets into scheduling, link-establishment, and data packets and forwards them to the main-processing module. In this module, there is one classifier per radio channel. The output-queue and channel-classifier module queue packets that have already been scheduled for transmission and forward them to the right radio channel at the transmission time indicated by their respective schedules. In this module, there is one output-queue per radio channel.

The main-processing module performs the following tasks:

- establishes incoming and outgoing links with each of the node's 1-hop neighbors (i.e., the links in the WiMAX-RBDS-Sim architecture are unidirectional)

- queues unscheduled data packets and forwards them to the output-queue and channel-classifier module once they are scheduled
- schedules for transmission data-packets generated at the logical-link layer
- forwards to the logical-link layer the received data-packets that are destined to the node

The main-processing module consists of several processors that are dedicated to specific tasks and a set of queues where unscheduled data packets are stored temporarily. There are three types of processors: main processor, scheduler, and link establisher. There is only one main processor in the module. The main processor controls the flow of packets within the module and creates and destroys scheduler and link-establisher processes. The schedulers are created and destroyed at the onset and at the end of the simulation respectively. There is one scheduler per channel. A scheduler generates and processes for its assigned channel the request, availability, and grant IEs transmitted and received by the node respectively. The link-establisher processes are created and destroyed dynamically by the main processor during the simulation. A link establisher is created when a 1-hop neighbor is detected for the first time. The link-establisher generates and destroys the link-establishment IEs transmitted and received by the node for establishing the incoming and outgoing links with the detected 1-hop neighbor. The main processor destroys the link-scheduler once these links have been established.

The flow of packets within the main-processing module consists of two directions. These correspond to the transmission and reception of packets. In the direction of transmission, data packets are generated by the logical-link layer and delivered to the main processor. The main processor stores them in the module's queues. These are the input queues. There is one input queue per outgoing link. Depending on the lengths of the input-queues, the main processor invokes the schedulers which generate the scheduling IEs (i.e., request IE, availability IE, and grant IE). With these IEs and based on the election algorithm, the main processor generates scheduling packets and calculates their schedules. The scheduling packets are forwarded to the output queues along with their schedules. When the schedulers are finished scheduling any packets in the input queues, the main processor removes those packets from the input queues and forwards them along with their schedules to their corresponding output queues. Also, the main processor generates link-establishment packets with the link-establishment IEs generated by the link-establishers and calculates their schedules based on the election algorithm. These packets are also forwarded along with their schedules to their corresponding output queues. In the direction of reception, the main processor first checks whether the received packets are directed to the node it belongs to. If that is the case, the main processor forwards data packets to the logical-link layer, reads IEs from scheduling and link-establishment packets, and destroys these types of packets (i.e., scheduling and link-establishment packets). Scheduling IEs are passed to their corresponding schedulers, and link-establishment IEs are passed to their corresponding link-establishers.

5.3.1 The Link-Establishers

The link-establisher process model is shown in Figure 5-3. The model is a state transition diagram in which actions are taken at every state, and state transitions take place when the main processor invokes the link-establisher. The link-establisher communicates with the main processor by writing on and reading from a buffer that can be accessed by both processes only.

The link-establisher is able to perform two different roles present in the three-way handshake. These are the challenger and the replier. The challenger corresponds to the link-establisher at the 1-hop neighbor with lower ID which initiates the handshake by sending a challenge. The replier corresponds to the link-establisher at the 1-hop neighbor with higher ID which replies to the challenge by sending a challenge-response. In the initial state, named init (Figure 5-3), the process decides its role. The challenger jumps from init to prep-challenge, and the replier jumps from init to prep-response.

Upon arrival to prep-challenge, the link-establisher prepares the link-establishment IE with the challenge that initiates the handshake and waits at state wait until the main processor is ready to process the link-establishment IE. When the main processor is ready, the link-establisher jumps to tx-challenge, writes the link-establishment IE on the buffer, and sets a timer for the retransmission of the link-establishment IE in case it receives no response from the 1-hop neighbor. Then, the link-establisher jumps to wait-response where it waits for a response. If the response is not the expected challenge-response, it jumps to abort where it aborts the handshake and destroys itself. In this case, the main processor will later create another link-establisher for establishing the links with the 1-hop neighbor. If the response is the expected challenge-response, the link-establisher, at state prep-accept, reads the incoming-link ID assigned by the 1-hop neighbor, and creates another link-establishment IE with an accept and outgoing-link ID. When the Link-Establishment is written on the buffer at state tx-accept, the link-establisher jumps to finish, notifies the logical-link layer that a new outgoing link has been established and destroys itself.

Upon arrival to prep-response (i.e., when the link-establisher takes the replier role), the link-establisher creates a link-establishment IE with a challenge-response and outgoing-link ID and waits at state wait until the main processor is ready to process the link-establishment IE. When the main processor is ready, the link-establisher jumps to tx-response, writes the link-establishment IE on the buffer, and sets a timer for a retransmission in case no response is received from the 1-hop neighbor. When the response is received, the link-establisher checks whether it is another challenge, the expected accept, or an unexpected response (i.e., reject, challenge-response). If it is another challenge, the link-establisher assumes that the previously transmitted challenge-response was not received and the link-establishment IE is retransmitted. If it is the expected accept, the link-establisher jumps to get-LinkID where it reads the incoming-link ID and jumps to finish. At state finish, it notifies the logical-link layer that a new outgoing link has been established and destroys itself. If it an unexpected response, the link-establisher jumps to abort.

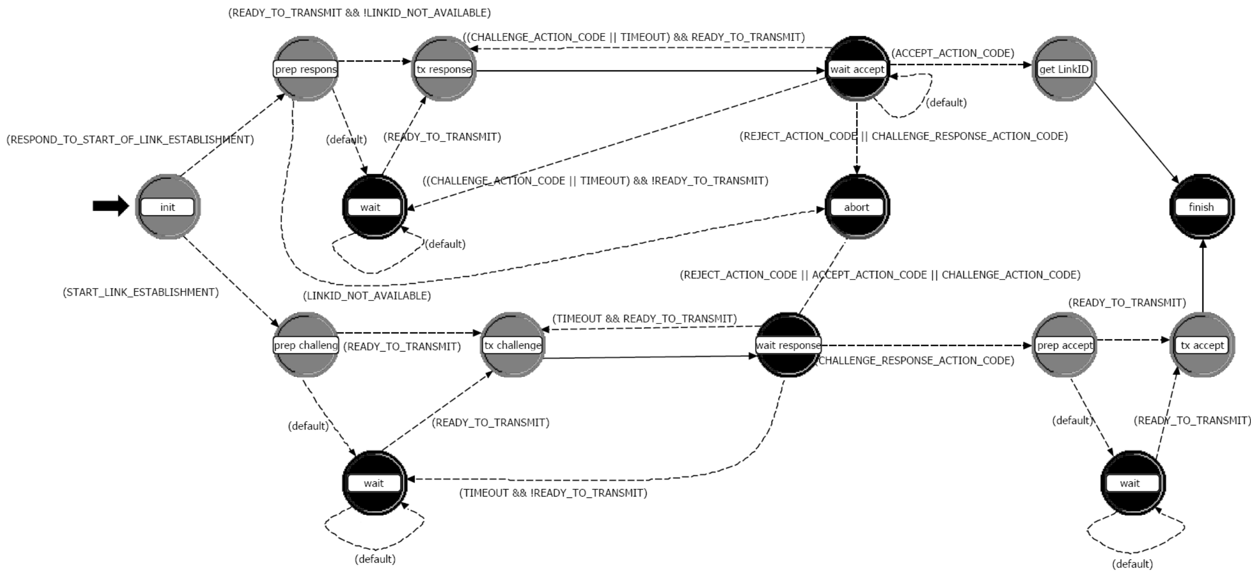


Figure 5-3. Link-establisher process model

5.3.2 The Schedulers

The schedulers implement the distributed-scheduling algorithm used for scheduling the data packets stored in all the input queues. To illustrate the scheduling process, we propose the Sliced-Greedy-Maximal-RBDS (Sliced-GM-RBDS) algorithm, based on the GM-RBDS policy proposed in Chapter 2. The Sliced-GM-RBDS algorithm uses sets of data slots like the ones shown in Figure 5-4. In order to specify these sets, the data subframes are all divided or sliced into a given number of data-slot groups of the same size. For example, in Figure 5-4, there are 12 data slots per frame numbered from 4 to 15 which are divided into 4 data-slot groups. These groups correspond to the data slots numbered 4-6, 7-9, 10-12, and 13-15. A set of data slots is specified with a data-slot group and a frame range. For example, if the group corresponds to data slots numbered from 7 to 9 and the frame range consists of frames numbered from 11 to 20, the set of data slots consists of 30 data slots (i.e., 10 frames and 3 data slots per frame). This set is shown in Figure 5-4 along with other three different sets. In WiMAX-RBDS-Sim, there are 256 minislots per frame as specified in the standard [1], and the size of the data-slot groups is a simulation parameter.

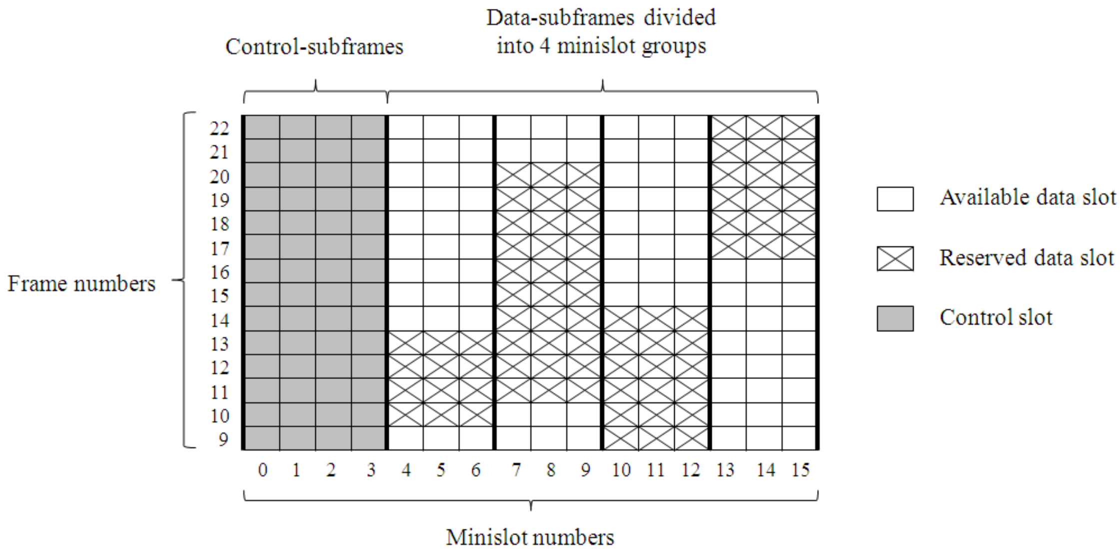


Figure 5-4. Data-slot reservation in the Sliced-GM-RBDS algorithm

The Sliced-GM-RBDS algorithm is based on the following scheduling policy. Whenever a node transmits a scheduling packet,

- for every outgoing link, request a set of data slots that covers the longest frame range that can be entirely covered with unscheduled data packets
- grant the longest requests that do not overlap with each other
- for every request and grant made, set its start frame (i.e., the first frame of the frame range covered by the request/grant) at the earliest possible frame such that overlaps are avoided

In the algorithm, a request IE (defined in Section 5.2.1) is always generated along with an availability IE. The request IE specifies the size of a set of data slots (i.e., the number of requested data slots). The availability IE specifies a set of data slots whose size is at least equal the size of the set of data slots specified in the request IE. The availability IE's set indicates all the data slots that the node's 1-hop neighbor is allowed to include in the grant for the request. Therefore, this set should not include any of the data slots that belong to any of the grants heard by the node sending the request, otherwise the grant generated at the node's 1-hop neighbor may overlap one or more of such grants, and the node will not be able to confirm the grant. On the other hand, the 1-hop neighbor is able to grant the request if it finds a set of data slots within the availability IE's set that does not overlap any of the grants that it has heard and that has the size specified in the request IE.

The scheduler process model is shown in Figure 5-5. It consists of two states. These are init and schedule. In state init, the scheduler initializes itself according to simulation parameters such as the number of data-slot groups. This initialization takes place only once when the scheduler is created by the main processor at the onset of the simulation. The tasks performed in state schedule take place every time the scheduler is invoked by the main processor. The main processor invokes the scheduler every time a scheduling packet is transmitted or received in order to generate and process MSH-DSCH messages respectively. The contents of the messages are generated and processed using the Sliced-GM-RBDS algorithm.

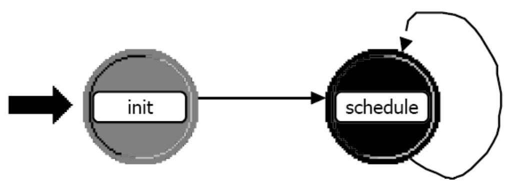


Figure 5-5. Scheduler process model

When the scheduler is invoked for generating the contents of an MSH-DSCH that is going to be transmitted, it generates grant, request, and availability IEs[5] .

Grant-IE generation: For every pending request, the set of overlapping requests is found. If the request is the longest in this set, it is granted and all the other requests in the set are discarded (i.e., they are not longer eligible for any grant). The grant for each of the granted requests is assigned a set of data slots that is determined as follows.

[5] The Sliced-GM-RBDS algorithm uses timers for discarding requests that expire.

The set's data-slot group is the one specified in the request's availability IE. The size of the set's frame range is the one specified in the request IE, and the number of the set's start frame is the maximum among the frame numbers of the following frames: the availability IE's start frame, the frame following the last frame granted in the data-slot group indicated by the availability IE, and the frame that is a pre-specified number of frames ahead of the current one. In this way, it is assured that the grant does not overlap any of the grants heard by the node and the 1-hop neighbor that made the request. Also, it is assured that the 1-hop neighbor is able to confirm the grant before the grant's start frame becomes the current frame.

Request and availability IEs generation: For every outgoing link, if there is a pre-specified minimum of unscheduled data packets in its input-queue, a request is made for a set of data slots whose length is the longest that can be entirely covered with unscheduled data packets. The availability of each request is generated according to the following four conditions.

It does not overlap any of the grants heard by the node.

If there is a data-slot group which has not been included in any of the availabilities heard by the node or any of the availabilities that were previously calculated for other outgoing links, that data-slot group is assigned to the availability being calculated, and the availability's start-frame number is the maximum among the frame numbers of the following frames: the frame following the last frame granted in the data-slot group, the frame when the next MSH-DSCH packet is transmitted by this node.

If there is not such data-slot group, the data-slot group that has been assigned to the lowest number of availabilities among all the availabilities heard by the node and the availabilities previously calculated for other outgoing links is assigned to the availability, and the availability's start-frame number is the maximum among the frame numbers of the following frames: the frame following the last frame granted across all the data-slot groups, the frame when the next MSH-DSCH packet is transmitted.

The number of the last frame of the availability's frame range is always made equal to infinity. In this way, the node that grants the request is able to use for the grant any of the frames following the availability's start frame. This is possible because none of the data slots that belong to the availability's data-slot group and to the frames that follow the availability's start frame has been granted according to the previous two items.

When the scheduler is invoked for processing the grant IEs of an MSH-DSCH message that has been received, it performs the following actions. For every grant IE in the MSH-DSCH message, the scheduler checks whether the grant is directed to the node. If it is, the scheduler looks for the ID of the outgoing link that connects to the 1-hop neighbor that sent the grant. Then, it schedules data packets that are in the input queue assigned to the outgoing link. These packets are scheduled in the set of data slots specified in the grant IE. Finally, it generates a grant IE for confirming the received grant. This grant confirmation is a copy of the received grant IE.

5.4 OPNET Implementation

The OPNET implementation of the WiMAX-RBDS-Sim architecture is shown in Figure 5-6. This implementation matches the WiMAX-RBDS-Sim architecture shown in Figure 5-2. The physical layer consists of a radio-receiver module and a radio-transmitter module. Each module is configured with 10 radio channels and any of the configuration modes (i.e., bandwidth, forward error correction, and modulation schemes) specified in the standard [1]. Also, each module is connected to 10 packet streams, which are OPNET elements used for communicating packets across modules. There is 1 packet stream per channel. The radio-receiver module connects to 10 packet-classifier modules through the packet streams, and each packet-classifier module connects to the main-processing module through 3 packet streams. Each of these packet streams carries packets of only one type (i.e., link-establishment, scheduling, or data packet). The main-processing module connects to the data-packet-source-sink module of the logical-link layer through 2 packet streams. The outgoing packet stream carries received data packets and notifications, in the form of packets, of new outgoing-link establishments. The incoming packet stream carries data packets generated by the data-packet-source-sink module. The main-processing module connects to 10 output-queue-channel-classifier modules with 1 packet stream per module. There is 1 of each of these modules per channel, and they are used for temporarily storing any type of scheduled packets. Finally, each of the output-queue-channel-classifier modules connects to the radio-transmitter module through 1 packet stream that carries scheduled packets of any type when they need to be transmitted.

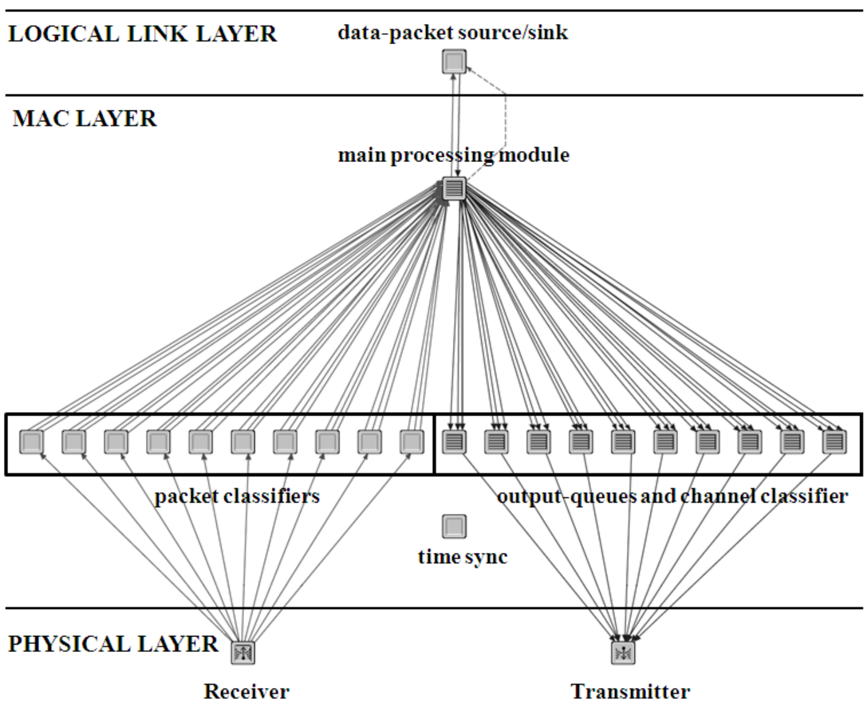


Figure 5-6. Implementation of the WiMAX-RBDS-Sim architecture

The main-processing module implements the main processor, link-establishers, and schedulers as shown in Figure 5-2. Also, it implements the input-queues in which data-packets generated at the logical-link layer are stored temporarily while they are scheduled.

The time-sync module synchronizes the main-processing and output-queue-channel-classifier modules with the frame structure shown in Figure 5-1. It interrupts these modules at the onset of every control and data slot and informs them of the current-time information such as the current frame number, current type of subframe (i.e., control or data subframe), and the current control or data slot number.

5.5 Simulation Results

An IEEE 802.16 mesh network with a grid topology of 7×8 nodes was simulated[6] . The number of slots in the control subframe was set to 9. The interference model considered was the protocol model [28] in which a reception is successful if the transmitter is closer to the receiver than any other node that transmits simultaneously. The traffic load was varied from 0 to 256 pk/s while accounting for the number of interfering links of every link. For example, if a link had 1 or 3 interfering links and the traffic load was set at 256 pk/s, the traffic generated for that link was 128 pk/s or 64 pk/s respectively (i.e., $\frac{256}{1+1}$ pk/s and $\frac{256}{1+3}$ pk/s). A total of 3000 frames were simulated.

The results obtained with WiMAX-RBDS-Sim are shown[7] in Figures 5-7, 5-8, 5-9, 5-10, 5-11, and 5-12.

Figure 5-7 shows the histogram of the control-subframe access delay. This is the delay a node experiences to gain access to the control subframe. When an IEEE 802.16 mesh node accesses the control subframe (i.e., it transmits either a link-establishment or scheduling packet), it competes for a future control slot to transmit its next link-establishment or scheduling packet. The time between two consecutive accesses to the control subframe is the access delay, and it directly

[6] The size of this network is representative of real WMNs [9].

[7] The WiMAX-RBDS-Sim framework also allows the analysis of the data-packet-delivery delay, throughput, and collision probability in the control and data subframes, but these results are not discussed here for the sake of brevity.

affects the time required to finish any handshake in control subframes. Therefore, the scheduling-handshake delay, which is the time required to perform the three-way handshake specified in Section 5.2.1 for negotiating a data-slot reservation, is affected by the shape of such histogram (i.e., the distribution of the control-subframe access delay). This shape can be modified by dynamically controlling the parameters of the election algorithm, which is the technique used in [6, 31, 40, 47, 78, 85–87] for the implementation of different QoS, collision avoidance, and congestion control schemes. Therefore, WiMAX-RBDS-Sim is a tool that can be used for the evaluation of such schemes by facilitating a simulation framework in which they can be readily integrated.

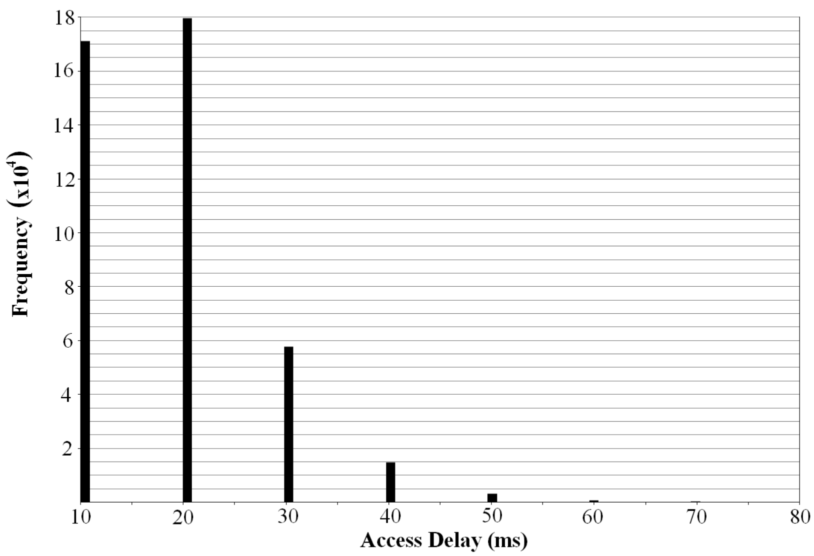


Figure 5-7. Control-subframe access delay histogram

Figure 5-8 shows the data-slot reservations of two 1-hop neighbors during an interval of 7 seconds. These reservations are calculated by the Sliced-GM-RBDS Algorithm so that overlaps between them are avoided. The algorithm was configured with 16 slices[8] . The blank spaces correspond to sets of data slots that were reserved for other nodes in the network. Ideally, there should not be any blank spaces nor

[8] The data-slot reservation example shown in Figure 5-4 considers only 4 slices.

overlaps when all the nodes that interfere with the two 1-hop neighbors are considered. Figure 5-8 is an example of the type of results that can be obtained for RBDS algorithms. In this example, the Sliced-GM-RBDS algorithm is being considered. However, other RBDS algorithms, such as the ones presented in [18, 21, 24, 42, 43, 48, 79, 88], can be integrated as scheduler processes (Figure 5-2) and evaluated.

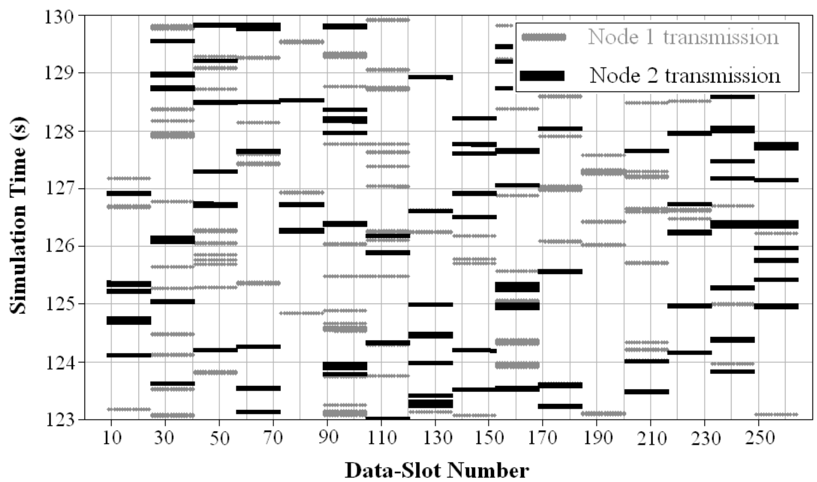


Figure 5-8. Data-slot reservation of two 1-hop neighbors in a network with grid topology of 7×8 nodes

Figure 5-9 shows the average length across the network of the output queues for a traffic load of 64 pk/s when the number of slices increases. The average output-queue lengths clearly depend on the number of slices. Therefore, the capacity of the network can be increased by configuring it with the optimal (i.e., minimum average output-queue length) number of slices. When the data subframes are not partitioned (i.e., when there is only 1 slice), the average output-queue length is not minimized. This is the configuration used in the GM-RBDS algorithm presented in Chapter 2. Therefore, the Sliced-GM-RBDS algorithm is an improved version of the GM-RBDS algorithm in terms of network capacity. The Sliced-GM-RBDS algorithm is able to maintain lower queue lengths, and as a consequence lower data-packet-delivery delays, than the GM-RBDS algorithm does. Specifically, the Sliced-GM-RBDS minimizes the average output-queue length when the number of slices is 16. Figure 5-10 shows the average output-queue length for different traffic loads when the number of slices is fixed at 16. The output-queue lengths are finite as long as the traffic load is below 128 pk/s. For higher traffic loads, the output-queue lengths increase indefinitely with time.

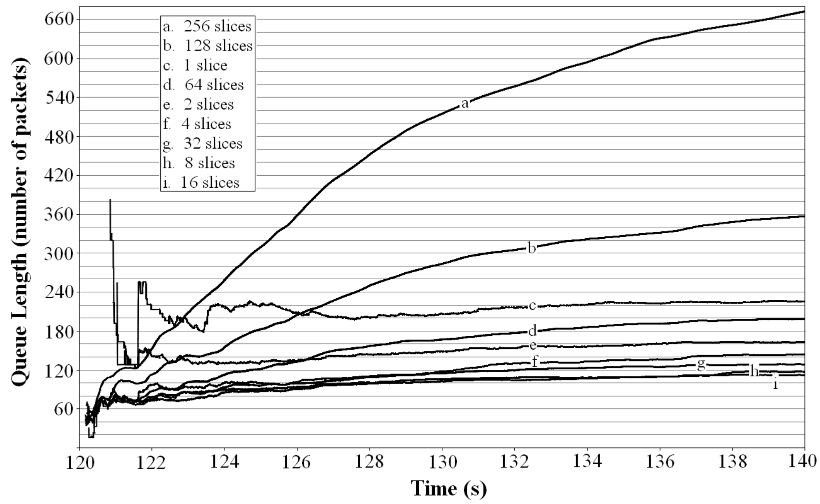


Figure 5-9. Average output-queue length for increasing number of slices (traffic load = 64 packets per second)

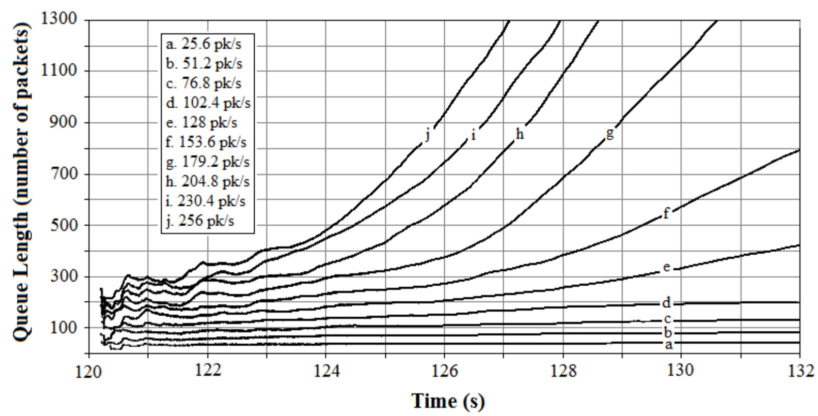


Figure 5-10. Average output-queue length for increasing traffic loads (number of slices = 16)

Figure 5-11 shows the instantaneous length of the input and output queues in the network when the traffic load is maximum (i.e., 256 pk/s). As expected from the results in Figure 5-10, the output queues are unstable (i.e., they increase indefinitely with time). However, the input queues are stable (i.e., they always return to the empty state at some points in time). Therefore, given that the input queues are always stable and the output queues are stable for traffic loads lower than 128 pk/s, the Sliced-GM-RBDS algorithm guarantees the stability of the network (i.e., all the queues in the network are stable) for the simulation scenario considered when the traffic load does not exceed 128 pk/s.

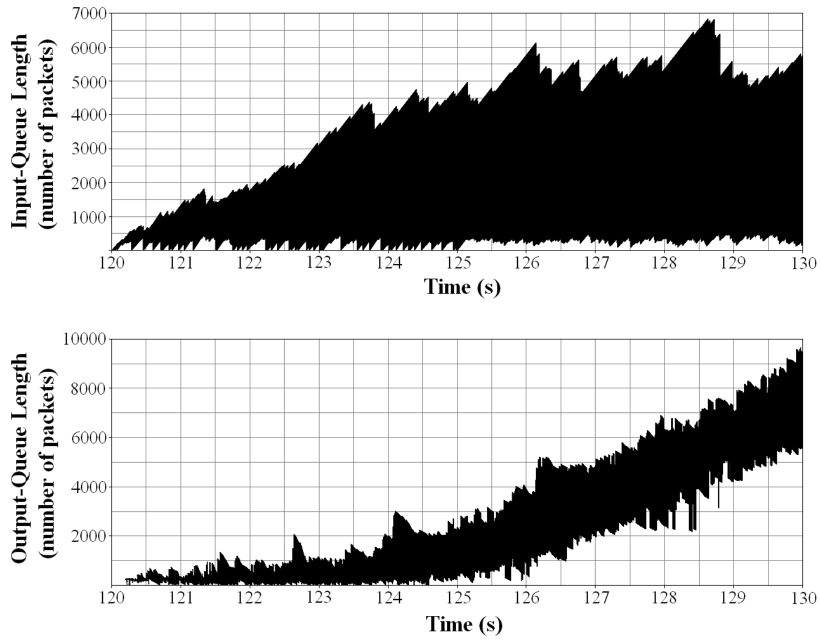


Figure 5-11. Input-queue and output-queue length comparison for maximum traffic load

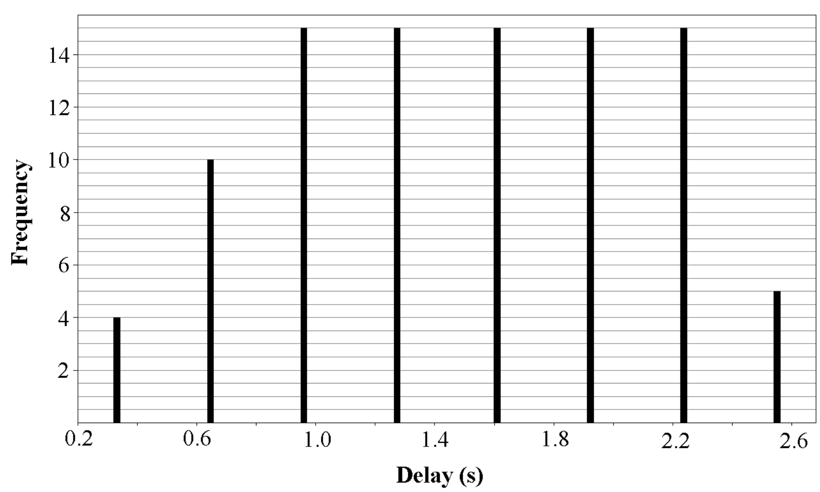


Figure 5-12. Link-establishment delay histogram

As a final example of the results and analysis that can be performed with WiMAX-RBDS-Sim, the histogram of the link-establishment delay is shown in Figure 5-12. This delay is the time required for establishing the incoming and outgoing links between

1-hop neighbors with the three-way handshake specified in Section 5.2.2. There is a total of 97 incoming-outgoing link pairs in the 7×8 grid topology. The results show that the link-establishment algorithm establishes a bidirectional link in no more than 2.6 s. This result is also affected by the distribution of the control-subframe access delay (Figure 5-7) given that the link-establishment handshake is performed with the exchange of packets in the control subframe. A scheme for reducing the link-establishment delay was proposed in [80].

5.6 Performance Evaluation

The performance of WiMAX-RBDS-Sim in terms of simulation speed and memory usage was evaluated for different network setups. The simulated network setups included 4 different grid topologies with increasing number of nodes (i.e., 4×8, 5×8, 6×8, and 7×8) with the same configuration used in Section 5.5. The simulations were performed in the sequential mode in which there is only one thread of execution[9] . Figures 5-13 and 5-14 show that the simulation speed decreases and the memory usage increases approximately linearly when the number of nodes increases.

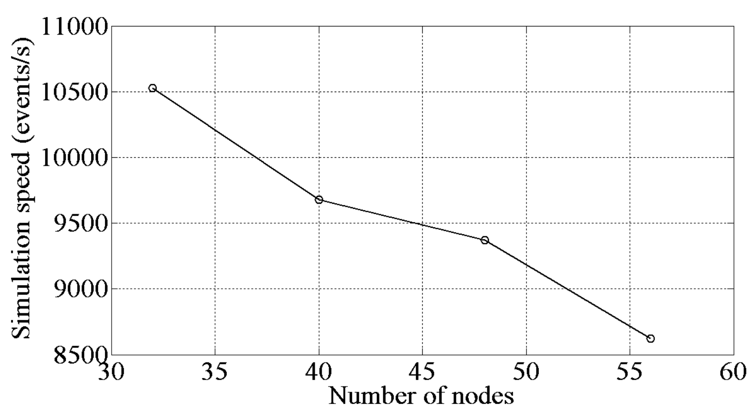


Figure 5-13. Simulation speed

[9] The platform used in the simulation included a processor Intel Core Duo at 3 GHz and 3 GB of RAM at 3 GHz.

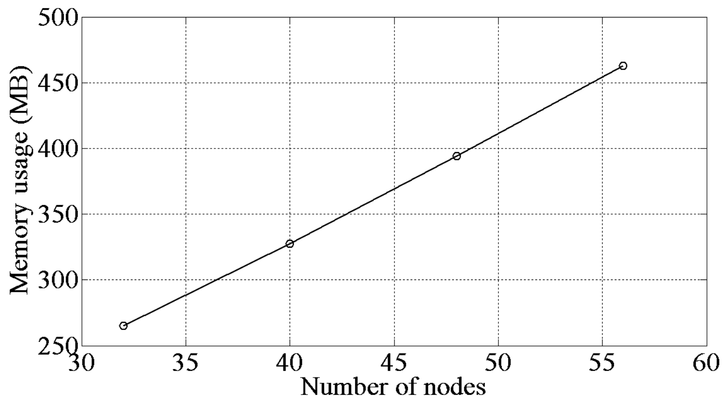


Figure 5-14. Memory usage

5.7 Summary

A new simulation model for IEEE 802.16 mesh networks with distributed scheduling (i.e., WiMAX-RBDS-Sim) was developed in OPNET. It provides interfaces for the implementation and evaluation of distributed scheduling policies and link-establishment algorithms. The policies may be based on the dynamic variation of the election-algorithm parameters or on the reservation of future data slots.

As an example of the use of WiMAX-RBDS-Sim, a new RBDS policy (i.e., Sliced-GM-RBDS) was implemented and evaluated with it. This policy is based on the concepts of input and output queues. The results showed that the Sliced-GM-RBDS policy outperformed the GM-RBDS policy in terms of capacity since the system of queues was stable for higher traffic loads when the Sliced-GM-RBDS was used. Also, a link-establishment algorithm was implemented and integrated with WiMAX-RBDS-Sim, and its performance was evaluated in terms of delay.

Finally, the performance of WiMAX-RBDS-Sim was evaluated in terms of simulation speed and memory usage for an increasing number of nodes.

CHAPTER 6
SUMMARY OF COMPLETED WORK

In Chapter 1, the research problem was stated. The main problem to be solved is to control the topology of wireless multihop networks (WMN) in order to maximize the performance of end-to-end data flows established on the networks. This performance is given in terms of the set of end-to-end data-packet rates that the network supports while guaranteeing stability. The topology is controlled by means of transmission-power control.

In order to solve the research problem, the work discussed in Chapters 2, 3, and 5 has been finished.

In Chapter 2, a new framework for the stability analysis of scheduling policies for wireless networks that allow the reservation of future data-subframes was proposed. In these policies, the nodes coordinate the assignment of data-subframes by exchanging scheduling packets. The concepts of input-queue and output-queue were introduced into the framework in order to account for the packets waiting to be scheduled and the schedules assigned to these packets. Based on these concepts, sufficient conditions for the stability of reservation-based-distributed-scheduling (RBDS) wireless networks were found.

Within the proposed framework, the greedy-maximal reservation-based-distributed-scheduling (GM-RBDS) policy was analyzed. The nodes implement this policy by exchanging scheduling packets using the Institute-of-Electrical-and-Electronics-Engineers (IEEE) 802.16 election algorithm. A region in which the proposed reservation-based scheduling policy is stable was found using the framework. It was shown that the size of this region depends on the factor ν which is determined by two characteristics of the network topology only (i.e., s^j and $|\mathcal{P}_{\mathrm{max}}|$). An IEEE 802.16 mesh network with the proposed scheduling policy (i.e., GM-RBDS) was simulated. It was shown that the policy always guaranteed the stability of the input-queues and that the output-queues were stable when the load was within $\frac{18}{24}$ of the optimal region. Finally, the performance of the GM-RBDS policy was compared with the W and enhanced-local-greedy-scheduling (ELGS) policies. It was shown that the GM-RBDS policy has an advantage over the W and ELGS policies in terms of the required overhead and its ability to reserve any future data-subframes.

In Chapter 3, the heuristic-stability-region-adaptation algorithm was proposed for transmission power control. This is a centralized algorithm that increases the input-packet rates that flows can support and decreases the end-to-end delays. It is is based on the adaptation of the stability region of a given link-scheduling policy when only the links that belong to a given set of flows are considered. The algorithm can be readily adapted to any link-scheduling policy whose stability region has been characterized, so it is not limited to any specific scheduling approach such as request-to-send/clear-to-send-based policies. The improvement on throughput achieved by our algorithm was evaluated by means of simulation for the min-hop routing algorithm and the GM-RBDS link-scheduling policy in IEEE 802.16 mesh networks. It was shown that it outperforms the classical solution of reducing transmission powers to increase spatial reuse. Also, its performance was evaluated. It was found that it depends on the flow density. The performance increases as the flow density decreases.

In Chapter 4, distributed topology-control algorithms for WMNs under GM-RBDS were designed considering the results in [73, 77]. The problem was solved in three steps. First, the topology-control optimization problem was formulated as a potential game in which the flows (i.e., players) collaborate for maximizing the packet rates they can support while guaranteeing stability. Second, in order to analyze the potential game, the objective functions (i.e., flows' utilities) of the topology-control optimization problem were heuristically modified such that the problem could be formulated as an integer linear program. This problem corresponded to the optimization of a linear potential function of the modified game. This linear potential function approximated the objective of the original problem (i.e., without the modifications). The reason for the linear formulation is that it guaranteed that the potential game converged to a Nash equilibrium that is also a global optimum of the modified objective function. However, given that a different objective function was optimized in the game due to the modifications, the transmission powers (TP) calculated by the flows were in some cases different to the optimal TPs. Third, the difference of the maximum total throughput achieved by means of the modified potential game and the optimal maximum throughput was characterized. This characterization was based on the worst case scenarios in which the modified utilities mislead the flows to decide strategies that did not optimize the maximum total throughput. The simulation results showed that for neworks with low flow density, the game-theoretical algorithms performed simalarly to the centralized algorithm of Chapter 3, for networks with medium flow density, they outperformed the centralized algorithm, and for networks with high flow density, they were outperformed by the centralized algorithm. The reason for this behavior was the view of the network that the algorithms have. The game-theoretical algorithms have limited view of the network due to their distributed nature, and the centralized algorithm has a global view of the network.

Finally, in Chapter 5, a new simulation model for IEEE 802.16 mesh networks with distributed scheduling was developed using optimized network engineering tools (OPNET). It was called worldwide-interoperability-for-microwave-access RBDS simulator (WiMAX-RBDS-Sim). It provides interfaces for the implementation and evaluation of distributed scheduling policies and link-establishment algorithms. The policies may be based on the dynamic variation of the election-algorithm parameters or on the reservation of future data slots.

As an example of the use of WiMAX-RBDS-Sim, a new RBDS policy (i.e., Sliced-GM-RBDS) was implemented and evaluated with it. This policy is based on the concepts of input and output queues. The results showed that the Sliced-GM-RBDS policy outperformed the GM-RBDS policy in terms of capacity since the system of queues was stable for higher traffic loads when the Sliced-GM-RBDS was used. Also, a link-establishment algorithm was implemented and integrated with WiMAX-RBDS-Sim, and its performance was evaluated in terms of delay.

The results obtained in Chapters 1 to 5 have been published as follows. The research problem discussed in Chapter 1 was originally formulated in [70]. The results discussed in Chapter 2 have been published in [74] and [71]. The results discussed in Chapter 3 have been published in [73] and [77]. The results obtained in Chapter 4 have been submitted for publication in [76]. The results discussed in Chapter 5 have been published in [75] and [72].

APPENDIX A
INPUT AND OUTPUT DATA-PACKET ARRIVAL RATE OF $\mathcal{S}_{\mathsf{S}}^{\mathcal{G}}$

The expected number of data-packet arrivals to node j's input-queue $Q_i^j(n)$ is derived as follows.

$$
\begin{aligned}
\lambda_{\mathsf{s}}^{(i,j)} \triangleq \mathsf{E}\big[A_{\mathsf{s}}^{(i,j)}\big] &= \mathsf{E}\left[\sum_{m=m_{n+1}-\Delta M_{m_n}^j}^{m_{n+1}} A'^{(i,j)}(m)\right] \\
&= \mathsf{E}\left[\sum_{l=0}^{N_{\mu_1}^j(m_n)-1} A'^{(i,j)}\left(\left(\left\lfloor \frac{m_{n+1}}{m_{\mathsf{cs}}}\right\rfloor - l\right) m_{\mathsf{cs}}\right)\right] \\
&= \mathsf{E}\left[\sum_{l=0}^{N_{\mu_1}^j(m_n)-1} \sum_{k=0}^{m_{\mathsf{ds}}-1} A^{(i,j)}\left(k_{\left(\left(\left\lfloor \frac{m_{n+1}}{m_{\mathsf{cs}}}\right\rfloor - l\right) m_{\mathsf{cs}}\right)} - k\right)\right] \\
&= \sum_{r=1}^{\infty} \mathsf{P}\left[N_{\mu_1}^j(m_n) = r\right] \mathsf{E}\left[\sum_{l=0}^{r-1} \sum_{k=0}^{m_{\mathsf{ds}}-1} A^{(i,j)}\left(k_{\left(\left(\left\lfloor \frac{m_{n+1}}{m_{\mathsf{cs}}}\right\rfloor - l\right) m_{\mathsf{cs}}\right)} - k\right)\right] \\
&= \sum_{r=1}^{\infty} \mathsf{P}_{\mu_1}^j[r] \sum_{l=0}^{r-1} \sum_{k=0}^{m_{\mathsf{ds}}-1} \mathsf{E}\left[A^{(i,j)}\left(k_{\left(\left(\left\lfloor \frac{m_{n+1}}{m_{\mathsf{cs}}}\right\rfloor - l\right) m_{\mathsf{cs}}\right)} - k\right)\right] \\
&= \sum_{r=1}^{\infty} \mathsf{P}_{\mu_1}^j[r] \sum_{l=0}^{r-1} \sum_{k=0}^{m_{\mathsf{ds}}-1} \lambda^{(i,j)} = m_{\mathsf{ds}} \lambda^{(i,j)} \sum_{r=1}^{\infty} \mathsf{P}_{\mu_1}^j[r] \sum_{l=0}^{r-1} 1 \\
&= m_{\mathsf{ds}} \lambda^{(i,j)} \sum_{r=1}^{\infty} r \mathsf{P}_{\mu_1}^j[r] = \overline{N_{\mu_1}^j} m_{\mathsf{ds}} \lambda^{(i,j)}
\end{aligned}
$$

APPENDIX B
PROOF OF THEOREM 2.1

Consider the Lyapunov functions $V_{\mathrm{s}}^{i}(n)$ and $V_{\mathrm{s}}^{o}(n)$ defined next for the queue processes $Q_i^j(n)$ and $Q_o^j(n)$ in $\mathcal{S}_{\mathrm{s}}^{\mathcal{G}}$.

$$V_{\mathrm{s}}^{i,o}(n) \triangleq \sum_{j\in\mathcal{N}} \left(Q_{i,o}^{j}(n)\right)^2$$

The following Lyapunov functions $V^i(m_n)$ and $V^o(m_n)$ for queue processes $Q_i^{(i,j)}(m)$ and $Q_o^{(i,j)}(m)$ in $\mathcal{S}^{\mathcal{G}}$ are found to be equivalent to $V_{\mathrm{s}}^{i}(n)$ and $V_{\mathrm{s}}^{o}(n)$ respectively as follows.

$$\begin{aligned}
V^i(m_n) &\triangleq \sum_{(i,j)\in\mathcal{L}} Q_i^{(i,j)}(m_n)\Bigg(\sum_{k\in\mathcal{S}_1^j} Q_i^{(k,j)}(m_n)\Bigg) \\
&= \sum_{(i,j)\in\mathcal{L}} \sum_{k\in\mathcal{S}_1^j} Q_i^{(i,j)}(m_n) Q_i^{(k,j)}(m_n) \\
&= \sum_{j\in\mathcal{N}} \sum_{i\in\mathcal{S}_1^j} \sum_{k\in\mathcal{S}_1^j} Q_i^{(i,j)}(m_n) Q_i^{(k,j)}(m_n) \\
&= \sum_{j\in\mathcal{N}} \Bigg(\sum_{i\in\mathcal{S}_1^j} Q_i^{(i,j)}(m_n)\Bigg)\Bigg(\sum_{k\in\mathcal{S}_1^j} Q_i^{(k,j)}(m_n)\Bigg) \\
&= \sum_{j\in\mathcal{N}} \left(Q_i^j(n)\right)^2 = V_s^i(n) \\
V^o(m_n) &\triangleq \sum_{j\in\mathcal{N}} \left(\max_{i\in\mathcal{S}_1^j} Q_o^{(i,j)}(m_n)\right)^2 = \sum_{j\in\mathcal{N}} \left(Q_o^j(n)\right)^2 = V_{\mathrm{s}}^{o}(n)
\end{aligned}$$

Therefore, $V^i(m_n)$ has negative drift if and only if $V_{\mathrm{s}}^{i}(n)$ has negative drift. The same relation holds for $V^o(m_n)$ and $V_{\mathrm{s}}^{o}(n)$.

The change in the value of $V_{\mathrm{s}}^{o}(n)$ is found as follows.

$$
\begin{aligned}
V_{\mathsf{s}}^{o}(n+1) - V_{\mathsf{s}}^{o}(n) &= \sum_{j\in\mathcal{N}} \big(Q_o^j(n+1)\big)^2 - \sum_{j\in\mathcal{N}} \big(Q_o^j(n)\big)^2 \\
&= \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n+1)\big)^2 + \sum_{j\notin\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n+1)\big)^2 - \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&\quad - \sum_{j\notin\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&= \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n) + |\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big)^2 + \sum_{j\notin\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&\quad - \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 - \sum_{j\notin\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&= \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n) + |\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big)^2 - \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&= \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 + \sum_{j\in\mathcal{N}_{\mu_2}(n)} |\mathcal{H}^j(n)|^2 + \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(N_{\mu_1}^j(n)\big)^2 \\
&\quad - 2\sum_{j\in\mathcal{N}_{\mu_2}(n)} Q_o^j(n) N_{\mu_1}^j(n) + 2\sum_{j\in\mathcal{N}_{\mu_2}(n)} Q_o^j(n)|\mathcal{H}^j(n)| \\
&\quad - 2\sum_{j\in\mathcal{N}_{\mu_2}(n)} N_{\mu_1}^j(n)|\mathcal{H}^j(n)| - \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(Q_o^j(n)\big)^2 \\
&= 2\sum_{j\in\mathcal{N}_{\mu_2}(n)} Q_o^j(n)\big(|\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big) + \sum_{j\in\mathcal{N}_{\mu_2}(n)} \big(|\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big)^2
\end{aligned}
$$

The drift of $V_{\mathsf{s}}^{o}(n)$ is found as follows, where C is some constant.

$$
\begin{aligned}
&\mathsf{E}_{Q_o^j(n)}\big[V_{\mathsf{s}}^{o}(n+1) - V_{\mathsf{s}}^{o}(n)\big] \\
&= 2\mathsf{E}_{Q_o^j(n)}\Bigg[\sum_{j\in\mathcal{N}_{\mu_2}(n)} Q_o^j(n)\big(|\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big)\Bigg] + C \\
&= 2\sum_{\mathcal{N}'\subseteq\mathcal{N}} \mathsf{E}_{Q_o^j(n),\mathcal{N}_{\mu_2}(n)}\Bigg[\sum_{j\in\mathcal{N}'} Q_o^j(n)\big(|\mathcal{H}^j(n)| - N_{\mu_1}^j(n)\big)\Bigg]\mathsf{P}\big[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\big] + C \\
&= 2\sum_{\mathcal{N}'\subseteq\mathcal{N}}\sum_{j\in\mathcal{N}'} Q_o^j(n)\big(\mathsf{E}\big[|\mathcal{H}^j(n)|\big] - \mathsf{E}\big[N_{\mu_1}^j(n)\big]\big)\mathsf{P}\big[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\big] + C
\end{aligned}
$$

In the previous equalities, the facts that $\mathsf{E}_{Q_o^j(n),\mathcal{N}_{\mu_2}(n)}\left[|\mathcal{H}^j(n)|\right] = \mathsf{E}\left[|\mathcal{H}^j(n)|\right]$ and $\mathsf{E}_{Q_o^j(n),\mathcal{N}_{\mu_2}(n)}\left[N_{\mu_1}^j(n)\right] = \mathsf{E}\left[N_{\mu_1}^j(n)\right]$ have been used. These facts are true because $|\mathcal{H}^j(n)|$ and $N_{\mu_1}^j(n)$ are independent of $Q_o^j(n)$ and $\mathcal{N}_{\mu_2}(n)$. $|\mathcal{H}^j(n)|$ is the output-queue-length increase, which corresponds to the total number of data-subframes included in the grants, blanks, and overlaps that are scheduled between the n^{th} and $(n+1)^{\text{th}}$ scheduling-packet transmissions of j (Section 2.3.2.4). This set of data-subframes is shown in Figure 2-5B. The distribution of grants, blanks, and overlaps depends on how the specific reservation-based-distributed-scheduling (RBDS) policy implemented on the network uses the length of input-queues for calculating requests. $N_{\mu_1}^j(n)$ is determined by the number of control-time-slots between these transmissions, and this number is equal to ΔM_m^j (Eq. 2–10).

Given that the network is stationary, the drift of $V_{\mathsf{s}}^o(n)$ is upper-bounded as follows.

$$
\begin{aligned}
&\mathsf{E}_{Q_o^j(n)}\left[V_{\mathsf{s}}^o(n+1) - V_{\mathsf{s}}^o(n)\right] \\
&= 2\sum_{\mathcal{N}'\subseteq\mathcal{N}}\sum_{j\in\mathcal{N}'} Q_o^j(n)\big(\overline{|\mathcal{H}^j|} - \overline{N_{\mu_1}^j}\big)\mathsf{P}\left[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\right] + C \\
&\le 2\sum_{\mathcal{N}'\subseteq\mathcal{N}}\sum_{j\in\mathcal{N}'} Q_o^j(n)\big(\max_{j\in\mathcal{N}'}\big(\overline{|\mathcal{H}^j|} - \overline{N_{\mu_1}^j}\big)\big)\mathsf{P}\left[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\right] + C \\
&= 2\sum_{\mathcal{N}'\subseteq\mathcal{N}} \mathsf{P}\left[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\right]\big(\max_{j\in\mathcal{N}'}\big(\overline{|\mathcal{H}^j|} - \overline{N_{\mu_1}^j}\big)\big)\sum_{j\in\mathcal{N}'} Q_o^j(n) + C \\
&\le 2\big(\max_{j\in\mathcal{N}}\big(\overline{|\mathcal{H}^j|} - \overline{N_{\mu_1}^j}\big)\big)\sum_{\mathcal{N}'\subseteq\mathcal{N}} \mathsf{P}\left[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\right]\sum_{j\in\mathcal{N}'} Q_o^j(n) + C
\end{aligned}
$$

By a similar analysis, it is found that the drift of $V_{\mathsf{s}}^i(n)$ is upper-bounded as follows.

$$
\begin{aligned}
&\mathsf{E}_{Q_o^j(n)}\left[V_{\mathsf{s}}^i(n+1) - V_{\mathsf{s}}^i(n)\right] \le \\
&\qquad 2\left(\max_{j\in\mathcal{N}}\left(\sum_{i\in\mathcal{S}_1^j}\lambda_{\mathsf{s}}^{(i,j)} - m_{\mathsf{ds}}\sum_{i\in\mathcal{S}_1^j}\overline{\left|G^{(i,j)}\right|}\right)\right)\sum_{\mathcal{N}'\subseteq\mathcal{N}} \mathsf{P}\left[\mathcal{N}_{\mu_2}(n) = \mathcal{N}'\right]\sum_{j\in\mathcal{N}'} Q_o^j(n) + C
\end{aligned}
$$

Therefore, the drifts of both $V_{\mathsf{s}}^{i}(n)$ and $V_{\mathsf{s}}^{o}(n)$ are negative if

$$\max_{j\in\mathcal{N}} \left(\overline{|\mathcal{H}^j|} - \overline{N^j_{\mu_1}}\right) < 0,$$
$$\max_{j\in\mathcal{N}} \left(\sum_{i\in\mathcal{S}_1^j} \lambda_{\mathsf{s}}^{(i,j)} - m_{\mathsf{ds}} \sum_{i\in\mathcal{S}_1^j} \overline{|G^{(i,j)}|}\right) < 0,$$

and the proof follows from this condition and Eq. 2–15.

APPENDIX C
PROOF OF LEMMA 1

Consider the input-queue given by the Markov process in Eq. 2–25, assume that the probability that the request for this queue is successfully granted $\big($i.e., $\mathsf{P}\big[I^{(i,j)}_{Q_{\max}}(n) = 1, I^{(i,j)}_{\mathrm{s}}(n) = 1\big]\big)$ is always equal to its lower-bound given by Eq. 2–28. Let s denote this lower-bound. The expected value of $Q^{(i,j)}_{\mathrm{s}}(n)$ is upper-bounded by the expected value of $Q^{(i,j)}_{\mathrm{s}}(n)$ found under this assumption.

Let a_x and δ_x be defined as follows.

$$a_x \triangleq \mathsf{P}[A^{(i,j)}_{\mathrm{s}}(n) = x]$$
$$\delta_x \triangleq \mathsf{P}[\Delta Q^{(i,j)}_{\mathrm{s}}(n) = x]$$

The probability that $Q^{(i,j)}_{\mathrm{s}}(n)$ moves from state x_1 to state x_2, where $x_1 \ \leq \ x_2$ is $s(a_n * \delta_n)[x_2] + (1-s)a_{x_2 - x_1}$. That is, the probability that the request is successfully granted and the number of packet arrivals $A^{(i,j)}_{\mathrm{s}}(n)$ plus the remaining packets in the queue $\Delta Q^{(i,j)}_{\mathrm{s}}(n)$ is x_2 or that the request is not successfully granted and there are $x_2 - x_1$ packet arrivals.

Therefore, the equilibrium equations are given as follows for all P_x, where P_x is the probability that $Q^{(i,j)}_{\mathrm{s}}(n)$ is in state x.

$$\begin{aligned}
P_x &= (s(a_n * \delta_n)[x] + (1-s)a_x)P_0 + (s(a_n * \delta_n)[x] + (1-s)a_{x-1})P_1 + \ldots \\
&= s(a_n * \delta_n)[x]\sum_{n=0}^{\infty} P_n + (1-s)\sum_{n=0}^{\infty} a_{x-n}P_n \\
&= s(a_n * \delta_n)[x] + (1-s)(a_n * P_n)[x]
\end{aligned}$$

The expected value of $Q^{(i,j)}_{\mathrm{s}}(n)$ is found by evaluating $\frac{dP(z)}{dz}\big|_{z=1}$, where $P(z)$ is defined as $\sum_{x=0}^{\infty} P_x z^x$.

$P(z)$ and $\frac{dP(z)}{dz}\big|_{z=1}$ are given as follows.

$$
\begin{aligned}
&P(z) = sA(z)\Delta(z) + (1-s)A(z)P(z) \\
&\frac{dP(z)}{dz}\Big|_{z=1} = \frac{1}{s}\frac{dA(z)}{dz}\Big|_{z=1} + \frac{d\Delta(z)}{dz}\Big|_{z=1} = \frac{1}{s}\frac{\mathsf{E}[A_s^{(i,j)}(n)]}{m_{\mathrm{ds}}} + \mathsf{E}[\Delta Q_{\mathrm{s}}^{(i,j)}(n)]
\end{aligned}
$$

From Eq. 2–15, Eq. 2–28, and the assumption that $\mathsf{P}\big[I_{Q_{\max}}^{(i,j)}(n) = 1, I_{\mathrm{s}}^{(i,j)}(n) = 1\big] = s$, the expected value of $Q_{\mathrm{s}}^{(i,j)}(n)$ can be upper-bounded as follows.

$$
\mathsf{E}\big[Q_{\mathrm{s}}^{(i,j)}(n)\big] \leq s^j \lambda^j \overline{N_{\mu_1}} + \mathsf{E}\big[\Delta Q_{\mathrm{s}}^{(i,j)}(n)\big]
$$

APPENDIX D
PROOF OF THEOREM 3.1

Let the following notation be defined for the presentation of the proof. Link (i,j) belongs to flow $f_n \in \mathcal{F}$ if node j is preceded by node i at any point in the flow's path p_n $\big($i.e., $(i,j) \in f_n$ if $p_n(m) = i$ and $p_n(m+1) = j$ for some $m = 1, 2, \dots, |p_n| - 1\big)$. The set of flows that link (i,j) belongs to is denoted by $\mathcal{F}_{(i,j)}$ $\big($i.e., $\mathcal{F}_{(i,j)} \triangleq \{f_n : f_n \in \mathcal{F}, (i,j) \in f_n\}\big)$. A node has three queues for every flow that it belongs to as shown in Figure D-1. For flow f_n on link (i,j) at node i, $Q^{(i,j)}_{\mathrm{r},f_n}(n)$, $Q^{(i,j)}_{\mathrm{i},f_n}(n)$, and $Q^{(i,j)}_{\mathrm{o},f_n}(n)$ are the regulator, input, and output queues respectively. A λ-regulator is a logical device with maximum service rate λ, i.e., at each time-slot, a λ-regulator checks its queue and, if there are any packets, it transfers a packet with probability λ. The following rule[1] is used for each flow f_n in $\mathcal{F}$: for the first hop along the flow's path (i.e., $p_n(1)$), there is a λ_n-regulator at its input queue; for the m^{th} hop (i.e., $p_n(m), m \geq 2$), there is a $(\lambda + (m-1)\epsilon)$-regulator, where $\epsilon > 0$. The packet departures of queues $Q^{(i,j)}_{\mathrm{r},f_n}$, $Q^{(i,j)}_{\mathrm{i},f_n}$, and $Q^{(i,j)}_{\mathrm{o},f_n}$ are denoted by $R^{(i,j)}_{f_n}(n)$, $G^{(i,j)}_{f_n}(n)$, and $D^i(n)$ respectively. The queue lengths and departures are indexed by n which denotes the n^{th} time that one or more than two nodes in the network transmit a scheduling packet. The link in flow f_n that follows link (i,j) is denoted by $(i,j)'$. For example, $Q^{(i,j)'}_{\mathrm{r},f_n}$ is the regulator queue for the link following link (i,j) in flow f_n, which is located at the destination node of link (i,j) (i.e., node j) as shown in Figure D-1. The total traffic rate on link (i,j) is denoted by $\lambda^{(i,j)}$, i.e., $\lambda^{(i,j)} = \sum_{\lambda_n \in \mathcal{F}_{(i,j)}} \lambda_n$.

Let the following Lyapunov function $V\big(\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{i}(n), \mathbf{Q}_\mathrm{o}(n)\big) = V_\mathrm{r}\big(\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{o}(n)\big) + V_\mathrm{i}\big(\mathbf{Q}_\mathrm{i}(n)\big) + V_\mathrm{o}\big(\mathbf{Q}_\mathrm{o}(n)\big)$ be defined for the system of queues given by $\mathbf{Q}_{\mathrm{r,\,i,\,o}}(n) \triangleq \big[Q^{(i,j)}_{\mathrm{r,\,i,\,o},f_n}(n)\big]_{f_n \in \mathcal{F}, (i,j) \in \mathcal{L}}$ as follows.

[1] This rule was originally used in [84] for the proof of the stability region of distributed greedy scheduling policies under regulated multihop traffic.

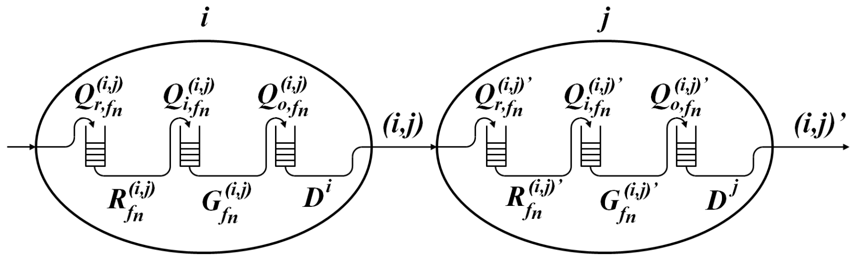


Figure D-1. Node queuing model per flow it belongs to

$$V_{\mathrm{r}}\big(\mathbf{Q}_{\mathrm{r}}(n),\mathbf{Q}_{\mathrm{o}}(n)\big) \triangleq \sum_{j,i,f_n}\Big(Q^{(i,j)}_{\mathrm{o},f_n}(n)+Q^{(i,j)'}_{\mathrm{r},f_n}(n)\Big)^2 \triangleq \sum_{j\in\mathcal{N}}\sum_{i\in\mathcal{S}^j_1}\sum_{f_n\in\mathcal{F}_{(i,j)}}\Big(Q^{(i,j)}_{\mathrm{o},f_n}(n)+Q^{(i,j)'}_{\mathrm{r},f_n}(n)\Big)^2$$

$$V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n)\big) \triangleq \sum_{(i,j)\in\mathcal{L}}\left(\left(\sum_{f_n\in\mathcal{F}_{(i,j)}} Q^{(i,j)}_{\mathrm{i},f_n}(n)\right)\left(\sum_{k\in\mathcal{S}^j_1}\sum_{f_n\in\mathcal{F}_{(k,j)}} Q^{(k,j)}_{\mathrm{i},f_n}(n)\right)\right)$$

$$V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n)\big) \triangleq \sum_{j\in\mathcal{N}}\left(\max_{i\in\mathcal{S}^j_1}\left(\sum_{f_n\in\mathcal{F}_{(i,j)}} Q^{(i,j)}_{\mathrm{i},f_n}(n)\right)\right)^2$$

The drift of $V\big(\mathbf{Q}_{\mathrm{r}}(n),\mathbf{Q}_{\mathrm{i}}(n),\mathbf{Q}_{\mathrm{o}}(n)\big)$, is defined as follows[2] .

$$E_{\mathrm{Q}_{\mathrm{r}}(n),\mathrm{Q}_{\mathrm{i}}(n),\mathrm{Q}_{\mathrm{o}}(n)}\big[\Delta V\big] =$$
$$E_{\mathrm{Q}_{\mathrm{r}}(n),\mathrm{Q}_{\mathrm{i}}(n),\mathrm{Q}_{\mathrm{o}}(n)}\Big[V\big(\mathbf{Q}_{\mathrm{r}}(n+1),\mathbf{Q}_{\mathrm{i}}(n+1),\mathbf{Q}_{\mathrm{o}}(n+1)\big)-V\big(\mathbf{Q}_{\mathrm{r}}(n),\mathbf{Q}_{\mathrm{i}}(n),\mathbf{Q}_{\mathrm{o}}(n)\big)\Big]$$

The proof is based on the sufficient conditions which guarantee that the drift of $V\big(\mathbf{Q}_{\mathrm{r}}(n),\mathbf{Q}_{\mathrm{i}}(n),\mathbf{Q}_{\mathrm{o}}(n)\big)$ is negative[3] . It proceeds as follows. First, the drifts of $V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n)\big)$ and $V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n)\big)$ are obtained from the results in Chapter 2. Then, the drift

[2] $E_Y[X]$ denotes the expected value of X given Y (i.e., $E_Y[X] \triangleq E[X|Y]$).

[3] According to Foster's theorem [5], a Markovian queueing system is stable if the drift of the Lyapunov function is negative.

of $V_{\mathrm{r}}\big(\mathbf{Q}_{\mathrm{r}}(n), \mathbf{Q}_{\mathrm{o}}(n)\big)$ is calculated. Finally, the drift of $V\big(\mathbf{Q}_{\mathrm{r}}(n), \mathbf{Q}_{\mathrm{i}}(n), \mathbf{Q}_{\mathrm{o}}(n)\big)$ is calculated from the drifts of $V_{\mathrm{r}}\big(\mathbf{Q}_{\mathrm{r}}(n), \mathbf{Q}_{\mathrm{o}}(n)\big)$, $V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n)\big)$, and $V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n)\big)$, and the sufficient condition in which the drift of $V\big(\mathbf{Q}_{\mathrm{r}}(n), \mathbf{Q}_{\mathrm{i}}(n), \mathbf{Q}_{\mathrm{o}}(n)\big)$ is negative is found.

According to Chapter 2, under the greedy-maximal reservation-based-distributed-scheduling (GM-RBDS) policy, the drift of $V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n)\big)$ is upper-bounded as follows, where C_1 and C_2 are positive constants independent of $\mathbf{Q}_{\mathrm{r}}(n)$, $\mathbf{Q}_{\mathrm{i}}(n)$, and $\mathbf{Q}_{\mathrm{o}}(n)$.

$$E_{\mathrm{Q}_{\mathrm{i}}(n)}\big[\Delta V_{\mathrm{i}}\big] = E_{\mathrm{Q}_{\mathrm{i}}(n)}\big[V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n+1)\big) - V_{\mathrm{i}}\big(\mathbf{Q}_{\mathrm{i}}(n)\big)\big] < -C_1 \sum_{j,i,f_n} Q_{\mathrm{i},f_n}^{(i,j)}(n) + C_2 \qquad \text{(D–1)}$$

Therefore, under the GM-RBDS policy, the input-queues are guaranteed to be always stable. Qualitatively, this result is a consequence of the fact that under the GM-RBDS policy, every time a node transmits a scheduling packet, it requests for every link as many future data-subframes as can be covered completely with unscheduled data packets. Given that these requests are always granted with some probability greater than zero (Chapter 2), the probability that the input-queues return to the empty state is always greater than zero.

According to Chapter 2, under the GM-RBDS policy, the drift of $V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n)\big)$ is upper-bounded as follows, where C_3 and η are constants independent of $\mathbf{Q}_{\mathrm{r}}(n)$, $\mathbf{Q}_{\mathrm{i}}(n)$, and $\mathbf{Q}_{\mathrm{o}}(n)$.

$$E_{\mathrm{Q}_{\mathrm{o}}(n)}\big[\Delta V_{\mathrm{o}}\big] = E_{\mathrm{Q}_{\mathrm{o}}(n)}\big[V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n+1)\big) - V_{\mathrm{o}}\big(\mathbf{Q}_{\mathrm{o}}(n)\big)\big] < -\eta \sum_{j,i,f_n} Q_{\mathrm{o},f_n}^{(i,j)}(n) + C_3 \qquad \text{(D–2)}$$

The constant C_3 is positive. The constant η is positive if the condition given by Eq. D–3 is satisfied (Chapter 2), where $\mathcal{P}_{\max}$ is the longest path among all the paths in the graphs induced by every node's 2-hop neighborhood and that originate at the node $\big($i.e., if $\mathcal{P}^i$ is the longest path that originates at node i in the graph induced by $\mathcal{S}^i_{\leq 2}$, then

$\mathcal{P}_{\mathsf{max}} = \mathrm{argmax}_{\{\mathcal{P}^i : i \in \mathcal{N}\}} \left(|\mathcal{P}^i|\right)$. The length of $\mathcal{P}_{\mathsf{max}}$ is defined as the number of links in it and denoted by $|\mathcal{P}_{\mathsf{max}}|$.

$$\sum_{i \in \mathcal{S}_1^j} \lambda^{(i,j)} \Big(1 + \big|\mathcal{S}_1^i \backslash \big(\mathcal{S}_1^j \cup j\big)\big|\Big) < \frac{1}{2|\mathcal{P}_{\mathsf{max}}| + 1} \quad \forall\ j \in \mathcal{N} \tag{D–3}$$

Therefore, under the GM-RBDS policy, the output-queues are guaranteed to be stable if Eq. D–3 is satisfied. Qualitatively, the condition given by Eq. D–3 guarantees the stability of the output-queues because it limits the amount of data traffic the every node receives. That is, when the total traffic that a node receives is limited, the grants generated by the node are guaranteed not to reserve future data-time-slots that are infinitely far away from the current data-time-slot. In this way, the probability that the system time reaches the schedule of the data packet scheduled the farthest in time is greater than zero. This is the probability that output-queues return to the empty state.

The drift of $V_{\mathsf{r}}\big(\mathbf{Q}_{\mathsf{r}}(n), \mathbf{Q}_{\mathsf{o}}(n)\big)$ is calculated as follows[4] .

$$\begin{aligned}
\Delta V_{\mathsf{r}} &= V_{\mathsf{r}}\big(\mathbf{Q}_{\mathsf{r}}(n+1), \mathbf{Q}_{\mathsf{o}}(n+1)\big) - V_{\mathsf{r}}\big(\mathbf{Q}_{\mathsf{r}}(n), \mathbf{Q}_{\mathsf{o}}(n)\big) \\
&= \sum_{j,i,f_n} \Big(Q_{\mathsf{o},f_n}^{(i,j)}(n+1) + Q_{\mathsf{r},f_n}^{(i,j)'}(n+1)\Big)^2 - \sum_{j,i,f_n} \Big(Q_{\mathsf{o},f_n}^{(i,j)}(n) + Q_{\mathsf{r},f_n}^{(i,j)'}(n)\Big)^2 \\
&= \sum_{j,i,f_n} \Big(Q_{o,f_n}^{(i,j)}(n+1) + Q_{r,f_n}^{(i,j)'}(n+1) - Q_{o,f_n}^{(i,j)}(n) - Q_{r,f_n}^{(i,j)'}(n)\Big)^2 \\
&\quad + 2 \sum_{j,i,f_n} \Big(\Big(Q_{o,f_n}^{(i,j)}(n) + Q_{r,f_n}^{(i,j)'}(n)\Big)\Big(Q_{o,f_n}^{(i,j)}(n+1) + Q_{r,f_n}^{(i,j)'}(n+1) - Q_{o,f_n}^{(i,j)}(n) - Q_{r,f_n}^{(i,j)'}(n)\Big)\Big)
\end{aligned}$$

Given that $Q_{o,f_n}^{(i,j)}(n+1) + Q_{r,f_n}^{(i,j)'}(n+1) = Q_{o,f_n}^{(i,j)}(n) + G_{f_n}^{(i,j)}(n) + Q_{r,f_n}^{(i,j)'}(n) - R_{f_n}^{(i,j)'}(n)$, ΔV_{r} is given as follows.

[4] This calculation is based on the proof for the stability region of distributed greedy scheduling policies under multihop traffic presented in [84].

$$\Delta V_{\mathrm{r}} = 2\sum_{j,i,f_n}\Big(\Big(Q^{(i,j)}_{o,f_n}(n) + Q^{(i,j)'}_{r,f_n}(n)\Big)\Big(G^{(i,j)}_{f_n}(n) - R^{(i,j)'}_{f_n}(n)\Big)\Big) + \sum_{j,i,f_n}\Big(G^{(i,j)}_{f_n}(n) - R^{(i,j)'}_{f_n}(n)\Big)^2$$

The drift of $V_{\mathrm{r}}\big(\mathbf{Q}_{\mathrm{r}}(n), \mathbf{Q}_{\mathrm{o}}(n)\big)$ is given as follows.

$$\begin{aligned} E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\big[\Delta V_{\mathrm{r}}\big] =& 2\sum_{j,i,f_n}\Big(\Big(Q^{(i,j)}_{o,f_n}(n) + Q^{(i,j)'}_{r,f_n}(n)\Big)E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[\Big(G^{(i,j)}_{f_n}(n) - R^{(i,j)'}_{f_n}(n)\Big)\Big]\Big) \\ &+ \sum_{j,i,f_n} E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[\Big(G^{(i,j)}_{f_n}(n) - R^{(i,j)'}_{f_n}(n)\Big)^2\Big] \end{aligned} \tag{D–4}$$

Given that the input-queues are guaranteed to be always stable according to Eq. D–1 and that $Q^{(i,j)}_{\mathrm{r},f_n}$'s output is connected to $Q^{(i,j)}_{i,f_n}$'s input only,

$$E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[G^{(i,j)}_{f_n}(n)\Big] = E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[R^{(i,j)}_{f_n}(n)\Big] \le E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[R^{(i,j)'}_{f_n}(n)\Big].$$

Therefore, the first term in the right-hand side of Eq. D–4 can be upper-bounded as follows, where $\overline{R^{(i,j)}_{f_n}}$ is the departure rate of regulator $Q^{(i,j)}_{\mathrm{r},f_n}$ if $Q^{(i,j)}_{\mathrm{r},f_n}$ is greater than zero, and C_4 is a positive constant independent of $\mathbf{Q}_{\mathrm{r}}(n)$, $\mathbf{Q}_{\mathrm{i}}(n)$, and $\mathbf{Q}_{\mathrm{o}}(n)$ such that $C_4 > \Big(\overline{R^{(i,j)'}_{f_n}} I_{Q^{(i,j)'}_{\mathrm{r},f_n}>0} - \overline{R^{(i,j)}_{f_n}}\Big) > 0$.

$$\begin{aligned} &2\sum_{j,i,f_n}\Big(\Big(Q^{(i,j)}_{o,f_n}(n) + Q^{(i,j)'}_{r,f_n}(n)\Big)E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\Big[\Big(R^{(i,j)}_{f_n}(n) - R^{(i,j)'}_{f_n}(n)\Big)\Big]\Big) = \\ &2\sum_{j,i,f_n}\Big(\Big(Q^{(i,j)}_{o,f_n}(n)+Q^{(i,j)'}_{r,f_n}(n)\Big)\Big(\overline{R^{(i,j)}_{f_n}} I_{Q^{(i,j)}_{\mathrm{r},f_n}>0} - \overline{R^{(i,j)'}_{f_n}} I_{Q^{(i,j)'}_{\mathrm{r},f_n}>0}\Big)\Big) < -2C_4\sum_{j,i,f_n}\Big(Q^{(i,j)}_{o,f_n}(n)+Q^{(i,j)'}_{r,f_n}(n)\Big) \end{aligned}$$

The second term in the right-hand side of Eq. D–4 can be upper-bounded as follows, where C_5 is a positive constant independent of $\mathbf{Q}_{\mathrm{r}}(n)$, $\mathbf{Q}_{\mathrm{i}}(n)$, and $\mathbf{Q}_{\mathrm{o}}(n)$.

$$\sum_{j,i,f_n} E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\left[\left(G_{f_n}^{(i,j)}(n) - R_{f_n}^{(i,j)'}(n)\right)^2\right] \leq \sum_{j,i,f_n} E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\left[\left(G_{f_n}^{(i,j)}(n)\right)^2 + \left(R_{f_n}^{(i,j)'}(n)\right)^2\right] < C_5$$

Therefore, the drift of $V_\mathrm{r}\big(\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{o}(n)\big)$ is upper-bounded as follows.

$$E_{\mathrm{Q_r}(n),\mathrm{Q_o}(n)}\left[\Delta V_\mathrm{r}\right] < -2C_4 \sum_{j,i,f_n}\left(Q_{o,f_n}^{(i,j)}(n) + Q_{r,f_n}^{(i,j)'}(n)\right) + C_5 \tag{D–5}$$

Finally, from Eq. D–1, Eq. D–2, and Eq. D–5, the drift of $V\big(\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{i}(n), \mathbf{Q}_\mathrm{o}(n)\big)$ can be upper-bounded as given by Eq. D–6, where C_6 is a positive constant independent of $\mathbf{Q}_\mathrm{r}(n)$, $\mathbf{Q}_\mathrm{i}(n)$, and $\mathbf{Q}_\mathrm{o}(n)$ such that $C_6 \geq C_2 + C_3 + C_5$.

$$E_{\mathrm{Q_r}(n),\mathrm{Q_i}(n),\mathrm{Q_o}(n)}\left[\Delta V\right] < -C_1 \sum_{j,i,f_n} Q_{\mathrm{i},f_n}^{(i,j)}(n) - \eta \sum_{j,i,f_n} Q_{\mathrm{o},f_n}^{(i,j)}(n) - 2C_4 \sum_{j,i,f_n}\left(Q_{o,f_n}^{(i,j)}(n) + Q_{r,f_n}^{(i,j)'}(n)\right) + C_6 \tag{D–6}$$

Therefore, the drift of $V\big(\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{i}(n), \mathbf{Q}_\mathrm{o}(n)\big)$ is guaranteed to be negative if η is guaranteed to be positive, and this condition is satisfied when Eq. D–3 holds.

In the multihop scenario, only the links that belong to at least one flow contribute to the queueing process $\mathbf{Q}_\mathrm{r}(n), \mathbf{Q}_\mathrm{i}(n), \mathbf{Q}_\mathrm{o}(n)$. Therefore, the sufficient condition for the stability of the queues given by Eq. D–3 can be expressed in terms of the active and direct 1-hop neighborhoods. That is, only the nodes in the direct 1-hop neighborhood send data to their next hop, and only the nodes in the active 1-hop neighborhoods participate in the scheduling process by granting requests, so the left-hand side of Eq. D–3 can be rewritten as follows.

$$\sum_{i \in \mathcal{S}_1^j} \lambda^{(i,j)} \Big(1 + \big|\mathcal{S}_1^i \backslash \Big(\mathcal{S}_1^j \cup j\Big)\big|\Big) = \sum_{i \in \mathcal{S}_\mathrm{d}^j} \lambda^{(i,j)} \Big(1 + \big|\mathcal{S}_\mathrm{a}^i \backslash \Big(\mathcal{S}_\mathrm{a}^j \cup j\Big)\big|\Big) =$$

$$\sum_{i \in \mathcal{S}_\mathrm{d}^j} \lambda^{(i,j)} \big|\mathcal{S}_\mathrm{a}^i \backslash \mathcal{S}_\mathrm{a}^j\big| = \lambda_\mathrm{f}^j \sum_{i \in \mathcal{S}_\mathrm{d}^j} d^{(i,j)} \big|\mathcal{S}_\mathrm{a}^i \backslash \mathcal{S}_\mathrm{a}^j\big|$$

Also, given that min-hop routing is assumed, $|\mathcal{P}_{\mathrm{max}}|$ can be at most 2 because there are two and only two links in the shortest path from any node to any of its 2-hop neighbors. These are the link from the node itself to a 1-hop neighbor that connects to the 2-hop neighbor and the link that connects the 1-hop neighbor to the 2-hop neighbor. Therefore, the bound given by Eq. D–3 is equivalent to

$$\lambda_\mathrm{f}^j \sum_{i \in \mathcal{S}_\mathrm{d}^j} d^{(i,j)} \big|\mathcal{S}_\mathrm{a}^i \backslash \mathcal{S}_\mathrm{a}^j\big| < \frac{1}{5} \quad \forall\ j \in \mathcal{N}.$$

APPENDIX E
FORMULATION OF THE STABILITY-REGION-ADAPTATION-FOR-THROUGHPUT-MAXIMIZATION PROBLEM AS A MIXED INTEGER PROGRAM WITH NON-LINEAR CONSTRAINTS

Let $\mathcal{K}(i,j)$ be the active 1-hop neighborhood of node i when all the nodes in $\mathcal{N}$ have their transmission ranges set at the maximum $r^{\max}$ (i.e., $\mathcal{K}(i,j) \triangleq \mathcal{S}_{\mathrm{a}}^{i}$ when $r^k = r^{\max} \ \forall \ k \in \mathcal{N}$).

Let $[x_{ij}]_{i,j \in \mathcal{N}}$ be the integer decision variables that indicate whether node j is within node i's transmission range (i.e., $x_{ij} = 1$ if $r^i \geq ||i,j||$, $x_{ij} = 0$ otherwise).

The minimum transmission range of node i that guarantees that none of the links used by the flows is broken is denoted by $r_{\min}^{i}$.

The problem of stability region adaptation for throughput maximization (SRA-TM) can be formulated as a mixed integer program with non-linear constraints as follows[1] .

$$
\begin{aligned}
\textit{maximize} \quad & \sum_{\lambda_n \in \mathcal{F}} \lambda_n \\
\textit{subject to} \quad & x_{ii} = 0 \ \ \forall \ \ i \in \mathcal{N} \\
& x_{ij} = 1 \ \ \text{if } i \text{ covers } j \text{ in the MinPower setup} \\
& x_{ij} = 0 \ \ \text{if } i \text{ does not cover } j \text{ in the MaxPower setup} \\
& x_{ij} \geq x_{ik} \quad \text{if} \quad ||i,j|| < ||i,k|| \\
& \lambda_n \leq \frac{1}{5 \sum_{i \in \mathcal{S}_{\mathrm{d}}^{j}} d^{(i,j)} \left(\sum_{k \in \mathcal{K}(i,j)} x_{ki} \left(1 - x_{kj}\right)\right)} \\
& \qquad\qquad\qquad\qquad \forall \ \ j \in \{p_n(m) : 2 \leq m \leq |p_n|\}
\end{aligned}
$$

The first set of constraints eliminates meaningless decision variables. The second set of constraints guarantees that none of the flows is broken by assuring the existence

[1] The terms MinPower and MaxPower used in the formulation are defined in Section 3.4.

of all the links present in the MinPower setup. The third set of constraints guarantees that the maximum transmission range of the nodes is not exceeded by any node. The fourth set of constraints guarantees that the coverage of every node is omnidirectional. The fifth set of constraints guarantees that the flow rates are within the lower-bound region given by Theorem 3.1.

The solution of the previous problem is given in terms of the decision variables $[x_{ij}]_{i,j\in\mathcal{N}}$, and the solution of the SRA-TM problem is given in terms of the transmission ranges $[r^i]_{i\in\mathcal{N}}$. Any set of transmission ranges that satisfies the solution $[x_{ij}]_{i,j\in\mathcal{N}}$ achieves the same objective-function value in the SRA-TM problem and its mixed-integer-program formulation.

APPENDIX F
CHAPTER 4'S NOTATION

Tables F-1 and F-2 summarize the notation used throughout Chapter 4.

Table F-1. Chapter 4's notation: network model

Symbol	Meaning
$\mathcal{N}^*$	The set of nodes that belong to at least one flow
$\mathcal{L}^*$	The set of links that belong to at least one flow
$\mathcal{H}$	Some subset of links, i.e., $\mathcal{H} \subseteq \mathcal{L}$
$\mathcal{M}_\mathcal{H}$	The set of all maximal schedules on $\mathcal{H}$
$Co(\mathcal{M}_\mathcal{H})$	The convex hull of $\mathcal{M}_\mathcal{H}$
$\mathcal{F}$	The set of flows
f_n	The n^{th} flow in $\mathcal{F}$
$\mathcal{P}^{f_n}$	The path of f_n
$\mathcal{P}^{f_n}_{\text{int}}$	The path of f_n without the source node
r^i	The transmission range of node i
r^i_{min}	The minimum r^i that does not break any link in $\mathcal{L}^*$
$\mathbf{r}$	The vector of transmission ranges of the nodes in $\mathcal{N}^*$
$\mathbf{r}_{\text{min}}$	The vector of minimum transmission ranges of the nodes in $\mathcal{N}^*$
$\mathbf{r}_{\text{max}}$	The vector of maximum transmission ranges of the nodes in $\mathcal{N}^*$
$\mathcal{S}^i_{\text{a}}$	The active 1-hop neighborhood of i, i.e., the 1-hop neighbors of i that are intermediate or destination nodes of at least one flow
$\mathcal{S}^i_{\text{d}}$	The direct 1-hop neighborhood of i, i.e., the 1-hop neighbors of i that send data packets to i
$\vert\vert i,j\vert\vert$	The Euclidean distance between nodes i and j
(i,j)	The link directed from node i to node j
$d^{(i,j)}$	The degree of link (i,j), i.e., the number of flows (i,j) belongs to
$\mathcal{E}^{(i,j)}$	The set of links that interfere with link (i,j)
$\mathcal{E}^{(i,j)}_{\text{r}}$	$\mathcal{E}^{(i,j)}$ as a function of $\mathbf{r}$
$\mathcal{E}^j_{\text{r}}$	The set of links that interfere with at least one incoming link of node j as a function of $\mathbf{r}$
$\mathcal{E}^{(i,j)}_{\text{max}}$	$\mathcal{E}^{(i,j)}_{\text{r}}$ evaluated at $\mathbf{r}_{\text{max}}$
$\lambda^{\mathcal{E}^{(i,j)}}$	The set of packet rates of the links in $\mathcal{E}^{(i,j)}$
$\boldsymbol{\lambda}$	The vector of packet rates of the links in $\mathcal{L}$
$\boldsymbol{\lambda}^*$	The vector of packet rates of the links in $\mathcal{L}^*$
λ^j_{max}	The highest packet rate that node j supports for each of its incoming flows that guarantees stability
$\lambda^j_{\text{max}}(\mathcal{E}^j_{\text{r}})$	λ^j_{max} as a function of $\mathcal{E}^j_{\text{r}}$
λ^{f_n}	The packet rate of flow f_n
$\lambda^{f_n}_{\text{max}}$	The highest packet rate that flow f_n supports while guaranteeing stability
$\lambda^{f_n}_{\text{max}}(\mathbf{r})$	$\lambda^{f_n}_{\text{max}}$ as a function of $\mathbf{r}$

Table F-2. Chapter 4's notation: potential game

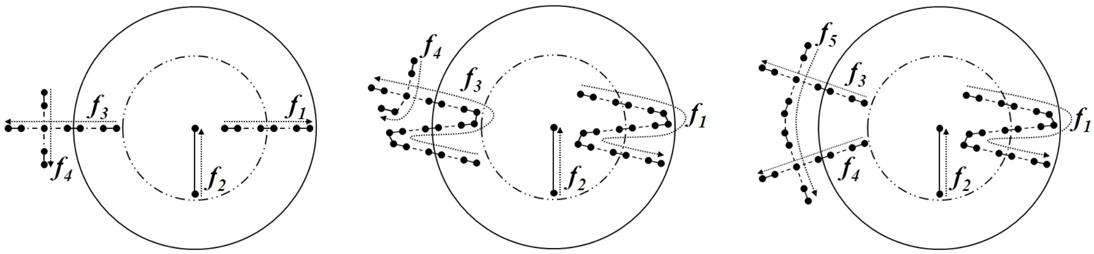
Symbol	Meaning
$\mathcal{S}^j$	The set of nodes that are able to affect λ^j_{max} with their transmission ranges (Eq. 4–4)
$\mathcal{S}^{f_n}$	The set of nodes that are able to affect $\lambda^{f_n}_{\text{max}}$ with their transmission ranges (Eq. 4–5)
$\mathcal{F}^n$	The set of flows whose highest supported packet rates are affected by any of the moves that f_n can make (Eq. 4–6)
$\mathcal{R}$	The game's action space: the set of feasible transmission ranges of the nodes controlled by the flows, i.e., the nodes in $\bigcup_{f_n \in \mathcal{F}} \mathcal{S}^{f_n}$
$\mathcal{R}_n$	The action space of f_n: the set of feasible transmission ranges of the nodes controlled by f_n, i.e., the nodes in $\mathcal{S}^{f_n}$
$\mathcal{R}_{-n}$	The set of feasible transmission ranges of the nodes not controlled by f_n, i.e., the nodes in $\left(\bigcup_{f_i \in \mathcal{F}} \mathcal{S}^{f_i}\right) \backslash \mathcal{S}^{f_n}$
$\mathbf{r}_{-n}$	The vector of transmission ranges of the nodes not controlled by f_n, i.e., the nodes in $\left(\bigcup_{f_i \in \mathcal{F}} \mathcal{S}^{f_i}\right) \backslash \mathcal{S}^{f_n}$
$\mu_n(\mathbf{r})$	The utility function of flow f_n as a function of $\mathbf{r}$ in the normalized-transport-capacity adaptation-problem (NTC-AP) game (Eq. 4–7)
$\boldsymbol{\mu}(\mathbf{r})$	The vector of utility functions $[\mu_1(\mathbf{r}), \mu_2(\mathbf{r}), ..., \mu_N(\mathbf{r})]$ in the NTC-AP game
$\mu^*_n(\mathbf{r})$	The utility function of flow f_n as a function of $\mathbf{r}$ in the linear NTC-AP (Lin-NTC-AP) game (Eq. 4–7)
$\lambda_{\text{T}}(\mathbf{r})$	An ordinal potential function (OPF) of the NTC-AP game (Theorem 4.1), and also the total throughput (Eq. 4–8)
$a^{(i,j)}_{\text{r}}$	The number of active hidden nodes of link (i,j), i.e., the number of nodes in $\mathcal{S}^i_{\text{a}} \backslash \mathcal{S}^j_{\text{a}}$, as a function of $\mathbf{r}$
$\mathbf{a}$	The vector $\left[a^{(i,j)}_{\text{r}}\right]_{(i,j) \in \mathcal{L}^*}$
$c^j(\mathbf{r})$	The contention level of node j (Eq. 4–14)
$c^{f_n}_{\text{T}}(\mathbf{r})$	The total contention experienced by f_n (Eq. 4–16)
$c^{f_n}_{\text{V}}(\mathbf{r})$	The contention variation experienced by f_n (Eq. 4–16)
$c_{\text{Lin}}(\mathbf{r})$	An OPF of the Lin-NTC-AP game, and also the total contention and contention variation experienced by all the flows (Eq. 4–17)
$c_{\text{NTC}}(\mathbf{a})$	The objective function of the NTC-AP as formulated in Eq. 4–15

APPENDIX G
PROOF OF THEOREM 4.5

In the normalized-transport-capacity adaptation-problem (NTC-AP), the goal is to minimize the maximum contention experienced by each of the flows (Eq. 4–15). Figure G-1 shows the cases in which transmission powers (TP) of active nodes determined from $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ do not minimize the maximum contention of a flow. The notation in Figure G-1 is as follows. The flow whose maximum contention is not minimized by $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ is f_1. The only TPs that are shown are the TPs of the active nodes that can be modified in order to minimize the maximum contention of f_1. The TPs shown with dashed lines are the optimal TPs of the linear NTC-AP (Lin-NTC-AP) (i.e., TPs determined from $\mathbf{a}_{\text{Lin}}^{\text{opt}}$), and the TPs shown with solid lines are the optimal TPs of the NTC-AP (i.e., TPs determined from $\mathbf{a}_{\text{NTC}}^{\text{opt}}$). In the case of one single flow whose maximum contention is not minimized by $\mathbf{a}_{\text{Lin}}^{\text{opt}}$, there are three possible TP-configurations in which the optimal Lin-NTC-AP TPs differ from the optimal NTC-AP TPs. In the general case, i.e., when the maximum contention experienced by two or more flows are not minimized by $\mathbf{a}_{\text{Lin}}^{\text{opt}}$, the TP-configuration for each of these flows corresponds to one of the three possible TP-configurations. Therefore, the three possible TP-configurations describe all the possible ways in which optimal Lin-NTC-AP TPs do not minimize the maximum contention experienced by one or more flows in the network. The first of these TP-configurations is shown in Figure G-1A, the second is shown in Figure G-1B and Figure G-1C, and the third is shown in Figure G-1D and Figure G-1E.

The following analysis is based on the following hidden nodes in Figure G-1. In Figure G-1A, Figure G-1B, Figure G-1C, only one hidden node is considered per figure. This node is the sink of f_2. In Figure G-1D and Figure G-1E, several hidden nodes are considered per figure. These nodes are the sinks of f_5, f_6, ..., $f_{5+|\mathcal{P}_{\text{int}}^{f_1}|}$.

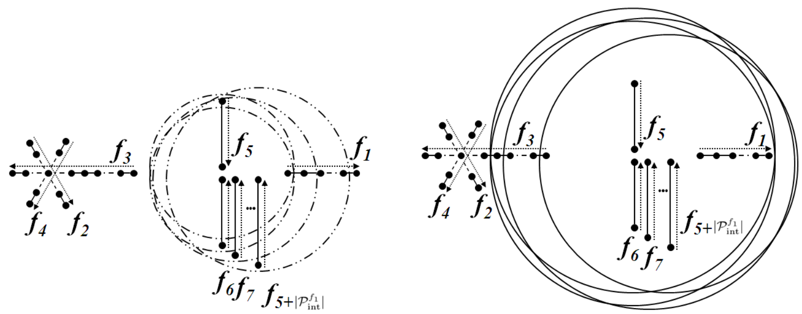
The contributions that the hidden nodes in Figure G-1A, Figure G-1B, Figure G-1C, Figure G-1D, and Figure G-1E make to the contention variation of f_1 and f_3 are denoted



A TP-configuration 1: single active node is a hidden node of a single link

B TP-configuration 2-single-flow: single active node is a hidden node of multiple links

C TP-configuration 2-multiple-flow: single active node is a hidden node of multiple links



D TP-configuration 3 (Lin-NTC-AP TPs): multiple active nodes are hidden nodes of multiple links

E TP-configuration 3 (NTC-AP TPs): multiple active nodes are hidden nodes of multiple links

Figure G-1. Suboptimal transmission-power (TP) configurations

by $\Delta_V^{f_1}$ and $\Delta_V^{f_3}$ respectively. The contribution that the hidden node in Figure G-1C makes to the contention variation of f_4 is denoted by $\Delta_V^{f_4}$. $\Delta_V^{f_1}$ is different from zero only when the TPs are determined from $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ (i.e., dashed-line TPs). When the TPs are determined from $\mathbf{a}_{\text{NTC}}^{\text{opt}}$ (i.e., solid-line TPs), the nodes are no longer hidden nodes of any of f_1's links, so they do not contribute to f_1's contention variation (i.e., $\Delta_V^{f_1} = 0$). On the other hand, $\Delta_V^{f_3}$ and $\Delta_V^{f_4}$ are different from zero only when the TPs are determined from $\mathbf{a}_{\text{NTC}}^{\text{opt}}$ (i.e., solid-line TPs). When the TPs are determined from $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ (i.e., dashed-line TPs), the nodes are no longer hidden nodes of any of the links of f_3 and f_4, so they do not contribute to the contention variation of f_3 and f_4 (i.e., $\Delta_V^{f_3} = \Delta_V^{f_4} = 0$).

It is assumed that the values taken by $\Delta_V^{f_1}$, $\Delta_V^{f_3}$, and $\Delta_V^{f_4}$ that are greater than zero meet the following inequalities. Otherwise, if this assumption is not made, the

Table G-1. Objective function values: Transmission-power configuration 1, 2-single-flow

	$\mathbf{a}_{\text{Lin}}^{\text{opt}}$	$\mathbf{a}_{\text{NTC}}^{\text{opt}}$
Lin-NTC-AP	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) + \Delta_{\text{V}}^{f_3} - \Delta_{\text{V}}^{f_1}$
NTC-AP	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - d_{\max}$

Table G-2. Objective function values: Transmission-power configuration 2-multiple-flows

	$\mathbf{a}_{\text{Lin}}^{\text{opt}}$	$\mathbf{a}_{\text{NTC}}^{\text{opt}}$
Lin-NTC-AP	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) + \Delta_{\text{V}}^{f_3}$ $+\Delta_{\text{V}}^{f_4} - \Delta_{\text{V}}^{f_1}$
NTC-AP	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - d_{\max}$

solution $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ becomes optimal. In TP-configuration 1 (Figure G-1A), TP-configuration 2-single-flow (Figure G-1B), and TP-configuration 3 (Figure G-1D and Figure G-1E), $\Delta_{\text{V}}^{f_3} > \Delta_{\text{V}}^{f_1}$, and in TP-configuration 2-multiple-flows (Figure G-1C), $\Delta_{\text{V}}^{f_3} + \Delta_{\text{V}}^{f_4} > \Delta_{\text{V}}^{f_1}$.

Tables G-1 to G-3 show the objective-function values of the Lin-NTC-AP and the NTC-AP evaluated at $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ and $\mathbf{a}_{\text{NTC}}^{\text{opt}}$ for the three TP-configurations. The values of the objective functions evaluated at $\mathbf{a}_{\text{NTC}}^{\text{opt}}$ (i.e., $c_{\text{Lin}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$ and $c_{\text{NTC}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$) are given in terms of the values of the objective functions evaluated at $\mathbf{a}_{\text{Lin}}^{\text{opt}}$ (i.e., $c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$ and $c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$). In this way, the factors that cause the difference between the objective-function values can be identified. For example, Table G-1 states that

$$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - c_{\text{Lin}}(\mathbf{a}_{\text{NTC}}^{\text{opt}}) = \Delta_{\text{V}}^{f_3} - \Delta_{\text{V}}^{f_1},$$
$$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - c_{\text{NTC}}(\mathbf{a}_{\text{NTC}}^{\text{opt}}) = d_{\max}.$$

The following analysis is divided into two parts. In the first part, the reason that the Lin-NTC-AP game reaches suboptimal solutions of the NTC-AP in the three TP configurations in Figure G-1 is proved. In the second part, the maximum difference

Table G-3. Objective function values: Transmission-power configuration 3

	$\mathbf{a}_{\text{Lin}}^{\text{opt}}$	$\mathbf{a}_{\text{NTC}}^{\text{opt}}$
Lin-NTC-AP	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) + \Delta_{\text{V}}^{f_3} - \Delta_{\text{V}}^{f_1}$
NTC-AP	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$	$c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - 2d_{\max}$

between the optimal and suboptimal solutions is calculated. Both parts are based on the effects of switching from the TP powers given by $\mathbf{a}^{\text{opt}}_{\text{Lin}}$ (i.e., dashed-line TPs) to the ones given by $\mathbf{a}^{\text{opt}}_{\text{NTC}}$ (i.e., solid-line TPs).

The Lin-NTC-AP game reaches suboptimal solutions of the NTC-AP in the three TP configurations in Figure G-1

In TP-configuration 1 (Figure G-1A and Table G-1), the dashed-line TP of the destination node of flow f_2 partially covers flow f_1 and does not cover f_3, and the solid-line TP covers f_1 completely and partially covers f_3. According to Eq. 4–17, the value of c_{Lin} changes due to changes on the total contention (i.e., $\sum_{f_n \in \mathcal{F}} c^{f_n}_{\text{T}}(\mathbf{r})$) and contention variation (i.e., $\sum_{f_n \in \mathcal{F}} c^{f_n}_{\text{V}}(\mathbf{r})$). The total contention is decreased by the value of the degree of the link of f_1 that is partially covered by the dashed-line TP, and it is increased by the value of the degree of the link of f_2 that is partially covered by the solid-line TP[1] . In the worst case scenario (i.e., when $c_{\text{NTC}}(\mathbf{a}^{\text{opt}}_{\text{Lin}}) - c_{\text{NTC}}(\mathbf{a}^{\text{opt}}_{\text{NTC}})$ is maximum), these two link degrees are equal to $d_{\max}$, so the total contention does not change. The contention variation is increased by $\Delta^{f_3}_{\text{V}} - \Delta^{f_1}_{\text{V}}$. Therefore, the total difference between $c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{NTC}})$ and $c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{Lin}})$ is $\Delta^{f_3}_{\text{V}} - \Delta^{f_1}_{\text{V}}$ (i.e., $c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{Lin}}) - c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{NTC}}) = \Delta^{f_3}_{\text{V}} - \Delta^{f_1}_{\text{V}}$). This result is the reason that, for TP-configuration 1, the Lin-NTC-AP game selects the dashed-line TP, so it reaches a suboptimal equilibrium that does not minimize c_{NTC}.

In TP-configuration 2-single-flow (Figure G-1B and Table G-1), the total contention is decreased by the values of the degrees of the two links of f_1 that are partially covered by the dashed-line TP, and it is increased by the values of the degrees of the two links of f_3 that are partially covered by the solid-line TP. In the worst-case scenario, the link degrees are equal to $d_{\max}$, so the total contention does not change. The contention variation is increased by $\Delta^{f_3}_{\text{V}} - \Delta^{f_1}_{\text{V}}$. Therefore, the total difference between $c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{NTC}})$ and $c_{\text{Lin}}(\mathbf{a}^{\text{opt}}_{\text{Lin}})$ is $\Delta^{f_3}_{\text{V}} - \Delta^{f_1}_{\text{V}}$. This result is the reason that, for TP-configuration 2-single-flow,

[1] See the example in Figure 4-2 for the explanation.

the Lin-NTC-AP game selects the dashed-line TP, so it reaches a suboptimal equilibrium that does not minimize c_{NTC}.

In TP-configuration 2-multiple-flows (Figure G-1C and Table G-2), the total contention is decreased by the values of the degrees of the two links of f_1 that are partially covered by the dashed-line TP, and it is increased by the values of the degrees of the link of f_3 and the link of f_4 that are partially covered by the solid-line TP. In the worst-case scenario, the link degrees are equal to d_{max}, so the total contention does not change. The contention variation is increased by $\Delta_{\text{V}}^{f_3} + \Delta_{\text{V}}^{f_4} - \Delta_{\text{V}}^{f_1}$. Therefore, the total difference between $c_{\text{Lin}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$ and $c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$ is $\Delta_{\text{V}}^{f_3} + \Delta_{\text{V}}^{f_4} - \Delta_{\text{V}}^{f_1}$. This result is the reason that, for TP-configuration 2-multiple-flows, the Lin-NTC-AP game selects the dashed-line TP, so it reaches a suboptimal equilibrium that does not minimize c_{NTC}.

In TP-configuration 3 (Figure G-1D, Figure G-1E, and Table G-2), the total contention is decreased by the values of the degrees of all the links of f_1 that are partially covered by the dashed-line TPs, and it is increased by the values of the degrees of the links of f_3 that are partially covered by the solid-line TPs. In the worst-case scenario, the link degrees are equal to d_{max}. Given that there are a total of $\mathcal{P}_{\text{int}}^{f_1} + 1$ TPs that cover partially the links of f_1 and f_3, the total decrease and increase are each equal to $(\mathcal{P}_{\text{int}}^{f_1} + 1)d_{\text{max}}$. Therefore, the contention does not change. The contention variation is increased by $\Delta_{\text{V}}^{f_3} - \Delta_{\text{V}}^{f_1}$. Therefore, the total difference between $c_{\text{Lin}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$ and $c_{\text{Lin}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$ is $\Delta_{\text{V}}^{f_3} - \Delta_{\text{V}}^{f_1}$. This result is the reason that, for TP-configuration 3, the Lin-NTC-AP game selects the dashed-line TPs, so it reaches a suboptimal equilibrium that does not minimize c_{NTC}.

The maximum difference between the optimal solution of the NTC-AP and the solution calculated by the Lin-NTC-AP game is bounded

The suboptimal equilibrium of the Lin-NTC-AP game reaches a value for c_{NTC} that is greater than the optimal (i.e., minimum) by a difference which is upper-bounded as follows.

In TP-configuration 1 (Figure G-1A and Table G-1), when the TP is changed from the dashed-line TP to the solid-line TP, the maximum contention experienced by f_1 is decreased by the value of the degree of the link of f_1 that is partially covered by the dashed-line TP. The maximum contention experienced by f_3 does not change because f_4 is the cause of the maximum contention of f_3 when the dashed-line TP is set, and when the solid-line TP is set, this maximum contention is not exceeded, i.e., when the solid-line TP is set, the nodes of f_3 whose incoming links are partially covered by the TPs of f_4 and f_2 experience the same maximum contention, which is equal to $2d_{\text{max}}$. This is shown in Figure G-2 in which the nodes of f_3 with maximum contention are highlighted and only the TPs that cause their contention are included. Therefore, given that the maximum contention experienced by f_1 is decreased by a value equal to a link degree and the maximum contention experienced by f_3 is not varied, $c_{\text{NTC}}(\mathbf{a}^{\text{opt}}_{\text{Lin}})$ is higher than $c_{\text{NTC}}(\mathbf{a}^{\text{opt}}_{\text{Lin}})$ by at most d_{max}.

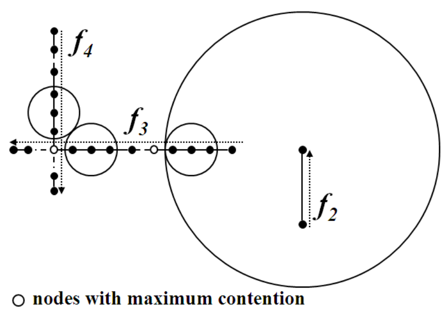


Figure G-2. Nodes of f_3 with maximum contention in TP-configuration 1

In TP-configuration 2-single-flow (Figure G-1B and Table G-1), the maximum contention experienced by f_1 is decreased by the value of the degree of one of the two links of f_1 that are partially covered by the dashed-line TP. The maximum contention experienced by f_3 does not change because f_4 is the cause of the maximum contention of f_3 when the dashed-line TP is set, and when the solid-line TP is set, this maximum contention is not exceeded, i.e., when the solid-line TP is set, the nodes of f_3 whose incoming links are partially covered by the TPs of f_4 and f_2 experience the same maximum contention, which is equal to $2d_{\max}$. Therefore, given that the maximum contention experienced by f_1 is decreased by a value equal to one link degree and the maximum contention experienced by f_3 is not varied, $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{Lin}})$ is higher than $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{Lin}})$ by at most $d_{\max}$.

In TP-configuration 2-multiple-flow (Figure G-1C and Table G-2), the maximum contention experienced by f_1 is decreased by the value of the degree of one of the two links of f_1 that are partially covered by the dashed-line TP. The maximum contentions experienced by f_3 and f_4 do not change because f_5 is the cause of the maximum contentions of f_3 and f_4 when the dashed-line TP is set, and when the solid-line TP is set, these maximum contentions are not exceeded, i.e., when the solid-line TP is set, the nodes of f_3 and f_4 whose incoming links are partially covered by the TPs of f_5 and f_2 experience the same maximum contention, which is equal to $2d_{\max}$. Therefore, given that the maximum contention experienced by f_1 is decreased by a value equal to one link degree and the maximum contentions experienced by f_3 and f_4 are not varied, $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{Lin}})$ is higher than $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{Lin}})$ by at most $d_{\max}$.

In TP-configuration 3 (Figure G-1D, Figure G-1E, and Table G-3), the maximum contention experienced by f_1 is decreased by twice the value of the degree of the link of f_1 that is partially covered by two dashed-line TPs. The maximum contention experienced by f_3 does not change because f_2 and f_4 are the cause of the maximum contention of f_3 when the dashed-line TPs are set, and when the solid-line TPs are set, this maximum contention is not exceeded, i.e., when the solid-line TPs are set, the two nodes of f_3 whose incoming links are partially covered by the TPs of f_2 and f_4 and by the TPs of f_5 and f_6 respectively experience the same maximum contention, which is equal to $3d_{\max}$. Therefore, given that the maximum contention experienced by f_1 is decreased by a value equal to two link degrees and the maximum contention experienced by f_3 is not varied, $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{Lin}})$ is higher than $c_{\mathrm{NTC}}(\mathbf{a}^{\mathrm{opt}}_{\mathrm{NTC}})$ by at most $2d_{\max}$.

In the general case, i.e., when the maximum contention experienced by two or more flows are not minimized by $\mathbf{a}_{\text{Lin}}^{\text{opt}}$, the TP-configuration for each of these flows corresponds to one of the three possible TP-configurations. The maximum possible difference between $c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}})$ and $c_{\text{NTC}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$ is achieved when TP-configurations 1 or 2 single-flow are repeated as many times as possible. The reason is that these configurations are the ones that use the less number of flows. Therefore, they are the ones that can be replicated the highest number of times. For each replication, the $c_{\text{NTC}}(\mathbf{a}_{\text{Lin}}^{\text{opt}}) - c_{\text{NTC}}(\mathbf{a}_{\text{NTC}}^{\text{opt}})$ value is increased by d_{max}. This is shown in Figure G-3 for TP-configuration 2-single-flow which is repeated the maximum number of times, i.e., $\frac{|\mathcal{F}|-2}{2}$ times.

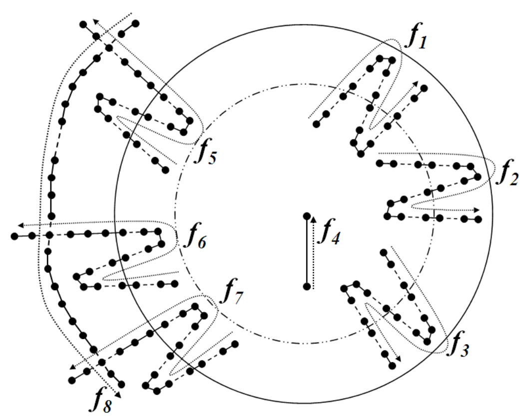


Figure G-3. Worst-case scenario

REFERENCES

[1] “IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed Broadband Wireless Access Systems.” IEEE Std. 802.16, 2004.

[2] “Network Simulator 2.” Network Simulation Software, 2009.

[3] “The OPNET Modeler.” 2009.

[4] “BARON Global Optimization Software.” 2010.

[5] Asmussen, Soren. *Applied Probability and Queues*. New York, NY: Springer, 2003.

[6] Bayer, Nico, Xu, Bangnan, Rakocevic, Veselin, and Habermann, Joachim. “Improving the Performance of the Distributed Scheduler in IEEE 802.16 Mesh Networks.” *Proc. 65th Vehicular Technology Conference, 2007 (VTC2007-Spring)*. IEEE, 2007, 1193–1197.

[7] Behzad, Arash and Rubin, Izhak. “Optimum Integrated Link Scheduling and Power Control for Multihop Wireless Networks.” *IEEE Transactions on Vehicular Technology* 56 (2007): 194–205.

[8] Belghith, Aymen and Nuaymi, Loutfi. “Design and implementation of a QoS-included WiMAX module for NS-2 simulator.” *Proc. 1st International Conference on Simulation Tools and Techniques (Simutools’08)*. ICST, 2008, 1–8.

[9] Bicket, John, Aguayo, Daniel, Biswas, Sanjit, and Morris, Robert. “Architecture and evaluation of an unplanned 802.11b mesh network.” *Proc. 11th Annual International Conference on Mobile Computing and Networking (Mobicom’05)*. ACM, 2005, 31–42.

[10] Birand, Berk, Chudnovsky, Maria, Ries, Bernard, Seymour, Paul, Zussman, Gil, and Zwols, Yori. “Analyzing the Performance of Greedy Maximal Scheduling via Local Pooling and Graph Theory.” *Proc. 29th Conference on Computer Communications. IEEE (INFOCOM’10)*. San Diego, CA, USA, 2010, 1–9.

[11] Brzezinski, Andrew, Zussman, Gil, and Modiano, Eytan. “Enabling Distributed Throughput Maximization in Wireless Mesh Networks: A Partitioning Approach.” *Proc. 12th Annual International Conference on Mobile Computing and Networking (MobiCom’06)*. Los Angeles, CA, 2006, 26–37.

[12] Bui, L., Eryilmaz, A., and Srikant, R. “Asynchronous Congestion Control in Multi-Hop Wireless Networks with Maximal Matching-Based Scheduling.” *IEEE/ACM Transactions on Networking* 16 (2008): 826–839.

[13] Cao, Min, Zhang, Qian, and Wang, Xiaodong. “Analysis of IEEE 802.16 Mesh Mode Scheduler Performance.” *IEEE Transactions on Wireless Communication* 6 (2007): 1455–1464.

[14] Chaporkar, P., Kar, K., and Sarkar, S. "Throughput Guarantees through Maximal Scheduling in Multi-hop Wireless Networks." *Proc. 43rd Annual Allerton Conference on Communications, Control, and Computing (Allerton'05)*. Urbana-Champaign, IL, 2005, 1–11.

[15] ———. "Achieving Queue-length Stability Through Maximal Scheduling in Wireless Networks." *Proc. Information Theory and Applications: Inaugural Workshop (ITA'06)*. San Diego, CA, 2006, 1–4.

[16] Chen, Lien Wu, Tseng, Yu Chee, Wang, You Chiun, Wang, Da Wei, and Wu, Jan Jan. "Exploiting Spectral Reuse in Routing, Resource Allocation, and Scheduling for IEEE 802.16 Mesh Networks." *IEEE Transactions on Vehicular Technology* 58 (2009).1: 301–313.

[17] Chen, Lijun, Low, Steven H., Chiang, Mung, and Doyle, John C. "Cross-layer Congestion Control, Routing and Scheduling in Ad Hoc Wireless Networks." *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'06)*. Barcelona, Spain, 2006, 1–13.

[18] Cheng, Shin-Ming, Lin, Phone, Huang, Di-Wei, and Yang, Shun-Ren. "A study on distributed/centralized scheduling for wireless mesh network." *Proc. International Conference on Wireless Communications and Mobile Computing, 2006 (IWCMC'06)*. ACM, 2006, 599–604.

[19] Chiang, Mung. "Balancing Transport and Physical Layers in Wireless Multihop Networks: Jointly Optimal Congestion Control and Power Control." *IEEE Journal on Selected Areas in Communications* 23 (2005): 104–116.

[20] Chiang, Mung, Tan, Chee Wei, Palomar, Daniel P., O'Neill, Daniel, and Julian, David. "Power Control by Geometric Programming." *IEEE Transactions on Wireless Communications* 6 (2007): 2640–2651.

[21] Cicconetti, Claudio, Akyildiz, Ian F., and Lenzini, Luciano. "FEBA: A Bandwidth Allocation Algorithm for Service Differentiation in IEEE 802.16 Mesh Networks." *IEEE/ACM Trans. Netw.* 17 (2009).3: 884–897.

[22] ———. "WiMsh: a simple and efficient tool for simulating IEEE 802.16 wireless mesh networks in ns-2." *Proc. 2nd International Conference on Simulation Tools and Techniques (Simutools'09)*. ICST, 2009, 1–10.

[23] Cicconetti, Claudio, Erta, Alessandro, Lenzini, Luciano, and Mingozzi, Enzo. "Performance Evaluation of the Mesh Election Procedure of IEEE 802.16/WiMAX." *Proc. 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems (MSWiM'07)*. ACM, 2007, 323–327.

[24] Cicconetti, Claudio, Gardellin, Vanessa, Lenzini, Luciano, and Mingozzi, Enzo. "End-to-End Bandwidth Reservation in IEEE 802.16 Mesh Networks." *Proc. Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007 (MASS'07)*. IEEE, 2007, 1–6.

[25] Gao, Yan, Zeng, Zheng, and Kumar, P. R. "Joint Random Access and Power Selection for Maximal Throughput in Wireless Networks." *Proc. 29th Conference on Computer Communications. IEEE (INFOCOM'10)*. San Diego, CA, USA, 2010, 1–5.

[26] Gupta, A., Lin, X., and Srikant, R. "Low-Complexity Distributed Scheduling Algorithms for Wireless Networks." *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*. Kobe, Japan, 1997, 1631–1639.

[27] Gupta, Gagan Raj and Shroff, Ness B. "Practical Scheduling Schemes with Throughput Guarantees for Multi-Hop Wireless Networks." *Computer Networks* 54 (2010): 766–780.

[28] Gupta, Piyush and Kumar, P. R. "The Capacity of Wireless Networks." *IEEE Trans. Inf. Theory* 46 (2000).2: 388–404.

[29] Hajek, B. and Sasaki, G. "Link Scheduling in Polynomial Time." *IEEE Transactions on Information Theory* 34 (1988): 910–917.

[30] Ho, Ivan Wang-Hei and Liew, Soung Chang. "Impact of Power Control on Performance of IEEE 802.11 Wireless Networks." *IEEE Transactions on Mobile Computing* 6 (2007): 1245–1258.

[31] Hu, Honglin, Zhang, Yan, and Chen, Hsiao-Hwa. "An Effective QoS Differentiation Scheme for Wireless Mesh Networks." *IEEE Network* 22 (2008).1: 66–73.

[32] Huang, Shiang-Ming, Sung, Ya-Chin, Wang, Shie-Yuan, and Lin, Yi-Bing. "NCTUns simulation tool for WiMAX modeling." *Proc. 3rd International Conference on Wireless Internet (WICON'07)*. ICST, 2007, 1–6.

[33] Jiang, Li Bin and Liew, Soung Chang. "Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks." *IEEE Transactions on Mobile Computing* 7 (2008): 34–49.

[34] Jiang, Libin and Walrand, Jean. "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks." *IEEE/ACM Transactions on Networking* 18 (2010): 960–972.

[35] Joo, C. "A Local Greedy Scheduling Scheme with Provable Performance Guarantee." *Proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'08)*. Hong Kong SAR, China, 2008, 111–120.

[36] Joo, C. and Shroff, N. "Performance of Random Access Scheduling Schemes in Multi-hop Wireless Networks." *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97)*. Kobe, Japan, 1997, 19–27.

[37] Joo, Changhee, Lin, Xiaojun, and Shroff, Ness B. "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multihop Wireless Networks." *IEEE/ACM Transactions on Networking* 17 (2009): 1132–1145.

[38] Karnik, Aditya, Iyer, Aravind, and Rosenberg, Catherine. "Throughput-Optimal Configuration of Fixed Wireless Networks." *IEEE/ACM Transactions on Networking* 16 (2008): 1161–1174.

[39] Kawadia, Vikas and Kumar, P. R. "Principles and Protocols for Power Control in Wireless Ad Hoc Networks." *IEEE Journal on Selected Areas in Communications* 23 (2005): 76–88.

[40] Kim, Bong Chan, Kwak, Dong Gu, Lee, Heecheol Song, and Ma, Joong Soo. "An adaptive holdoff algorithm based on node state for IEEE 802.16 mesh mode with coordinated distributed scheduling." *Proc. 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008 (PIMRC'08)*. IEEE, 2008, 1–5.

[41] Kim, Tae-Suk, Lim, Hyuk, and Hou, Jennifer C. "Understanding and Improving the Spatial Reuse in Multihop Wireless Networks." *IEEE Transactions on Mobile Computing* 7 (2008): 1200–1212.

[42] Kong, Peng-Yong, Wang, Haiguang, Ge, Yu, Ang, Chee-Wei, Pathmasuntharam, J.S., Su, Wen, Zhou, Ming-Tuo, and Harada, H. "Distributed Adaptive Time Slot Allocation for WiMAX Based Maritime Wireless Mesh Networks." *Proc. Wireless Communications and Networking Conference, 2009 (WCNC'09)*. IEEE, 2009, 1–6.

[43] Kuran, Mehmet S., Gur, Gurkan, Tugcu, Tuna, and Alagoz, Fatih. "Cross-layer routing-scheduling in IEEE 802.16 mesh networks." *Proc. International Conference on Mobile Wireless Middleware, Operating Systems, and Applications (Mobilware'08)*. ICST, 2008, 1–6.

[44] Le, Long Bao and Mazumdar, Ravi R. "Control of Wireless Networks with Flow Level Dynamics under Constant Time Scheduling." *Wireless Networks* 16 (2010): 1355–1372.

[45] Leconte, Mathieu, Ni, Jian, and Srikant, Rayadurgam. "Improved Bounds on the Throughput Efficiency of Greedy and Maximal Scheduling in Wireless Networks." *Proc. 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'09)*. Louisiana, LA, USA, 2009, 165–174.

[46] Li, Bo, Boyaci, Cem, and Xia, Ye. “A Refined Performance Characterization of Longest-Queue-First Policy in Wireless Networks.” *Proc. 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc’09)*. Louisiana, LA, USA, 2009, 65–74.

[47] Li, Yun, Wei, Dengyu, Zhuang, Hongcheng, Wang, Haibao, and Wang, Ping. “A New Congestion Control Method for IEEE 802.16 Mesh Mode.” *Proc. International Conference on Communication Software and Networks, 2009 (ICCSN’09)*. IEEE, 2009, 726–730.

[48] Lin, Hao-Min, Chen, Whai-En, and Chao, Han-Chieh. “A Dynamic Minislot Allocation Scheme Based on IEEE 802.16 Mesh Mode.” *Proc. Second International Conference on Future Generation Communication and Networking, 2008 (FGCN’08)*. IEEE, 2008, 288–293.

[49] Lin, X. and Shroff, N. “The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks.” *IEEE/ACM Transactions on Networking* 14 (2006): 302–315.

[50] Lin, Xiaojun and Rasool, Shahzada B. “Constant-Time Distributed Scheduling Policies for Ad Hoc Networks.” *Proc. IEEE Conference on Decision and Control (CDC’06)*. San Diego, CA, 2006, 1258–1263.

[51] ———. “Distributed and Provably Efficient Algorithms for Joint Channel-Assignment, Scheduling, and Routing in Multichannel Ad Hoc Wireless Networks.” *IEEE/ACM Transactions on Networking* 17 (2009): 1874–1887.

[52] Liu, Shenghai, Feng, Suili, Ye, Wu, , and Zhuang, Hongcheng. “Slot Allocation Algorithms in Centralized Scheduling Scheme for IEEE 802.16 based Wireless Mesh Networks.” *Computer Commununications* 32 (2009).5: 943–953.

[53] Monderer, Dov and Shapley, Lloyd S. “Potential Games.” *Games and Economic Behavior* 14 (1996): 124–143.

[54] Monks, Jeffrey P., Bharghavan, Vaduvur, and mei W. Hwu, Wen. “A Power Controlled Multiple Access Protocol for Wireless Packet Networks.” *Proc. 20th Conference on Computer Communications. IEEE (INFOCOM’01)*. Anchorage, AK, 2001, 219–228.

[55] Msadaa, Ikbal C., Filali, Fethi, and Kamoun, Farouk. “An 802.16 model for NS2 simulator with an integrated QoS architecture.” *Proc. 1st International Conference on Simulation Tools and Techniques (Simutools’08)*. ICST, 2008, 1–10.

[56] Muqattash, Alaa and Krunz, Marwan. “Power Controlled Dual Channel (PCDC) Medium Access Protocol for Wireless Ad Hoc Networks.” *Proc. 22nd Conference on Computer Communications. IEEE (INFOCOM’03)*. San Francisco, CA, 2003, 470–480.

[57] ———. “A Single-Channel Solution for Transmission Power Control in Wireless Ad Hoc Networks.” *Proc. 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04)*. Roppongi, Japan, 2004, 210–221.

[58] Neely, Michael J., Modiano, Eytan, and Rohrs, Charles E. “Dynamic Power Allocation and Routing for Time-Varying Wireless Networks.” *IEEE Transactions on Selected Areas in Communication* 23 (2005): 89–103.

[59] Papadimitriou, C. and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*. Upper Saddle River, NJ: Prentice-Hall, 1982.

[60] Penttinen, Aleksi, Koutsopoulos, Iordanis, and Tassiulas, Leandros. “Low-Complexity Distributed Fair Scheduling for Wireless Multi-Hop Networks.” *Proc. Workshop on Resource Allocation in Wireless Networks (RAWNET'06)*. Boston, MA, 2006, 1–6.

[61] Sanghavi, S., Bui, L., and Srikant, R. “Distributed Link Scheduling with Constant Overhead.” *Proc. International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'07)*. San Diego, CA, 2007, 313–324.

[62] Santi, Paolo and Blough, Douglas M. “The Critical Transmitting Range for Connectivity in Sparse Wireless Ad Hoc Networks.” *IEEE Transactions on Mobile Computing* 2 (2003): 25–39.

[63] Sarkar, S., Chaporkar, P., and Kar, K. “Fairness and Throughput Guarantees with Maximal Scheduling in Wireless Networks.” *Proc. 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'06)*. Boston, MA, 2006, 1–13.

[64] Sayenko, A., Alanen, O., Martikainen, H., Tykhomyrov, V, Puchko, O., and Hamalainen, T. “WINSE: WiMAX NS-2 extension.” *Proc. 2nd International Conference on Simulation Tools and Techniques (Simutools'09)*. ICST, 2009, 1–10.

[65] Sharma, G., Joo, C., and Shroff, N. “Distributed Scheduling Schemes for Throughput Guarantees in Wireless Networks.” *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton'06)*. Monticello, IL, 2006, 1–10.

[66] Sharma, G., Mazumdar, R., and Shroff, N. “On the Complexity of Scheduling in Wireless Networks.” *Proc. Annual International Conference on Mobile Computing and Networking (MobiCom'06)*. Los Angeles, CA, 2006, 227–238.

[67] Sridhar, Adarsh and Ephremides, Anthony. “Energy Optimization in Wireless Broadcasting through Power Control.” *Ad Hoc Networks* 6 (2008): 155–167.

[68] Tassiulas, L. "Linear complexity algorithms for maximum throughput in radio networks and input queued switches." *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'98)*. San Francisco, CA, 1998, 533–539.

[69] Tassiulas, L. and Ephremides, A. "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks." *IEEE Transactions Automatic Control* 37 (1992): 1936–1948.

[70] Vejarano, Gustavo and McNair, Janise. "An intelligent wireless mesh network backbone." *Proc. 3rd International Conference on Wireless Internet (WICON'7)*. Austin, TX, 2007, 1–5.

[71] ———. "Stability Analysis of Reservation-Based Distributed Scheduling Policies in Wireless Networks." IEEE Transactions on Parallel and Distributed Systems, 2009. Accepted for publication.

[72] ———. "Modeling and Evaluation of IEEE 802.16 Wireless Mesh Networks with Distributed Scheduling." Tech. rep., Wireless and Mobile Systems (WAMS) Laboratory, Department of Electrical and Computer Engineering, University of Florida, 2010.

[73] ———. "Queue-stability-based transmission power control in wireless multihop networks." *Proc. Global Communications Conference, IEEE (GLOBECOM'10)*. Miami, FL, 2010, 1–5.

[74] ———. "Reservation-based distributed scheduling in wireless networks." *Proc. 11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'10)*. Montreal, QC, Canada, 2010, 1–9.

[75] ———. "WiMAX-RBDS-Sim: An OPNET Simulation Framework for IEEE 802.16 Mesh Networks." *Proc. 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools'10)*. Torremolinos, Malaga, Spain, 2010, 1–10.

[76] Vejarano, Gustavo, Wang, Dexiang, Dubey, Ritwik, and McNair, Janise. "Flow-Level Collaboration in Wireless Networks for Throughput Maximization." IEEE Transactions on Selected Areas in Communication, 2011. Submitted for publication.

[77] Vejarano, Gustavo, Wang, Dexiang, and McNair, Janise. "Stability Region Adaptation using Transmission Power Control for Transport Capacity Optimization in IEEE 802.16 Wireless Mesh Networks." Computer Networks (Elsevier), 2010. Accepted for publication.

[78] Wang, Shie-Yuan, Lin, Chih-Che, Chu, Han-Wei, Hsu, Teng-Wei, and Fang, Ku-Han. "Improving the Performance of Distributed Coordinated Scheduling in IEEE 802.16 Mesh Networks." *IEEE Transactions on Vehicular Technology* 4 (2008): 2531–2547.

[79] Wang, Shie-Yuan, Lin, Chih-Che, and Fang, Ku-Han. “Improving the Data Scheduling Efficiency of the IEEE 802.16(d) Mesh Network.” *Proc. Global Telecommunications Conference, 2008 (Globecom'08)*. IEEE, 2008, 1–5.

[80] Wang, Shie-Yuan, Lin, Chih-Che, Fang, Ku-Han, and Hsu, Tens-Wei. “Facilitating the Network Entry and Link Establishment Processes of IEEE 802.16 Mesh Networks.” *Proc. Wireless Communications and Networking Conference, 2007 (WCNC'07)*. IEEE, 2007, 1842–1847.

[81] Wang, Wei, Srinivasan, Vikram, and Chua, Kee-Chaing. “Power Control for Distributed MAC Protocols in Wireless Ad Hoc Networks.” *IEEE Transactions on Mobile Computing* 7 (2008): 1169–1183.

[82] Wong, Tan. “Asymptotic CFAR Network Stability Test.” Tech. rep., Wireless Information Networking Group (WING), Department of Electrical and Computer Engineering, University of Florida, 2009.

[83] Wu, X. and Srikant, R. “Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks with Node-Exclusive Spectrum Sharing.” *Proc. IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC'05)*. Seville, Spain, 2005, 5342–5347.

[84] Wu, X., Srikant, R., and Perkins, J. “Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks.” *IEEE Transactions on Mobile Computing* 6 (2007): 595–605.

[85] Zhang, Yan, Zheng, Jun, and Li, Wei. “A Simple and Effective QoS Differentiation Scheme in IEEE 802.16 WiMAX Mesh Networking.” *Proc. Wireless Communications and Networking Conference, 2007 (WCNC'07)*. IEEE, 2007, 3216–3220.

[86] Zhang, Yan, Zhou, Mingtuo, Xiao, Shaoqiu, and Fujise, Masayuki. “An Effective QoS Scheme in WiMAX Mesh Networking for Maritime ITS.” *Proc. 6th International Conference on ITS Telecommunications*. IEEE, 2006, 612–616.

[87] Zhu, Hua, Tang, YatKwan, and Chlamtac, Imrich. “Unified Collision-Free Coordinated Distributed Scheduling (CF-CDS) in IEEE 802.16 Mesh Networks.” *IEEE Transactions on Wireless Communications* 7 (2008).10: 3889–3903.

[88] Ziller, Andreas, Mogre, Parag S., Hollick, Matthias, and Schwingenschlogl, Christian. “Reliable Broadcast Mechanism for the IEEE 802.16 Mesh Extension.” *Globecom Workshops*. IEEE, 2008, 1–5.

[89] Zussman, Gil, Brzezinski, Andrew, and Modiano, Eytan. “Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks.” *Proc. 27th Conference on Computer Communications. IEEE (INFOCOM'08)*. Phoenix, AZ, 2008, 1139–1147.

BIOGRAPHICAL SKETCH

Gustavo Vejarano earned his Bachelor of Science (BS) degree from Universidad del Valle, Cali, Colombia in 2005, and his Master of Science (MS) and Doctor of Philosophy (PhD) degrees from University of Florida, Gainesville, Florida, United States in 2009 and 2011 respectively. Gustavo's research interests are Wireless Multihop Networks (WMN), Cyber-Physical Systems: Adaptation Mechanisms for WMNs to Human Activity and Behavior, Self-Organized Communication Networks, Cognitive Radios and Networking, Multimedia Communications, Cross-Layer Design, and 3rd and 4th Generation Cellular Networks. He has been member of the Wireless and Mobile Systems Laboratory, under the direction of Dr. Janise McNair, since 2007.