4.2.2 Rate Region for Uplink Scenario

In the uplink scenario, suppose that nodes 1 and 2 both need to transmit information to node B. Let r_1 and r_2 denote the rate of reliable information delivery from nodes 1 and 2 to node B, respectively. Also let P_{1max} and P_{2max} be the maximum power used by nodes 1 and 2, respectively, whereas let P_1 and P_2 be the actual power used by each of these nodes ($P_1 \leq P_{1max}$ and $P_2 \leq P_{2max}$). In this case, similar to the downlink scenario, we will consider three approaches:

- *Simple approach:* A simple approach is for node B to treat signal from node 2 as noise when decoding the information from node 1, and vice-versa. Thus, in this case,

$$r_1 = W \log\left(1 + \frac{P_1\, g_{1B}}{P_2 g_{2B} + N_0 W}\right) \tag{4.3}$$

$$r_2 = W \log\left(1 + \frac{P_2\, g_{2B}}{P_1 g_{1B} + N_0 W}\right) \tag{4.4}$$

- *Time-sharing:* In this case, at any given time, exactly one of nodes 1 and 2 transmits. Thus, the two transmissions will be orthogonal, and not pose interference to each other. It should be easy to see that, if α fraction of time is given to node 1, the following rates can be obtained:

$$r_1 = \alpha W \log\left(1 + \frac{(P_1/\alpha)\, g_{1B}}{N_0 W}\right) \tag{4.5}$$

$$r_2 = (1-\alpha) W \log\left(1 + \frac{(P_2/(1-\alpha))\, g_{2B}}{N_0 W}\right) \tag{4.6}$$

  In the above expression, the power used by node 1 is P_1/α but only for α fraction of the time, resulting in the average power over the entire duration being P_1. Thus, in this case, P_{1max} is treated as a constraint on average power, rather than the maximum power. Similarly, the power used by node 2 for $1-\alpha$ fraction of time is $P/(1-\alpha)$. Note that when $\alpha = 0$, we will have $r_1 = 0$, and when $\alpha = 1$, r_2 will be 0.

  The same result can also be achieved by splitting the bandwidth between the two transmitters, instead of splitting time. In particular, if nodes 1 and 2 use bandwidth αW and $(1-\alpha)W$, the above rates will be obtained. In this case, nodes 1 and 2 transmit all the time, using power P_1 and P_2, respectively. The noise at nodes 1 and 2 will now be $\alpha N_0 W$ and $(1-\alpha) N_0 W$, respectively, again yielding the above rate equations.

- *Successive interference cancellation:* In successive interference cancellation, we have two choices, depending on which signal is first decoded by node B. Suppose that node B first decodes the signal from node 1, considering the signal from node 2 as interference. In this case, we will have

$$r_1 = W \ \log\left(1 + \frac{P_1\, g_{1B}}{P_2 g_{2B} + N_0 W}\right)$$

Having determined the signal from node 1, now node B can cancel the interference posed to the reception from node 2, and obtain rate

$$r_2 = W \ \log\left(1 + \frac{P_2\, g_{2B}}{N_0 W}\right)$$

Observe that, in this case,

$$r_1 + r_2 = W \ \log\left(1 + \frac{P_1\, g_{1B} + P_2\, g_{2B}}{N_0 W}\right)$$

Alternatively, the receiver can decode the signal from node 2 first, and then the signal from node 1. It is easy to see that, in this case as well, we will obtain the above expression for $r_1 + r_2$.

4.2.3 Operating Points

All rate vectors in a rate region are said to be *feasible* rate vectors. In the above discussion, we saw that some of the rate vectors correspond to *operating points*, while others do not. The rate vector corresponding to an operating point can be achieved within a given slot, by appropriately choosing parameters such as power and allocated bandwidth. On the other hand, some of the rate vectors cannot be achieved in any one slot, but can be achieved as an average over longer time intervals (essentially, by time-sharing between different operating points).

Let us denote by $\widetilde{\Gamma}$ the set of all rate vectors corresponding to the feasible operating points. Physical layer constraints will determine the set of feasible operating points. $\widetilde{\Gamma}$ may not necessarily be convex. However, as we have seen before, by using time-sharing between the operating points in $\widetilde{\Gamma}$, we can obtain the convex-hull of $\widetilde{\Gamma}$. This convex hull is the rate region for the network, and will be denoted as Γ.

A rate vector $\mathbf{r}$ in a set of rate vectors is said to be *maximal* in that set if no other rate vector in that set dominates $\mathbf{r}$. Thus, $\widetilde{\Gamma}$ will contain some maximal rate vectors, while other rate vectors may not be maximal. If performance metric of interest is an increasing function of flow throughputs, then clearly it is always desirable to use a maximal rate vector. An operating point that yields a maximal rate vector will be said to be a maximal operating point.

4.2.4 Practical Considerations and Approximate Rate Regions

The discussion above considered rate regions for the relatively simple uplink and downlink scenarios. Computation of the rate region is a difficult problem in general. In practice, it may not be easy for any node to obtain all the information (particularly, channel gains) necessary for computing the rate region. Besides, channel gains change with time, thus the actual channel gain for a future transmission may be different from the prior channel estimates.

For such reasons, practical systems may constrain the manner in which the wireless resources are shared. For instance, in a IEEE 802.11-based wireless LAN, an access point uses a channel that is not used by any nearby access point. In this case, transmissions to or from different access points can be treated as being orthogonal (since they pose negligible interference to each other), and each access point may independently schedule the transmissions on its allocated channel. Similarly, although successive interference cancellation (SIC) can yield better performance, some practical protocols, such as IEEE 802.11, do not use SIC.

In multi-hop wireless networks, it is particularly difficult for any node to precisely know the channel gain for all the links in the network. In such cases, scheduling algorithms are often designed using approximate estimates of the rate region. We now discuss one such approximate model named *conflict graph model*, which is often used in the design of scheduling algorithms.

The conflict graph model relies on the notion of a *conflict* between links. A link l_2 is said to conflict with link l_1 if scheduling l_2 while l_1 is scheduled is likely to result in an unreliable transmission on link l_1. Recall from our physical layer discussion that whether a transmission is successful or not depends on the SINR as well as the chosen transmission rate. Thus, the conflict definition assumes a particular set of parameters (such as transmission rate and power) for the links. Also, the interference from link l_2 may not make every transmission on l_1 unreliable, since the errors may be introduced probabilistically. Thus, the conflict relation may be viewed as a deterministic approximation of a probabilistic phenomenon.

The conflict is defined here for a pair of links. In reality, the reliability of transmission on l_1 depends on interference due to interference from *all* the other transmissions that are scheduled along with l_1. For instance, even if transmissions on link l_1 may be sufficiently reliable when either l_2 or l_3 (but not both) transmit simultaneously with l_1, l_1 may not be adequately reliable if l_2 and l_3 both transmit along with l_1. The pair-wise conflict relation does not allow us to capture this behavior. Thus, we can only approximately model the interference relationship between the various links using the conflict graph model.

The conflict graph model defines for each link l a subset of other links $I(l)$ such that if l is scheduled for transmission then none of the links in $I(l)$ should be scheduled simultaneously. Thus, all the links in $I(l)$ "conflict" with link l. We will call $I(l)$ the conflict set of link l.

In general, an interference conflict may not be symmetric. That is, if links l and m are scheduled simultaneously, one of the transmissions, but not both, may be unreliable. However, for simplicity, in our discussion of conflict graphs, we will assume that the conflict relationship is symmetric.

The conflicts may be defined differently for the two directed links in opposite directions between a pair of hosts. Thus, link (A,B) may possibly have different conflicts than link (B,A).

Our discussion above used interference to define the conflict relationship. We will refer to such a conflict as an *interference conflict*. Conflicts can arise due to hardware sharing as well. For instance, if an interface can be used to transmit to only one receiver, then *interface conflicts* also arise. For instance, links (A,B) and (A,C) cannot be used simultaneously, if they must share an interface at host A. In this case, links (A,B) and (A,C) have an interface conflict, and the conflict set of link (A,B) will include link (A,C), and vice-versa. Links causing interface conflict with a link l are also included in $I(l)$.

The conflict relationship between links is represented by the *conflict graph*. A link in the wireless network will be represented by a node in the conflict graph. An edge is drawn between two nodes in the conflict graph if the corresponding two links in the wireless network conflict with each other. Such a conflict graph is useful to determine the set of links that may be scheduled for transmission simultaneously without causing conflicts: in particular, a set of links do not include any conflicting links if and only if the set of corresponding nodes in the conflict graph do not contain any nodes that are adjacent in the conflict graph.

Two special cases of the conflict-based model are of particular interest.

- *Primary interference model:* The primary interference model assumes that two links conflict if and only if the two links share a common host as an endpoint. Thus, links (A,B) and (A,C) have a primary interference conflict. This model is useful when each host is equipped with a single interface, and each each link in the network is assigned a different channel. Alternatively, links may be assigned channels such that two links using the same channel are so far apart that they pose negligible interference to each other. In this case, only interface conflicts occur, and conflict set $I(l)$ for link l consists of links that need to use one of the end-hosts that is also used for link l.

- *Distance-Based Model:* In this model, two links are considered to be conflicting if they are "too close" to each other. Depending on how the conflict is defined exactly, we get different models: we discuss the hop-based and physical distance models.

  - *Hop-based model*: Consider two links (A,B) and (C,D). Consider the minimum hop route between the end-hosts of (A,B) and end-hosts of (C,D). If at least one of these four routes contains $h-1$ hops, and the other routes contains no less than $h-1$ hops, then the distance between links (A,B) and (C,D) is defined as h hops.

The hop-based model stipulates that, for some positive integer d, two links operating on the same channel conflict if and only if they are at distance at most d hops from each other. This model may be applied to IEEE 802.11, since its virtual carrier sensing mechanism attempts to ensure that no two links at distance up to 2 hops will transmit simultaneously.

- *Physical distance model:* In this case, physical distance between nodes is used to determine whether two links conflict or not. For instance, we can define an *interference range* such that a link (C,D) is likely to interfere with the transmission from A to B on link (A,B) if node B, which is the receiver on link (A,B), is within interference range of node C, which is the transmitter on link (C,D). Similar to the above models, physical distance model is also an approximation of the reality. For instance, the notion of interference range is not quite accurate in practice, since cumulative interference from all interference sources affects reliability, not just the interference from one source. Assuming a *circular* interference range is also not quite accurate. However, such approximate models often make network analysis simpler, and allow us to gain insights that can be useful in practice.

The conflict graph provides a means for the scheduler to determine the subset of links that can simultaneously transmit reliably in a given slot. Thus, the conflict graph can be viewed as a representation of the operating points in $\widetilde{\Gamma}$.

4.3 Centralized Scheduling

Let us assume an ideal centralized scheduler that can somehow learn the operating points for the network. Let us assume that the time is divided into time *slots*, and the scheduler must decide which links should be *active* in each slot: an active link can transmit packets in that slot at a rate specified by the scheduler.

The scheduling decisions are made on a per-slot basis: this means that the scheduler uses one particular operating point in each slot. There is no time-sharing *within* each time slot. Thus, the rate vector used for each slot belongs to $\widetilde{\Gamma}$. In our discussion of scheduling protocols, we will primarily consider the conflict graph model. Thus, $\widetilde{\Gamma}$ is represented in the form of a conflict graph.

For each link, we assume that there is a queue containing packets that need to be transmitted on that link. The queue may possibly be empty. Let q_l denote the queue for link l (we will also refer to size of this queue as q_l).

Example:

Consider the network shown in Figure 4.1, and assume that each link conflicts with links within distance 1 hop. The resulting conflict graph is shown in Figure 4.5 Let us

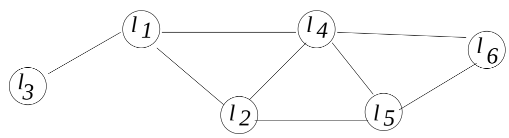

Figure 4.5 A conflict graph

assume that all transmissions occur at rate R in the absence of a conflict. The set of links in this network is $\{l_1, l_2, l_3, l_4, l_5, l_6\}$ and the following maximal rate vectors are in $\widetilde{\Gamma}$.

$[R, 0, 0, 0, R, 0]$, $[R, 0, 0, 0, 0, R]$, $[0, R, R, 0, 0, R]$, $[0, 0, R, R, 0, 0]$, $[0, 0, R, 0, R, 0]$

The above rate vectors are feasible operating points. The other operating points are obtained by replacing one ore more R entries in these vectors by 0. The convex hull of the above maximal rate vectors and [0,0,0,0,0,0] is the rate region Γ for this case.

How should the scheduler choose the schedule in each slot? The answer to this question depends on the goal the scheduler is attempting to achieve. For instance, let us consider the case where we would like to achieve rate $R/2$ on links l_1 and l_6 both, and rate $R/3$ on links l_2 and l_3 both. This goal can be achieved by using the following "cyclic" schedule repeatedly:

($[R, 0, 0, 0, R, 0]$, $[R, 0, 0, 0, 0, R]$, $[0, R, R, 0, 0, R]$)

With this cyclic schedule, in the first slot, the network is operated at the operating point $[R, 0, 0, 0, R, 0]$ (that is, only links l_1 and l_5 transmit). In the second and third slots, operating points $[R, 0, 0, 0, 0, R]$ and $[0, R, R, 0, 0, R]$ are used, respectively. This schedule is then used repeatedly, repeating once every three slots. Observe that link l_1 is scheduled every 2 slots out of 3 slots, yielding a rate of $2R/3$, which exceeds the desired rate of $R/2$. Similarly, the above schedule achieves or exceeds the goal for the other links as well. It should be clear that there exist many other possible schedules that can also satisfy the flow rate requirements.

In the above example, we assumed a certain desired rate vector for the various flows, and found a schedule that can support the corresponding rates on the various links. In Section 4.5, we will discuss a generic algorithm that can determine an appropriate schedule provided that the desired traffic rate vector is inside the rate region. Although we used rate region defined using the conflict graph model in the example above, it should be easy to see that such schedules may potentially be computed for arbitrary rate regions as well.

4.4 Scheduling For Always Heavily Backlogged Flows

A *flow* is presently defined as traffic that originates at one node, and is intended for an adjacent node. The packets from a flow are transmitted on the link between its source and its destination. We will assume that there is one flow on each directed link.

We will say that the flow on link l is *heavily backlogged* if there are always enough packets pending in q_l such that, if link l is scheduled for transmission during a time slot at a certain rate then there is enough data in q_l to fully utilize the slot duration. The heavily backlogged flows may be viewed as being greedy in that the flows generate enough packets to maintain an adequate backlog. Clearly, there are many different ways of choosing the schedule in each slot. Let us consider the *max-sum-rate* scheduler below.

Max-Sum-Rate Scheduler

As the name implies, the scheduler tries to pick a schedule that maximizes the sum of rates of the scheduled transmissions. Formally, the scheduler picks a rate vector $\mathbf{r}$ with rate r_l for link l such that

$$\mathbf{r} = arg \ \max_{\mathbf{s} \in \widetilde{\Gamma}} \sum_l s_l \tag{4.7}$$

The summations of the form $\sum_l$ in this case should be interpreted as summation over all the links that have flows scheduled on them. *arg max* above is the argument (or rate vector) $\mathbf{s}$ that maximizes the sum $\sum_l s_l$. The intuition here is that we want to pick a rate vector that will maximize the aggregate rate at which data will be transmitted. Recall that we are considering heavily backlogged flows.

While such a strategy seems intuitive, is it a good strategy? The strategy is good if our performance metric is *aggregate* over all the flows. But the strategy is not too good if we care about *fairness* – fairness means that each flow should get its "fair" share. We haven't quite defined what "fair" means, but it should be intuitive that maximizing the sum of rates might lead to some flow starving. For our example in Figure 4.5, using the above algorithm will result in using the operating point $[0, R, R, 0, 0, R]$ in all the slots, starving flows on links l_1, l_4 and l_5.

What should we do if we care about fairness? The first step then is to define what we mean by fairness. An intuitive definition is that the schedule is fair when each link gets identical throughput. However, it is not always possible to satisfy this goal without underutilizing some of the links in the network. As a simple example, consider a network of three links l_1, l_2, l_3 such that l_2 and l_3 conflict, but l_1 does not conflict with any other link. In this case, the maximal rate vectors in $\widetilde{\Gamma}$ are $[R, 0, R]$ and $[R, R, 0]$. There is a flow on each of the three links. To achieve identical throughput for all three flows, the scheduler must use a non-maximal schedule in some slots. For instance, scheduling $[0, 0, R]$ and $[R, R, 0]$ with equal frequency results in equal throughput $(R/2)$ for the three flows. However, the non-maximal operating point $[0, 0, R]$ does not fully utilize the available spectrum. For instance, by scheduling $[R, 0, R]$ and $[R, R, 0]$ alternately we achieve throughput R for the first flow, and $R/2$ for the remaining two flows, which is an improvement over throughputs achieved by scheduling $[0, 0, R]$ and $[R, R, 0]$ with equal frequency.

Defining "fair" as "equal rate" is not always appropriate if we care about efficient use of channel resources, as the example above illustrates. To fully utilize the resources, we must use *maximal* schedules in each slot. Thus, an alternative definition is necessary to define fairness. One possible approach is to use a *utility* function to define fairness. In particular, for each flow f, let us define utility U_f as a function of the average throughput r_f achieved by flow f. Then, network utility U is defined as the sum of the utility of all flows in the network. That is,

$$\text{network utility } \ U = \sum_f U_f(r_f)$$

With this definition of utility, we will say that the scheduler is "fair" if it maximizes the network utility. As per this definition, the max-sum-rate scheduler is fair if utility of each flow is equal to its throughput. In general, different utility functions will lead to different allocation of wireless resources to the flows by the centralized scheduler.

4.5 Scheduling for Flows Within Rate Region

In the previous section, we assumed that the flows are always heavily backlogged. Thus, clearly it is not possible to serve all data that all the flows can possibly produce. In this section, we assume that the flows produce traffic at rates that are, in fact, schedulable. In general, the flows may produce data at time-varying rates, and the rate region Γ itself may also be time-varying. However, in our present discussion, we will assume that the flows generate data at a constant rate, and the rate region is independent of time. In this section, we consider flows for which the rate vector is "inside" the rate region Γ. Meaning of the term "inside" will be made more precise later. The scheduler discussed here is said to be *Throughput-Optimal*, since it can schedule any flows inside Γ.

Recall that presently we are considering only single-hop flows. q_l denotes the size of the queue for directed link l. q_l will change with time, as new data arrives from the source, or when data from the corresponding queue is transmitted. We may use the notation $q_l(t)$ to indicate that q_l varies with time t. The goal of the scheduler is to make scheduling decisions such that the flow demands are met [3].

Throughput-optimal (TO) scheduler: Choose the rate vector as a function of queue sizes as follows:

$$\mathbf{r} = arg\ max_{\mathbf{s} \in \widetilde{\Gamma}} \sum_l q_l\, s_l \tag{4.8}$$

Thus, if the TO-scheduler chooses rate vector $\mathbf{r}$, then

$$\sum_l q_l\, r_l \ \ \geq \ \ \sum_l q_l\, s_l \ \text{ for all } \mathbf{s} \in \widetilde{\Gamma} \tag{4.9}$$

Observe that Equations 4.7 and 4.8 are similar except that the rate is weighted by the queue size in Equation 4.8.

Since Γ is the convex hull of $\widetilde{\Gamma}$, and $\sum_l \ q_l\, s_l$ is a linear operation on the rates, Equation 4.9 implies that

$$\sum_l q_l\, r_l \ \ \geq \ \ \sum_l q_l\, s_l \ \text{ for all } \mathbf{s} \in \Gamma \tag{4.10}$$

Stability property of throughput-optimal (TO) scheduler [3]:

Let us assume that the time is slotted, with the slot duration being 1 time unit (thus, rates are specified using slot duration as the time unit). The slots are numbered 0, 1, 2, etc. To analyze the stability of the TO-scheduler, let us define a function $L(t)$ as follows, where t denotes the slot number.

$$L(t) = \sum_l q_l^2(t)$$

and define

$$\Delta L(t) = L(t+1) - L(t) = \sum_l q_l^2(t+1) - \sum_l q_l^2(t)$$

Now, in slot t, the TO-scheduler chooses rate vector $\mathbf{r}(t)$ defined by Equation 4.8. Thus, queue q_l is served at rate $r_l(t)$ for 1 slot duration (1 time unit). λ_l is the traffic arrival rate for link l. Therefore, given the slot duration of 1 time unit, λ_l units of data is added to q_l in slot t. Assume that only data that is *already* in queue q_l at the start of slot t may be served during slot t (that is, transmitted on link l during slot t at rate $r_l(t)$). Then queue size at the end of slot t, or equivalently the queue size at the start of slot $t+1$ is given by

$$q_l(t+1) = \lambda_l + max(q_l(t) - r_l(t), 0) \tag{4.11}$$

The $max(q_l(t) - r_l(t), 0)$ term indicates that if $q_l(t)$ is less than $r_l(t)$, then at most $q_l(t)$ data may be served in slot t. Now, for any non-negative real numbers v, u, r and a, it can be shown that, if $v \leq a + max(q-r, 0)$ then

$$v^2 \leq q^2 + r^2 + a^2 - 2q(r-a) \tag{4.12}$$

From Equations 4.11 and 4.12, we get that

$$\begin{aligned}
q_l^2(t+1) &\leq q_l^2(t) + r_l^2(t) + \lambda_l^2 - 2q_l(t)\left(r_l(t) - \lambda_l\right) \\
\Rightarrow q_l^2(t+1) - q_l^2(t) &\leq r_l^2(t) + \lambda_l^2 - 2q_l(t)\left(r_l(t) - \lambda_l\right) \\
\Rightarrow \sum_l \left(q_l^2(t+1) - q_l^2(t)\right) &\leq \sum_l \left(r_l^2(t) + \lambda_l^2 - 2q_l(t)\left(r_l(t) - \lambda_l\right)\right)
\end{aligned}$$

From the above inequality, and the definition of $\Delta L(t)$, we have that

$$\Delta L(t) \leq \sum_l \left(r_l^2(t) + \lambda_l^2 - 2q_l(t)\left(r_l(t) - \lambda_l\right)\right)$$

Now, let us suppose that λ_{max} is an upper bound on λ_l for all l, and let r_{max} be an upper bound on rate r_l for all l. Assume that r_{max} and λ_{max} are both finite. Define

$$B = \sum_l \left(r_{max}^2 + \lambda_{max}^2\right)$$

Then it follows that B is a finite positive constant, and

$$\Delta L(t) \leq B - 2\sum_l q_l(t)\left(r_l(t) - \lambda_l(t)\right) \tag{4.13}$$

Suppose that there exists a positive value ϵ such that $\mu_l = \lambda_l + \epsilon$ and rate vector $\mu \in \Gamma$. This is what we mean by the traffic rate vector being "inside" Γ. Recall from Equation 4.10 that

$$\sum_l q_l r_l(t) \geq \sum_l q_l s_l \quad \text{for all rate vectors } \mathbf{s} \in \Gamma$$

Therefore,

$$\begin{aligned}
\sum_l q_l r_l(t) &\geq \sum_l q_l \mu_l \quad \text{since } \mu \in \Gamma \\
\Rightarrow \sum_l q_l r_l(t) &\geq \sum_l q_l(\lambda_l + \epsilon) \\
\Rightarrow \sum_l q_l(r_l(t) - \lambda_l) &\geq \sum_l q_l \epsilon \\
\Rightarrow -\sum_l q_l(r_l(t) - \lambda_l) &\leq -\epsilon \sum_l q_l
\end{aligned}$$

The above inequality and Equation 4.13 imply that

$$\Delta L(t) \leq B - 2\epsilon \sum_l q_l$$

Then from the definition of $\Delta L(t)$, we have that

$$L(t+1) - L(t) \leq B - 2\epsilon \sum_{l} q_l$$

Since B is a constant, if the queue for any one link is larger than $\frac{B}{2\epsilon}$ then the right hand side of the above inequality will be negative, implying that $L(t+1)$ will be smaller than $L(t)$. Since $L(t+1)$ is sum of square of the queue lengths, this in turn implies that, with the TO-scheduler, the queues cannot grow unboundedly. Thus, the queues remain "stable", and thus the scheduler is able to schedule any flows *inside* the rate region.

An interesting property of TO-scheduler is that although it is able to schedule the flows inside the rate region, it does not actually need to know the actual flow rates. What is even more remarkable is that the scheduler can schedule the flows even if the flow rates vary with time, provided that the average flow rates are inside the rate region. In addition, the scheduler can schedule the flows when the rate region itself is also time-varying, so long as the average flow rates are inside the average rate region. In case of time-varying flow rates and rate regions, the flow rates and rate regions need to satisfy certain constraints, which we will not elaborate on here. Here we also do not discuss the proof of stability under time-varying conditions.

4.6 Summary

In this chapter, we introduced the notion of a rate region. The rate region is constrained by parameters such as maximum transmit power and available bandwidth, and also by hardware constraints (for instance, a wireless interface may be able to transmit to only one node at any given time). A scheduler should only choose rate vectors that are within the available rate region. However, due to the difficulty in identifying the exact rate region, a practical scheduler may use an approximate definition of the rate region. The chapter also discussed some centralized scheduling algorithms.

REFERENCES

[1] N. Abramson, "The ALOHA system - another alternative for computer communication," in *1970 Fall Joint Computing Conference Proceedings*, pp. 281–285, 1970.

[2] A. Chandra, V. Gummalla, and J. O. Limb, "Wireless medium access control protocols," *IEEE Communications Surveys*, 2000.

[3] L. Georgiadis, M. Neely, and L. Tassiulas, "Resouce allocation and cross layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, 2006.

[4] IEEE, "IEEE std. 802.11 - wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 1999.

[5] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Prentice-Hall, 2002.

[6] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 2 ed., 2002.

[7] F. A. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution," *IEEE Transactions on Communications*, vol. COM-23, no. 12, pp. 1417–1433, 1975.

[8] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge Press, 2005.

[9] X. Yang, *Efficient Packet Scheduling in Wireless Multihop Networks*. PhD thesis, ECE Department, University of Illinois at Urbana-Champaign, 2005 December.

[10] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *ACM MobiHoc*, 2002.