“You can’t serve all of the people all of the time” Resource Allocation In Wireless Networks

Harish Ganapathy Siddhartha Banerjee

June 14, 2009

Abstract

We study the problem of resource allocation in wireless networks, with an emphasis on the role of limited feedback. Feedback in wireless systems is the process by which the receiver communicates information about the wireless channel back to the transmitter which enables the latter to make suitable transmit decisions. Traditionally in wireless networks, feedback resources have been assumed to be unlimited while developing scheduling and resource allocation algorithms. However, as advancements in wireless technology have allowed the realization of these algorithms in real systems, it is becoming increasingly clear that the gains promised by these algorithms can never be realized due to fundamental limits on the availability of feedback resources. Thus, the proper modeling of feedback, and the development of algorithms to optimally allocate feedback, is one of the main challenges faced by the wireless networking community today.

The study of limited feedback in wireless networks, although much younger than other topics in the field, is a fairly well developed area on its own. In order to understand this topic, we first present the canonical resource allocation problem, and thereby introduce some of the primary concepts involved in the design of wireless networks. Next, we identify and present some of the seminal papers in this field to study various approaches which are taken towards modeling and solving this problem. Finally, we explore open problems in the field, and future directions of research.

1 An Overview of Wireless Communications Networks

1.1 Basics of Wireless Communications

We start with a quick introduction to wireless communications networks in order to define the underlying model and the parameters that we are interested in. In the process we hope to identify the features of wireless networks which differentiate them from other communication media (wireline, optical fiber) and make them interesting from a design perspective. For the sake of brevity, we keep technical details to a minimum in order to highlight the topics of interest. For a more detailed overview of wireless networks, refer to Tse and Vishwanath [1].

In comparison to other communications media, the wireless medium is unique due to the fact that there is no fundamental orthogonalization between different transmitter-receiver pairs. Thus, while other media allow designers to analyze and optimize each link separately, and share resources in a controlled manner (multiplexing), the wireless medium is by default a shared resource on which designers must impose some structure and order in order to allow for meaningful communication. The situation is akin to trying to hold a conversation in a crowded party where different people are talking among themselves.

This open nature of the wireless medium shows up at the physical level in the manner in which the medium scales any signal sent through it. Various electromagnetic phenomena such as reflection and diffraction result in the constructive and destructive interference of the transmitted signals. Thus, even a single transmitted signal on the wireless medium is received as several randomly time shifted copies scaled by random scaling factors. The wireless channel however has been very well studied, and there exist excellent statistical models to describe it. We comment further on this in later sections.

From a multiuser networking perspective, the effect of the wireless medium can be distilled into two main concepts:

- Broadcast: There are no clear channels between a transmitter and a receiver, and thus a message transmitted by one transmitter can be received by several receivers (either as data, or interference) with different signal strengths.
- Superposition: When several signals from different transmitters arrive at a receiver, they are superimposed on each other. For point to point links, this causes interference at the receiver. However, this can be utilized to receive several signals at a common receiver via 'multiple access' schemes.

Because of these two effects, links in a wireless network are never isolated but instead interact in complex ways, thus making it a shared resource. This is quite unlike the wired world where each transmitter-receiver pair can often be thought the of as isolated point-to-point links.

1.2 Description of resources?

In wireless networks, shared resources can be broadly classified into four categories; time, frequency, space and code [1]. Users that communicate using the same resource block experience interference from each other resulting in degraded signal quality. As is quite intuitive, the signal quality, typically measured using the signal-to-interference-plus-noise power ratio (SINR), is directly related to the maximum data rate R (bits/sec) that a transmitter-receiver pairs can support through Shannon's capacity formula $R = W \log_2(1 + \text{SINR})$ bits/sec. Here, W represents the bandwidth or set of frequencies occupied by the transmission. It is of primary interest in wireless networks to determine how

[1] For the wireless literate, we use *code* to describe systems such as code-division-multiple-access (CDMA) and superposition coding.

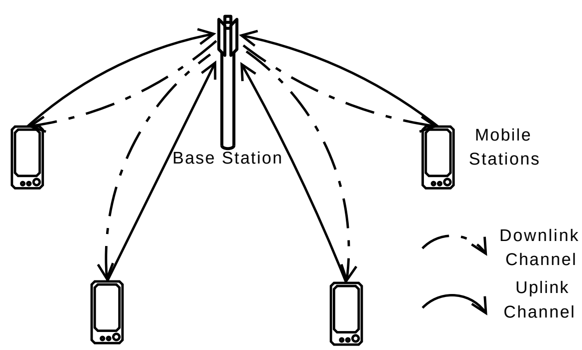
to allocate resources amongst users in a manner that maximizes some utility function. We explicitly mention data rate as a measure of resource allocation efficiency because a typical utility function used in a wireless system belongs to the class of functions that is strictly non-decreasing in data rate.

At this point, we find it useful to begin describing the entire network through the use of a mathematical model. We assume that source i, $i = 1, 2, \ldots, N$, transmits data packets that can be modelled as a stationary random process $\{x_i[n]\}_{n=1}^{\infty}$ with average transmit power $E[|x_i[n]|^2] = P_i$. Each data packet, which occupies a (time, frequency, space, code) resource block, is created through a mapping from a collection of bits to a collection of complex-valued *symbols* that are transmitted over the wireless channel. For simplicity of exposition, we will adopt a more classical treatment through the remainder of this paper by ignoring the code and space dimensions, which are relatively recent advances. We feel that adding these dimensions would not modify significantly the main concepts that are presented in this paper. Assuming that the users' packet transmissions are time-synchronized, occupy a sufficiently small bandwidth, use the same code and are done over a single antenna, the input-output relationship of the entire network at time n can be decribed succinctly as

$$\mathbf{y_n} = \mathbf{H_n x_n} + \mathbf{v_n} \in \mathbb{C}^{N \times 1} \tag{1}$$

where $h_{ij}[n]$ is a complex value that represents the channel gain between source j and destination i; $\mathbf{v}[n]$ denotes complex additive white Gaussian noise that is used to model thermal noise that often affects the front-end of a receiver. The channel matrix $\mathbf{H}[n]$ completely captures the broadcast and superposition effects of a wireless network which were mentioned in Section 1.1. Without subjecting the reader to excessive detail, we would like to point out that complex channel, symbol and noise representations arise out of the fundamental fact that, at any given frequency f_c, two natural/orthogonal transmission channels exist in the form of the sinusoids $\sin(2\pi f_c t)$ and $\cos(2\pi f_c t)$.

In the following, we focus on a specific type of network called *cellular network* in which mutiple mobile cell phone users are served (according to some chosen resource allocation scheme) by a base-station as shown below in Fig. 1.2. In a cellular network, information flow occurs in two directions:



(i) *Downlink*: Data is transmitted from the base-station to the mobile stations (MS). In this case, the data vector $\mathbf{x_n}$ in (1) is transmitted by the base-station.
(ii) *Uplink*: Data is transmitted from the MSs to the base-station. In this case, i-th element of $\mathbf{x_n}$ in (1) is transmitted by user i and vector $\mathbf{y_n}$ is received at the base-station.

In cellular networks, downlink and uplink information is organized and transmitted in accordance with the cellular standard that is adopted by the network provider (ex. ATT, Sprint, etc.). In almost all standards, downlink and uplink information is either *duplexed* in time, which is called time-division duplexing (TDD) or in frequency, which is called frequency-division duplexing (FDD). In the former, downlink and uplink frames are ordered in time. A simple ordering scheme that is often used is downlink and uplink frames are transmitted in even and odd time slots respectively. In designing transmission policies for TDD systems, it is often assumed that the channel $\mathbf{H}[n]$ is approximately the same (termed reciprocity) for both downlink and uplink due to memory in the channel. In the latter, downlink and uplink frames are not ordered in time but are transmitted as entire streams on two separate frequencies. The reciprocity assumption on the channel matrix is typically does not hold for such systems. The two duplexing rules are described below in Fig. 1. We now focus our attention

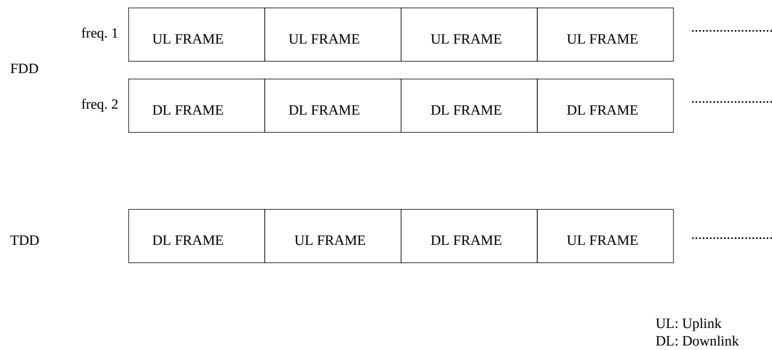


Figure 1: TDD/FDD frame structure

on the contents of a typical frame (downlink or uplink) in standards such as WiMAX and Long Term Evolution (LTE), which employ Orthogonal Frequency Division Multiple Access (OFDMA) as their preferred multiple access scheme. We are now able to define the word "resource" that we have used at liberty thus far. Each OFDMA frame is a contiguous collection of time-freqency slots or blocks in which a subset of blocks or *resource* is assigned to a particular user at the discretion of the base-station. This process is called user *scheduling* or *resource allocation*. The base-station informs each user of its assignment through control segments, which are time-frequency blocks that are located at the beginning of each frame. Fig. 2 illustrates the frame structure for WiMAX TDD. The control segment of the downlink frame consists of the *preamble*, *UL MAP* and *DL MAP*. The preamble contains, amongst other information, a sequence of *pilot* bits that are known to the receiver apriori, which can then be used to estimate the channel, which is integral to the successful decoding of a data packet. The UL MAP and DL MAP contain uplink and downlink resource (time-frequency block) assignments respectively. *DL Bursts* refer to user data that is arranged by the base-station according to the DL MAP while the user transmits *Bursts* on the uplink according to the UL MAP. This brings us to an important question the answer to which forms the subject of the remainder of this paper.

This is the question of how the base-station must allocate time-frequency sub-blocks or resources to each user. Furthermore, what information does the base-station require in order to make these scheduling decisions? The short answer to the latter question is that the base-station needs uplink and downlink network state parameters specific to each user prior to every scheduling instant. The precise nature of these parameters will be futher explained in Section 2. While some network state parameters are readily accessible to the base-station, the remaining parameters are communicated back to the base-station by the users through a procedure that can formally be classified *feedback*. Upon receiving all required parameters or state information, the base-station makes a scheduling decision in order to maximize a utility function that is chosen by the network provider. In WiMAX, users feedback network state information through the *Ranging* sub-block and CQICH in the uplink frame as shown in Fig. 2. The base-station uses this information to make a scheduling decision for the subsequent downlink and uplink frame which is communicated to the users through the UL and DL

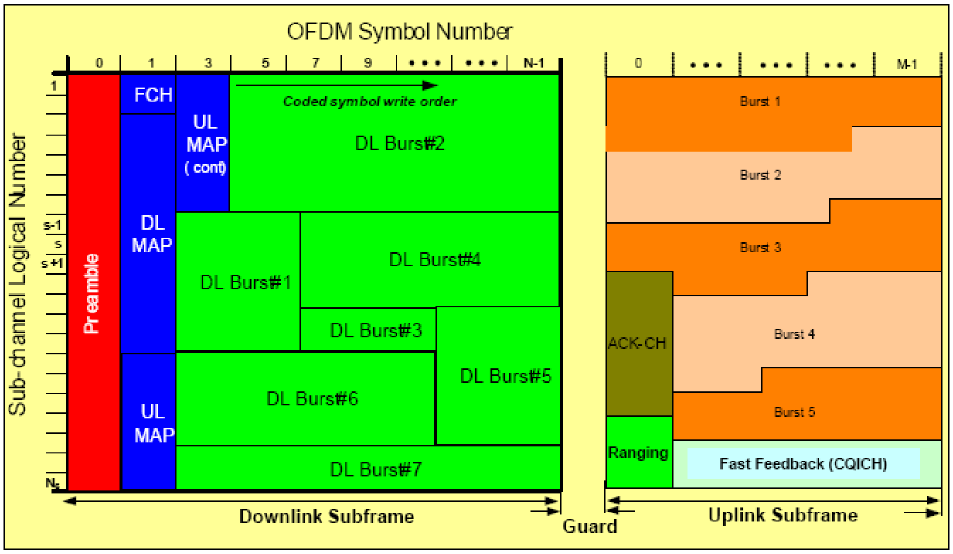


Figure 2: TDD Frame Structure for WiMAX [2]. The preamble contains, amongst other information, a sequence of *pilot* bits that are known to the receiver apriori, which is used to estimate the channel, which is integral to the successful decoding of a data packet. The *UL MAP* and *DL MAP* contain uplink and downlink resource (time-frequency block) assignments respectively. *DL Bursts* refer to user data that is arranged by the base-station according to the DL MAP while the user transmits data on *Bursts* on the uplink according to the UL MAP. The *Ranging* and CQICH sub-blocks contain network state information that is used by the base-station for scheduling in the next downlink and uplink frame. If the packet intended for a user is received and decoded correctly, the user sends an acknowledgement or ACK in the *ACK-CH* segment while if the packet is not decoded correctly, the user sends a negative acknowledgement or NACK in the *ACK-CH*.

MAP as mentioned earlier.

Scheduling or resource allocation forms the emphasis of Section 2.

2 The Canonical Resource Allocation Problem in Wireless Communications Networks

In this section we study the traditional resource allocation problem in wireless communications networks. We consider the model given by Huang et al. [7], as it succinctly captures the essential features of the problem while minimizing the extraneous details that arise from real world considerations. In the process, we hope to expose the following concepts:

- The modeling of resources and constraints on them imposed both by the medium as well as by various physical restrictions at transmitter and receiver.
- The idea of time scale decomposition, wherein various tasks are assumed to be carried out under different resolutions of time discretization. From a physical viewpoint, the idea of time scale decompositions can be viewed as arising out of the notion of a channel coherence time over which the channel state remains roughly correlated.
- The choice of utility functions for the optimization problem. These utility functions depend

upon the design philosophy and the requirements imposed upon the system. However, one of the main ideas that we exploit is that the allocation algorithms can be made independent of the choice of objective function by suitable time scale decompositions. We discuss this in detail in section 2.1.

- The online and real-time nature of the algorithms, which imposes complexity requirements on them.

2.1 The Case for Dynamic Resource Allocation

Dynamic resource allocation algorithms aim to maximize a 'utility function', that represents both system-wide service requirements, through the optimal allocation of resources. The primary resource under consideration is the resource which is utilized for multiple access. We continue to consider OFDMA systems, and so the primary resource is a set of time-frequency block. The allocation is dynamic in nature as decisions are made at the beginning of every frame. In addition, there are other resources which arise from physical limitations at the transmitter and receiver. The primary resource of this type is power, which arises from physical considerations at the transmitter. For an uplink channel, the power constraint is a per-user constraint, whereas in downlink, it is a shared resource across all the downlink paths. Other resources will be highlighted when we consider them in later sections.

The allocation algorithms for downlink assume the availability of network state information at the BS as mentioned earlier. The exact definition of network state information varies. At a minimum, the BS requires channel state information (CSI) and depending on the chosen utility function, the BS may require some additional information such as the queue states, the average rate seen so far, etc. After acquiring all required state information, the BS then decides the optimal allocation and then allocates resources accordingly. In the downlink, CSI needs to be fed back from the MSs while the queue states are readily accesible. In the case of uplink, the MSs need to fed back only queue state information. In all these algorithms, the feedback is assumed to be perfect and infinite, and hence the results obtained by these algorithms are an upper bound to the system performance when feedback limitations are considered.

Dynamic resource allocation is essential in data networks in order to maximize their performance under a broad class of utility functions. Unlike voice communications, where there are strict requirements on quality of service metrics such as delay during the duration of a session, data communications allows for more flexibility which can therefore be exploited to maximize chosen system utility functions. In wireless communications, one key example of the benefit of this flexibility is the concept of 'Multi-User Diversity' (refer [1]) whereby a broadcast or multiple access channel behaves not as the average channel experienced by the users, but as the 'best-case' channel (in probabilistic terms, as the maximum of a set of random variables) through the adoption of appropriate policies.

2.2 Utility Functions and Gradient-Based Scheduling

Utility function is an umbrella term used to describe various system-wide objectives that are considered while designing a wireless system. These objectives are usually long-term objectives, and thus a designer is interested in the ergodic value, i.e., the long term time average, of these functions. However, the random nature of the channel can be separated from the resource allocation by choosing appropriate time scale decompositions. The fundamental idea behind this is that the wireless medium, although randomly fluctuating, does so in a correlated manner across time. Thus, it is possible to define an interval called a coherence time within which the signal scaling is for all practical purposes a constant. We refer to this as the networking time scale/time slot. Now, if the scheduling decisions are made within one networking time slot, then one can estimate the channel at the beginning of the time slot and use it as an input for the algorithm. A subtle point here is that several of the constraints considered, in particular, the per-user power constraints, are also on their average usage, and in addition, the information theoretic rate can only be achieved by coding over large blocks. However, we assume that the data is transmitted in smaller packets on a physical layer time scale (or information theoretic time scale), which is several orders smaller than the networking time scale [2]. Thus there are a sufficient number of physical layer time slots within one networking time slot so as to allow us to achieve rates close to capacity under average power assumptions *within* a networking time slot. From an information theoretic perspective, this is akin to saying that the noise random process varies from one physical layer time slot to another, while the channel scaling random process remains fixed over a networking time slot.

Huang et al. [7] argue that a large class of utility maximizing scheduling algorithms can be viewed as 'gradient-based' algorithms, i.e., they select the transmission rate vector that maximizes the projection onto the the gradient of the system's total utility. In other words, maximizing the utility function over time can be shown to be same as maximizing a weighted sum of rates within one networking time slot, where the weights are determined by the gradient of the utility function. Formally, in a system with K users and in time slot $n \in \mathbb{Z}_+$, given the channel state information $\mathbf{h_n}$, we associate an achievable rate region $\mathcal{R}(\mathbf{h_n})$ and a rate vector $\mathbf{r_n} = (r_{1,n}, r_{2,n}, \ldots, r_{K,n}) \in \mathcal{R}(\mathbf{h_n})$ where $r_{i,n}$ is the rate that can be achieved by user $i \in \{1, 2, \ldots, K\}$. Next, given the average throughput $(W_{i,n})$ and queue states $(Q_{i,n})$ of the users, we define a utility function $U(\mathbf{W_n}, \mathbf{Q_n}) := \sum_{i=1}^{K} U_i(W_{i,n}, Q_{i,n})$. Finally, under a suitable definition of $U'(.)$, the sum of the derivative of $U(.)$ with respect to the throughput and the derivative of $U(.)$ with respect to the queue states, we define the output of the scheduling algorithm as

$$\mathbf{r_n}^* := \arg \max_{\mathbf{r_n} \in \mathcal{R}(\mathbf{h_n})} \nabla U(\mathbf{W_n}, \mathbf{Q_n})^T \cdot \mathbf{r_n}$$

Thus the problem of optimal resource allocation to achieve a long term objective reduces to choosing an appropriate vector of weights and maximizing a corresponding weighted sum of rates within each networking time slot. In the rest of this section we consider the intra time slot scheduling problem. The problem of choosing appropriate weights is taken up in Section 3.

2.3 The Resource Allocation Problem for OFDMA

An OFDMA system has been described in earlier Section 1.2. The primary resource is frequency, which is divided into a number of sub-bands or tones. In traditional OFDMA systems, only one user is associated with each tone, but we generalize this by assuming that a fraction of each tone can be alloted to a user via time-sharing within a networking time slot. In addition, each user has maximum power constraints on the transmit signal power. Thus $\mathcal{R}(\mathbf{h_n})$ is parameterized by the allocation of tones to users and allocation of powers across tones. For simplicity, we have dropped the dependence on time for ease of notation, as the allocation problem has to be re-solved within each time slot.

Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of tones and $\mathcal{K} = \{1, 2, \ldots, K\}$ the set of users. For each tone $j \in \mathcal{N}$ and user $i \in \mathcal{K}$, let h_{ij} be the received signal-to-noise ratio (SNR) per unit transmit power[3]. We denote the transmit power allocated to user i on tone j by p_{ij}, and the fraction of that tone allocated

[2]To get a feel for these time scales, consider the WiMAX standard. At 5.8 MHz operating frequency and 100kmph speeds, the coherence time, and therefore the networking time slot of the channel is roughly 2ms. The typical rates achieved on uplink and downlink are of the order of > 10Mbps. Even if we consider a 5Mbps rate, this allows us to send 104 bits in one networking time slot. This is a large enough block length such that we can achieve close to channel capacity using sophisticated coding techniques.

[3]We have earlier used h_{ij} to denote the channel scaling, but this is equivalent to the SNR if we scale the channel coefficient by the noise variance

to user i as x_{ij}. Thus for a given allocation with perfect CSI, user i's feasible rate on tone j is given by Shannon's formula for the capacity of an additive Gaussian noise channel as-

$$r_{ij} = x_{ij} B \log\left(1 + \frac{p_{ij} h_{ij}}{x_{ij}}\right)$$

Here B is the bandwidth of tone j which we can assume to be 1 without loss of generality as all tones have equal bandwidth.

We consider the case of the uplink network, but note that the model can be generalized to a downlink network with the per-user power constraints replaced by a sum power constraint. For the uplink network, the feasible rate region $\mathcal{R}(\mathbf{h_n})$ is given by-

$$\mathcal{R}(\mathbf{h_n}) = \left\{ \mathbf{r} \in \mathbb{R}_+^{\mathbb{K}} : r_i = \sum_{j \in \mathcal{N}} r_{ij}, \forall i \in \mathcal{K} \right\}$$

where $(\mathbf{x}, \mathbf{p}) \in \mathcal{X}$ are chosen subject to

$$\sum_i x_{ij} \leq 1, \forall j \in \mathcal{N} \quad (2)$$

$$\sum_j p_{ij} \leq P_i, \forall i \in \mathcal{K} \quad (3)$$

and the set

$$\mathcal{X} := \left\{ (\mathbf{x}, \mathbf{p}) \geq 0 : 0 \leq x_{ij} \leq 1, p_{ij} \leq \frac{x_{ij} s_{ij}}{h_{ij}}, \forall i, j \right\}$$

where s_{ij} is a maximum SINR constraint on the sub-channel j for user i which can be used to model various physical constraints on the system. Constraint (2) ensures that the time-fraction allocations per sub-channel do not exceed one while constraint (3) ensures that the total power allocated to each user i across sub-channels j does not exceed the total available power P_i for that user .

Finally, we define the resource allocation problem as-

$$\max_{\mathbf{r} \in \mathcal{R}(\mathbf{h_n})} \sum_{i \in \mathcal{K}} w_i r_i$$

Where the weights w_i are chosen according to the utility function.

2.4 Optimal Solution to the Resource Allocation Problem

The optimal solution to the problem proposed above is obtained using the technique of Lagrange Multipliers. This duality approach is possible since the optimization problem of interest satisfies Slater's condition. Slater's constraint qualification condition requires that the interior of the constraint set be non-empty. In the resource allocation problem of interest, the power allocation $p_{ij} = 0, \forall i, j$, is a valid allocation that yields rates $r_{ij} = 0, \forall i, j,$, which is in the interior of our constraint set. Proceeding further, we associate dual variables $\lambda = (\lambda_i)_{i \in \mathcal{K}}$ with the per-user power constraints, and $\mu = (\mu_j)_{j \in \mathcal{N}}$ with the fractional tone sharing constraints. Thus, from duality theory, it follows that the optimal solution to the problem is given by

$$\min_{(\lambda, \mu) \geq 0} \max_{(\mathbf{x}, \mathbf{p}) \in \mathcal{X}} L(\lambda, \mu, \mathbf{x}, \mathbf{p}).$$

where the Lagrangian is written as

$$L(\lambda, \mu, \mathbf{x}, \mathbf{p}) := \sum_{i,j} w_i x_{ij} \log\left(1 + \frac{p_{ij} h_{ij}}{x_i j}\right)$$
$$+ \sum_i \lambda_i (P_i - \sum_j p_{ij}) + \sum_j \mu_j (1 - \sum_i x_{ij}).$$

This optimization problem is solved in several steps. First, we analytically find the optimal $\mathbf{p}$ and $\mathbf{x}$ given fixed values of the dual variables. Then, after substituting the optimal values of the primal variables, we find the optimal μ using a search for the maximum value of a per-user metric on each subchannel. Finally, the optimal value of λ is obtained using a numerical search.

Huang et al. [7] note, however, that calculating the optimal solution to this problem requires substantial computation, in particular due to the numerical search procedure for the optimal λ, as well as the complexity of breaking ties in allocations which arise often in typical problem instances. Thus, they have used the intuition derived from the optimal model to obtain low complexity suboptimal algorithms, and they have used simulations to compare the performance of these algorithms versus the optimal solution.

3 Optimal Scheduling with Queues

In the previous sections, and in particular in Section 2.2, we argue that the design of a wireless network to meet system-wide as well as per-user objectives can be broken down into two separate sub-problems:

- Intra time-slot scheduling and resource allocation in order to maximize a weighted sum of rates.
- Choice of the weights in terms of available user and system statistics in order to achieve the desired objective.

The main advantage of this decomposition is that it allows us to separate out the channel randomness, which is used to choose the appropriate weights for the scheduling algorithm. These weights, along with the availability of CSI, allows us to formulate the resource allocation problem in a completely deterministic manner. The formulation and algorithms for the intra time-slot optimization problem are covered in the previous section. One can argue that this decomposition is not possible for all objectives and scenarios, but historically most of the significant objectives studied have been shown to result in algorithms of this form. The decomposition also requires certain assumptions for the system model, but we argue in Section 1 that these are reasonable assumptions for real-life systems.

In this section, we see how the weights in Section 2.2 are chosen in order achieve certain objectives. In particular, we consider the problem of scheduling with queues. This problem was first considered in a landmark paper by Tassiulas and Ephremides [3], in which they proposed an algorithm which we refer to as the backpressure algorithm (BP), as well as developed techniques for analyzing it. The subsequent development of data-only protocols for wireless networks led to a renewed interest in this class of algorithms in the early part of this decade. Andrews et al. [4] introduced a new formal framework for results of this type and proposed a new way to derive them via fluid limits.

The proofs in this section will make use of basic Markov Chain theory and in particular, the idea of Lyapunov functions and Foster's theorem. We start of by giving a brief overview of the necessary results before moving on to the problem formulation and solution. By the end of this section, we hope to introduce the following concepts

- The queuing model for a wireless communications network.

- The idea of a queue stabilizing policy, and its associated throughput region for flows into the network.
- The idea of static-split rules which are used to derive an outer bound for the best achievable throughput region.
- The optimality of the backpressure algorithm.
- The use of fluid limits to derive the optimality of an algorithm.

The complete proofs of the results presented are quite technical, and we simplify them by showing how they can be derived for simple examples, as well as giving an intuition behind their general structure. For further details, refer to Tassiulus et al. and Andrews et al. [3, 4].

3.1 Overview of Markov Chains

In this section, we assume a knowledge of basic Markov Chain theory, and in particular, of the definitions of irreducibility and positive recurrence of Markov Chains. The main result presented in this section is known as Foster's criterion, and is the primary tool that we use in proving the stability of our scheduling policies. It is a sufficient condition for the positive recurrence of a Markov Chain, and is very useful in cases where the complicated nature of the queuing system does not allow us to calculate the stationary distribution for the chain and thereby check for positive recurrence in a more direct manner.

Theorem 1. ***(Foster's Criterion)****: Suppose $\mathbf{X} = \{X_n\}_{n\in\mathbf{N}}$ is an irreducible, discrete-time Markov chain on a countable state space $\mathcal{X}$ with transition probability matrix P. Let $\mathcal{L} : \mathcal{X} \to \mathcal{R}_+$ and define a drift of X with respect to $\mathcal{L}$ as*

$$d_{\mathcal{L}}(x) = \mathbb{E}\left[\mathcal{L}(X_{n+1}) - \mathcal{L}(X_n)|X_n = x\right], x \in \mathcal{X}. \tag{4}$$

If there is a finite set $A \subset X$ and constants $\varepsilon, \beta > 0$ such that

$$d_{\mathcal{L}}(x) < \beta, \forall x \in A \text{ and } d_{\mathcal{L}}(x) < -\varepsilon, \forall x \in \mathcal{X} \setminus A, \tag{5}$$

then $\mathbf{X}$ is positive recurrent.

Proof: Recall that if a Markov Chain is irreducible, then

$$\lim_{t\to\infty} \frac{1}{t} \sum_{n=0}^{t-1} \mathbf{1}\left(X_n = i\right) = \begin{cases} \pi(i), & \forall i \in \mathcal{X} \text{ if } \mathbf{X} \text{ is positive recurrent} \\ 0, & \forall i \in \mathcal{X} \text{ else.} \end{cases} \tag{6}$$

Note that this is true pathwise, irrespective of the initial distribution, which implies that the above holds true if one takes expectations. We start by re-writing (5) as

$$d_{\mathcal{L}}(x) < -\varepsilon + (\beta + \varepsilon)\mathbb{1}_{(x\in\mathcal{A})}. \tag{7}$$

So if $X_o = x$, then

$$\mathbb{E}\left[\mathcal{L}(X_n) - \mathcal{L}(X_o)|X_o = x\right] = \mathbb{E}\left[\sum_{i=1}^{n} \mathcal{L}(X_i) - \mathcal{L}(X_{i-1})|X_o = x\right] \tag{8}$$

$$= \mathbb{E}\left[\sum_{i=1}^{n} \mathbb{E}\left[\mathcal{L}(X_i) - \mathcal{L}(X_{i-1})|X_{i-1}\right]|X_o = x\right] \tag{9}$$

$$= \mathbb{E}\left[\sum_{i=1}^{n} d_{\mathcal{L}}(X_{i-1})|X_o = x\right] \tag{10}$$

$$< \mathbb{E}\left[\sum_{i=1}^{n} -\varepsilon + (\beta+\varepsilon)\mathbb{1}_{(X_{i-1}\in\mathcal{A})}|X_o = x\right] \tag{11}$$

$$= \sum_{i=1}^{n} -\varepsilon + (\beta+\varepsilon)\mathbb{E}\left[\mathbb{1}_{(X_{i-1}\in\mathcal{A})}|X_o = x\right] \tag{12}$$

Since $\mathcal{L}$ is a positive function $\mathbb{E}\left[\mathcal{L}(X_n) - \mathcal{L}(X_o)|X_o = x\right] \geq \mathbb{E}\left[-\mathcal{L}(X_o)|X_o = x\right] = -\mathcal{L}(x)$. Hence, re-arranging (12), we get

$$-\frac{\mathcal{L}(x)}{n(\beta+\varepsilon)} + \frac{\varepsilon}{(\beta+\varepsilon)} \leq \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(X_{i-1}\in\mathcal{A})}|X_o = x\right] \tag{13}$$

and it follows that

$$\liminf_{n\to\infty} \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{(X_{i-1}\in\mathcal{A})}|X_o = x\right] > \frac{\varepsilon}{\beta+\varepsilon} > 0. \tag{14}$$

Since $\mathbf{X}$ is an irreducible Markov chain and A is a finite set, it follows that for the average time spent in A to be positive, the Markov chain must be positive recurrent from (6). □

The function $\mathcal{L}$ is called the Lyapunov function of the Markov chain. Intuitively, we can think of $\mathcal{L}$ as a potential function on the state space of the chain. Thus the conditions of the theorem can be viewed as trying to prevent $\mathcal{L}(X)$ and thereby X from escaping to infinity. Note though that this is only a sufficient condition for positive recurrence, and also that there is no clear procedure as to how to choose $\mathcal{L}$. In the subsequent sections, we will argue that the throughput optimal policies fall out from particular choices of $\mathcal{L}$ and this therefore means that they are not unique (although still optimal, since that is shown by means of a matching outer bound on the rate region).

3.2 The Queuing Model of a Wireless System

The modeling of wireless networks as queuing systems is a comparatively recent idea in the history of wireless communications. There are two reasons for this-

- Wireless networks were till very recently predominantly used for voice communications. Thus, these networks were primarily based on a circuit switching framework, wherein some minimum resources were guaranteed to users engaged in a call. However, with the development of data-only protocols, network designers are able to impose a packet switching paradigm on these networks. This allows for buffering of data streams at various points in the network. We discussed this in detail earlier in Section 2.1.

- Wireless systems design was viewed as a physical and MAC layer issue, and hence separated from queuing which was a networks layer issue. However, it is becoming increasingly clear that there are gains to be realized from treating the resource allocation problems as a cross layer optimization problem.

At the most abstract level, a wireless network is modeled from the queuing perceptive in the following manner

- **Network Model:** The network is modeled as a graph $G = (V, E)$ where V is a set of nodes representing wireless nodes and E a set of undirected edges representing the wireless links between them.

- **Channel Model:** In every time slot t, each edge $e_{ij} \in E$ has an associated weight of $M_{ij}[t]$ representing the maximum service rate, i.e., maximum transfer of data that can be supported by that link in the present time slot. This is derived from information theoretic considerations, and in this section we assume that the supportable link rates are independent of the scheduling policy (although this can be generalized). Although $M_{ij}[t]$ is in general a random variable, we assume that at the beginning of each tie slot, the channel parameters are estimated and thus the scheduler has available a realization $M_{ij}[t] = \mu_{ij}[t]$ for each edge.

- **Interference Model:** The interference property of wireless networks is modeled in the form of independent set constraints, i.e., by specifying a set $\mathcal{C}$ consisting of every permissible set of edges that can be activated simultaneously. This serves as a constraint set over which the optimization is performed. Although we will deal with simple 'on-off' type links in this section, the results can be generalized and thus the set $\mathcal{C}$ is in most general of the form of the constraint set described in the Section 2.3.

- **Traffic Model:** In the most general scenario, the input traffic to the network consists of a vector of flows $(A_s^d[t], s, d \in V)$ consisting of single packet streams from source s to destination d. These flows are assumed to be i.i.d random variables with an associated first moment vector $\lambda = (\lambda_s^d), s, d \in V, \lambda_s^d = \mathbb{E}[A_s^d[t]]$, which is used to define the throughput region.

- **Queue Model:** Each node $i \in V$ has upto $|V| - 1$ FIFO buffers Q_i^d, one each for flows to all other possible destinations $d \in V \setminus \{i\}$, where the incoming packets destined for that node are stored until transmitted onwards. The buffer size is assumed to be infinite. Finally, we define $X[t] = (Q_i^d[t])_{i,d \in V}$ as the state of the system, and observe that under a policy that depends only on the current queue and channel states for scheduling, $X[t]$ is a Markov chain.

- **Information and Control Model:** At the beginning of each time slot, complete CSI, in the form queue states and channel states, are input to the scheduling policy or algorithm, which calculates the appropriate weights for Section 2.3.

3.3 Optimal Throughput Region

Given a scheduling algorithm, we can find a set of input rate vectors λ such that for any input vector lying in this set, it can be shown that the system is positive recurrent under the chosen policy. This set is called the achievable throughput region of the policy. The primary tool for showing stability of flows in this region is Foster's criterion. However, due to the fact that Foster's criterion is a sufficient result and not a necessary result, it does not guarantee optimality of the scheduling algorithm. Hence we need some other means to characterize the best achievable throughput region, i.e., the union of throughput regions under all possible policies, based on both current information as well as complete history, in order to have a benchmark for the performance of a given policy.

Andrews et al. [4] present a formal framework for constructing outer bounds for problems of this type. The primary idea behind their technique involves characterizing what they call Static Service Split Rules (or SSS rules). These are scheduling rules which depend only upon the channel state and not the queue state. We illustrate this with the example given in [4].

Consider an $N \times 1$ uplink channel, i.e., there are N MS trying to transmit to a single BS on a single channel. The channel state vector is assumed to exist in M possible states from set $\mathcal{M}$, where each state $m \in \mathcal{M}$ is a vector $(\mu_i^m)_{i \in \{1,2,\ldots,N\}}$ of rates for each MS in that channel state, and the channel exists in state m with probability π_m in each time slot. For this channel, Andrews et al. [4] define a SSS rule as being parametrized by a stochastic matrix ϕ. When the channel is in state m, the SSS rule chooses to serve MS i with probability ϕ_{mi}. Thus the SSS rule only depends on the channel state and not queue lengths.

Using this framework of SSS rules, the paper goes on to define the long-term rate offered to user i by a given rule ϕ as

$$\nu_i(\phi) = \sum_m \pi_m \phi_{mi} \mu_i^m$$

.

Finally, the optimal throughput region of this system is characterized using the following theorem.

Theorem 2. ***(Characterization of Optimal Throughput Region)*** *A scheduling rule exists under which the system is stable if and only if there exists an SSS rule ϕ such that:*

$$\lambda < \nu(\phi). \tag{15}$$

One subtle point here is that the theorem seems to suggest (and in fact this is made explicit in the proof) that given an input rate vector λ, the systems is stabilized by operating using any SSS rule that satisfies (15). Hence, the optimal rate region can be written as

$$\mathcal{V} = \{\nu(\phi) : \phi \text{ stochastic}\}$$

. It has become evident that achieving stability using SSS policies requires knowledge of the arrival rates of the flows. While one can estimate these arrival rates, our objective in the following is to find develop policies (preferrably simple in nature) that have no apriori knowledge of the flows. In the next section, we discuss a policy that implicitly uses the queue length to acquire information about the arrival rates and thus is able to adapt to varying arrival rates within the rate region.

3.4 The Backpressure Algorithm

The main contribution of Tassiulas and Ephremides [3] is a queue-based algorithm that achieves the optimal throughput region described in the previous section. This is the celebrated backpressure algorithm, which we will outline in this section, and give a sketch of the proof of optimality in the subsequent sections.

From our discussion on SSS rules, it would appear to the reader that the ideal scheduling policy given an input rate vector λ would be to choose an SSS rule that dominates the input rate vector and operate at that rule to stabilize the system. The difficulty in this policy is that it assumes a knowledge of the input rate vector beforehand, and also that the policy has to change whenever there is a change in the input flows. What one ideally wants however is a policy that stabilizes all flows within the throughput region without any a priori knowledge of λ. Queue based policies are attractive due to this precise property of being able to stabilize any flow within the region without requiring any knowledge of its distribution. Intuitively, one can visualize such a policy as one which uses the queue information as a proxy to learn the input rate vector and iteratively find the optimal SSS to operate at.

The BP algorithm takes as input the current channel states and queue states in a time slot t, and outputs a set of edges that are to be activated, as well as the precise flow which should be transferred on each activated edge. The algorithm can be broken down into three stages-

- In the first stage, each edge e_{ij} calculates a weight w_{ij} using the following formula

$$w_{ij}[t] = \max_{d \in V \setminus \{i,j\}} Q_i^d[t] - Q_j^d[t] \tag{16}$$

  This quantity is often called the backpressure across a link, which is how the algorithm gets its name. The algorithm also stores a destination d_{ij}^* which corresponds to the destination of the flow which maximizes equation 16. Note that by default a destination for a flow is assumed to have a size 0 buffer for that flow, i.e., $Q_d^d = 0 \forall d \in V$.

- In the next stage, the algorithm chooses a set of links $C^*[t] \in E$ such that-

$$C^*[t] = \arg \max_{C \subseteq V, C \in \mathcal{C}} \sum_{(i,j) \in C} w_{ij} \min(\mu_{ij}, Q_i^{d_{ij}^*}[t]) \tag{17}$$

  In other words, we find the set of edges that can be activated simultaneously and maximizes the weighted sum of rates that can be transferred across these links. The min is present in the formula to take care of buffer underflow.

- Finally, after finding the optimal $C^*[t]$, the algorithm instructs each edge $e_{ij} \in C^*[t]$ to activate and drain the queue $Q_i^{d_{ij}^*}[t]$ corresponding to the maximum backpressure flow across the link.

3.5 Optimality of BP

To show that the BP algorithm is optimal, we need to show that given an input rate vector λ that lies in the stability region, the system is positive recurrent under the BP algorithm. This is shown by choosing an appropriate candidate Lyapunov function and using Foster's criterion. The Lyapunov function used by Tassiulas and Ephremides is the quadratic Lyapunov, i.e.

$$\mathcal{L}(X[t]) = \sum_{i,d \in V} \left(q_i^d[t] \right)^2 \tag{18}$$

The complete proof is quite technical due to the generality of the model, but the basic intuition behind it is that when the difference equation for the change in the expected value of the Lyapunov function is expanded out, it can be shown that all the positive terms which are independent of queue state can be upper bound, and that the negative terms associated with the queue state are precisely those which are maximized by the BP algorithm. Now, by choosing an input rate vector λ that lies inside the throughput region obtained in Section 3.3, the function can be shown to be a suitable Lyapunov function, and hence the system is positive recurrent by Foster's criterion. In the next section, we present an alternate approach for this proof, which although requiring many more technical lemmas, is useful for building intuition regarding these proofs.

3.6 The Fluid Limits Approach

Andrews et al. [4] give an alternate method for deriving the stability regions of scheduling policies by studying the properties of an alternate fluid process for the queuing system. The basic idea behind this approach is that under a certain time and amplitude scaling, the sample paths of the queuing process can be shown to converge almost surely to a deterministic process which is called the fluid limit of the system. Further, the dynamics of the fluid model is identical in form to the dynamics of the discrete model except that it is now given by a differential equation rather than a difference equation. This deterministic system can therefore be studied using traditional Lyapunov analysis, and stability of its paths under different initial conditions and input processes can be determined. Finally, the authors use a theorem of Malyshev and Menshikov to show that the stability of the fluid process can be used to demonstrate positive recurrence of the underlying queuing process.

The development of the fluid limit is very technical and does not provide much intuition into the stability analysis itself. The main result that emerges from the analysis is that the queues in the fluid process are differentiable, and the dynamics of the queuing system is given by-

$$\dot{q}_i^d(t) = \lambda_i^d(t) + \left[\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right]^+ \tag{19}$$

Where $t \in \mathbb{R}_+$ is now a continuous time, and we define the following fluid processes

$q_i^d(t)$: The queue at node i for data packets destined for node d

$\lambda_i^d(t)$: The average external input flow at node i of data packets destined for node d

$\mu_{\text{in}(i)}^d(t)$: The sum of flows in to node i from neighbouring nodes of data packets destined for node d

$\mu_{\text{out}(i)}^d(t)$: The sum of flows out from node i to neighbouring nodes of data packets destined for node d

$\mu_{ij}^d(t)$: The flow across link (i,j) of data packets destined for node d

We define $X(t) = (q_i^d(t))_{i,d\in V}$ to represent the state of the system at time t, whose dynamics is governed therefore by the above equations. Finally we define the equivalent scheduling policy as-

$$\arg\max_{C\in\mathcal{C}} \sum_{(i,j)\in C, d\in V} \max_{d\in V}\left[\mu_{ij}^d(t)\left(q_i^d(t) - q_j^d(t)\right)\right] \tag{20}$$

Finally we present a sketch of the Lyapunov analysis that is required to prove the stability of the fluid process. We define $\mathcal{L}(X(t)) = \sum_{i,d\in V}\left(q_i^d(t)\right)^2$. Thus we have-

$$\begin{aligned}
\dot{\mathcal{L}}(X(t)) &= \sum_{i,d\in V} 2q_i^d(t)\dot{q}_i^d(t) \\
&= \sum_{i,d\in V} 2q_i^d(t)\dot{q}_i^d(t) \\
&= \sum_{i,d\in V} 2q_i^d(t)\left(\lambda_i^d(t) + \mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right) \\
&= \sum_{i,d\in V} 2q_i^d(t)\lambda_i^d(t) + \sum_{i,d\in V} 2q_i^d(t)\left(\mu_{\text{in}(i)}^d(t) - \mu_{\text{out}(i)}^d(t)\right) \\
&= \sum_{i,d\in V} 2q_i^d(t)\lambda_i^d(t) - 2\sum_{(i,j)\in E, d\in V} \mu_{ij}^d(t)\left(q_i^d(t) - q_j^d(t)\right)
\end{aligned}$$

Note however that the last term is precisely what is being maximized by the policy in equation 20. Thus, given a λ which lies in the rate region, we can show that $\mathcal{L}(X(t))$ is a valid candidate Lyapunov function for this system, and hence the underlying queuing system is stable.

References

[1] D. Tse and P. Viswanath, "Fundamentals of wireless communication", Cambridge University Press, 2005.

[2] J. G. Andrews, A. Ghosh and R. Muhamed, Fundamentals of WiMAX, Prentice-Hall, June 2007.

[3] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Trans. Automatic Control*, vol. 37, pp. 1936-1949, Dec. 1992.

[4] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar and P. Whiting, "Scheduling in a queueing system with asynchronously varying service rates", *Probability in the Engineering and Informational Sciences*, vol. 18 , pp. 191-217, Apr. 2004.

[5] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao and M. Andrews, "An overview of limited feedback in wireless communication systems", *IEEE Journ. Sel. Areas Commun.*, vol. 26, pp. 1341-1365, Oct. 2008.

[6] J.Chen, R. A. Berry and M. L. Honig, "Large system performance of downlink OFDMA with limited feedback",*IEEE Intern. Symp. Info. Theory*, Seattle, WA, July 2006.

[7] J. Huang, V. Subramanian, R. Agrawal, and R. Berry, "Joint scheduling and resource allocation in uplink OFDM Systems for broadband wireless access networks", *IEEE Journ. Sel. Areas Commun.*, vol. 27, pp- 226-234, Feb. 2009.