A DYNAMIC RESOURCE ALLOCATION STRATEGY FOR SATELLITE COMMUNICATIONS

Aradhana Narula-Tam
MIT Lincoln Laboratory
Lexington, MA

Thomas Macdonald
MIT Lincoln Laboratory
Lexington, MA

Eytan Modiano
MIT LIDS
Cambridge, MA

Leslie Servi
MIT Lincoln Laboratory
Lexington, MA

Abstract

The focus of this paper is dynamic resource allocation algorithms for sharing the limited uplink resources of a future satellite system among many bursty users with varying QoS requirements. The data rates provided to each terminal are selected to differentiate between multiple QoS priority levels, provide fairness, and to maximize system capacity under time-varying channel conditions and traffic loads. The proposed resource schedulers are compared to alternative approaches and shown to provide dramatic improvements in both the average data rates and delay characteristics experienced by the users.

I. Introduction

Future protected military satellite communications will continue to use high transmission frequencies to capitalize on the large amounts of bandwidth that are available at these frequencies. However, these future systems will communicate Internet-like packet traffic, rather than the circuit-switched communications that are prevalent today. One of the main distinguishing features between packet-switched and circuit-switched traffic is that the packet traffic is bursty (i.e., the data rate needed to support the service is not constant). In addition to the variation in the demands placed on the system due to the bursty nature of the packet traffic, there are numerous other system variations. These include changes in the link quality experienced by each terminal due to weather, mobility, jamming, and other factors. At the frequencies that the satellite systems operate rain along the propagation path can result in many decibels of additional attenuation. However, such rain events only occur a small fraction of the time. The data rates achievable by each terminal are dependent not only the capabilities of the terminal but also the current link conditions and the resources allocated to that terminal. In this paper a technique is presented for dynamic resources allocation for a future satellite communications system. The details presented correspond to the satellite uplink, but the allocation algorithm could be applied to either the uplink or downlink.

The goal of the dynamic allocation scheme presented in this paper is to support bursty packet traffic, maximize system capacity, insure fair allocation of resource across the terminal population, differentiate among quality of service (QoS) queues, and allow operation through variations in the system environment. One technique for supporting bursty traffic and compensating for adverse weather conditions is to vastly overprovision the resources given to each user. In fact, this is how the current system operates. However, such a scheme limits the capacity of the satellite system, both in terms of the data rate granted to each user and the number of users in the system. Future systems will have the ability to adapt both the terminal's information transfer rate and transmission time based on link conditions and traffic loads [1]. A dynamic algorithm for allocating resources is presented in this paper. This algorithm is opportunistic [2], in that it exploits channel variations to give terminals more resources when their link quality is high, thereby achieving greater overall capacity for the system and greater throughput for each user. The proposed algorithm is also throughput optimal, which means that the algorithm will stabilize the queues whenever the input rates are within the system's stability region. In other words, if there exists any algorithm that can stabilize all the queues in the system, this algorithm will also stabilize the queues. The performance of the new algorithm is evaluated and compared against a number of other allocation schemes for a variety of metrics such as throughput, delay, fairness, and QoS differentiation.

The paper is organized as follows. A description of the satellite system is given in Section II and an overview of the proposed dynamic allocation schemes are given in Section III. In Section IV the proposed allocation schemes are compared against a number of other allocation schemes in terms of average throughput and average delay. Issues associated with achieving QoS using the proposed schedulers are discussed in Section V.

This work was sponsored by the United States Air Force under Air Force Contract #F19628-00-C-0002. Opinions, interpretations, recommendations and conclusions are those of the authors and are not necessarily endorsed by the United States Government.

II. System Description

One of the challenges with a dynamic allocation algorithm for satellite systems is the long propagation delay. Military satellite communications typically employ satellites in geosynchronous orbit, and so the round trip delay along a single link is approximately 250 msec. The system under consideration uses both time and frequency multiplexing on the uplink. All of the users are synchronized to the time source in the satellite and system access is coordinated via a reservation-based scheme. In order to accommodate the long propagation delay and allow for sufficient processing, an allocation cycle or epoch of 640 msec is employed. This epoch is broken up into a series of 4 msec time slots. During each epoch a terminal will be allocated an integer number of time slots. The system also has a number of different channel symbol burst rates, modulation formats, and forward error correction coding rates. The triplet of modulation format, coding rate and burst rate is referred to as a *mode*. There are five different burst rates, two code rates, and four modulation formats.

Table I gives the number of bits transferred in a time slot for each mode of operation. The fifteen modes of operation are separated into five columns corresponding to the burst rate associated with each mode. The first column of the table shows the symbol alphabet size for the corresponding modes. The second column shows the Forward Error Correction coding rate associated with the modes. The relative signal-to-noise ratio P_r/N_0, needed to support each of the fifteen modes is given in Table II. The parameter P_r/N_0 for each mode is the required signal-to-noise ratio relative to the signal-to-noise ratio necessary to support the lowest data rate mode. All of the values in the tables are given for completeness, but the important information to note is that this a complicated system that supports a wide variety of information transfer rates. Notice that there is factor of 2000 increase in data rate from the lowest rate mode to the highest rate mode, and that this highest rate mode requires 35 dB more power than the lowest rate mode. It is not anticipated that a single terminal would be able to span this entire range, but it is anticipated that there will be a wide variety of terminal types that will access the system and that there will be a wide range in the capabilities of the terminals. The different burst rates also require different allocations of channel bandwidth. Let γ denote the bandwidth required for burst rate 1. The bandwidth required for burst rate 2 is 8γ, burst rate 3 is 28γ, burst rate 4 is 84γ, and burst rate 5 is 336γ. As the symbol alphabet size increases (traversing the table downward), the bandwidth efficiency also increases. It may seem prudent to always operate in the most bandwidth efficient mode in order to maximize total system throughput. However, a terminal may not always have sufficient signal-to-noise ratio (SNR) to achieve the desired bit rate while operating in the most bandwidth efficient regime.

TABLE I
Information transfer (bits/time slot) for each mode

Sym Alph	Code Rate	Burst Rate 1	Burst Rate 2	Burst Rate 3	Burst Rate 4	Burst Rate 5
2	1/2	256	2048	8192	24576	-
4	1/2	512	4096	16384	49152	196608
8	2/3	-	8192	32768	98304	393216
16	2/3	-	-	-	131072	524288

TABLE II
Required relative P_r/N_0 in dB for each mode

Sym Alph	Code Rate	Burst Rate 1	Burst Rate 2	Burst Rate 3	Burst Rate 4	Burst Rate 5
2	1/2	0	7	12	17	-
4	1/2	3	10	15	19	26
8	2/3	-	16	22	26	33
16	2/3	-	-	-	28	35

Due to the reservation based nature of the system access, each terminal is granted at least one time slot per allocation epoch. This time slot allows the terminal to send in requests for the next time slot. In addition, for burst rate 1 the minimum number of time slots that can be granted per epoch is increased to guarantee that a useful amount of information can be transferred in an epoch. For the modulation format with the symbol alphabet of size 2 a minimum of four time slots per epoch are granted for burst rate 1 and for the 4-ary symbol alphabet a minimum allocation of two time slots per epoch is enforced for burst rate 1. In order to reduce the complexity of the earth terminals, an additional system constraint is imposed that requires all the time slots that are granted to a terminal in an epoch to be in the same mode. The mode can change between epochs, but within an epoch a terminal only uses a single mode.

III. Allocation Algorithms

The allocation algorithm under consideration in this paper partitions resources among R terminals within a single uplink beam. Each terminal has Q different QoS queues. The allocation algorithm resides in the satellite payload and during each epoch each terminal transmits the amount of information stored in each of its queues to the payload. In parallel, the payload computes the link quality for each terminal based on transmissions from that terminal. The link quality estimation is not the focus of this paper, so it is assumed that accurate link quality information is available to the payload. This assumption has been validated through other system design activities.

There are several other system parameters. Let $U_{r,q}$ denote the queue length of the QoS queue q of terminal r. Define b_m as the burst rate (bits/slot) of a terminal in mode m and let β_m be the bandwidth required for a terminal to operate in mode m. Define the total bandwidth available as β and the total number of slots in each epoch as S. To differentiate between the QoS levels of each queue, a weight parameter is used. Define the weight of QoS level q as w_q. Higher priority queues are given larger weights. Define $L_{r,q,m}$ as the number of slots assigned to queue q of terminal r. The allocation algorithm establishes $L_{r,q,m}$, the number of slots assigned to each queue on an epoch by epoch basis. Note that for each terminal and queue only one $L_{r,q,m}$ will be nonzero, since the terminal only operates in one mode per epoch.

This scheduling problem is similar to the scheduling problem in cellular wireless communication systems. One notable difference, however, is that in the satellite system, the scheduler assigns slots and bandwidth on an epoch by epoch basis, rather than on a slot by slot basis. As mentioned above, for the satellite system there are several constraints that apply on an epoch time scale. For example, the terminal mode, and hence its burst data rate is fixed during each epoch. Also, each terminal must be given a minimum allocation during each epoch. Finally, there are multiple queues at each terminal, all of which transfer data at the same burst rate.

For slotted wireless communication systems, several algorithms that are throughput optimal have been developed for both satellite [3][4]and HDR data services [5][6]. A throughput optimal scheduling algorithm, the maximum weight rule, which services users based on the channel state and queue backlog was proposed and studied for a multi-beam satellite downlink channel in [4]. In [4], the limited resource, power, was allocated on a slot by slot basis. In the satellite system studied here, time and bandwidth are the limiting resources. The maximum weight rule in the context of limited bandwidth corresponds to selecting the users service rates at each time slot in order to maximize the sum of the weighted queue length of the users served:

$$\max_{L_{r,q,m}} \sum_{r,q} w_q U_{r,q} L_{r,q,m} b_m, \tag{1}$$

subject to a constraint on the total bandwidth available. Here $L_{r,q,m}$ is either one or zero, depending on whether the corresponding queue is operating in mode m and whether it is assigned a time slot. Multiple terminals can be serviced in each time slot by allocating orthogonal frequency channels to each terminal. The number of terminals that can be serviced is dependent on the terminal bandwidth requirements and the total bandwidth available. This algorithm can not be applied directly to the satellite system because epoch constraints, such as requiring each terminal to use the same mode for the entire epoch, are difficult to enforce on a scheduler which operates on a slot basis. Alternatively, applying the rule of Equation (1) directly to each epoch is also suboptimal. In the satellite, the number of slots assigned to each queue, $L_{r,q,m}$, is at most S. Unfortunately, during each epoch, the maximum weight rule assigns the maximum number of slots to the queues with the largest weighted queue length. Each queue that is given service, is given a maximum number of slots, while other queues are given no resources for the entire epoch. For example, consider a scenario with two users (two terminals, each with a single queue) and a single bandwidth channel. Let each epoch consist of 100 slots. Now suppose user 1 has a queue length of 100 and user 2 has a queue length of 99. If the weights and signal-to-noise ratio are the same for both users, then user 1 will be assigned all 100 slots in this epoch rule, whereas ideally, 50 slots should be assigned to each of the two users.

A. Equitable Weighted Queue Length Scheduler

In this section an allocation algorithm is proposed that ensures fair service to queues of equal priority *within* each epoch. Therefore, instead of maximally serving only the queues with the largest weighted delays first, the proposed scheduler services the queues in a manner that equalizes the weighted queue lengths at the end of each epoch. Thus terminals are given larger allocations when their channel is better, but the queue lengths are not allowed to grow too large and similar service levels are provided to all queues of the same weight. Formally, the rule allocates slots to minimize the weighted square of the end-of-epoch queue length:

$$\min_{L_{r,q,m}} \sum_q w_q \sum_r (U_{r,q} - \sum_m L_{r,q,m} b_m)^2 =$$
$$\max_{L_{r,q,m}} \sum_{r,q,m} w_q L_{r,q,m} b_m (2U_{r,q} - L_{r,q,m} b_m). \qquad (2)$$

The resulting objective function is quadratic. The scheduler, referred to as the Equitable Weighted Queue Length (EWQL) scheduler, can be implemented as a quadratic mixed integer program as described below.

The objective function, Equation (2), is maximized subject to the following constraints.

1. A constraint on the total resources (bandwidth × slots) used:

$$\sum_{r,q,m} L_{r,q,m} \beta_m \leq S\beta. \qquad (3)$$

2. A constraint on the total number of slots assigned to any terminal. For each terminal:

$$\sum_q \sum_m L_{r,q,m} \leq S. \qquad (4)$$

3. The service at each queue must not exceed the queue length:

$$\sum_m L_{r,q,m} b_m \leq U_{r,q}. \qquad (5)$$

4. Each terminal must be given a minimum number of time slots depending on its mode. Let $1/\alpha_m$ be the minimum number of slots assigned to a terminal that is using mode m.

$$\sum_{q,m} \alpha_m L_{r,q,m} \geq 1. \qquad (6)$$

5. All queues at each terminal must use the same mode within an epoch. Let $I_{r,m}$ be an indicator variable which is 1 if terminal r is in mode m and 0 otherwise.

$$L_{r,q,m} \leq S I_{r,m} \qquad (7)$$

6. Each terminal uses only one mode for a particular epoch.

$$\sum_m I_{r,m} = 1. \qquad (8)$$

It is also ensured that the signal-to-noise ratio at each terminal is sufficient to support the selected mode. The mode of each terminal and the number of slots assigned to each queue are determined by the algorithm. Note that in this formulation the total resources (slots × bandwidth) are constrained. However, slots and bandwidth are not interchangeable, and hence there may be situations where the resulting solution can not be packed into the time/frequency allocation space. Other investigations have shown that for typical problems, the packing problem is very tractable and there is little consequence of using this simplification.

The quadratic mixed integer program above can be easily solved for small problems, but as the number of terminals increases the complexity becomes untractably large. In Section IV, simulation results for a system with 4 terminals and 8 QoS levels are presented. Described below is a lower complexity heuristic which can be used to solve much larger problems.

B. Heuristic Scheduler

The difficulty in solving the scheduling algorithm above is due to both the integer variables, corresponding to the terminal mode selection, and the quadratic objective function. Described below is a heuristic allocation algorithm which simplifies the problem by dividing it into two programs, the first of which is a mixed integer linear program and the second is a quadratic program with only continuous variables.

In the first step of the heuristic the mode of each terminal is determined. In this step, the maximum weight objective function, Equation (1), is applied on an epoch by epoch basis. The epoch constraints outlined above are are integrated to form a mixed integer linear program that solves quickly. The time slot allocations produced from this program are ignored. Simulations show, however, that the mode selections from this step are quite reasonable. Intuitively, appropriate mode selection may be facilitated by the constraint that each terminal requires at least one slot per epoch. Hence the scheduler attempts to put each terminal in a reasonable mode. In the second step, the mode of each terminal is fixed according to the results of the mixed integer linear program. Hence, the indicator variables $I_{r,m}$ are inputs to the second program rather than variables. The slot variables, $L_{r,q,m}$ are relaxed to be continuous and hence all variables are continuous. The objective function for the second program is the quadratic objective function of Equation (2). Both programs solve considerably faster than the quadratic mixed integer program. The heuristic runs more than 50 times faster than the quadratic mixed integer program. The performance of the two schedulers is compared in the following section.

This scheduler is referred to as the Heuristic scheduler.

IV. Simulations and Comparison with Other Schedulers

The proposed schedulers are compared to two alternative system schedulers. The first is a Fixed Rate system, where the mode and time slots dedicated to each terminal are fixed. The mode is selected to ensure transmission is possible in poor signal-to-noise ratio scenarios even though these conditions occur very rarely. This is representative of the way the current system operates. Essentially a fixed data rate circuit is appropriated to each terminal. The queues at each terminal are serviced in strict priority order.

The second scheduler implemented is a Strict Priority Round Robin scheduler. The queues are examined in priority order. The mode of each terminal and the time slot allocation are first selected to empty the highest priority queue at that terminal. If resources are available after ensuring service to the current priority level queues at each terminal, then queues at the next priority level are examined. The mode of each terminal and the slot allocation are modified to empty all queues up to the current QoS level until all resources are allocated. The service order of the terminals is randomly selected to ensure fairness.

Simulation results are presented for a system with 4 terminals and 8 QoS levels. The arrival rate of each queue is modeled as Poisson with mean of 1357 kbits/epoch. The total bandwidth available is 336γ. The time-varying nature of the channel is modeled using a Markov model. Each terminal, depending on its location, may experience different weather conditions. A terminal's channel is assumed constant during an epoch and varies from epoch to epoch according to the Markov model in Figure 1. Each state change corresponds to a 2 dB change in signal-to-noise ratio, for a total range of 12 dB. The model is representative of rain moving across the region of operation. For simplicity, a single terminal type is assumed; the signal-to-noise ratio varies from 16 dB to 28 dB.

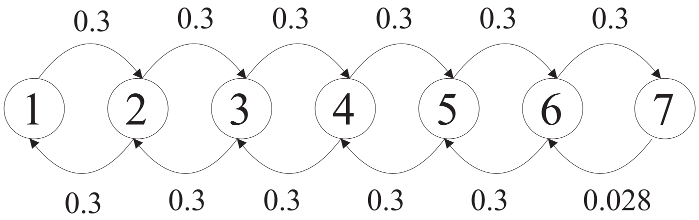

Fig. 1. The seven-state Markov model of the channel state.

For the EWQL and Heuristic schedulers, the values of the priority weights w_q must be selected to ensure sufficient differentiation for each of the QoS levels. In the simulations below, it is assumed $w_{q-1} = 10w_q$, (i.e., bits of each priority level are 10 times more important than bits in the level below). Different weight levels result in different treatment for the different quality of service levels, and hence affect the delay and rate experienced by these queues.

The schedulers were employed for 10,000 epochs. The average rate of the four scheduling algorithms is shown in Figure 2. The Fixed Rate system is only able to provide sufficient rate to service the highest QoS level queue. The top three QoS level queues are given sufficient rate under the Strict Priority Round Robin (SPRR) scheduler. With the EWQL and Heuristic schedulers, the average service rate at *all* queues is sufficient to meet demand. The EWQL and Heuristic schedulers are able to provide higher average rates by giving more service to terminals when their channel conditions are favorable. This is illustrated in time sample of Figure 3, which shows the signal-to-noise ratio of terminal 2 and the bit rate allocated to the terminal under the four scheduling rules as a function of the epoch number.

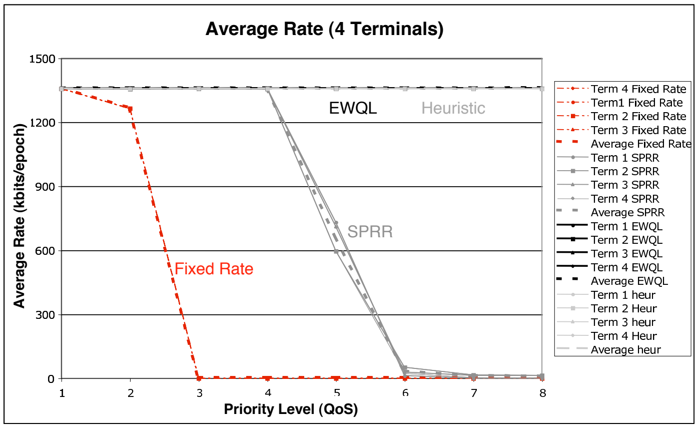

Fig. 2. The average service rate for each QoS level queue.

In Figure 4, the average delays of the queues under the four scheduling algorithms are compared. Only the EWQL and Heuristic schedulers are able to stabilize all eight QoS queues. For the other schemes, the lower priority queues grow without bound. The average delay is slightly lower for the highest priority queue using the Fixed Rate and Strict Priority Round Robin schedulers, because these algorithms are designed to service the highest priority queue to completion first. Of course, increasing the weight of the highest priority queue in the EWQL and Heuristic schedulers can decrease the average delay experienced by the highest priority queue at the expense of slightly increasing the delays of the other queues.

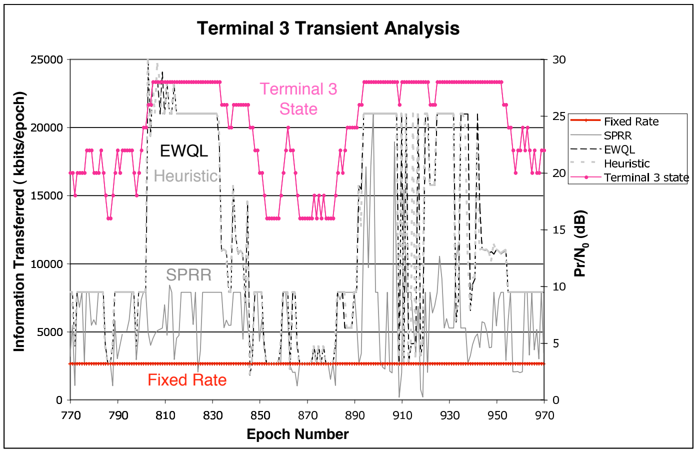

Fig. 3. The rate allocation and P_r/N_0 for terminal 2 for a representative series of epochs.

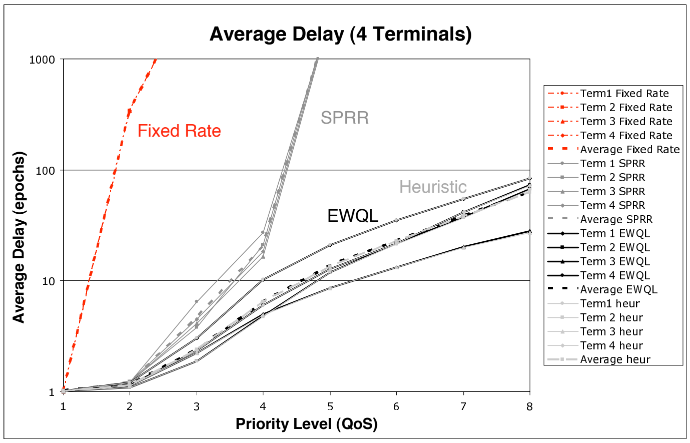

Fig. 4. The average delay experienced by each QoS level is shown under the four allocation algorithms.

The average delay and rate achieved by the Heuristic scheduler is very similar to that of EWQL for this example. The two rules performed similarly for other simulation cases as well, including scenarios where the maximum weight epoch rule alone would assign the maximum number of slots to the terminals serviced and the minimum number to other terminals. Due to its reduced complexity, the heuristic allows larger problems to be scheduled.

V. Achieving QoS

The Heuristic and EWQL schedulers can be used to provide QoS to multiple queues in a realistic satellite system. By assigning different weights w_q to the queues at each terminal, service differentiation among the various queues can be achieved. The scheduler provides fairness among queues with the same weight and QoS level. There are multiple ways of achieving a guaranteed rate service. To achieve a guaranteed *average* rate, virtual queue lengths can be used to request a fixed number of bits every epoch. As described in [7], these virtual queues are given large weights to ensure their required data rate is received. Alternatively, to achieve a guaranteed rate per epoch for some queues, a constraint can be added to the program to ensure a minimum fixed rate is given to certain queues. A minimum rate guarantee for a queue can be achieved by dividing the real queue into two queues consisting of a virtual queue and a second queue. The virtual queue length is determined by the minimum rate requirement, while the second queue length is the non-negative difference between the real queue length and the virtual queue length [7]. Again, the virtual queue is given a large weight to ensure the minimum rate is allocated.

The proposed schedulers can guarantee QoS and stabilize the system when the arrival rates of all queues are within the feasibility region of the system. To ensure arrival rates are feasible, connection admission control in combination with policing must be used to prevent the system from becoming overloaded [8].

VI. Conclusion

Future satellite systems will carry bursty packet switched traffic. To efficiently utilize the uplink it is necessary to consider opportunistic schedulers which provide more resources to terminals under favorable channel conditions. An opportunistic allocation algorithm for a satellite system with epoch constraints is developed which stabilizes the queues of all users whenever the set of rates are feasible under any rule. This rule ensures fairness and allows QoS differentiation. The achievable rates are shown to be significantly larger than those achievable with non-opportunistic scheduling schemes. A lower complexity heuristic rule that provides very similar performance is also presented.

References

[1] C. E. Fossa Jr. and T. G. Macdonald, “Dynamic resource allocation for satellite communications,” in *IEEE Wireless Communications and Networking Conference*, Mar. 2004.

[2] X. Liu, E. Chong, and N. Shroff, “A frameowrk for opportunistic scheduling in wireless networks,” *Computer Networks*, vol. 41, no. 4, pp. 451–474, Mar. 2002.

[3] M. Neely, E. Modiano, and C. Rohrs, “Power and server allocation

and routing in a multibeam satellite with time-varying channels," in *IEEE Infocom*, June 2002.

[4] M. Neely, E. Modiano, and C. Rohrs, "Power allocation and routing in multibeam satellites with time-varying channels," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 138–152, Feb. 2003.

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

[6] S. Shakkotai and A. L. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: the exponential rule," *Analytic Methods in Applied Probability*, pp. 185–202, 2002.

[7] S. Shakkotai and A. L. Stolyar, "Scheduling algorithms for a mixture of real-time and non-real-time data in HDR," in *17th International Teletraffic Congress*, Sept. 2001, pp. 793–804.

[8] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar, "Providing quality of service over a shared wireless link," *IEEE Communications Magazine*, pp. 150–154, Feb. 2001.