Ocelot's Knapsack Calculations for Modeling Power Amplifier and Walsh Code Limits

Kenneth L. Clarkson†, and John D. Hobby*
* Bell Laboratories, Alcatel-Lucent
† IBM Research (work was done at Bell Laboratories)

***Abstract*— We give a model for the performance impact on wireless systems of the limitations of certain resources, namely, the base-station power amplifier and the available OVSF codes. These limitations are readily modeled in the *loss model* formulation as a *stochastic knapsack*. A simple and well-known recurrence of Kaufman and Roberts allows the predictions of the model to be efficiently calculated. We discuss the assumptions and approximations we have made that allow the use of the model. We have included the model in Ocelot, a Alcatel-Lucent tool for modeling and optimizing cellular phone systems. The model is fast to compute, differentiable with respect to the relevant parameters, and able to model broad ranges of capacity and resource use. These conditions are critical to our application of optimization.**

I. Introduction

There are many conditions that reduce the performance of cellular phone systems. Several of these conditions are limitations of shared resources. The theory of *loss model systems* studies the properties of multiple services contending dynamically for a common resource. This note describes the assumptions and approximations the authors have made to apply the loss model formulation to two such common resources: the base-station power amplifier (or *PA*), and the set of OVSF codes used in the forward radio link (from base-station to mobile phone). These resources limit both *circuit* and *packet* services; here we will be mainly concerned with circuit services, with some discussion of how their modeling interacts with the modeling of packet services.

Although the loss model setting is natural and appropriate for this modeling task, it is not a perfect fit. Section II-A below gives some assumptions and approximations that we must make to use loss model results to capture the performance effects of the power amplifier and the OVSF codes. For example, the power needs of a call can vary continuously, but the simplest loss model results assume resources are measured in discrete units. Some discretization is therefore needed; our approach to that task is outlined in Section II-B. Next, after giving a more formal description of the loss model calculations for each resource separately, in Section II-C, we describe the assumptions and approximations that are needed to allow the combined effects of the limitations of these resources, in Section II-D. Finally, Section II-E discusses packet data and gives the relevant assumptions.

Section III discusses our overall performance estimate. Section IV discusses a simple way to model a system having circuit services with two particular priority types. While not used in Ocelot, such a scheme shows the flexibility of the modeling scheme used. Then Section V tests our assumptions and evaluates the accuracy of the performance estimates.

II. Loss Model Systems

Loss model results apply to the following situation. (See, for example, [1], [2].)

There is a collection of "jobs," or "calls," contending for a resource; there are M units of the resource, for some M, and there are K distinct kinds of jobs, where job type k requires b_k units of the resource. Once *admitted*, a job uses the resource until the job is done. If admission of a job would raise the total number of units of the resource above M, the job is not admitted: it is *blocked*. It is assumed that jobs arrive at random, and take a random amount of time to be done. We are interested in estimating the *blocking probability* under these conditions, the probability that when a job arrives, too few units of the resource remain unused.

We next review the assumptions needed to apply this framework to the power amplifier, and to OVSF codes (hereafter referred to colloquially as *Walsh* codes, although we mean the more general class of codes).

A. Preliminary Assumptions

While a call (that is, job) of a circuit service uses a constant number of Walsh codes over the time the job is in the system, this is only true approximately for PA usage under power control: fading, both slow and fast, results in random variation in the power demands of the call. It is possible to model this variation, but for now we assume that power remains at a fixed level for the duration of a call. Although there are "continuous stochastic knapsack" models, we simply use a finite K and discretize power demand.

Approximation 2.1: **Power discretization.** Power will be discretized so that M^A units of power, called *bins*, will be available from the PA. The power demands of the calls will be further discretized to use units of $b_i^A = 1, 2, 4 \ldots$ bins.

Since calls far away from the base-station (that is, at high pathloss) use more power, we cannot assume that the power is fixed for each service; even for voice, some calls may take 1 bin, others 2, others 4, and so on. That is, the appropriate "job type" for analyzing power is not service but what we will call the *power demand class*. Section II-B has more details on the way this discretization is done

The coarse rounding we do is partly justified by a study [3] showing that blocking results are relatively insensitive to whether we model calls by different job sizes, or simply by the mean of those sizes; on the other hand, the range of power demands we will consider is quite high, so it seems inadvisable to ignore per-call variations in power demand.

Approximation 2.2: **Walsh code additivity.** If the total number of Walsh codes requested is below a given limit, then the requests can be satisfied.

This is an approximation, because a service needing 2^j Walsh codes will be allocated a set of codes of the form $(k-1)2^j, (k-1)2^j+1, \ldots, k2^j-1$, for some k; it cannot be allocated an arbitrary set of 2^j codes. Moreover, the allocation decision must be done "on-line," as calls arrive.

As our experimental results of Section V show, this assumption has a substantial effect, but an empirical adjustments to the admission threshold help a great deal.

Assumption 2.3: **Poisson arrival.** Jobs arrive as a Poisson process, and their completion time is a random process. If a job is blocked, it goes away (is *cleared*).

A Poisson process has an associated parameter, its mean λ, and we assume that the completion time has mean $1/\mu$; the key parameter for us is $\rho \equiv \lambda/\mu$, the *load*, and the load of expected jobs of type k is ρ_k, where $\rho_k = \lambda_k/\mu_k$.

B. Continuous versus Discrete Power Demands

In practice, loads are not naturally divided into discrete power demand classes. Instead, we have a series of incremental load contributions with some way of computing power requirements for each. For example, a load contribution $\bar{\rho}$ may require some non-integer number of power bins $\bar{b}$. We cope with this by defining a weighting function w_i for each power demand class i, and contributing $w_i(\bar{b})\bar{\rho}$ to each power demand class i whose weighting function is nonzero at $\bar{b}$.

There are many ways to define the functions w_i. For brevity, we shall just list their main properties: they should be smooth and continuous; they should add up to 1; few of them should be nonzero simultaneously; and wherever possible. $\bar{b} = \sum_i b_i^A w_i(\bar{b})$.

C. Loss Models

Up to the approximations we have discussed, the loss model systems we have are two instances of a *stochastic knapsack*. The blocking probabilities, and other properties, of the stochastic knapsack can be computed provably and exactly using an efficient calculation, which we next review. This model, and calculation, has seen many applications in network modeling. It has also been applied to model the limitations of wireless systems with respect to reverse-link interference,[4] an area in which we have not applied it.

We have a collection of K kinds of jobs contending for a resource of M units, and job type k needs b_k units. Jobs arrive Poisson, with load ρ_k. For the Walsh codes, the job type is the type of service; for the PA, the job type is the power demand class.

The stochastic knapsack calculation (Kaufman-Roberts recurrence) allows us to find the steady-state probability distribution of the number of resources in use, and the blocking probabilities per job type.[5], [6], [2]

Let $g(c)$ satisfy $g(c) = 0$ for $c < 0$ or $c > M$, and $g(0) = 1$, and let

$$g(c) = \frac{1}{c}\sum_k \rho_k b_k g(c - b_k) \tag{1}$$

for $c = 1 \ldots M$. Then the steady-state probability that c resource units are used is $\hat{g}(c) \equiv g(c)/G$, where $G \equiv \sum_c g(c)$. The blocking probability for a circuit service needing b_k units is then

$$R_k \equiv \sum_{c<b_k} \hat{g}(M-c),$$

or

$$R_k = 1 - G(M-b_k)/G(M). \tag{2}$$

where $G(c) \equiv \sum_{c' \le c} g(c')$. We will use the recurrence to compute the "passing probability"

$$P_k \equiv 1 - R_k = G(M-b_k)/G(M).$$

D. Assumptions for Integrated Analysis

So far, discussion has been about analyzing the PA blocking probability in isolation, and similarly the Walsh code blocking probability, in isolation. Moreover, discussion has not addressed packet data QoS analysis. This subsection discusses the approximations and assumptions we make to analyze the joint effect of the Walsh code and PA blocking.

Approximation 2.4: **Cascade model.** We will first compute the blocking per service due to Walsh code limitations, and then compute the blocking due to the PA of the resulting reduced load, on a service-by-service basis.

That is, high blocking of a given service by Walsh code limitations implies a reduced load when considering that service with respect to PA limitations. The reduction in PA load for that service is assumed to be uniform across different power demand classes for that service. (This is expressed symbolically as (3) below.)

This is an approximation, because even when the PA is highly loaded, and a service might be blocked as a result, we will still consider the service at full load for the Walsh code calculation.

The *reduced-load approximation* (also known as the *Erlang fixed-point approximation*) would avoid this approximation, but it generally requires a fixed point iteration that is much slower than our simple cascade. Although there is always a unique solution in the $K = 1$ case, [7], [8]; the general result is that it has at least one solution. [9] Another difficult is that our need for a smooth function with derivatives would probably require continuing the fixed point iteration to machine accuracy.

Assumption 2.5: **Equal priority circuit service.** All circuit services (including voice and data) have equal priority.

That is, we don't model a policy where circuit data services are thrown off in overload. This assumption can be avoided if there is no packet data as sketched out in Section IV.

Approximation 2.6: **Activity factors.** In addition to loads, we also have *activity factors* that specify another form of variation in the use of resources.

The activity factor models the use of the resource during a job, giving an indication of the *average* use of the resource. The role of activity factors is different for different services and for the two resources. While an inactive voice or circuit call uses less (or no) power, it does use its allocated Walsh codes, so the "activity factor" for circuit Walsh code usage is one, even if the general activity factor for PA usage is less than one.

E. Packet Data

Packet data services do not satisfy, even approximately, the assumptions of the stochastic knapsack calculation. From our assumptions, however, the results of the stochastic knapsack calculation for the circuit services can be used to more accurately predict the system performance for packet data services.

Assumption 2.7: **Packet doesn't affect circuit.** The circuit services affect the resources available to the packet services, but *not* the other way around.

Approximation 2.8: **Packet service quality function.** For given power available from the PA, all users of a packet data service will see the same performance, which is a function of the available power and of the average power needed per call by the users of that packet service.

From this assumption, we model the expected QoS for a packet-data service as follows: we use a quality function $P_p(X, Z)$, taking values in $0 \ldots 1$, where X is the expected number of packet users, and Z is the ratio of available units of the resource to the average need of that resource by a user. (So Z is the "number of channels".) The function $P_p()$ is analogous to P_k for circuit services: we want $P_p()$ as large as possible. The expected quality of service for packet-data services is then estimated as

$$\bar{P_p} \equiv \sum_c \hat{g}(c) P_p(\hat{\rho}_p, \hat{\rho}_p(M-c)/D_p),$$

where $\hat{\rho}_p$ is the expected number of packet users and D_p is the expected total demand for the resource by the packet user. If we define $\tilde{\rho}_k$ as the expected number of packet data users for the job type k, then

$$\hat{\rho}_p = \sum_k \tilde{\rho}_k = \sum_{i,j \notin C} \rho_{ij} \quad \text{and} \quad D_p \equiv \sum_k b_k \tilde{\rho}_k$$

So $D_p/\hat{\rho}_p$ is an estimate of the average resource need per packet-data user and the c resource units used by circuit services leave $M - c$ are available for packet data.

III. Performance Estimate

First we describe the loss model calculations, and then the overall performance estimate.

A. Per-service Performance

As described above, we will do a loss model calculation for the Walsh codes, and then use the resulting reduced load to do a loss model calculation for the PA, and use the results of those calculations to compute performance estimates for packet data services.

Each service j, such as voice, circuit data, packet data, etc., will be modeled as having offered load ρ_{ij} for power demand b_i^A and Walsh code usage b_j^W; that is, ρ_{ij} expected users using service j will need b_i^A bins of the PA and b_j^W Walsh codes.

The load ρ_j^W of circuit service j for the Walsh-code loss-model calculation is $\rho_j^W \equiv \sum_i \rho_{ij}$, and we can take ρ_j^W to be zero for $j \notin C$, where C is the set of indices of circuit services (that is, voice or circuit data). Having done the loss model calculation for the Walsh codes, we have Walsh code usage probabilities $\hat{g}^W(c)$ and passing probabilities P_j^W. As discussed above, we use the $\hat{g}^W(c)$ values to compute $\bar{P_p}^W$, the normalized relative throughput of packet data services due to Walsh code limitations. Here, for packet data service j, the load for service j (job type j) is $\tilde{\rho}_j^W = \sum_i \rho_{ij}$, and $\tilde{\rho}_j^W = 0$ if $j \in C$. This implies

$$\hat{\rho}_p^W = \sum_j \tilde{\rho}_j^W = \sum_{i,j \notin C} \rho_{ij}$$

and

$$D_p^W \equiv \sum_{j \notin C} b_j^W \tilde{\rho}_j = \sum_{i,j \notin C} b_j^W \rho_{ij}$$

For the PA loss-model calculation, the load values ρ_i^A are

$$\rho_i^A \equiv \sum_{j \in C} P_j^W \rho_{ij}. \tag{3}$$

Together with the bin requirements b_i^A, these yield PA bin usage probabilities $\hat{g}^A(c)$ and passing probabilities P_j^A. The usage probabilities are used to compute $\bar{P_p}^A$, the normalized relative throughput of packet data services due to PA limitations. Here packet data load for power demand class (PA job type) i is $\tilde{\rho}_i^A = \sum_{j \notin C} \rho_{ij}$.

Note that since our model of QoS for packet data is based on delay, there is no modeled reduction of PA demand by packet calls due to Walsh code limitations. We have

$$\hat{\rho}_p^A = \sum_i \tilde{\rho}_i^A = \sum_{i,j \notin C} \rho_{ij},$$

so indeed $\hat{\rho}_p^A = \hat{\rho}_p^W = \hat{\rho}_p$, and

$$D_p^A \equiv \sum_i b_i^A \tilde{\rho}_i = \sum_{i,j \notin C} b_i^A \rho_{ij}$$

B. Overall Performance

We can now join together the blocking probabilities and performance estimates to obtain an overall performance estimate.

We merge together the packet data estimates $\bar{P_p}^W$ and $\bar{P_p}^A$ using a "smooth min" function $\mathcal{M}(.,.)$ to obtain a packet performance estimate

$$\bar{P_p}^O \equiv \mathcal{M}(\bar{P_p}^W, \bar{P_p}^A).$$

If we used the usual min instead of the smooth min, the estimate would not be differentiable everywhere, and a derivative discontinuity would interfere with optimization.

Let $\mathcal{I}_j$ be a weighting factor indicating the "importance" of service j. We will combine the estimates together by weighting using $\mathcal{I}_j$ and using the appropriate loads. Let $\mathcal{L}_p \equiv \sum_{j \notin C} \sum_i \mathcal{I}_j \rho_{ij}$. Our measure of overall performance is $T/\mathcal{L}$, where

$$\begin{aligned} T &\equiv \sum_{j \notin C} \sum_i \mathcal{I}_j \rho_{ij} \bar{P_p}^O + \sum_{j \in C} \sum_i \mathcal{I}_j \rho_{ij} P_j^W P_i^A \\ &= \mathcal{L}_p \bar{P}_p + \sum_{j \in C} \sum_i \mathcal{I}_j \rho_{ij} P_j^W P_i^A \end{aligned}$$

and

$$\mathcal{L} \equiv \sum_j \sum_i \mathcal{I}_j \rho_{ij}.$$

IV. Two-Priority Systems

As given above as Assumption 2.5, we assume that all circuit services have the same priority. It is not unusual, however, to have circuit data services at lower priority than voice services, where the data services are thrown off in overload. If there is no packet data, we can model this by solving (1) twice: once without the circuit data services, and once for just those services. This gives a function $R_k^D(c)$ like (2), but for data services only. Then the service k blocking is a weighted average of $R_k^D(M-c)$ based on the voice service steady state probabilities $\hat{g}^V(c)$.

V. Experimental Studies

Since it is hard to compare our model directly with the real world, we built a Monte Carlo simulator that models call arrivals and departures, Walsh code allocation and deallocation, and blocking due to Walsh and power limitations. It does not operate in discrete time steps, but rather uses exponentially-distributed random variables to decide when the next event happens. Each simulation ran for at least 30,000 events with one half or one third of that reserved for "warm-up time" not used in gathering statistics. Furthermore, each statistic reported below is averaged over at least 10 such simulation runs.

A. Walsh Code Additivity

Running the Monte Carlo simulator with the available power M^A set very high allows us to compare Approximation 2.2 (Walsh code additivity) to the popular *crowded-first* allocation scheme [10], [11], also called *crowded-first-code* [12]. One would naturally expect the additivity assumption to be optimistic with respect to the overall blocking probability, because it amounts to assuming that there is never any *code blocking* (blocking calls due to otherwise sufficient free Walsh codes not forming a large enough contiguous block). However, Figure 1 shows that the assumption actually becomes *pessimistic* at high loads. Here code blocking affects more calls that demand many Walsh codes. For a lower overall blocking, it is good strategy to block such "large jobs" unnecessarily if this is likely to prevent many small jobs from being blocked.

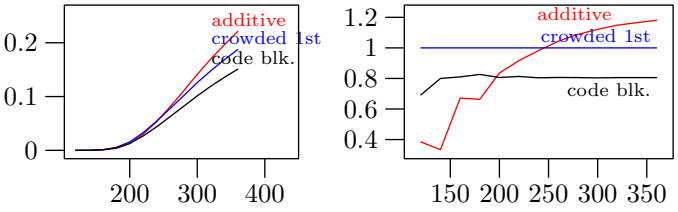

Fig. 1. (a) Blocking versus average Walsh code demand ℓ under the additivity assumption and crowded-first allocation, with the code blocking component graphed separately; (b) the ratio of each type of blocking to crowded-first blocking. Each call is equally likely to demand 1, 2, 4 or 8 Walsh codes.

When the distribution of Walsh code demands is skewed so there are fewer large jobs and more small jobs, the additivity assumption is more pessimistic at high loads. (Figure omitted for brevity.)

To get a better idea of what Walsh code demand qualifies a call as "large," note that each value for the total free space σ in the Walsh tree leads to a probability distribution for how often the maximum contiguous block of available Walsh codes is 1, 2, 4, 8, 16, etc. Figure 2 shows $\gamma := 2^{E[\lg \beta | \sigma]}$ as a function of σ, where β is the maximum contiguous available Walsh size. Here $E[\lg \beta \mid \sigma]$ is the conditional expectation of $\lg \beta$ given σ, for some specific ρ and b^W values, and $\lg n := \log_2 n$.

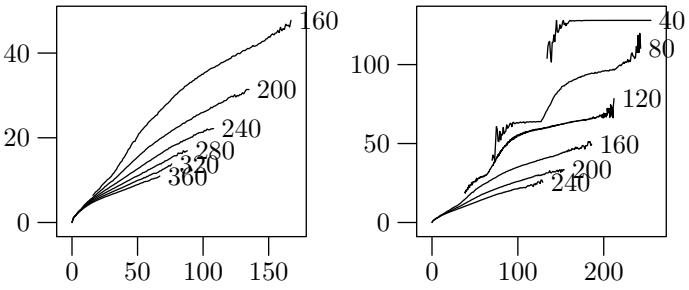

Fig. 2. Graphs of maximum Walsh codes per call γ versus total Walsh tree free space σ for $z = 4.61$ and various total Walsh loads ℓ. (a) shows only σ values with ≥ 1000 occurrences in the simulator runs; (b) uses a lower limit of 10 occurrences for $\ell = 40, 80$ and a limit of 100 for larger ℓ values.

Figure 2 suggests that γ is roughly linear in σ, but depends on the Walsh code demands of incoming calls, and is bounded by the largest power of 2 not exceeding σ. Thus an empirical estimate for γ can be of the form,

$$\gamma \approx \min(3 + f(z,\ell) \cdot \sigma,\ 2^{\lfloor \lg \sigma \rfloor}), \tag{4}$$

where $\ell := \sum_j \rho_j^W b_j^W$ and $z := \sqrt{\sum_j \rho_j^W (b_j^W)^2 / \sum_j \rho_j^W}$, the RMS mean of the Walsh demands b_j^W weighted by the associated traffic ρ_j^W. Since it can be useful to be able to compute estimated γ values, we shall use the rather arbitrary

empirical formula

$$f(z,\ell) = \frac{7.5 + 8.53z + 0.157z^2}{\ell + \max(0, 24z - 100)}. \quad (5)$$

It is not hard to base call blocking on (4) and (5) instead of just using the additivity assumption, and simulator runs showed that this significantly reduces the error relative to crowded first blocking (e.g., from 18% to 4.1% for $\ell = 360$ and equal Walsh demands). The crossover from optimism to pessimism tends to make these errors smaller at the 1% or 2% blocking rates typical of a loaded cell phone system. Note that it is impractical to seek further improvements by incorporating $f(z,\ell)$ into (1), because this requires a much more complicated notion of resource consumption.

B. The cascade model

One would expect the cascade model (Approximation 2.4) to perform well when Walsh limitations are much more important, and this is demonstrated by the scatter plots in Figure 3. These show the model's prediction for overall performance, versus corresponding simulator results. Here there were two services and two power demand classes with $M^A = 100$,

$$M^W = 64,\ \mathcal{I}_1, \mathcal{I}_2 = 1, 2,\ b_1^W, b_2^W = 16, 1,\ b_1^A, b_2^A = 1, 4,$$

and 81 different problem instances were obtained by trying all possible load matrices where

$$\rho_{1,1}, \rho_{2,1} \in \{2, 8, 32\}, \quad \rho_{1,2}, \rho_{2,2} \in \{1, 4, 16\}. \quad (6)$$

Since Figure 3b shows such a close match between the cascade model and the simulator results, almost all the disagreement in Figure 3a must be due to the Walsh additivity assumption.

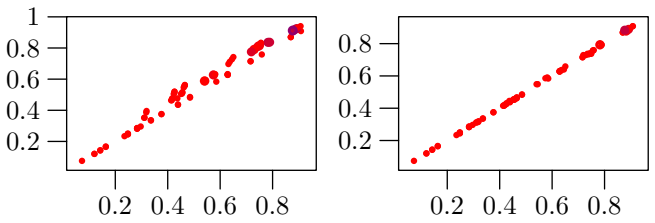

Fig. 3. (a) Overall performance predicted by the Monte Carlo simulator for various Walsh-limited scenarios versus the corresponding $T/\mathcal{L}$ from the cascade model; (b) the same except with the simulator using Walsh additivity assumption for admission control.

Now consider 81 power-limited scenarios where the 2 services and 2 power demand classes have $M^A = 100$,

$$M^W = 256,\ \mathcal{I}_1, \mathcal{I}_2 = 1, 2,\ b_1^W, b_2^W = 8, 1,\ b_1^A, b_2^A = 11, 4,$$

and (6) gives 81 sets of $\rho_{i,j}$ values. In this case the Walsh-additivity assumption does not matter and we get good agreement between simulator results and the cascade model as shown in Figure 4.

Tweaking the scenarios so that $M^A = 100$,

$$M^W = 64,\ cI_1, \mathcal{I}_2 = 1, 2,\ b_1^W, b_2^W = 8, 1,\ b_1^A, b_2^A = 8, 3,$$

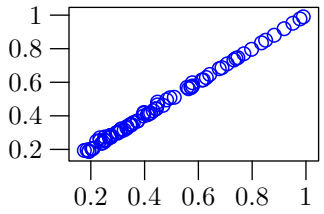

Fig. 4. Overall performance predicted by the Monte Carlo simulator for various power-limited scenarios versus the corresponding $T/\mathcal{L}$ from the cascade model.

while still using the 81 sets of $\rho_{i,j}$ values from (6) gives Figure 5. Many of these are scenarios where Walsh and power limitations both matter, yet there isn't a lot of scatter in the figure, and the small circles for scenarios where both limitations matter do not appear particularly problematical. In fact, comparing 5a to 5b shows more scatter due to the Walsh additivity assumption than due to the cascade model.

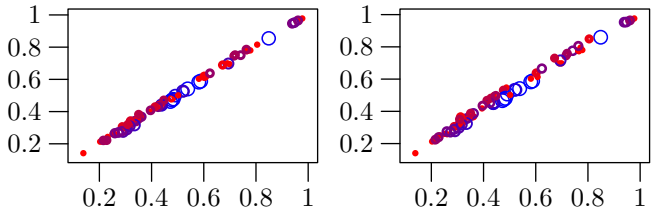

Fig. 5. (a) Overall performance predicted by the Monte Carlo simulator for various scenarios versus the corresponding $T/\mathcal{L}$ from the cascade model without the empirical correction; (b) the same except with the simulator using Walsh additivity assumption. Dots like those in Figure 3 denote Walsh-limited scenarios; circles like those in Figure 4 denote power-limited scenarios; and anything in-between denotes a scenario where both limitations matter.

REFERENCES

[1] F. P. Kelly, "Loss networks," *Ann. Appl. Prob.*, vol. 1, pp. 319–378, 1991.

[2] K. W. Ross, *Multiservice loss models for broadband telecommunication networks*. Springer, 1995.

[3] P. Whiting, personal communication.

[4] M. Jaber, S. A. Hussain, and A. Rouz, "Modified stochastic knapsack for UMTS capacity analysis," *FUJITSU Sci. Tech. J.*, vol. 38, no. 2, pp. 183–191, Dec. 2002.

[5] J. S. Kaufman, "Blocking in a shared resource environment," *IEEE Trans. Commun.*, vol. 29, pp. 1474–1481, Aug. 1981.

[6] J. W. Roberts, *A service system with heterogeneous user requirements*. North-Holland, 1981, pp. 423–431.

[7] W. Whitt, "Blocking when service is required from several facilities simultaneously," *AT&T Tech. J.*, vol. 64, no. 8, p. 18071857, 1985.

[8] F. P. Kelly, "Blocking probabilities in large circuit switched networks," *Adv. Appl. Probabil.*, vol. 18, pp. 473–505, 1986.

[9] S.-P. Chung and K. W. Ross, "Reduced load approximations for multirate loss networks," *IEEE Trans. Communications*, vol. 41, no. 8, pp. 1222–1231, Aug. 1993.

[10] C.-M. C. Y.-C. Tseng, "Code placement and replacement strategies for wideband cdma ovsf code tree management," in *Proc. of IEEE GLOBECOM*, vol. 1, 2001, pp. 562–566.

[11] A. N. Rouskas and D. N. Skoutas, "OVSF codes assignment and reassignment at the forward link of W-CDMA 3G systems," in *Proc. of IEEE PIMRC*, vol. 5, 2002, pp. 2404–2408.

[12] C.-M. Chao, Y.-C. Tseng, and L.-C. Wang, "Reducing internal and external fragmentations of OVSF codes in WCDMA systems with multiple codes," *Wireless Communications and Networking*, vol. 1, pp. 693–698, March 2003.