BRICKS: a Slotted CDMA Protocol for Efficient Integration of Voice and Data

Hongjun Zhang & Pawel Gburzynski
University of Alberta, Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8

Keywords: mobile networks, bandwidth allocation, quality of service

Abstract

We present a novel radio channel structure based on slotted CDMA technology intended for carrying traffic with diverse bandwidth/QoS requirements. The essence of our approach is a combination of flexible slotting with allocation of multiple codes to high-bandwidth mobiles. We demonstrate that the proposed scheme efficiently integrates multiple traffic classes into a unified CDMA system.

INTRODUCTION

To support services within the bit rate ranges offered by the third-generation mobile systems, the scarce resource of the radio channel must be shared in a highly flexible and efficient manner. Two types of solutions have been proposed for such systems, one class based on the variable spreading gain technology (VSG-CDMA) [2, 3], and the other on allocating multiple codes to a single session (MC-CDMA) [1, 4, 5].

In this paper, we introduce an MC-CDMA protocol, dubbed BRICKS. As two integral components of our solution, we discuss a quick code acquisition system and an air interface in which the uplink signaling channel is based on code-domain minislots. We also propose a method to explicitly exploit silent periods in voice activity. As indicated by our performance studies, BRICKS efficiently accommodates multiple traffic classes with different bandwidth requirements and QoS expectations.

SYSTEM PREREQUISITES

We consider a network with a single base station (BS) and a variable number of mobile stations (MS). Since the performance of a mobile network is limited by the capacity and flexibility of the link from MS to BS, only the structure of the *uplink* is discussed in detail. The protocol assumes the same prerequisites as WISPER [4], with a similar structure of the mobile transmitter and receiver. The carrier modulation is binary PSK.

All transmissions are carried out at the fixed *basic* rate R_b. A single mobile can transmit $1 \leq m \leq M$ packets simultaneously, using different spreading codes C_i, $(i = 1, \ldots, m)$ [1]. The transmission power P_t expedited by a mobile must increase along with the transmission rate m to provide the same signal-to-interference ratio (S/I) for each of the m parallel channels. A quick and reliable code acquisition scheme is essential for the correct operation of BRICKS. We propose a parallel acquisition system [6, 7] built around a matched filter and utilizing the maximum likelihood strategy. The acquisition procedure consists of two phases: searching and verification. The search block consists of N parallel matched filters [8].

Fast code acquisition at the base is more critical because of the compact organization of the uplink frame. To facilitate it, the mobile transmits a fixed length unmodulated PN sequence (the acquisition preamble). The uncertain region L for code acquisition at the base station is determined by the maximum distance D between the base and the mobile plus the uncertainty δ of the signal processing time: $L = ((2D/c) + \delta)/T_c$, where L is the duration of the uncertain region expressed in the number of PN chips, c is the speed of light, and T_c is the PN chip duration.

Assume that the code synchronization procedure starts at time zero. Using N parallel passive non-coherent PN matched filters (PN-MFs), the uncertain region L is divided into N subsequences, each of length $K = L/N$. Each PN-MF is loaded with one of the N subsequences. The number of search cells on each delay line in the matched filter is K/Δ with the delay of ΔT_c between successive taps.[1] In KT_c seconds, NK/Δ cells are searched, with each cell corresponding to one of the possible NK/Δ phases in the uncertain region.

[1] The standard recommended value of Δ is $1/2$.

The largest sample and the corresponding code phase from each of the N parallel PN-MFs are stored and compared. If the sample exceeds a threshold γ_1, a tentative *hit* is assumed, and the corresponding phase is used to initiate the correlator and start the verification process. The searching continues until a true *hit* is declared in the verification process, or the preamble runs out, whichever happens first. In the former case, code tracking is started; otherwise, the code synchronization is lost.

Every LT_c seconds, the N PN-MFs are reset with a new portion of the PN code shifted by LT_c seconds. If a certain specified number of all tests exceed the threshold γ_2, code acquisition is assumed and the code tracking system takes over the code synchronization. Otherwise, a false alarm is declared. If a new tentative hit was found already, say at time T_h, the verification process is immediately restarted with the phase shift of $t - T_h$, where t is the current time. Otherwise, the verification is aborted and postponed until the next hit. The verification is also aborted and reset if within the current search interval of KT_c seconds a new sample is found that is bigger than the one that triggered the previous hit. After that interval, however, the correlator is never reset unless it detects a false alarm.

The performance of a single matched filter is analyzed in [6]. With N parallel filters (matched to multiple segments of the code), the acquisition probability is much higher than the estimate in [6], and of order $1-(1-P)^N$, where P is the probability for a single filter. Additionally, as subsequent slots transmitted by the same mobile will tend to be located closely in time domain, the uncertainty interval for code acquisition can be centered around the exact phase found for the previous slot. Also, this interval can be set much tighter than the upper bound L, depending on how late the current slot follows the last slot for which the code was successfully acquired. This is because D will tend to change very little between consecutive acquisitions.

PROTOCOL DESCRIPTION

Every uplink frame is partitioned into a number of logical channels spanning two dimensions, i.e., time and code. The frame can be envisioned as consisting of multiple layers of slots resembling a brick wall, as in Figure 1. The amount of power assigned to a channel corresponds to the height of the corresponding brick.

Except for the *RA* slots, the height of the remaining slots may vary across the code dimension. This is different, e.g., from [4], where different slots occurring at the same time have identical properties. The *RA* slots play in our protocol a similar role to the contention slots in [4]. The remaining portion of the frame is built of four slot types. The granularity of bandwidth allocation is determined by the size of the standard (basic) slot, *TA*, whose duration is selected to accommodate exactly one active voice session. The second slot type, *TS*, is used to build signaling channels, needed by the mobiles admitted to the system to provide the base with feedback regarding their dynamic bandwidth requirements and received power. These slots are allocated from the beginning of the frame. They cannot appear too close to the end, because the base must be able to process the information contained in them before announcing the layout of the next frame. As they are shorter than the *TA* slots, the last signaling slot may be followed by an unusable gap.

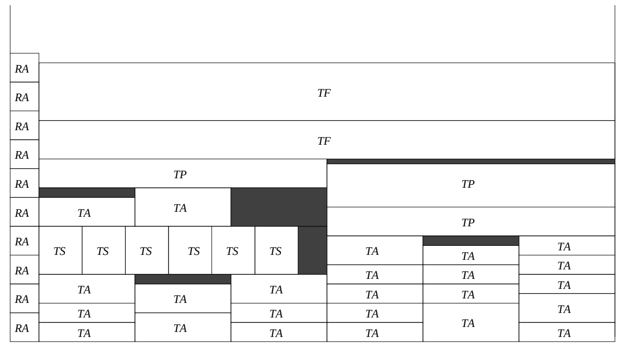

Figure 1: BRICKS frame structure

A slot of the third type spans the entire frame space and is intended for high-bandwidth sessions. We call it a *flat slot* (or a flat channel) and denote by *TF*. To increase the flexibility of flat allocation, we admit *partial flat slots*, denoted by *TP*, spanning the width occupied by several *TA* slots, but less than the whole frame. As the allocation of partial flat slots is considerably more complex than for the remaining slot types, it makes sense to impose restrictions on their size and/or position within the frame. In our virtual implementation of the protocol, we have assumed that a partial flat slot is half the size of the (full) flat slot, and that it must be aligned at a half-frame boundary.

When a mobile wants to initiate a session, it randomly selects one of the *RA* channels and sends a request to the base station using a modified ALOHA protocol. The access request packet transmitted in the *RA* channel consists of the following components: mobile ID, service type, resource requirements, delivery deadline, and transmitted power level. The exact specification of resource requirements depends on the service type and may include transmission rate, message length, or be empty. If the base station correctly receives the access request packet, it will assign to the mobile an uplink signaling channel. Depending on the available bandwidth, the mobile may be also immediately assigned a traffic channel.

As part of the control packet announcing the layout of the forthcoming frame, the base broadcasts to the mobiles the current *contention permission status*, indicating which traffic classes are allowed to compete for bandwidth. With this mechanism, the base is able to inhibit lower priority sessions when bandwidth becomes scarce. By thresholding the perceived noise level in the *RA* channels, the base may also selectively restrict contention to high priority classes if that level appears to be too high.

To provide a satisfactory reliability of transmission for a given service, the system must ensure that the bit error rate (BER) does not exceed the service-specific maximum [9, 11]. The BER specification can be mapped to the bit energy to noise spectral density ratio E_b/N_0 [10]. In contrast to [10], an allocated channel in BRICKS is always active; thus, we have the following admission constraint:

$$\sum_{k=1}^{K} \alpha_k < 1, \qquad \alpha_k = \frac{(E_b/N_0)_k}{W/R_b + (E_b/N_0)_k}$$

where K is the number of simultaneous code channels in the time slot, $(E_b/N_0)_k$ is the E_b/N_0 requirement for k-th code channel, W is the total spread bandwidth and R_b is the base rate. Consequently, the minimum power assignment is $P_i = \alpha_i(\eta + P)$, for $i = 1, \ldots, K$, where η is the background noise, and the minimum total received power is $P = \left(\sum_{k=1}^{K} \alpha_k \eta\right) \Big/ \left(1 - \sum_{k=1}^{K} \alpha_k\right)$.

While building the structure of the next uplink frame, the base keeps an allocation table indexed by time slots located at the time boundaries of *TA* channels—see Figure 1. Each entry in that table stores the height of the brick wall across the slot (i.e., the sum of all α_k falling in the slot) and the list of requests accommodated at that location. A new *TA* channel is allocated at the location with the smallest height, which results in the update of a single entry in the allocation table, while the addition of a new *TF* channel simply raises the height of the entire wall by the same amount.

Formally, the optimal assignment of channels to the frame is NP-hard. However the maximum error incurred by the simple greedy approach is bounded by $\max \alpha_k$, which seems quite acceptable. We postulate that high priority sessions be restricted to basic (*TA*) and full flat (*TF*) channels, whose allocation is straightforward. Following this "rigid" allocation phase, the leftover frame space can be partitioned among the more flexible lower priority sessions. This way, instead of trying to solve a computationally difficult task of partitioning the remaining chunks of bandwidth into a number of rigid chunks, we reverse the problem and allocate whatever chunks come out handy to flexible sessions.

The bandwidth scheduler operates in cycles. Every cycle starts with the reception of all *RA* slots through which new mobiles register their sessions with the base. These new requests are appended at the end of the respective queues. Then the status of the sessions in progress is updated based on the received contents of the signaling channels *TS*. Having received the last signaling slot, the base is ready to process the request queues, allocate bandwidth, and announce the layout of the next frame.

VIRTUAL IMPLEMENTATION

We assume that our cellular system offers four types of service listed in the decreasing order of priority: voice, video conferencing, file transfer, and SMS.

The Radio Channel

The basic rate R_b of our radio channel is 500 kbps and the transmission rate needed to sustain a voice session in its active phase is 22 kbps. The total length of a single uplink frame is $l_f = t_g + p + l_{ra} + N_{ta} \times (t_g + p + l_{ta})$, where t_g is the guard time (8 bits = 16 μs), p is the acquisition preamble length (32 bits or 64 μs), l_{ra} is the length of the contention slot *RA* (96 bits or 192 μs), l_{ta} is the length of the basic slot *TA* (384 bits or 768 μs–ATM payload size), and N_{ta} is the number of *TA* slots in a single layer of the frame (20). All these numbers add up to l_f = 8616 bits or 17.24 ms. The payload length of one flat slot *TF* is $l_{tf} = N_{ta} \times (t_g + p + l_{ta}) - t_g - p = 8440$ bits.

The total amount of bandwidth available within a frame is determined by the basic rate R_b, the total spread bandwidth of the channel W, and the E_b/N_0 requirements of the individual sessions. In our model, we assume $W = 20$ MHz as the target bandwidth, although we consider some other values for comparison.

Signaling Channels

The total length occupied by a single signaling slot *TS* is 72 bits, with the payload restricted to 32 bits. These bits are partitioned into a 6-bit bandwidth specification, 10 bits for received power indication, and 8 bits left for extensions. All signaling slots occupy up to two layers (codes) of the first half of the frame space. The number of signaling channels per one layer (code) is limited by 58, and the maximum total number of signaling channels is 116. The E_b/N_0 ratio for signaling channels is 6 dB.

Voice Traffic

Our variant of the "on-off" voice model is similar to the one considered in [12], with the following parameters: *source rate* = 22 kbps, *mean call duration* = 3 minutes, *mean talkspurt length* = 1 second, *mean silence length* = 1.35 seconds, *deadline* = 10 seconds, *threshold for disabling voice requests* = 10%, *QoS* (E_b/N_0) = 5 dB. The arrival process of new calls is Poisson.

When a voice session enters the silent state, its traffic channel is temporarily released, but the connection sustains itself through the signaling channel. When the session gets back to the talkspurt mode, its traffic channel is reassigned in the next frame.

The bandwidth temporarily released by a voice session that has entered the silent phase can be assigned to lower priority sessions, but it cannot be used to accommodate a new voice session. This way, the bandwidth scheduler makes sure that all admitted voice sessions can always be accommodated in their active states.

The access contention permission flag for voice traffic is cleared if the population of unserviced voice requests exceeds 10% of all requests in the voice queue. The amount of bandwidth assigned to an active voice session is always the same and equal to one basic slot *TA* at $E_b/N_0 = 5$ dB.

Video Traffic

This traffic is described by a DAR(1) (Discrete Autoregressive) model [13] with the following parameters: *mean source rate* = 128 kbps, *variance* = 5536, *correlation* = 0.98, *mean call duration* = 30 minutes, *call deadline* = 10 seconds, *threshold for disabling video requests* = 10%. $E_b/N_0 = 5$ dB.

Every admitted video session is guaranteed a certain minimum amount of bandwidth b_{min}. Any bandwidth requested in excess of the minimum is scheduled in a fair manner using the equal degradation approach, with the service grade $G = \min(1, B_a/B_r)$, where B_a is the total maximum bandwidth available for teleconferencing service, after the higher-priority voice sessions have been accounted for, and B_r is the total extra bandwidth requested by the admitted video sessions. The amount of extra bandwidth assigned to a video mobile is equal to $\min(G, b_r)$, where b_r is the extra bandwidth requested by the station.

While admitting a new video session, the bandwidth scheduler assumes that the session can start with the minimum bandwidth b_{min}, and admits the session if that much bandwidth is available. For this calculation, all active video sessions are counted with their minimum bandwidth b_{min} and all voice sessions are assumed to be active. The assignment part of bandwidth scheduling for video sessions is trivial: it consists in adding one or more *TF* slots to the brick wall, appropriately updating its height everywhere.

File Transfers and SMS

File transfers have no inherent delay requirements, and they can use any rate physically available to the mobile. Our primary objective in handling file transfers is to maintain a reasonable degree of fairness while avoiding unnecessary fragmentation.

File transfers occur in bursts, with burst duration and inter-burst periods being exponentially distributed. During a burst, a new file for transmission is generated at exponentially distributed intervals. The length of every file is exponentially distributed as well. The numerical parameters assumed in our model are as follows: *mean burst duration* = 30 minutes, *mean inter-burst interval* = 30 minutes, *mean interval between files in a burst* = 36 seconds, *mean file length* = 104 KB, *threshold for disabling file transfer requests* = 10%. $E_b/N_0 = 6$ dB.

File transfer session receive bandwidth with the *preferred granularity* of one *TF* slot, in a Round-Robin fashion. If a mobile needs less than a full *TF* slot to complete its request, it is assigned a partial flat slot *TP* or a basic slots *TA*. The procedure continues until all bandwidth has been assigned or all mobiles have been satisfied. If no more *TF* channels can be allocated, the preferred granularity is downgraded to *TP* and then to *TA*.

An SMS message is treated exactly as a file to transfer, except that the transfer is scheduled with the lowest possible priority. The numerical parameters are as follows: *mean message length* = 6250 bytes, *mean interval time* = 11 seconds, *deadline* = 2 hours, *threshold for disabling SMS requests* = 20%, $E_b/N_0 = 6$ dB. Both the interarrival time and message length are exponentially distributed.

SELECTED SIMULATION RESULTS

The performance of BRICKS has been investigated by simulation and compared to the performance of WISPER [4] and VSG-CDMA [2]. All protocols were implemented in the same virtual radio environment in which the sole criterion of a successful reception was the bit energy to noise density ratio E_b/N_0.

Figure 2 illustrates how the link bandwidth in BRICKS is shared among the four traffic classes. This graph has been obtained for the total spread bandwidth

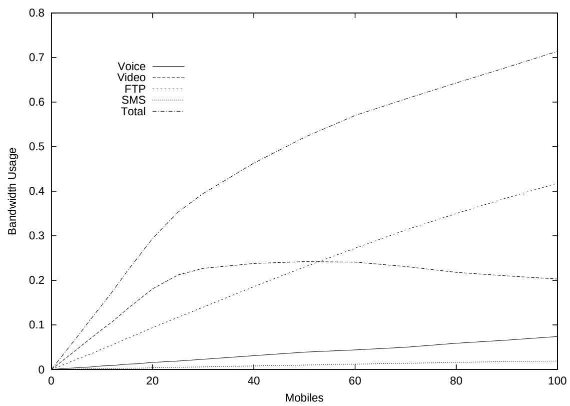

Figure 2: Bandwidth utilization in BRICKS ($W = 20\,\mathrm{MHz}$)

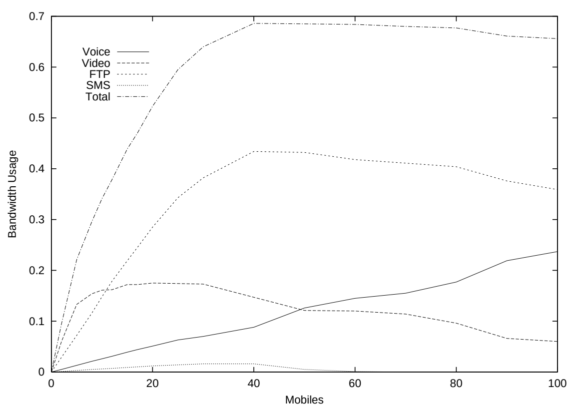

Figure 3: Bandwidth utilization in BRICKS ($W = 5\,\mathrm{MHz}$)

$W = 20\,\mathrm{MHz}$. A single point on the X-axis indicates the number of mobiles engaged in sessions of a given type (the same for all four traffic classes).

Although file transfers and SMS sessions have lower priorities than video sessions, their assigned bandwidth tends to increase until the very end of the investigated range of traffic conditions. There are two reasons for this behavior. First, there exist silent periods in voice sessions that cannot be reused to set up new voice or video connections, but are available to UBR/ABR traffic classes. Moreover, because of the inherent variability in video load, an equivalent of silent periods also occurs in video sessions. Second, the bandwidth requirements of a video session are stringent and must be granted in multiples of TF channels. Consequently, there exist chunks of bandwidth unusable by video sessions, but available to the much less picky file and SMS transfers.

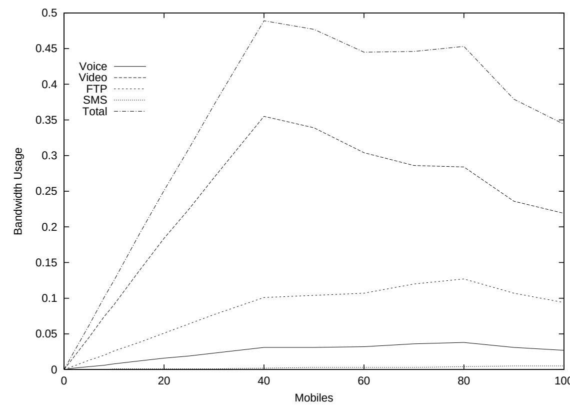

Figure 4: Bandwidth utilization in WISPER ($W = 20\,\mathrm{MHz}$)

The QoS tradeoffs in BRICKS are somewhat better visible in a network with smaller total spread bandwidth W, which leaves less room for the low priority traffic to sneak in. Figure 3 has been obtained for a network with $W = 5\,\mathrm{MHz}$. One can clearly see in it how all three lower priority classes yield to voice.

The performance of WISPER under identical offered load is shown in Figure 4. Notably, the protocol achieves lower maximum channel utilization than BRICKS, and the total bandwidth utilization tends to drop when the system becomes saturated. This trend is followed by all traffic classes (except SMS, whose bandwidth is too small to be significant), which means that the prioritization of the four traffic types does not fulfill its purpose very well. For example, an increase in the number of admitted video sessions results in a drop in the portion of bandwidth effectively available to voice sessions. This is caused by two reasons: the rigidity of bandwidth allocation, which requires that a given time slot be filled with traffic of the same class, and the relatively poor performance of the contention resolution part of the protocol (whose impact was neglected in the original analysis of WISPER presented in [4]).

A similar and more pronounced trend can be observed in VSG-CDMA whose performance is shown in Figure 5. As it turns out, a realistic implementation of this protocol exhibits instabilities. Under light load, the network operates without transmission control, and all ready stations transmit with probability 1. Having detected a congestion in one frame, the base turns on the controlled mode. Consequently, in the next frame all active users transmit with some probability p, which solely depends on the number of active

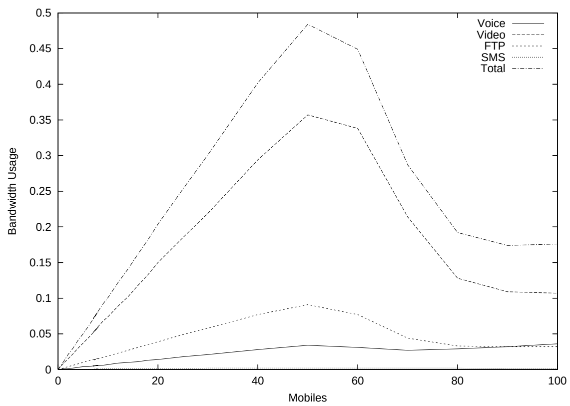

Figure 5: Bandwidth utilization in VSG-CDMA ($W = 20\,\mathrm{MHz}$)

mobiles. That frame tends to be underloaded, which forces the network back to the uncontrolled mode, and so on. These oscillations do not necessarily occur on a frame-by-frame basis, but their impact on the overall performance of the scheme is noticeable.

CONCLUSIONS

We have presented a CDMA protocol aimed at accommodating traffic classes with different QoS requirements. The protocol is flexible with bandwidth allocation, yet the complexity of its bandwidth scheduler seems to be reasonable. We have demonstrated that our protocol well caters to traffic classes with diverse QoS requirements and efficiently accommodates data traffic without compromising the quality of service for voice and video. This property makes it a good candidate for future mobile networks, in which non-voice traffic will constitute a considerably more significant component than it does today.

Our approach of admitting only as many voice sessions as can be sustained simultaneously runs against the commonly accepted policy that relies on statistical multiplexing to offer more voice bandwidth to the users. Although one can only guess about the load patterns of future PCS networks, it is rather obvious that the contribution of voice sessions to those patterns will tend to decrease. Consequently, it will be pointless to try to accommodate as many voice sessions as physically possible, and the focus will shift toward efficient coexistence of voice with other session types.

REFERENCES

[1] Chih-Lin I and R.D Gitlin, "Multi-code CDMA Wireless Personal Communication Networks," in Proceedings of the IEEE International Conference on Communications, Seattle, WA, vol. 2, pp. 1060–1064, June 1995.

[2] Chih-Lin I, K.K. Sabnani, "Variable spreading gain CDMA with adaptive control for true packet switching wireless network," in proceeding of ICC '95 Seattle, Vol.2, pp.725 -730, June 1995.

[3] R. Fantacci, S. Nannicini, "Multiple access protocol for integration of variable bit rate multimedia traffic in UMTS/IMT-2000 based on wideband CDMA," IEEE Journal on Selected Areas in Communications, Vol.18, No.8, pp.1441 -1454, Aug. 2000.

[4] Ian F.Akyildiz, David A.Levine, and Inwhee Joe, "A slotted CDMA protocol with BER scheduling for wireless multimedia networks", IEEE/ACM Transactions on Networking, Vol.7, pp.146 - 158, Apr. 1999.

[5] Sunghyun Choi and Kang G. Shin, "An uplink CDMA system architecture with diverse QoS guarantees for heterogeneous traffic", IEEE/ACM Transactions on Networking, Vol.7, pp. 616 - 628, Oct. 1999.

[6] U. Madhow, M.B. Pursley, "Mathematical modeling and performance analysis for a two-stage acquisition scheme for direct-sequence spread-spectrum CDMA," IEEE Transactions on Communications, Vol.43 No.9, pp.2511 -2520, Sept. 1995.

[7] Hyung-Rae Park, Bub-Joo Kang, "On the performance of a maximum-likelihood code-acquisition technique for preamble search in a CDMA reverse link," IEEE Transactions on Vehicular Technology, Vol.47, No.1, pp.65 -74, Feb. 1998.

[8] A.Polydoros and C.L. Weber, "A unified approach to serial search spread-spectrum code acquisition - Parts II: A Matched-Filter Receiver," IEEE Transactions on Communications, Vol.32, No.5, pp.550-560, 1984.

[9] A.M. Viterbi, and A.J. Viterbi, "Erlang capacity of a power controlled cellular CDMA system," IEEE Journal on Selected Areas in Communications, Vol.11, No 6, pp.892-900, August, 1993.

[10] L. C. Yun and D. G. Messerschmitt, "Power control for variable qos on a cdma channel," In Proceedings of IEEE MILCOM, Vol.1, pp.178-182, 1994.

[11] A. Sampath, P. S. Kumar and J. M. Holtzman. "Power Control and Resource Management for a Multimedia Wireless CDMA System". In PIMRC'95, September 1995.

[12] Xiaoxin Qiu, Victor O. K.Li, and Ji-Her Ju, "A multiple access scheme for multimedia traffic in wireless ATM," Mobile Networks and Applications, 1(3):259 - 272, Dec. 1996.

[13] Daniel P. Heyman and T. V. Lakshman, "Source models for VBR broadcast-video traffic," IEEE/ACM Transactions on Networking, 4(1):40 - 48, Feb. 1996.