

Throughput Competitiveness of WCDMA Channel Assignment

Jiun-Horng Cherng, Fen-Fen Shueh, and Wen-Shyen Eric Chen

Institute of Computer Science, National Chung-Hsing University
250, Kuo-Kuang Road, Taichung, 402, Taiwan, Republic of China
Phone: +886-4-2284-0497 ext 917, Fax: +886-4-2359-7915
{cherng, fenfen, echen}@cs.nchu.edu.tw

Abstract. Efficient and effective resource utilization is important in the bandwidth allocation of the communication network, especially when the resources are scarce such as the radio channels in the wireless links. Channel assignment in Wideband CDMA transmission schemes, i.e., the bandwidth allocation in the 3G wireless links, can be transformed into the Orthogonal Variable Spreading Factor code assignment to the connection requests. The code assignment problem presents a typical on-line feature that the decision to accept/reject the request has to be made at the arrival time of the request without the knowledge of future requests. An algorithm for an on-line problem is considered *competitive* if its performance is within some constant fraction of the performance of any other algorithm for the same input sequences. The study in deriving a competitive algorithm is valuable as it guarantees a lower bound of the system performance in all circumstances. This paper presents an on-line competitive algorithm called Left_Right_Aggressive (LRA) for the OVSF code assignment problem in the wireless links regarding system throughput as the performance indicator. With preemption allowed, the proposed algorithm achieves the constant competitive ratio of $2/(1-2\beta)$, where β is the fraction of the maximum requested bandwidth to the system capacity. The simulation results demonstrate that it is promising to achieve better throughput performance and lower preemption ratio in request patterns of Poisson distribution.

1 Introduction

High-rate multimedia data transmissions and multiple services to users are significant characteristics in the third-generation (3G) mobile communication systems. International Mobile Telecommunications-2000 (IMT-2000) sets up the air interface objectives [1] to support wideband data services as high as 2Mb/s for limited, and 144Kb/s to 384Kb/s for full coverage and mobility, while keeping the same transmission quality as in fixed telecommunication network [2]. Orthogonal Variable Spreading Factor (OVSF) [3] codes were proposed and adopted by IMT-2000 as the channelization codes on the forward radio link for the Wideband CDMA [4,5] radio transmission scheme.

The efficient and effective utilization of the scarce radio resources is the main theme for data transmissions over the air, and system throughput, which is the aggregated occupied bandwidth over time, is usually regarded as the performance indicator. Since the connection requests arrive one by one over time, it manifests the typical on-line feature [6,7], i.e., the system has to make the acceptance/rejection decision at the arrival time of the request without the advanced knowledge of the future requests. To achieve higher bandwidth utilization and/or lower blocking rate of the system, the preemptive feature [8,9,10] is deployed as to provide more “valuable” connections at the expense of less profitable ones already in services, when the total bandwidth requested exceeds the system capacity so that not all requests can be served at the same time. When and how to reject/preempt connections, and simultaneously guarantee the throughput performance, are more challenging in the on-line problem domain as there is no contribution to the throughput performance if the request is rejected/preempted. Authors in [11] demonstrated that an average 20% gain of throughput efficiency can be achieved if reassignment scheme is deployed.

We present an on-line preemptive algorithm *Left_Right_Aggressive* (LRA) for OVFSF code assignment in this paper that can achieve the constant competitiveness in the system throughput aspect. Algorithm LRA is an improvement of algorithm LR [10] in reducing the unnecessary preemption and increasing the system throughput. For any $\beta < 1/2$ where β is the ratio of the maximum bandwidth request to the system capacity, the algorithm is $2/(1 - 2\beta)$ -competitive. Our simulation demonstrates that the LRA algorithm is capable of achieving both higher throughput and at the same time keeping the low preemption ratio.

The remainder of the paper is organized as follows. Section 2 delineates the background and the related work of OVFSF code assignment, on-line problem and competitiveness, and preemption. In Section 3, the proposed algorithm LRA, with both preemptive and code reassigning features, is presented and proved to achieve the constant competitive ratio. The comparison of the throughput among LRA algorithm and others are demonstrated in Section 4. We present the discussion and the future work in Section 5.

2 Preliminaries

We describe the background and the related work of OVFSF code assignment in the first subsection, followed by the introduction and definition of on-line problem and competitiveness. Finally, the preemptive operation and related work is delineated.

2.1 OVFSF Code Assignment

Three Direct Sequence Code Division Multiple Access (DS-CDMA) transmission schemes [12,13,14], namely multi-code (MC-CDMA) method, OVFSF code method, and hybrid method, had been proposed to achieve variable and high

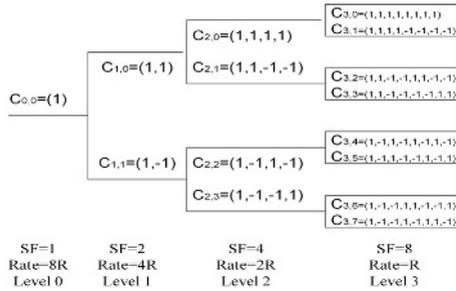


Fig. 1. An OVSF code tree

data-rates as required by 3G air interface objectives. The OVSF codes, a modified Hadamard transformation, were adopted by IMT-2000 as the spreading code generation scheme on the forward radio link as the channelization codes.

Through the recursive method in orthogonal code tree generation [13], the OVSF code tree with 4 levels (level 0 ~ level 3) is shown in Figure 1. $C_{L,S}$ uniquely identifies an OVSF code in the code tree, where L indicates the level of the code and S is the sequence number in that level. For a $(B + 1)$ -level OVSF code tree, the spreading factor (SF) of codes in level L is 2^L . Let the codes in the last level (level B), be capable of carrying R data-rate information, the codes in level L can carry data with rate $2^{B-L}R$.

The spreading codes are generated recursively as follows. Let $C_L = \{C_{L,S} | S = 0 \text{ to } 2^L - 1\}$ denotes a matrix representation of the set of 2^L binary spreading codes with spreading factor 2^L . The code length of the L^{th} level is 2^L chips and the code is used for transmitting symbols at the rate of 2^L times lower than the chip rate. Assuming that the \bar{C} is the binary complement of C , the C_L codes can be easily constructed from C_{L-1} as $C_L = \begin{bmatrix} C_{L-1} & C_{L-1} \\ C_{L-1} & \bar{C}_{L-1} \end{bmatrix}$.

The OVSF code tree is basically a full binary tree. In the OVSF code tree, all codes are orthogonal to each other, except the pairs that one code is the ascendant (descendant) of the other. It is thus evident that one code in level L can be assigned if and only if no code on the path from root to this code or any code in the subtree of this specific code is assigned. The restriction is called the *orthogonality constraint*. The request is blocked and rejected even when the system has the adequate bandwidth capacity, and the scenario is called *code blocking* in differentiating from the *system blocking* which results from the inadequacy of system capacity to serve the request.

A simple example in Figure 2 illustrates the impact of code blocking. In Figure 2(a), a request with data rate $2R$ cannot be accepted because any code in level 2 is orthogonal to the existing codes of $C_{1,0}$, $C_{3,4}$, and $C_{3,6}$. It is accepted only after the reassignment, say code $C_{3,6}$ is reassigned to $C_{3,5}$, and Code $C_{2,3}$ is available to be used as shown in Figure 2(b).

The authors in [11] devised a dynamic code assignment method on OVSF codes. Under the code constraint property, they proposed a code assignment

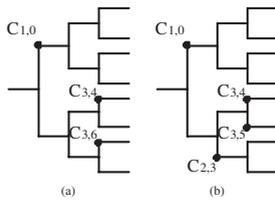


Fig. 2. OVSF code assignment (a) code blocking, (b) code reassignment

scheme intending to minimize the number of necessary reassignments of the occupied codes. The detailed description of the algorithm was not given, especially in the recursive searching steps. The comparison between more heuristic garbage collection methods with the recursive one was studied in [15], and the result implied that garbage collection is well suited for code reassignment when OVSF code blocking occurs.

Some research efforts focused on code assignments and reassignments for QoS purpose or by MC-CDMA and hybrid schemes [16,17,18,19]. Another approach [20] was the protocol design in supporting bursty data traffic by statistical multiplexing in the forward radio link of W-CDMA instead of assigning a fixed-rate code to a user.

2.2 Online Problem and Competitiveness

The main characteristic of the on-line problem is to make the acceptance/rejection decision at the time of the request without the knowledge of the future requests. A competitive algorithm for an on-line problem was discussed in [7]. It states that an algorithm for an on-line problem is called *competitive* if its performance is within some constant fraction of the performance of any other algorithm on the same input sequence, even an off-line algorithm. Let $P_A(S)$ and $P_B(S)$ represent the performance function of algorithms A and B respectively on a input sequence of requests S . For $c \geq 1$, algorithm A is called *c-competitive* if the following conditions are satisfied: (a) A is an on-line algorithm; (b) B is any other algorithm, even the off-line algorithm; and (c) A constant d exists for any sequence S such that $P_A(S) \geq 1/c * P_B(S) + d$.

Authors in [8] also demonstrated that, for the connection admission problems, if the holding time of the request is unknown when the request arrives, it is impossible to find a c -competitive algorithm if throughput is considered as the performance function. In this paper, we assume that the holding time of the request (the duration of each request) becomes known when the request arrives at the system.

2.3 Preemption

For real-time tasks, preemption techniques are often deployed. Some “weighting functions” or “values” are set for comparison in the preempting processes. The bandwidth request preemption in [8,9] considered several forms of values

such as the uniform profit for all requests, routing path counts and duration of the request. The values are gained only when the requests are completed, and no value counts if the request is rejected or preempted as the penalty of rejection/preemption. The study in [8] also indicates that if no penalty is paid for preemption, it is trivial to achieve the 1-competitive algorithm.

When throughput is considered as the performance function, the “Left_Right” (LR) strategy [10] which deploys preempting operations in the bandwidth allocation problems achieves the constant competitive ratio of $2/(1-2\beta)$, where β is the fraction of the maximum requested bandwidth to the bandwidth of the channel, i.e., system capacity. This is in contrast to the poor logarithmic competitive ratio that the non-preemptive algorithm [21] can obtain. The authors in [10] showed that the traditional throughput (bandwidth times the duration of the request) considerations as the parameter of preemption was not desirable, no matter in “greedy” or “double the gain” situations.

An alternative method considering both the periods of time of the requests that have been served and will be served by the system was suggested, yet this strategy is too conservative in that it may unnecessarily preempt connection requests. Further discussion and improvement will be entailed in the next section.

3 LRA Scheme for OVSF Code Assignment

3.1 LR Strategy

For an on-line communication system, there are times that the proper preemption of requests is allowed for obtaining more valuable “profit”. Let the profit be the system throughput, and let the request k is of 3-tuple form (a_k, d_k, r_k) , where a_k and d_k specifies the arrival time and duration of request k , and r_k is the data-rate requirement. The goal is to derive an algorithm that, with preemption allowed, is capable of achieving the guaranteed system throughput. The authors in [10] suggested that, with system capacity W and $\beta = \max \{r_k/W\}$ for all k , the procedure LR as depicted in Figure 3, can achieve $2/(1-2\beta)$ -competitive. That is, its throughput performance is at least $(1-2\beta)/2$ times of the best algorithm that may not exist.

The procedure starts, upon arriving of request k , by sorting all requests currently in the system in the ascending order of the starting times, and then selecting the top ones of requests into a set L such that the total bandwidth of those requests do not exceed half of system capacity. The same is done by sorting the ending times of the requests in descending order and selecting the top ones into a set R with the same $W/2$ -bandwidth constraint. Those requests that are neither in L nor R can be preempted or rejected.

The process is demonstrated using an example in Figure 4(a). Set system bandwidth $W=10$, and $\beta = 0.4$. Let requests 3, 4, 7 be in services while request 8 arrives at time $t=13$. Each request is with the input parameters $(8,12,4)$, $(9, 9, 2)$, $(12, 5, 2)$, and $(13, 6, 4)$, respectively, with total occupied bandwidth 8 at $t=13$. According to the procedure LR, $L=\{3\}$, $R=\{3\}$, and $P = \{4,7,8\}$,

```

Procedure LR
Let  $Z'$  be the set of requests currently in service;
Let  $P$  be the set of requests that can be preempted or rejected.
 $W$  is the system bandwidth;
Upon the request  $k$  with bandwidth request  $r_k$ , DO:
BEGIN
 $Z = Z' \cup \{k\}$ ;
Sort  $Z$  by increasing order of starting time;
 $L =$  top requests of  $Z$  such that the total bandwidth is at most  $W/2$ ;
Sort  $Z$  by decreasing order of ending time;
 $R =$  top requests of  $Z$  such that the total bandwidth is at most  $W/2$ ;
 $P = Z - (L \cup R)$ ;
Preempt/reject the requests in  $P$ ;
END
    
```

Fig. 3. Procedure LR for determining the requests to be rejected and/or preempted

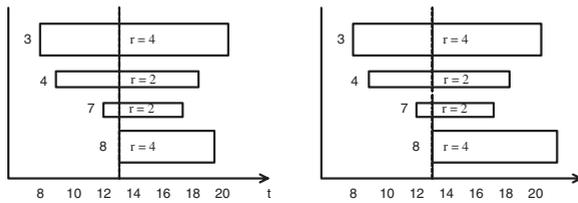


Fig. 4. Request 8 arrives at time $t=13$ while requests 3, 4, and 7 are in service

the request 4, 7 are preempted and request 8 is rejected. The procedure LR guarantees that for all the time periods of request 4, 7, and 8, the occupied bandwidth ratio is always greater than $(1/2 - \beta)$. The requests that are rejected and/or preempted have no contribution to the throughput.

Figure 4(b) illustrates another conservative behavior of procedure LR. Let the input sequence be the same as in Figure 4(a) with the exception that the input parameter of request 8 is $(13, 8, 4)$. In this example, $L=\{3\}$, $R=\{8\}$, and requests 4 and 7 are preempted. As can be easily verified that request 4 is preempted unnecessarily since input sequence $\{3, 4, 8\}$ can coexist in the system for the time period $t \in [8, 21]$ without exceeding the system capacity.

The LR technique, trying to keep the requests that have been served and will be serviced for longer period, is similar to the concern of the requests' holding times. If the duration of the request is large enough, it will certainly be kept in either L or R, even though there is no guaranteed competitive ratio. In Section 4, the simulation is also conducted for the DUR algorithm which, upon request k 's arrival, sorts the duration of the requests in descending order and selects those long-duration requests with total bandwidth less than or equal to W , and preempts/rejects those unselected.

```

Procedure LRA for OVSF code assignment
Let  $Z'$  be the set of requests currently in service;
Let  $P$  be the set of requests that can be preempted or rejected.
 $W$  is the system bandwidth;
 $U$  is the system bandwidth currently occupied.
Upon the request  $k$  with bandwidth request  $r_k$ , DO:
BEGIN
  IF (  $r_k > W - U$  ) THEN           //insufficient system resource (i)
     $Z = Z' \cup \{k\}$ ;
    Sort  $Z$  by decreasing order of ending time;
     $R =$  top requests of  $Z$  such that the total bandwidth is at most  $W/2$ ;
    IF request  $k$  is not included in  $R$  THEN goto end;           // (ii)
     $V =$  total bandwidth aggregated in  $R$ ;
    Sort  $Z$  by increasing order of starting time;
     $L =$  top requests of  $Z$  such that the total bandwidth is at most  $W-V$ ; // (iii)
     $P = Z - (L \cup R)$ ;
    Preempt/reject the requests in  $P$ ;
    Release the OVSF codes previously assigned to requests in  $P$ ;
  ENDIF
  Assign request  $k$  by OVSF code assignment procedure;
END

```

Fig. 5. Procedure LRA for OVSF code assignment

3.2 Algorithm LRA

To improve the throughput performance of LR algorithm, the more aggressive one called LRA (Left_Right_Aggressive) is designed and depicted in Figure 5. The bandwidth utilization is considered only in two conditions: when request k is rejected, or when request k is accepted while others are preempted.

The procedure LRA, upon the arrival of request k with bandwidth request r_k , checks the occupied bandwidth U and decides if the system has enough capacity. If the resource is available, the request is assigned an OVSF code by the OVSF code assignment procedure that will be presented later. If there is insufficient resource, the process begins sorting the ending times of all the requests in service. If request k is not included in set R , as shown in the example of Figure 4(a), the request k alone is rejected without affecting other requests in service. Since request k has the earliest ending time compared to all other requests currently in the system and definitely the latest starting time among them, the total bandwidth ratio of all others requests in service exceeds $(1 - \beta)$ for the period $t \in [a_k, a_k + d_k]$ for request k , the bandwidth utilization is thus guaranteed to be greater than $(1 - \beta)$ in this period.

If request k is included in R , as example shown in Figure 4(b), and let the total bandwidth in R be V , the set L could accommodate the aggregated bandwidth up to $(W - V)$ instead of $(1/2 - \beta)W$.

As emphasized in Section 2.1, the OVSF code assignment of the $(B + 1)$ -level OVSF code tree may encounter the code blocking that prohibits the allocation of bandwidth to the request. In this paper, the First-Fit algorithm is carried out to find an available code $C_{L,S}$ from $S = 0$ in level L when request bandwidth $r = 2^{B-L}R$, where R is the basic data rate. If no available code found, the garbage collection operation is then executed from the bottom level B , up to

level $(L - 1)$ so that it is guaranteed to get an available code in level L for request k . These operations result in code reassignment in compensation for the gain in throughput performance.

The LRA algorithm thus guarantees to achieve the competitive ratio of $1/(1/2 - \beta) = 2/(1 - 2\beta)$, yet this is only the worst-case scenario, the real implementation achieves much higher value as demonstrated in the next section.

4 Simulation and Results

Throughput performance, as well as the preemption ratio, is compared among the three algorithms: LR algorithm, LRA algorithm, and DUR (preemption algorithm based on the duration of the requests.) Event driven simulation is performed. Each run produces 30,000 requests with different patterns of data rate requirement. System bandwidth W is assumed $256R$ for an OVSF code tree with $B=8$, which complies with the IMT-2000 specification. The data rate $r \in [R, 2R, 4R, 8R, 16R, 32R, 64R]$. Five data patterns are generated: UR (Uniform Rate) with equal probability in each data rate, and the other four rate patterns P_8, P_16, P_32, and P_64, with Poisson distribution of rate request with mean rate $8R, 16R, 32R$ and $64R$ respectively.

4.1 Parameters

Input parameters are λ , μ , and r , and important output values are throughput, the preemption ratio, and the reassignment ratio. They are defined as follows:

λ : new request rate. The inter-arrival time of each request is assumed exponentially distributed with mean value of $1/\lambda$.

μ : the mean service rate of each connection. $1/\mu$ is the mean service time (duration) of each connection, and is assumed of exponential distribution.

λ/μ : the ratio of request rate to the service rate, and is called the *traffic load* of the system.

r : bandwidth request. r is assumed $R, 2R, 4R, 8R, 16R, 32R$, and $64R$. The frequency on different rate depends on the rate patterns.

Throughput: the aggregated product of requested bandwidth and the duration of the requests that are completed without being rejected/preempted.

Preemption ratio: the fraction of preempted requests to the accepted requests. It indicates the ratio of requests that were previous accepted to the system but later preempted with no contribution to the system throughput.

Reassignment ratio: the number of reassignment per accepted request. The code reassignment is executed when the code blocking occurs during the OVSF code assignment procedure. This ratio indicates how often one request will be reassigned a new OVSF code.

4.2 Results

The system throughput is the major goal of the preemption algorithms. The LR and LRA algorithms are both competitive while DUR algorithm is not. The

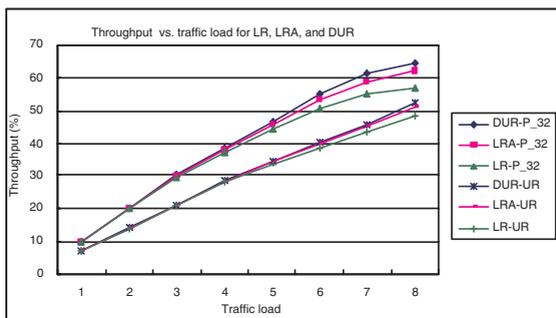


Fig. 6. Throughput comparison vs. traffic load for LR, LRA and DUR algorithms

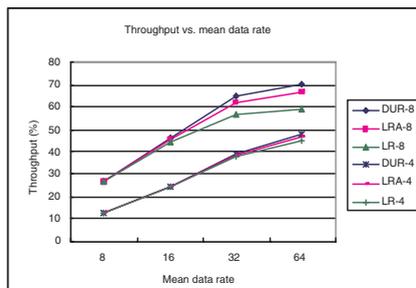


Fig. 7. Throughput comparison vs. mean data rate for LR, LRA and DUR algorithms.

throughput comparison of the three algorithms using the uniform data pattern (UR) and Poisson data rate with mean 32 (P_32) is illustrated in Figure 6. The throughput performance of LRA is better than LR, about 5% throughput improvement when traffic load approaches 8 in P_32 data pattern is gained, and 2% throughput improvement gained for uniform rate. The improvement increases as traffic load and mean data rate increase. The DUR algorithm performs quite well under the Poisson traffic, even though there is no guarantee for other traffic patterns. The throughput of DUR algorithm is higher than LRA for about 1~2% as shown in Figure 6.

Figure 7 depicts the throughputs of the three algorithms when traffic loads (λ/μ) are 4 and 8 respectively, as the mean data rates grow from $8R$ to $64R$ (P_8, P_16, P_32, and P_64). The LRA algorithm performs better than LR, and the difference grows as the mean data rate increases. Again the DUR algorithm performs slight better than the LRA algorithm.

High preemption ratio indicates the inefficiency of the algorithm since it accepts the requests but easily abandons them later. The preemption ratios of the three algorithms under different traffic load are demonstrated in Figure 8. While the DUR algorithm performs well in throughput aspect, it is quite poor in this category. The preemption ratio is close to 20% when traffic load approaches between 7 and 8 in the P_32 data pattern. The LRA algorithm, on the other hand, keeps the ratio below 13% even for when traffic load is 8. The LRA algorithm

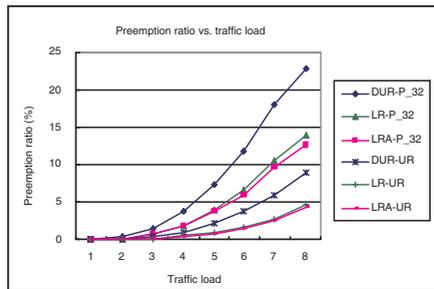


Fig. 8. Preemption ratio comparison vs. traffic load for LR, LRA and DUR algorithms

Table 1. Reassignment ratios under uniform data rate pattern

Traffic load	LR Algorithm	LRA Algorithm	DUR Algorithm
3	0.0028	0.0029	0.0030
4	0.0092	0.0097	0.0121
5	0.0175	0.0178	0.0200
6	0.0312	0.0325	0.0387
7	0.0475	0.0510	0.0572
8	0.0686	0.0733	0.0851

obtains lower preemption ratio than LR algorithm, and at the same time, higher throughput performance.

Reassignment ratios (the fraction of reassigned requests to the accepted requests) for different algorithms under the uniform data rate pattern are shown in Table 1. As the value are all very low, the LR algorithm outperforms the other two as the system may find available OVSF code more easily since the bandwidth utilization is comparatively lower.

5 Conclusions and Future Work

The competitiveness for an on-line program is valuable in that it guarantees the performance in all input sequences. To achieve better throughput performance, it is sometimes suitable to reassign and/or preempt the previous accepted requests. This paper presents a modified version of competitive algorithm called LRA. The LRA algorithm for OVSF code assignment problem is proved and demonstrated to improve the original LR algorithm by reducing the unnecessary preemptions to the requests, and also achieving higher throughput performance than the theoretical value.

Traffic patterns other than Poisson distribution may cause different behavior of the three algorithms in discussion. This is the interesting topic to work into. As MC-CDMA and hybrid method other than OVSF code assignment schemes are paid much attention, the further studies of the throughput competitive analysis in both non-preemption and preemption scenarios are also worthwhile.

References

1. Special Issue, IMT-2000: Standards Efforts of the ITU, IEEE Pers. Commun., vol. 4, Aug. 1997
2. R. Prasad and T. Ojanpera, "An Overview of CDMA Evolution Towards Wideband CDMA," IEEE Communications Surveys, vol. 1, no. 1, 4-th Quarter, pp. 2–29, 1998
3. TR25.922, "Radio Resource Management Strategies," 3GPP, V3.1.0, pp. 25–27, March 2000
4. A. J. Viterbi, CDMA: Principle of Spread Spectrum Communications, Addison-Wesley, 1995
5. A. Baier, U.-C. Fiebig, W. Granzow, W. Koch, P. Teder, and J. Thielecke, "Design Study for a CDMA-based Third Generation Mobile Radio System," IEEE J. Sel. Areas Commun., vol. SAC-12, pp. 733–43, May 1994
6. D. D. Sleator and R. E. Tarjan, "Amortized Efficiency of List Update and Paging Rules," Communications of ACM, vol. 28, pp. 202–08, 1985
7. M. Manasse, L. McGeoch and D. D. Sleator, "Competitive Algorithms for On-line Problems," Proc. ACM Symposium on Theory of Computing, pp. 322–33, May 1988
8. J. A. Garay and I. S. Gopal, "Call Preemption in Communication Networks," Proc. INFOCOM '92, vol. 3, pp. 1043–50, 1992
9. J. A. Garay, I. S. Gopal, S. Kutten, Y. Mansour, and M. Yung, "Efficient On-line Call Control Algorithms," Proc. 2nd Israel Symp. on the Theory and Computing Systems, pp. 285–93, 1993
10. A. Bar-Noy, R. Canetti, S. Kutten, Y. Mansour, and B. Schieber, "Bandwidth Allocation with Preemption," Proc. ACM Symp. on Theory of Computing, pp. 616–25, May 1995
11. T. Minn and K.-Y. Siu, "Dynamic Assignment of Orthogonal Variable Spreading Factor Codes in W-CDMA". IEEE J. Sel. Areas Commun., vol. 18, no. 8, pp. 1429–40, Aug. 2000
12. C.-L. I and R. D. Gitlin, "Multi-code CDMA Wireless Personal Communications Networks," Proceedings of ICC'95, vol. 2, pp. 1060–64, 1995
13. F. Adachi, M. Sawahashi, and K. Okawa, "Tree-structured Generation of Orthogonal Spreading Codes with Different Length for Forward Link of DS-SS Mobile Radio," Electronics Letters, vol. 33, no. 1, pp. 27–28, Jan. 1997
14. S. H. Kim., S. H. Shin, and K. S. Kwak, "A Hybrid Multi-rate Scheme for WCDMA," Proceedings of TENCON' 99, vol. 2, pp. 1224–27, Dec. 1999
15. F. Shueh, and W.-S. E. Chen, "Minimizing Reallocation on OVFS Channel Assignment," in preparation
16. W.-T. Chen, Y.-P. Wu, and H.-C. Hsiao, "A Novel Code Assignment Scheme for W-CDMA Systems," Proceedings of VTC'2001 Fall, vol. 2, pp. 1182–86, 2001
17. R.-G. Cheng and P. Lin, "OVFS Code Channel Assignment for IMT-2000," Proceedings of VTC'2000 Spring, vol. 3, pp. 2188–92, 2000
18. F. Shueh, Z.-E. P. Liu, and W.-S. E. Chen, "A Fair, Efficient, and Exchangeable Channelization Code Assignment Scheme for IMT-2000," IEEE ICPWC'2000, pp. 429–33, Dec. 2000
19. F. Shueh and W.-S. E. Chen, "Code Assignment for IMT-2000 on Forward Radio Link," Proceedings of VTC'2001 Spring, vol. 2, pp. 906–10, 2001
20. A. C. Kam, T. Minn, and K.-Y. Siu, "Supporting Rate Guarantee and Fair Access for Bursty Data Traffic in W-CDMA," IEEE J. Sel. Areas Commun., vol. 19, no. 11, pp. 2121–30, Nov. 2001
21. B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-Competitive On-line Routing," IEEE Proc. Foundations of Computer Science, pp. 32–40, 1993