ISSN:0975-9646

A Comparative Analysis of Proposed Schemes for Increasing Channel Utilization In 3G Networks

Shekhar Tyagi[1], **Prof. Hariom Tyagi**[2]

[1] Research Scholar M.TECH, Deptt. Of Computer Science, R.D. Engineering College, Ghaziabad, Uttar Pradesh, India
[2] Professor, Department Of Computer Science, R.D.Engineering College, Ghaziabad, UP, India

Abstract- In recent years fast increase of the traffic of the non telephone services in a cellular phone is remarkable and in the next generation of wireless system, it is expected to be the era of the multimedia communications . In order to provide such integrated services plateform , the system must be able to multiplex users with different transmission rates for different service types . In this background , the next generation mobile communication standard so called IMT-2000 based on WCDMA provides the multimedia communication. In this paper, we have proposed different algorithms and their evaluation for OVSF and NOVSF codes for their performance evaluation in the sense that all codes are orthogonal to each other and no code blocks the assignment of any other NOVSF code.By evaluating the performance of OVSF and NOVSF codes in W-CDMA, by calculating their Blocking chances and number of calls rejected against new call arrival rate (request per minute) at different data rates (R, 2R, 4R and 8R where R can be either 8kbps or 16 kbps) . NOVSF is a new code assignment scheme that greatly reduces the code blocking problem. There are a number of techniques of NOVSF code generation. We have proposed three of them . The common purpose of all of them is to minimize the blocking probability and the reallocation codes cost so that more of new arriving call requests can be supported. This work is related with nonblocking OVSF (NOVSF) codes, to increase substantially the utilization of channelization codes without having the overhead of code reassignments.

1. INTRODUCTION: OVSF-NOVSF CODES

Orthogonal variable spreading factor (OVSF) codes are used for differentiating users in W-CDMA access technology of IMT-2000. OVSF codes have the advantage of supporting variable bit rate services which is important for emerging multimedia with different bandwidth requirements. OVSF codes are employed as channelization codes in W-CDMA. Any two OVSF codes are orthogonal if and only if one of them is not a parent code of the other. Therefore, when an OVSF code is assigned, it blocks entire ancestor and descendant codes from assignment because they are not orthogonal to each other . This code-blocking problem of OVSF codes can cause a substantial spectrum efficiency loss. Efficient channelisation code management results in high code utilization and increased system capacity. NOVSF is a new code assignment scheme that greatly reduces the code blocking problem. There are mainly three techniques of NOVSF code generation. The common purpose of all of them is to minimize the blocking probability and the reallocation codes cost so that more of new arriving call requests can be supported

2. MULTIPLE ACCESS SYSTEMS

Wireless telecommunications has dramatically increased in popularity, resulting in the need for technologies that allow multiple users to share the same frequency. These are called "multiple access systems." The three types of multiple access system are:

- Frequency Division Multiple Access (FDMA)
- Time Division Multiple Access (TDMA)
- Code Division Multiple Access (CDMA)

For radio systems there are two resources, frequency and time. Division by frequency, so that each pair of communicators is allocated part of the spectrum for all of the time, results in Frequency Division Multiple Access (FDMA). Division by time, so that each pair of communicators is allocated all (or at least a large part) of the spectrum for part of the time results in Time Division Multiple Access (TDMA). In Code Division Multiple Access (CDMA), every communicator will be allocated the entire spectrum all of the time. CDMA uses codes to identify connections.

3. WCDMA IN 3RD Generation

Analog cellular systems are commonly referred to as first generation systems. The main first generation standards are AMPS, TACS, and NMT. The digital systems currently in use, such as GSM, PDC, cdma One (IS-95) and US-TDMA (IS-136) are second generation systems. These systems have enabled voice communications to go wireless in many of the leading markets, and customers are increasingly finding value also in other services such as text messaging and access to data networks, which are starting to grow rapidly

Third generation systems are designed for multimedia communication: with them person-to-person communication can be enhanced with high quality images and video, and access to information and services on public and private networks.

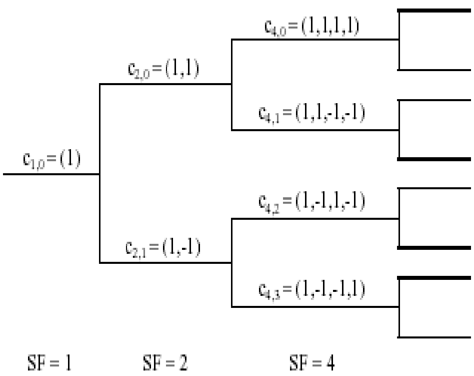
4.CHHANELIZATION CODES

Flexible communication capabilities of third generation systems, together with the continuing evolution of the second-generation systems, will create new business opportunities not only for manufacturers and operators, but also for the provider.

In the standardization forums, WCDMA technology has emerged as the most widely adopted third air interface. Its specification has been created in 3GPP (The Third Generation Partnership Project), which is the joint standardization project of the standardization bodies from Europe, Japan, Korea, the USA and China. Within 3GPP, WCDMA is called UTRA (Universal Terrestrial Radio Access) FDD (Frequency Division Duplex) and TDD (Time Division Duplex), the name WCDMA being used to cover both FDD and TDD operation.

The channelization code for the Primary CPICH is fixed to $c_{256,0}$ and the channelization code for the Primary CCPCH is fixed to $c_{256,1}$.The channelization codes for all other physical channels are assigned by UTRAN.

When compressed mode is implemented by reducing the spreading factor by 2, the OVSF code of spreading factor SF/2 on the path to the root of the code tree from the OVSF code assigned for normal frames is used in the compressed frames. For the case where the scrambling code is changed during compressed frames, an even numbered OVSF code used in normal mode results in using the even alternative scrambling code during compressed frames, while an odd numbered OVSF code used in normal mode results in using the odd alternative scrambling code during compressed frames.The ovsf codes can be defined as-



In Figure , the channelization codes are uniquely described as $C_{SF,k}$, where SF is the spreading factor of the code and *k* is the code number, 0 <=*k* <=SF-1.

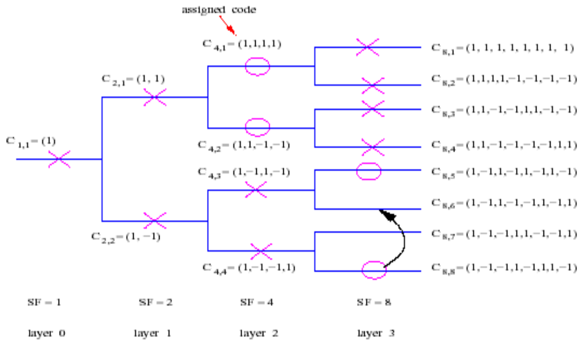
generation method for the channelization code is defined as:

$$c_{1,0} = 1,$$

$$\begin{bmatrix} C_{2,0} \\ C_{2,1} \end{bmatrix} = \begin{bmatrix} C_{1,0} & C_{1,0} \\ C_{1,0} & -C_{1,0} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} C_{2^{(n+1)},0} \\ C_{2^{(n+1)},1} \\ C_{2^{(n+1)},2} \\ C_{2^{(n+1)},3} \\ \vdots \\ C_{2^{(n+1)},2^{(n+1)}-2} \\ C_{2^{(n+1)},2^{(n+1)}-1} \end{bmatrix} = \begin{bmatrix} C_{2^n,0} & C_{2^n,0} \\ C_{2^n,0} & -C_{2^n,0} \\ C_{2^n,1} & C_{2^n,1} \\ C_{2^n,1} & -C_{2^n,1} \\ \vdots & \vdots \\ C_{2^n,2^n-1} & C_{2^n,2^n-1} \\ C_{2^n,2^n-1} & -C_{2^n,2^n-1} \end{bmatrix}$$

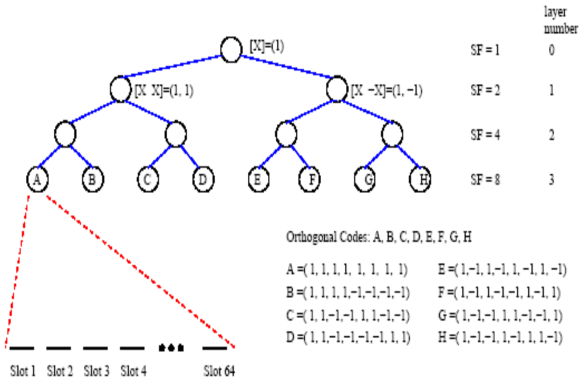
4.1 Orthogonal code tree generation and blocking problem An OVSF code tree is a binary tree with ten layers, labeled from 0 to 9 starting with the root node, such that SF of codes at layer *k* is equal to 2*k*. As stated earlier, any two OVSF codes are orthogonal if and only if one of them is not a parent code of the other. Therefore, when an OVSF code is assigned to a channel, it blocks its entire ancestor and descendant codes from assignment because they are not orthogonal to each other. For instance, the assignment of code *C*4;1 shown in Fig. blocks the assignment of its ancestor codes (i.e., *C*2;1 and *C*1;1) and descendant codes(i.e., *C*8;1 and *C*8;2). The circle and cross signs on the links indicate the assigned and blocked codes, respectively. For instance, the assignment of code *C*4;1 blocks the assignments of *C*2;1, *C*1;1, *C*8;1, and *C*8;2 because they are either ancestors or descendants of*C*4;1. Code*C*4;4 can be prevented from being blocked by freeing*C*8;8 and reassigning code *C*8;6 to the channel of *C*8;8.



5. NON BLOCKING OVSF CODES

The main objective of these codes is to improve the utilization of OVSF codes without the overhead of code reassignments. To achieve this, only a single layer of OVSF codes with SF is taken into consideration and time multiplexing is applied to share them among channels. This implies that both time and code multiplexing are used in NOVSF codes. Note that all OVSF codes of the same layer are orthogonal to each other and, therefore, do not block each other. Each code may be shared in time among more than one channel. The number of time slots in an OVSF code with SF 8 can be variable or fixed. If it is variable, then we need to introduce a variable, say cycle length, to indicate the number of time slots, which requires that receiver be informed about the cycle length during transmission. We assume that the number of time slots is fixed and equal to 64. In this case, assigning one time slot of an OVSF code with SF 8 would be equivalent to assigning an OVSF code with SF 512 to a channel without any time multiplexing. Similarly, when all 64 time slots of an OVSF code are assigned to the same channel, the supported data rate becomes the same as the one that would be obtained in case of assigning an OVSF code with SF 8 without any time multiplexing. Thus, if all 64 time slots of a code are not assigned to the same channel, the data over the channel are transmitted intermittently. Figure illustrates 8 OVSF codes with SF 8, namely, *A*, *B*, *C*, *D*, *E*, *F*, *G*, and *H*. Each code has 64 time slots, each corresponding to a sequence of 8 chips. Hence, there are 64 chip sequences in all 64 time slots, resulting in a total of 512 = 64_8 chips. The date rate supported by each time slot is equivalent to the data rate that an OVSF code with SF 512 can support. Each time slot of the WCDMA standard frame can carry 2560 chips, which implies that there are 5 transmissions of 64 time slots in a frame. If the date rate supported by a time slot is denoted by *R*, then data rate supported by *K* time slots equals *R_K*. The time slots that are assigned to a channel do not have to be consecutive. Fig.2.8 illustrates how the first 8 time slots of two NOVSF codes, namely, *A* and *B*, may be shared in time among five different channels at some point of time. The data rates corresponding to the OVSF codes with SF 512, 256, 128, 64, 32, 16, and 8 are obtained by 6 assigning 1, 2, 4, 8, 16, 32, or 64 time slots, respectively, that are a power of 2. Indeed, since any number of time slots may be assigned to a channel, many intermediate data rates can be supported in channels when NOVSF codes are employed. Moreover, the way that NOVSF codes are shared in time is somewhat similar to how an OVSF code is shared in time among more than one interactive and/or background traffic users in Channel

Figure -The NOVSF code-tree containing eight orthogonal OVSF codes with SF=8 each. Each of these eight codes has 64 time slots.



6. Performance Of Code Assignment Algorithms For Ovsf And Novsf Codes

Dynamic code assignment schemes have the ability to enhance statistical multiplexing and spectral efficiency of WCDMA systems. Code assignment schemes determine how to allocate codes to different channels. Because OVSF codes are very valuable resources in WCDMA, they should be managed properly to support as many users with different QoS requirements as possible. A number of code assignment algorithms for OVSF codes are introduced in the literature. The code assignment algorithm assigns codes to low data rate users in a manner that maximizes the available number of low SF codes corresponding to high data rate codes. They assume that a user can be assigned multiple OVSF codes, requiring multiple RAKE receivers. In code blocking problem is mitigated by reassigning existing users to new codes in a manner to maximize the available number of low SF codes, without addressing to support different types of traffic. This algorithm may lead to a chain of code reassignments, resulting in a lot of overhead for informing receivers about the change of code assignments. The code assignment algorithm supports both real-time and non-real-time traffic, without addressing the problem of efficiently sharing OVSF codes between a numbers of bursty traffic users. Another code assignment algorithm shares bandw/. idth between bursty traffic sources with different QoS requirements by dynamically changing the spreading code and bandwidth at the cost of increased complexity.

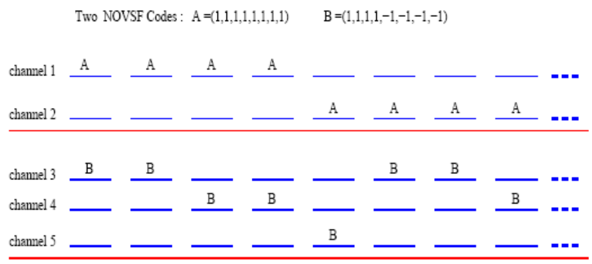


Figure shows Two NOVSF codes are shared in time among at least five different channels.

Due to limited space, only few time slots of each code's 64 time slots are shown. The first time slots of codes *A* and *B* are shared by two and three, respectively, channels. Code reassignments have substantial overhead, and there may not be sufficient network resources to accomplish code reassignments. First of all, code reassignments require all those receivers whose channels are involved with code reassignments have to be informed about the assignments of new OVSF codes. In signaling radio bearers (SRBs) are proposed to minimize the signaling overhead that occur while receivers are informed about new code assignments. According to the radio resource control (RRC) messages are sent on four particular radio bearers called signaling radio bearers (SRBs) whose total data rate equals 3.4 kbps and the transmission time interval (TTI) is 40 ms. But, SRBs are also used in handover, negotiation and renegotiation of QoS parameters. This implies that SRBs may not be available at the time of code reassignments. In addition, more than one call may need chains of code reassignments at the same time. Hence, code reassignments may not be accomplished even if the network has excess capacity to support a new call. If SRBs are available at the time of code reassignments, code reassignments may take as much as 1,600 m sec . As the code reassignments are performed for one call, those new calls requiring code reassignments are queued or dropped. The proposed NOVSF codes enhance the utilization of codes by improving spectral efficiency up to 25% in WCDMA without doing any code or time-slot reassignments. I have compared the performance of OVSF code reassignment techniques with a preliminary code allocation technique for NOVSF codes. Where the number of new call rejects determines the call blocking probability.

Check whether there is any increase on the total cell interference when NOVSF codes with time multiplexing are employed. The simulation results indicate that the aggregate cell interference in NOVSF codes with time multiplexing is not more than that in OVSF codes, assuming that the same amount of traffic is supported in both cases. Note that there are normally at most eight users at any point of time in case of NOVSF codes with time multiplexing.

7. Type 2 Of Novsf Codes

This type of NOVSF codes can be described in three different cases. In all cases, OVSF codes are reorganized in code trees such that all the codes of code tree are orthogonal to each other. The reason why the codes in the first two cases are orthogonal is as follows. There are initially $X1;X2; : : : ;Xi$ orthogonal codes with the same spreading factor (SF) that is equal to *i*, where either *i* = 4 or *i* = 8. Let code *Xj, j _ i*, generate *nj* orthogonal codes with the *same* *SF*, where *n j* is a power of 2. All of these *n j* orthogonal codes with the same SF are placed on the same distinct layer of a code tree. Therefore, all the codes of the resulting code tree are still orthogonal to each other. Case 3 starts with four codes as in Case 1, but the descendants of a code can be assigned to more than one tree layer.

Case 1: *NOVSF codes with four initial orthogonal codes.*
In this case, there are initially four orthogonal codes, namely, *A*, *B*, *C*, and *D*. Using these four orthogonal codes, a binary code tree is constructed as follows. Code *A* is made the root code with *SF* = 4 in the layer 1 of the tree. For the tree layer 2, the following two orthogonal codes with *SF* = 8 are generated from code *B*: (*B*;*B*) and (*B*;_*B*). Similarly, four codes are generated from code *C* and are placed on layer 3 of the tree. Finally, eight generated codes from *D* are placed on layer 4 of the tree. All the codes of the tree are orthogonal to each other and, they can be very desirable codes for broadband fixed wireless networks where maximum SF should not exceed 32. Indeed, what is required is to have a code tree of four layers in this case, but the SF of codes at any one of these four layers can be equal to any power of 2 ranging between 4 to 512, depending on the requested data rates of users. For instance, the SFs of the code tree could be 16, 4, 32, and 64 at some instant of time.

Case 2: *NOVSF codes with eight initial orthogonal codes with SF from 8 to 512.*
In this case, as shown in Figure, there are initially eight orthogonal codes, namely, *A*, *B*, *C*, *D*, *E*, *F*, *G*, and *H*. Using the first seven orthogonal codes, a binary code tree is constructed as follows. Code *A* is made the root code with *SF* = 8 in the layer 1 of the tree. For the tree layer 2, the following two orthogonal codes with *SF* = 16 are generated from code *B*: (*B*;*B*) and (*B*;_*B*). Similarly, four codes are generated from code *C* and are placed on layer 3 of the tree. As illustrated in Figure 7, codes *D*, *E*, *F*, and *G* generate 8, 16, 32, and 64 codes, respectively, and are placed on layers 4, 5, 6, and 7, respectively. Code *H* can be used as a standby code in any tree layer whenever more codes are needed. Indeed, each one of the eight codes *A*, *B*, *C*, *D*, *E*, *F*, *G*, and *H*. can have any spreading factor depending on the requested data rates. For instance, if there are eight users.

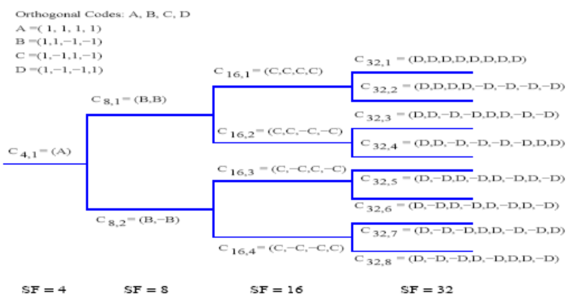


Figure showing NOVSF codes with four initial orthogonal codes.

In this figure, it is assumed that SF ranges from 4 to 32. But, SF can indeed range from 4 to 512. For instance, the SFs of the tree layers may be 4, 8, 32, and 128. requesting codes with *SF* = 8, then each layer is assumed to be assigned a code with *SF* = 8. In general, any layer of the tree can have *X*=8 orthogonal codes with the spreading factor of *X*, where *X* is a power of 2 ranging from 8 to 512. This implies that, without considering the standby code *H*, there may be at most 64 codes in each layer. When *H* is also considered, one layer can have as many as 128 codes.

Case 3: *NOVSF codes with four initial orthogonal codes with SF from 4 to 512.*
In this case, there are initially four orthogonal codes, namely, *A*, *B*, *C*, and *D* as in Case 1. It is the same as Case 1 except that the descendants of a code in this case can be assigned more than one layer with the condition that only orthogonal descendants can be assigned.

7.2 Type 3 of NOVSF Codes

This type of NOVSF codes are generated systematically when there is no limit on the upper bound of SF [1]. To describe the systematic generation of all orthogonal codes for *SF* _ 4, we first define BOVSF codes and then NOVSF codes. **BOVSF codes**: *1)* Let *A* =[1] be the root BOVSF code, as *A* =[1] is also the root OVSF code. *2)* Use each BOVSF code *X* to generate two orthogonal codes: [*X*; *X*; *X*; *X*] and [*X*; _*X*], where _*X* is the inverted sequence of *X*. Using this procedure recursively, generate all BOVSF codes that can be represented as nodes of a balanced binary tree. BOVSF codes have the same property as OVSF codes, that is, *i*) all BOVSF codes of the same layer of the BOVSF code-tree are orthogonal to each other, and *ii*) any two codes of different layers are orthogonal except for the case that one of the two codes is a parent code of the other.

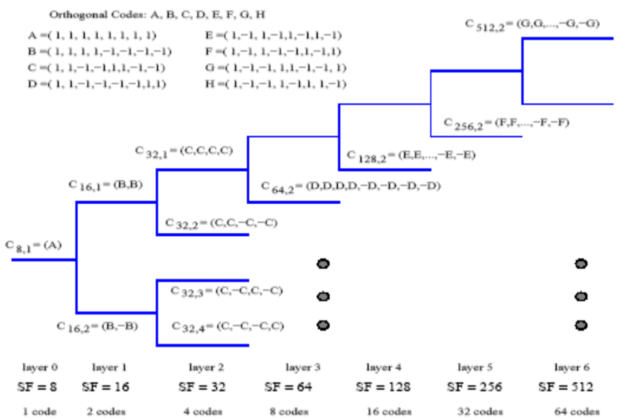


Figure showing The binary code tree for NOVSF codes with 8 _ *SF* _ 512.
(Only one NOVSF code is illustrated in layers 4 to 7 due to space limitations.

8. Proposed Algorithm For Novsf

8.1 Algorithm for Type 1 NOVSF

1. Start.
2. Generate the NOVSF code tree layer 4; having 8 element (codes), here each element is an array of 1*64,

showing 64 slots of each code. All the element of the arrays are zero initially.

3. Initialize an array of new calls arriving with values 20, 40, 60,….., 500.
4. Repeat the following for each element of the array of new calls:

4(a) Divided the element of the array into 4 equal groups

4(b) Repeat the following

(1) for group no. 1

(I) Search tree layer for free slots by checking the value of array element/slots. If 0 then assign the code slots and make the value of slot 1, else keep on following this till either no user is left without code or all the code slots of each code are assigned.

(II) Check the number of users assigned, if this number is less then the number of user to be assigned take this difference as number of blocked code and add them with last computed numbers of blocked codes.

(2) for group no. 2

(I) Search tree layer for two free slots in the same code by checking the value of array element/slots. If 0 assign both the code slots and make value of slots 1, else keep on following this step till either no user is left without code or all the code slots of each code are assigned.

(II) Check the number of users assigned, if this number is less then the number of users to be assigned take this difference as number of blocked codes and ad them with last computed numbers of blocked codes.

(3) for group no. 3

(I) Search tree layer for four free slots in the same code by checking the value of array elements/slots. If 0 assign both the cod slots and make the value of slots 1, else keep following this step till either no user is left without code or all the code slots of each code are assigned.

(II) Check the number of users assigned, if this number is less then the number of user to be assigned take this difference as number of blocked codes and add them with last computed numbers of blocked codes.

(4) for group no. 4

(I) Search the layer for eight free slots in the same code by checking the value of array element/slots. If 0 assign both the code slots and make the value of slots 1, else keep on following this step till either no user is left without code or all the code slots of each code are assigned.

(II) Check the number of users assigned, if this number is less then the number of user to be assigned take this difference as number of blocked codes and add them with last computed numbers of blocked codes.

4(c) Sum the values of the codes blocked and assign it to an array which is equal in dimension with new call array.

5. Calculate the blocking probability by dividing each element of code blocked array.
6. Plot the graph blocking probability Vs new call arrival rate (request/minute)
7. End

8.2 Algorithm for OVSF

1. Start
2. Generate the OVSF code tree using a structure with fields sf blocked, assigned, parent, right and left.
3. Initialize an array of new calls arriving with values 20,40,60,….., 500.
4. Repeat the following for each element of the array of new calls:

4(a) Divide the element of the array into 4 equal groups

4(b) Repeat the following

(1) for group no.1 (from left to right)

(I) Search the leaf layer "(i.e. 9th layer) for free codes by checking the block and assigned field; if blocked and assigned are 0 assign the code and make the values of fields 1, else increment the value of blocked code by 1.

(2) for group no. 2 (from left to right)

(I) Search the eight layer for free codes by checking the blocked and assigned field; if blocked and assigned fields are 0 then assign the code and make the values of fields 1, else increment the value of blocked code by 1.

(3) for group no.3 (from left to right)

(I) Search the seventh layer for free codes by checking the blocked and assigned field; if blocked and assigned fields are 0 then assign the code and make the values of fields 1, else increment the value of blocked code by 1.

(4) for group no.3 (from right to left)

(I) Search the sixth layer for free codes by checking the blocked and assigned field; if blocked and assigned fields are 0 the assign the code and make the values of fields 1, else increment the value of blocked code by 1.

4 (c) Sum the value of the codes blocked and assigns it to an array which is equal in dimension with new call array.

Calculate the blocking probability by dividing each element of code blocked array by corresponding element of the new call array.

8.3 Algorithm for Type 2 NOVSF

1. Start.
2. Generate the NOVSF code tree using a structure with field's sf, assigned, parent, right and left.
3. Initialize an array of new calls arriving with values 20, 40, 60,….., 500.
4. Repeat the following for each element of the array of new calls:

(a) Divide the element of the array into 4 equal groups

(b) Repeat the following

(1) for group no. 1 (from left to right)

(I) search the leaf layer (i.e.8th layer) for free codes by checking the Assigned field; if assigned field is 0 then assign the code and Make the value of field 1, else increment the value of blocked Code by 1.

(2) for group no. 2

(I) Search the seventh layer for free codes by checking the assigned Fields, if assigned field is 0 then assign the code and make the value of field 1, else increment the value of blocked code by 1.

(3) for group no. 3(from left to right)

(I) Search the sixth layer for free codes by checking the assigned field; if Assigned field is 0 then assign the code and makethe value of field 1, Else increment the value of blocked code by 1.

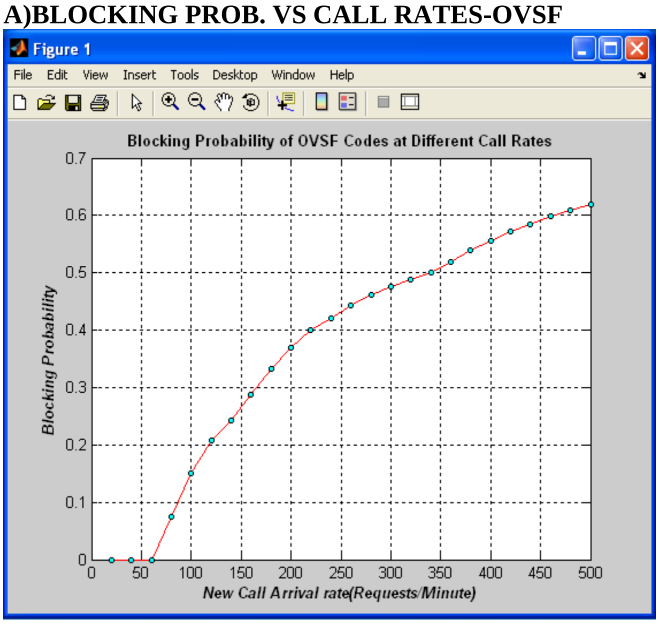
(4) for group no. 4 (from right to left)

(I) Search the fifth layer for free codes by checking the assigned field; if Assigned field is 0 then assign the code and make the value of field Else increment the value of blocked code by 1.

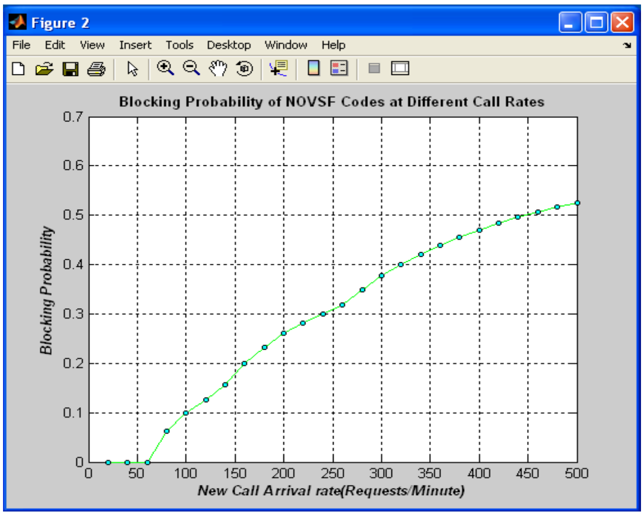
(c) Sum the value of the codes blocked and assign it to and array which is equal in dimension with new call array

9. SIMULATION RESULTS

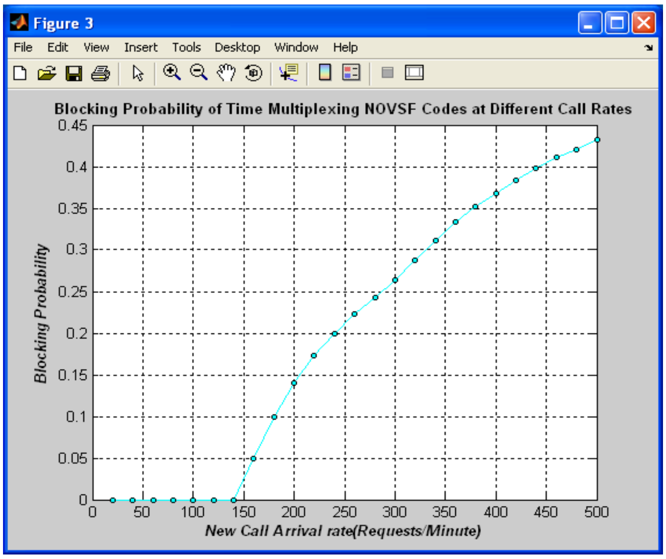
A)BLOCKING PROB. VS CALL RATES-OVSF



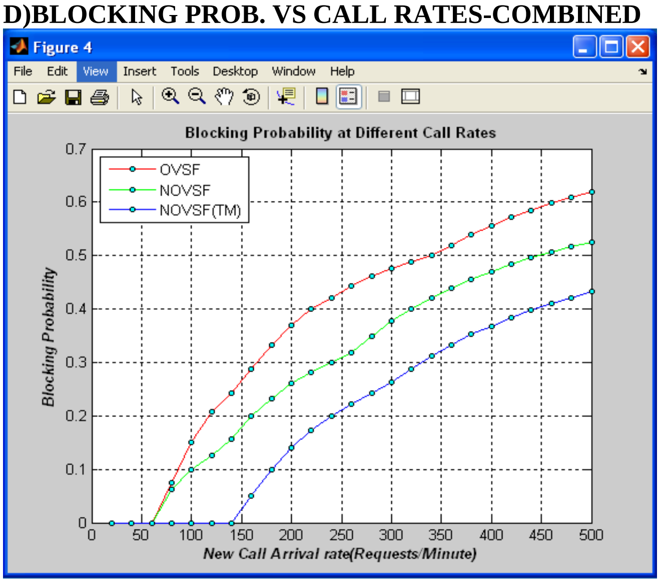
B)BLOCKING PROB. VS CALL RATES-TYPE1 NOVSF



C)BLOCKING PROB. VS CALL RATES TYPE2 NOVSF



D)BLOCKING PROB. VS CALL RATES-COMBINED



10.CONCLUSION AND FUTURE WORK

In order to evaluate the performance of OVSF and NOVSF codes, three algorithms are proposed and implemented for OVSF and two different types of NOVSF codes (Type 1 and Type 2) . The individual performance as well as relative performance is calculated in terms of Blocking probabilities of calls and Numbers of calls rejected against different requested call rates.

As the graphs of Blocking probability Vs New call arrival rate (request/minute) show that Time Multiplexing NOVSF codes produce less code blocking then NOVSF codes with eight initial orthogonal codes at different call rates, which again produce less code blocking than OVSF codes. So NOVSF codes of type-2 and Time Multiplexing NOSF codes lead to increased system capacity and high code utilization. Therefore, Time Multiplexing Non Blocking OVSF codes are better option for channelization codes in W-CDMA system in near future.

As Non-Blocking orthogonal Variable Spread Factor (NOVSF) codes are non-blocking in nature, any code can be assigned for channelization purpose in W-CDMA system. In future different algorithms may be developed giving better system capacity and high code utilization.

REFERENCES

[1] H. Cam, "Nonblocking OVSF codes and enhancing network capacity for 3G Wireless and beyond systems",Computer Communications, Vol. 26, No. 17, pp. 1907-1917, (Nov. 2003).

[2] Mann A. S. A l-Adwany* , Mustafa B. Ayoub ,Performance Evaluation Of Nonblocking OVSF Codes in WCDMA Systems , 2010

[3 Mehdi Askari, Reza Saadat, Mansour Nakhkash , The Comparison of Various Code Assignment Schemes in Wideband CDMA , 2008

[4] Shailendra Mishra , Non Blocking OVSF Codes .

[5] "Nonblocking OVSF Codes and Enhancing Network Capacity for 3G Wireless and Beyond System" by Hasan Cam,(2002).

[6] Davinder S.Saini and S.V.Bhooshan, "Adaptive assignment scheme for OVSF**",** in proc of IEEE ICWMC'06, Bucharest", July 2006.

[7] Esmal H. Diman and Bijan Jabbari, "Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks," IEEE Communications Magazine, vol.36, pp. 48-54, September 1998.

[8] Third Generation Partnership Project Technical Specification Group Radio Access Network Working Group 1, "Multiplexing and Channel Coding (FDD)." TS 25.212 V5.0.0 (2002-03).

[9] Third Generation Partnership Project Technical Specification Group Radio Access Network Working Group 1, "Physical Layer Procedure (FDD)," TS 25.214 V5.0.0 (2002-03).

[10] Chapman, S.J. 2002. MATLAB® Programming for Engineers. Brooks/Code: Pacific Grove.

[11] Performance Analysis of Non blocking OVSF Codes in WCDMA" by Haan Cam and Kiran Vadde.