# 行政院國家科學委員會專題研究計畫　成果報告

## 限制獨立支配問題之研究(2/2)

計畫類別：個別型計畫
計畫編號：NSC92-2213-E-216-001-
執行期間：92 年 08 月 01 日至 93 年 07 月 31 日
執行單位：中華大學資訊工程學系

計畫主持人：俞征武

報告類型：完整報告

處理方式：本計畫可公開查詢

中　華　民　國 93 年 12 月 1 日

# 行政院國家科學委員會專題研究計畫成果報告

## 隨意網路上之 OVSF 碼管理問題
## OVSF Code Management Schemes on Ad Hoc Networks

計畫編號：NSC 92-2213-E-216-001
執行期限：92 年 8 月 1 日至 93 年 7 月 31 日
主持人：俞征武　執行機構及單位名稱:中華大學資訊工程學系

### 一、 中文摘要

在此計劃中我們提出隨意網路中六種 Orthogonal Variable Spreading Factor (OVSF)碼的管理機制。因為 OVSF 碼是原先為 IMT-2000 設計的集中管理 DS-CDMA 系統。故不可直接應用於分散式的隨意網路上。我們所提出的六種 OVSF 碼的管理機制方法包含重新分配碼和碼的放置策略。實驗結果驗證我們的方法顯著地降低阻塞率。

關鍵字： 隨意網路、OVSF 碼、演算法

Abstract

In the project, we present the Orthogonal Variable Spreading Factor (OVSF) code management schemes on Ad Hoc networks. Ad Hoc networks are wireless networks without fixed infrastructure. Each mobile node in the network may move arbitrarily, and therefore network topology changes frequently and unpredictably. Since the OVSF codes are originally used as the channelization codes in the DS-CDMA system of IMT-2000, previous schemes are centralized and cannot be applied directly to fully distributed systems such as Ad Hoc networks. Totally, six distributed code management schemes are proposed in the project, and three of them heavily exploit two techniques: code reassignment and code tree management. Simulation results show that our schemes, with the help of the techniques, reduce the call-blocking rate dramatically.

Keywords-code assignments, code replacement, Ad Hoc networks, distributed algorithms, OVSF-CDMA

### 二、緣由與目的

The third generation wireless standards UMTs/IMT-2000 [1, 8, 21] use wide-band CDMA (W-CDMA) to address the higher and variable rate requirements of multimedia application. Three different schemes of DS-CDMA transmission were proposed: single orthogonal variable-spreading-factor code (OVSF-CDMA) [1], multicode CDMA (MC-CDMA), and the hybrid method [7]. MC-CDMA requires multiple transceiver units, but OVSF-CDMA requires only a single transceiver unit. Therefore, in terms of hardware complexity, OVSF-CDMA is preferred over MC-CDMA. However, the code-blocking problem in OVSF-CDMA results in a higher call-blocking rate for higher data rate users. Thus the OVSF code management becomes an important design consideration in wireless networks, and has received a lot of attention [5, 6, 17, 20].

Ad Hoc networks are wireless networks without fixed infrastructure. Each mobile node in the network functions as a router that discovers and maintains routes for other nodes. These nodes may move arbitrarily, and therefore network topology changes frequently and unpredictably. Other limitations of Ad Hoc networks include high power consumption, low bandwidth, and high error rates [19]. Applications of ad hoc networks are emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, data acquisition operations in inhospitable terrain, and automated battlefield [19].

The project discusses the OVSF code management problem on Ad Hoc networks. Since the OVSF codes are originally used as the channelization codes in the DS-CDMA system of IMT-2000, previous schemes proposed are centralized and cannot be directly applied to fully distributed systems such as Ad Hoc networks. In the project, we propose six distributed code management schemes, and three of them heavily exploit two techniques: code exchange and code tree management. We also conduct simulations by using ns-2 [9], and simulation results demonstrate that our schemes reduce the call-blocking rate dramatically.

The rest of the project is organized as follows. In the next section, some preliminaries are introduced. In Section 3, we briefly survey related work. In Section 4, six distributed code management schemes are presented. Simulation results are described and discussed in Section 5. Finally, Section 6 concludes the project.

### 三、研究報告應含的內容

The OVSF code tree, the basic data structure of our OVSF code management schemes, is presented first. Then, interference issues in CDMA networks are discussed. At last, we describe Hu's deadlock free orientation code assignment scheme [15], which will be used as a basic step in some of our proposed schemes.

3.1 The OVSF code tree

Spectrum spreading is achieved by mapping each data bit (1 or –1) into an assigned code sequence. The length of the code sequence per data bit is called the *spreading factor*. The possible OVSF codes can be represented as nodes in a complete binary tree called the *OVSF code tree* [2] as shown in Figure 1. Some features of the OVSF code tree are listed as follows. Suppose $c_i$ is the father of $c_j$ in the tree.

(1) Every node $c_j$ except root node $c_1$ (=[1]) can be generated from its father $c_i$. Suppose the code sequence of $c_i$ is [$x$], then we have $c_j$=[$x$, $x$] if $c_j$ is the left child of $c_i$. Or we have $c_j$= [$x$, -$x$] if $c_j$ is the right child of $c_i$. Where [-$x$] is the bitwise complement of [$x$].

(2) If the spreading factor of $c_i$ is $2^k$, the spreading factor of $c_j$ is $2^{k+1}$. Consequently, if the data rate of $c_i$ is $2^r$, then the data rate of $c_j$ is $2^{r-1}$. Hence leaf nodes have the minimum data rate and root node has the maximum data rate.

(3) If a code is used, then simultaneous use of its descendants or ancestors cannot be allowed because their encoded sequences are indistinguishable. For example, when $c_4$ is used, then none of {$c_1$, $c_2$, $c_8$, $c_9$} can be assigned simultaneously.

It is possible that a new call cannot be served even though the total available code capacity can fulfill the acquired transmission rate of the call. For example, in Figure 1, suppose that {$c_4$, $c_{11}$, $c_{14}$} are assigned, and {$c_1$, $c_2$, $c_3$, $c_5$, $c_7$} are thus blocked. Let the data rate supported by leave nodes be $R$. Note that currently the total available code capacity is 4$R$. However, a new call with 4$R$ data rate will be blocked, because none of available codes can support such data rate. If we replace $c_{14}$ with $c_{10}$, then $c_3$ and $c_7$ are unblocked. Thus we have an available code $c_3$, which can support a call with 4$R$ data rate.

In summary, code blocking in OVSF-CDMA may result from the fragmentation of available codes. To reduce code-blocking rate, some schemes resorted to code replacement [17]. Other schemes try to place assigned codes in the code tree properly [5, 20]. We will show that both of code placement and code replacement are considered in our schemes in Section 4.

### 3.2. Interference issues in CDMA networks

In CDMA networks, there are two kinds of interference: direct interference and secondary interference [15]. Both lower the system throughput and increase the average packet delay. To avoid secondary interference, several code assignment schemes require that no set of stations that are two hops away have the same code [15].

Define a binary relation $\rightarrow$ between codes in an OVSF code tree as follows. We have $c_i \rightarrow c_j$ if $c_i$ is an ancestor of $c_j$ in the OVSF code tree. Note that the binary relation $\rightarrow$ is a partial ordering relation [16]. We say that $c_i$ *conflicts with* $c_j$ (denoted by $c_i \leftrightarrow c_j$) if $c_i \rightarrow c_j$ or $c_j \rightarrow c_i$. In OVSF-CDMA systems, two codes cannot be used in the same station if they conflicts with each other. Moreover, no set of stations that are two hops away have conflicted codes.

### 3.3. Hu's deadlock free orientation

Hu proposed two-phase algorithms to assign and reassign CDMA codes to transmitters, receivers, or pairs of stations [15]. The *deadlock-free orientation* (DFO) in their second phase is to exploit parallelism, when assigning codes, without deadlocks. A deadlock is a directed cycle of nodes in which nodes waits permission from the next node, which will never come. Let $V$ denote the node set, and a unique priority $p_i$ is given to every node $i$ in $V$. We creates an arc [$a$, $b$] (from $a$ to $b$) in $E$ if $a$ and $b$ chares a common neighboring node and $p_a < p_b$. Since each node has a unique priority and no circular priority list is permissible, we can view DFO as a cycle-free directed graph $G$=($V$, $E$). We describe DFO for transmitter-based code assignment (TCA) [15] as follows:

Algorithm DFO for TCA:

Step 1. Each node broadcasts its unique priority to its neighbors and collects priority information from its neighbors. Thereafter, each node arranges a chain of neighboring nodes with decreasing priority, and broadcasts the chain to its neighbors.

Step 2. If a node is the head of all priority chains calculated by neighbors, it picks a code for itself and broadcasts a token packet to its children in all chains for acknowledgement.

Step 3. A node can select a code only upon receiving tokens from the parents in all chains. After selecting a code, the node broadcasts a token to its children in all chains.

### 3.5 Related work

In [17], Min and Siu proposed an algorithm to minimize the number of OVSF codes that must be reassigned to support a new call. Their algorithm is based on the concept of assigning a cost function to each candidate branch of the OVSF code tree, and identifying a branch with a minimal cost by using an efficient search. Their algorithm, however, does not consider the code placement problem.

In [6], Cheng and Lin considered another different problem that uses multiple OVSF codes to support multi-rate services. They used a code word to record the number of available codes in each level of the OVSF code tree. With the help of the code word, their heuristic algorithm selects codes according to the following two rules: (1) Preserve more small-spreading-factor codes after selection. (2) Use the least codes. Their algorithm also does not consider the code placement problem.

In [5], Chen et al. devised a dynamic code assignment scheme that statically partitions the entire code tree into two groups, and each group serves traffics with various quality-of-service (QoS) requirements. Code assignments and reassignments are dynamically performed on new and released calls over each separated code group, respectively.

Tseng et al. have proposed the crowded-first strategy to assignment and reassignment codes in WCDMA [20]. When there is one or more than one code in the code tree with the desired rate, the strategy will pick the one whose ancestor code has the least free capacity to accommodate the new call.

Note that all schemes described in the section are centralized, which cannot be directly applied to distributed systems like Ad Hoc networks.

### 3.6 Distributed OVSF code management schemes

We will present six distributed OVSF code management schemes in the section. To avoid interference, our schemes require that no set of stations that are two hops away have codes conflict with each other. Hello exchange mechanism is used in our schemes to collect code use information of two-hop-away neighbors. The first three schemes are simple and straightforward so we described them as follows.

(1) *Random Scheme* (RS): As a new call with a rate *R* is requested, RS randomly picks a feasible code in the code tree until no conflict occurs. When conflicts occur as nodes move, the request of lower priority node will be blocked. Moreover, once a node has been blocked, its status will be sustained to the end even though a free code may exist.

(2) *Deadlock-Free Orientation scheme* (DFOS): In DFOS, each node takes turns to select codes according to the priority chain constructed in Algorithm DFO (described in Subsection 2.3). When selecting a code, a node is required to pick a spare code that does not conflict with code used by its two-hop neighbors. If no free code can be found, the corresponding request will be blocked. When conflicts occur as nodes move, the request of lower priority node will be blocked. Moreover, once a node has been blocked, its status will be sustained to the end even though a free code may exist.

(3) *Local Search Scheme* (LSS): LSS is an improvement on DFOS. As network topology changes, nodes in LSS will alter its blocking status immediately when finding a feasible free code that does not conflict with codes used by its two-hop neighbors at the moment.

The remaining three schemes are improvements on LSS, and will be presented in the following subsections separately.

### 3.7 Global Reassignment Scheme

A request of node *x* denied in LSS indicates that no feasible spare code can be found in *x*'s code tree at that moment. If there exits a two-hop neighbor *w* (of *x*) whose code can be reassigned to *x* and replaced with another code without introducing new conflicts, then the code blocking can be eliminated. *Global reassignment scheme* (GRS) is an improvement on LSS by exploiting this kind of code reassignment when necessary.

We define some useful notations as follows. Let $N_2[x]$ denotes the two-hop neighbors of *x*. Formally, let $N_2[x]=\{y \mid (x, z)$ and $(z, y)$ are two pairs, where the former is a neighboring node of the latter in each pair, and $y \neq x\}$. A code is *spare* with respect to a node *x* if the code does not conflict with codes used by any node in $N_2[x] \cup x$. In GRS, node *x* try to inquire a specific subset $C_{x,2}$ of $N_2[x]$ for finding spare codes. Set $C_{x,2}$ must satisfy the following three conditions:

(1) *The has-assigned condition*: $[\forall y \in C_{x,2}$ (*y* owns codes with the desired level)]. Every node in the set must own at least a code with the desired bandwidth.

(2) *The loop-free inquiry condition*: $[\forall y \in C_{x,2}$ $(p_y < p_x)]$. A *loop inquiry* is a group of inquires that form a loop. For example, $a_1$, $a_2$, $a_3$, …, $a_k$ form a loop inquiry if $a_1$ inquires $a_2$, $a_2$ inquires $a_3$, …, and $a_k$ inquires $a_1$. The loop inquiry may result in deadlock, invalid inquiries, and useless code replacements. No loop inquiry can exist if $C_{x,2}$ satisfies the loop-free inquiry condition, because each node has a unique priority and a circular priority list is obviously a contradiction.

(3) *The safe replacement condition*: $[\forall y \in C_{x,2}$ $(\forall z \in N_2[x]-y$, ($c_y$ does not conflict with $c_z$))]. Where $c_k$ denotes the code assigned to node *k*. The condition ensures that the code assigned to each node *y* in $C_{x,2}$ does not conflict with every code used by nodes in $N_2[x]-y$. No new conflicts will be introduced if node *y* in $C_{x,2}$ releases its code to grant the request of *x*.

GRS uses above conditions to find feasible candidates for possible code reassignments. The algorithmic form of GRS is described as follows.

*The algorithmic form of GRS:*

Step 1. At the beginning, each node performs like LSS.

Step 2. When the request of a node *x* is denied in Step 1, we compute set $C_{x,2}$ based upon three conditions described above. Then node *x* inquire nodes in $C_{x,2}$ one by one according to their priority values decreasingly until a spare code is found. On the other hand, when a node *y* with code $c_y$ in $C_{x,2}$ receives the inquiring packet, GRS checks its code tree to find a spare code $c_z$ for supporting the request of *y*. If the answer is positive, GRS reassigns the spare code to *y* by replacing $c_y$ with $c_z$. Then *y* notices the inquiry initiator *x* to use its previously assigned code $c_y$. Finally, the request of *x* is granted.

Step 3. Let $C_{x,4}=\{z \mid z \in C_{y,2}$ and $z \notin C_{x,2}$, where $y \in C_{x,2}\}$. When the request of a node *x* is denied in Step 2, *x* will inquire nodes in $C_{x,4}$ one by one according to their priority values decreasingly until a spare code is found.

Step 4. If node *x* still find no feasible spare code, its request is blocked at the moment.

### 3.8 Select-by-Votes Scheme (SVS)

We have shown that code blocking may result from the fragmentation of spare codes in Section 2.1. In the section, we propose *Select-by-Votes Scheme* (SVS) to select the desired code by votes from two-hop neighbors.

Define a function $\psi$: $\{T\times R\times I\}\rightarrow Z^+\cup\{0\}$, where $T$ denotes a set of OVSF code trees, $R$ denotes an integer which represents a tree level, $I$ denotes a code in OVSF code trees, and $Z^+$ denotes the set of positive integer respectively. Suppose that $t$ is an OVSF code tree. Let $\psi(t, r, i)$ denote the number of blocked codes which are in the subtree rooted at internal node $i$ of $t$ and at tree level $r$. For example, let $t$ denote the tree shown in Figure 1. Then we have $\psi(t, 3, c_2)=|\{c_4, c_5\}|=2$ and $\psi(t, 3, c_1)=|\{c_4, c_5, c_7\}|=3$. Later, we will use function $\psi$ to aggregate blocked codes and assigned codes together.

The main idea of SVS is described as follows. When a node $x$ requests a code with tree level $r$, SVS searches $x$'s own code tree from the root until finding the desired code. Let $L$ ($R$) denote the left (right) child of the visited node, and let $t_L$ ($t_R$) denote the subtree rooted at $L$ ($R$). If there is exactly one of $t_L$ and $t_R$ own a spare code with tree level $r$, SVS selects the node as the next visited node. If both of $t_L$ and $t_R$ own such a code, SVS selects one of $L$ and $R$ as the next visited node by counting votes from every two-hop neighbor of $x$. Every two-hop neighbor (of $x$) $y$ votes for $L$ if $\psi(t_y, r, L)\geq\psi(t_y, r, R)$. If $\psi(t_y, r, L)<\psi(t_y, r, R)$, node $y$ votes for $R$.

Iterate the similar procedure until the desired code is found in $t_x$.

### 4.3 Backup Code Scheme (BCS)

The last scheme we proposed is called *Backup Code Scheme* (BCS). BCS tries to select a code so that the resulting neighboring nodes' OVSF code trees owning the maximal number of spare codes with the same bandwidth. We also call these codes *backup codes*. Backup codes preserved in neighboring nodes can be used to replace blocked codes when code conflicts occur.

Suppose node $x$ requests a code with level $k$. Let $N(x)=\{y_1, y_2, y_3, \ldots, y_k\}$ is the set of two-hop neighbors of $x$. Let $a_{i,j}$ denote the number of spare codes with level $k$ in the $y_i$'s code tree after $x$ selecting code $j$. BCS selects a code to maximize $Min\{a_{i,j} | \text{where } y_i\in N(x)$, where $j$ is a spare code in $x$'s code tree with level $k\}$.

## 四、結論

In this project, we propose six distributed schemes and three of them heavily exploit two techniques: code reassignment (LSS and GRS) and code tree management (SVS and BCS). Simulation results show that our schemes, with the help of the techniques, reduce the call-blocking rate dramatically. One of our future works is to apply the techniques to other different systems like MC-CDMA. We also note that there has to be a trade-off between system overhead and call admission rate. How to design a code management scheme that has low blocking rate and low overhead at the same time is an interesting research topic.

## References


1. F. Adachi, M. Sawahashi, and H. Suda, "Wideband CDMA for next-generation mobile communications systems," *IEEE Commun. Mag.*, vol. 36, pp. 56-69, 1998.
2. F. Adachi, M. Sawahashi, and K. Okawa, "Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA mobile radio," *Electron. Lett.*, vol. 33, pp. 27-28, 1997.
3. A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 34, pp. 441–449, 1995.
4. J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, 1976.
5. Wen-Tsuen Chen, Hung-Chang Hsiao, and Ya-Ping Wu, "A novel code assignment scheme for W-CDMA systems," in *Proceedings of IEEE Vehicular Technology Conference*, 2001, vol. 2, pp. 1182-1186.
6. Ray-Guang Cheng and Phone Lin, "OVSF Code Channel Assignment for IMT-2000," in *Proceedings of IEEE Vehicular Technology Conference*, 2000, vol. 3, pp. 2188-2192.
7. I. Chih-Lin *et al.*, "IS-95 enchancements for multimedia services, " *Bell Labs. Tech. J.*, pp. 60-87, 1996.
8. E. Dahlman, B. Gudmundson, M. Nilsson, and J. Skold, "UMTS/IMT-2000 based on wideband CDMA," IEEE Commun. Mag., vol. 36, pp. 70-80, 1998.
9. K. Fall and K. Varadhan, editors. *ns notes and documentation*, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, November 1999. Available from http://www-mash.cs.berkeley.edu/ns/.
10. S. Fiorini and R. J. Wilson, *Edge-Colourings of Graphs*, 1977.
11. J. J. Garcia-Luna-Aceves and J. Raju, "Distributed assignment of codes for multihop packet-radio networks," in *Proceedings MILCOM*, 1997, vol. 1, pp. 450–454.
12. Limin Hu, *Distributed Algorithms for Multihop Packet Radio Networks*, Ph.D. dissertation, University of California Berkeley, 1990.
13. Limin Hu, "A novel topology control for multihop packet radio networks," in *Proceedings INFOCOM*, 1989, pp. 1084-1093.
14. Limin Hu, "Distributed code assignment algorithms for cdma packet radio networks," in *Proc. INFOCOM*, 1991, pp. 1500-1509.
15. Limin Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 668–677, Dec. 1993.
16. C. L. Liu, Elements of Discrete Mathematics, second edition, McGraw-Hill, 1998.
17. Thit Minn and Kai-Yeung Siu, " Dynamic assignment of orthogonal variable-spreading-factor codes in W-CDMA," *IEEE Journal on Selected Areas in Communications*, vol. 188, no. 8, pp. 1429 –1440, 2000.
18. Timucin Ozugur, Mahmoud Naghsineh, Parviz Kermani, C.Michael Olsen, Babak Rezvani, and John A, "Balanced media access methods for wireless networks," *ACM/IEEE Int'l. Conf. On Mobile Computing and Networking*, October 1998.
19. E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks, " *IEEE Personal Communication*, pp. 46-55, 1999.
20. Yu-Chee Tseng and Chih-Min Chao, "Code placement and replacement strategies for wideband CDMA OVSF code tree management," in *Proceedings The 8th Mobile Computing Workshop*, pp.188-195, 2002.