# A proposal of a minimal-state processing search algorithm for isochronous channel reuse problems in DQDB networks

Nobuo Funabiki
Okayama University

Ayako Sugano
Okayama University

Teruo Higashino
Osaka University, Osaka

# A Proposal of a Minimal-State Processing Search Algorithm for Isochronous Channel Reuse Problems in DQDB Networks

Nobuo Funabiki[1] Ayako Sugano[1] Teruo Higashino[2]

[1]Department of Communication Network Engineering, Okayama University
funabiki@cne.okayama-u.ac.jp, ayako@sec.cne.okayama-u.ac.jp
[2]Graduate School of Information Science and Technology, Osaka University
higashino@ist.osaka-u.ac.jp

## Abstract

The IEEE 802.6 MAC standard protocol defines the distributed-queue dual bus (DQDB) for metropolitan area networks (MANs). The isochronous channel reuse problem (ICRP) has been studied for the efficient use of DQDB. Given a set of established connections and a set of connection requests, the goal of ICRP is to maximize the number of satisfied requests by finding a proper channel assignment, such that no established connection is not only reassigned a channel, but also any pair of active connections does not interfere each other. In this paper, we propose a minimal-state processing search algorithm for ICRP (*MIPS_ICRP*). The simulation results show that MIPS_ICRP always provides near-optimum solutions.

## 1. Introduction

The IEEE 802.6 MAC standard protocol is designed for the slotted, dual-bus configuration in the distributed-queue dual bus (DQDB) network (Fig. 1) for MAN. The slots in DQDB are classified into two categories, pre-arbitrated (PA) slots for isochronous traffics and queued-arbitrated (QA) slots for asynchronous ones. Each PA slot retains 48 octets for payload, where one octet provides a 64-kbps isochronous channel. Each station can occupy one channel or multiple channels, if they are free. A *connection* represents packet transmissions from a source station to a destination through a bus by occupying the necessary number of channels on the intermediate sections between two stations. The number of channels for each connection is determined by the bandwidth required for the traffic. A channel may be reused by two or more connections, if they do not intersect each other. If a connection has an overlapped section on a bus with another connection, they intersect each other. Otherwise, they do not intersect. The proper channel assignment to connections is essential to improve the transmission efficiency in the DQDB network by reusing as many channels as possible. As a result, the *isochronous channel reuse problem (ICRP)* has been extensively studied. ICRP has been proved to be $NP$-hard in general [3].

In this paper, we present a *minimal-state processing search algorithm for ICRP (MIPS_ICRP)* through the modification of the algorithm *MIPS_CLR* for the graph coloring problem in [7]. MIPS_ICRP is composed of three stages to produce high-quality solutions in short computation time. The first stage gives the lower bound on the number of unsatisfied connection requests. The second stage greedily generates an initial minimal state. The last stage iteratively improves the state by evolving minimal states with help of the state shuffle scheme for global convergence. The performance is evaluated through simulations in static instances to find a channel assignment for one time configuration.

## 2. Problem Formulation of ICRP

ICRP in this paper basically follows the problem formulation defined by Kassotakis et al. in [5]. Let $X = \{x_1, x_2, ..., x_{|X|}\}$ be a set of currently established $|X|$ isochronous connections in the DQDB network, where $x_i = (s_i^x, d_i^x, c_i^x)$ represents the $i$th established one between source station $s_i^x$ and destination $d_i^x$ using channel $c_i^x$. In this paper, we focus on a simple case of ICRP where every connection has the same band width equal to a single channel (64 kbps) as in [5]. Let $Y = \{y_1, y_2, ..., y_{|Y|}\}$ be a set of $|Y|$ connection requests, where $y_j = (s_j^y, d_j^y)$ represents the $j$th request between stations $s_j^y$ and $d_j^y$.

A channel in the DQDB network can be reused by two or more connections, if they do not intersect each other. Here, we only deal with the intersections between upstream connections using bus $A$ without loss of generality. Then, if two upstream connections $z_i = \{s_i, d_i\}$ and $z_j = \{s_j, d_j\}$ satisfy the relationship $s_i = s_j$ or $s_i < s_j < d_i$ or $s_j < s_i < d_j$, they intersect each other, because they must pass through the same bus section.

Proceedings of the17 th International Conference on Advanced Information Networking and Applications (AINA'03)
0-7695-1906-7/03 $17.00 © 2003 IEEE

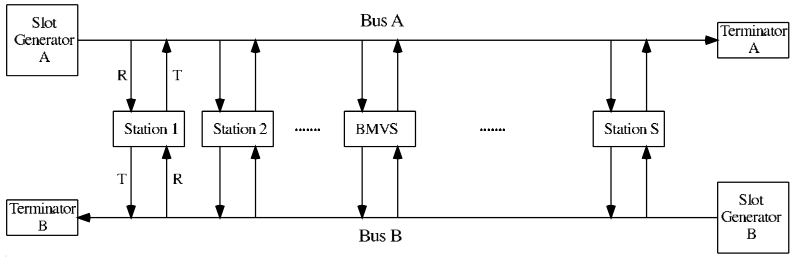


Figure 1: A dual-bus network topology.

Note connection $z$ may be an established one or a request. To realize feasible operations, any activated connection must not cause the intersection with others. Besides, any established connection in $X$ must not be reassigned channels. The goal of ICRP is to maximize the number of satisfied connection requests from $Y$ by assigning proper channels. The intersection between two connections is described by the $N \times N$ ($N = |X| + |Y|$) compatibility matrix $M$. The $ij$th element $m_{ij}$ in $M$ represents whether two connections $i$ and $j$ intersect each other or not. $m_{ij}$ for $i = 1, ..., N$ and $j = 1, ..., N$ is given by:

$$m_{ij} = \begin{cases} 1, & \text{if } s_i = s_j (i \neq j) \\ & \text{or } s_i < s_j < d_i \text{ or } s_j < s_i < d_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Then, ICRP is mathematically defined as follows:

**Input** A set of established connections $X = \{x_1, ..., x_{|X|}\}$ ($x_i = (s_i^x, d_i^x, c_i^x)$), a set of connection requests $Y = \{y_1, ..., y_{|Y|}\}$ ($y_i = (s_i^y, d_i^y)$), the number of channels $C$, and the number of stations $S$.

**Output** A set of channel assignments $(c_1, ..., c_N)$ to $X$ and $Y$.

**Constraint** $c_i = c^x_{i-|Y|}$ for $i = |Y| + 1, ..., N$, and $c_i \neq c_j$ for $\forall i, j$ with $1 \leq c_i, c_j \leq C$ and $m_{ij} = 1$.

**Objective** Maximize $\sum_{i=1}^{|Y|} I(c_i)$, where function $I(x)$ returns 1 if $1 \leq x \leq C$ and 0 otherwise.

## 3. MIPS Approach for ICRP

MIPS_ICRP is composed of three stages to find high-quality solutions through visiting only *minimal state*, where the full set of established connections and a subset of connection requests are assigned channels without violations, and any additional assignment to remaining requests in the *unsatisfied request list* $L_{request}$ cause violations. A state transition is designed to minimize $E(s)$ by moving to a best neighbor minimal state:

$$\begin{aligned} E(s) &= A\sum_{i=1}^{|Y|}(1 - I(c_i)) + B\sum_{i=1}^{|Y|}\deg_i \cdot (1 - I(c_i)) \\ \deg_i &= \sum_{j=1, j\neq i}^{N} m_{ij} \end{aligned} \quad (2)$$

where $A$ and $B$ are constant coefficients to satisfy $A >> B$, and $deg_i$ represents the number of connections that cannot be assigned the same channel as connection $i$. The $A$-term encourages more requests to be assigned channels. The $B$-term suggests that state $s_\alpha$ is better than state $s_\beta$, if the unsatisfied requests in $s_\alpha$ has less number of conflicting connections than state $s_\beta$, although $s_\alpha$ and $s_\beta$ have the same number of satisfied requests. The channel assignment to less conflicting requests is usually easier than the assignment to more conflicting requests. If $E(s) = 0$ is achieved, every connection request is satisfied in state $s$.

The first stage of MIPS_ICRP computes the lower bound on the number of unsatisfied connection requests $LB$ to terminate the iterative computation. Let *section* $j$ represent the bus interval between stations $j$ and $j+1$. Let $pass_j$ represent the total number of connections both established or requested passing through section $j$. Because every connection passing through section $j$ must be assigned a different channel to avoid the intersection, ($pass_j - C$) requests cannot be assigned channels if it is positive. If $(pass_j - C) > 0$, we call section $j$ an *over-section*. Then, the lower bound on the number of unsatisfied requests is equal to the minimum number of requests that must be removed from the instance so that no over-section exists. The first stage counts this number by removing a request one by one, that passes through the maximum number of over-sections until every over-section disappears. If two or more requests become candidates simultaneously, it selects one that has the maximum *over-area*, to minimize the difference of

$pass_j$ between sections so that it can select a request passing through a larger number of over-sections in later selections. The over-area for a request is given by the sum of $(pass_j - C)$ in the over-sections that the request passes through. Then, the second stage generates an initial minimal state by a simple greedy algorithm in [6].

The third stage evolutes minimal states by repeatedly generating *best neighbors* in terms of the cost function $E(s)$, starting from the initial state in the second stage. A next state generation is initiated by randomly selecting one unsatisfied request from $L_{request}$ to avoid biased movements. Then, a channel that minimizes the increase of $E(s)$, is selected for its assignment. Since $A$ in $E(s)$ is much larger than $B$, the channel conflicting with the least number of existing assignments becomes the candidate. The tiebreak is resolved randomly. Here, two auxiliary conditions are imposed. One is the prohibition of selecting a channel in a *tabu list* to avoid cyclic state transitions. The tabu list describes the channels that have been selected within the predefined number of iteration steps $T_{tabu}$ since its last selection to the corresponding request. Another is the prohibition of selecting any channel conflicting with established connections in $X$, to avoid removing them from the network. If any channel assignment to every request in $L_{request}$ conflicts with established connections, the third stage is terminated. To provide a hill-climbing capability, a channel satisfying the two auxiliary conditions is randomly selected regardless of the increase of $E(s)$, which is called the *random selection*. The random selection is actually applied when $E(s)$ has not been improved during the constant number of iterations $T_{random}$. Then, to retrieve a minimal state, any conflicting request is sought a new feasible channel assignment. If no feasible channel is found, such a request is inserted into $L_{request}$. After that, each request in $L_{request}$ may be assigned the selected channel if it is feasible.

The third stage induces the *state shuffle scheme* for big state changes while maintaining the best-achieved solution quality, if $T_{shuffle}$ trials of the random selection do not provide enough state fluctuations to escape from local minimum. Firstly, the best-achieved state in terms of $E(s)$ is retrieved as the current state to be evolved into different directions. The best state that has been visited is always memorized. Then, one seed channel (let channel $a$) that has not been tried at this trial is randomly selected, and its pairing channel (let channel $b$) is randomly selected such that a subset of connections with channel $a$ and a subset of connections with channel $b$ can be swapped in their channels. If there is no channel to satisfy this condition, the swap operation for channel $a$ is skipped. Here, these swapped connections must be requests in $Y$. This swap operation is terminated when no channel can be selected as a seed channel. After the scheme is done, each request in $L_{request}$ is sought a channel assignment, to retrieve a minimal state, and the tabu list is cleared.

## 4. Performance Evaluation

In our simulations, static ICRP instances in [5] are solved by MIPS_ICRP. Besides, larger instances are simulated to evaluate the scalability of MIPS_ICRP. In a static instance in [5], the number of stations $S$ and the number of isochronous channels $C$ are 10 and 50 respectively, the number of established connections $|X|$ is 50 where each one is assigned a different channel, the number of connection requests $|Y|$ is varied from 90 to 150, the source station $s_i$ and the destination one $d_i$ for connection $i$ are uniformly randomized between 1 and $S$, to satisfy $s_i < d_i$. In a larger static instance, each number for $S$, $C$, $|X|$, and $|Y|$ is tripled.

To show the modification of the constraint such that the channel reassignment of any established connection is allowed ($1 \leq c_i \leq C$ for $i = |Y|+1, ..., N$), can reduce the computation time and improve the solution quality, we have modified MIPS_ICRP such that the state shuffle scheme may reassign channels to established connections, and simulated it for static and dynamic instances. This ICRP formulation is called "General ICRP" in this paper.

For each size, a total of 100 runs are repeated with different random numbers in each size, and their average results are evaluated to obtain statistically rational results. Table 1 shows the instance size index, the number of established connections $|X|$, the number of requests $|Y|$, the upper bound $UB$ on the number of satisfied requests obtained by $UB = |Y| - LB$, the number of satisfied requests in solutions, the computation time (seconds) on Pentium-IV ($1.8GHz$) by MIPS_ICRP. Besides, the results for General ICRP are also summarized. For reference, the best number of satisfied requests and the computation time (seconds) on a 630 MIPS machine by the hybrid genetic algorithm (HGA) in [5] are shown there, where "-" indicates no corresponding result there. Note that exact comparisons are impossible between our results and their results because simulated instances are independently generated in each paper.

Table 1 indicates that MIPS_ICRP always finds near-optimum solutions that are close to the upper bound on the number of satisfied requests. The comparison between MIPS_ICRP and the existing HGA for smaller instances suggests that MIPS_ICRP provides better solutions with shorter computation time. For General ICRP, MIPS_ICRP finds the optimum solution in any instance except for size index 14 with the shorter time. The flex-

Table 1: Simulation results for static instances.

| Index | $\|X\|$ | $\|Y\|$ | MIPS_ICRP | | | General ICRP | | HGA in [5] | |
|---|---|---|---|---|---|---|---|---|---|
| | | | $UB$ | solution | time (s) | solution | time (s) | solution | time (s) |
| 1 | 50 | 90 | 63.66 | 63.65 | 0.083 | 63.66 | 0.005 | 63 | - |
| 2 | | 100 | 68.19 | 68.17 | 0.154 | 68.19 | 0.007 | 68 | - |
| 3 | | 110 | 72.95 | 72.92 | 0.521 | 72.95 | 0.018 | 72 | 2.030 |
| 4 | | 120 | 77.72 | 77.69 | 0.351 | 77.72 | 0.032 | 75 | 2.200 |
| 5 | | 130 | 82.61 | 82.60 | 0.317 | 82.61 | 0.042 | 81 | 2.310 |
| 6 | | 140 | 86.15 | 86.14 | 0.421 | 86.15 | 0.058 | 86 | 2.800 |
| 7 | | 150 | 90.70 | 90.66 | 1.020 | 90.70 | 0.079 | 89 | 2.778 |
| 8 | 150 | 270 | 209.36 | 209.31 | 28.824 | 209.36 | 3.523 | - | - |
| 9 | | 300 | 224.19 | 224.13 | 34.067 | 224.19 | 4.383 | - | - |
| 10 | | 330 | 240.27 | 240.25 | 34.058 | 240.27 | 4.011 | - | - |
| 11 | | 360 | 254.92 | 254.92 | 28.233 | 254.92 | 10.376 | - | - |
| 12 | | 390 | 269.01 | 268.98 | 85.186 | 269.01 | 14.672 | - | - |
| 13 | | 420 | 282.85 | 282.80 | 111.809 | 282.85 | 12.589 | - | - |
| 14 | | 450 | 296.63 | 296.61 | 140.273 | 296.61 | 36.360 | - | - |

ibility of the channel reassignment to established connections contributes the significant speed up the convergence by MIPS_ICRP. The results confirm the high capability of MIPS_ICRP for finding one time configuration of channel assignments.

### 5. Conclusion

This paper has presented MIPS_ICRP, a three-stage heuristic algorithm for the NP-hard isochronous channel reuse problem in DQDB networks. The performance is evaluated through solving static instances, where comparisons in the solution quality to bounds and the existing algorithm confirm the extensive search capability and the efficiency. Some improvement in the computation time and the solution quality is noticed when the channel reassignment of any established connection is allowed. In further studies, MIPS_ICRP should be extended to more practical situations where only limited stations may function as erasure nodes, and connections may have various bandwidth.

## References


[1] R. M. Newman and J. L. Hullett, "Distributed queuing: a fast efficient packet access protocol for QPSX," Proc. ICCC, pp. 294-299, 1986.

[2] R. M. Newman, Z. L. Budrikis, and J. L. Hullett, "The QPSX man," IEEE Commun. Magazine, vol. 26, no. 4, pp. 20-28, 1988. IEEE/ACM Trans. Networking, vol. 2, no. 1, pp. 89-100, Feb. 1994. IEEE/ACM Trans. Networking, vol. 5, no. 3, pp. 410-419, June 1997.

[3] N. F. Huang and H. I. Liu, "A study of isochronous channel reuse in DQDB metropolitan area networks," Proc. INFOCOM, pp. 1318-1325, 1997. metropolitan area networks," National Tsing Hua Univ., Taiwan, Tech. Rep. NTHU/CS, Feb. 1996.

[4] N. F. Huang and H. I. Liu, "A study of isochronous channel reuse in DQDB metropolitan area networks," IEEE/ACM Trans. Networking, vol. 6, no. 4, pp. 475-484, Aug. 1998. networks," Proc. Int. Conf. Network Protocols, pp. 244-251, 1996.

[5] I. E. Kassotakis, M. E. Markaki, and A. V. Vasilakos, "A hybrid genetic approach for channel reuse in multiple access telecommunication networks," IEEE J. Selet. Areas Commun., vol. 18, no. 2, pp. 234-243, Feb. 2000. 37, pp. 21-25, Jan. 1991. MAN," Proc. INFOCOM, pp. 636-643, 1990.

[6] D. S. Johnson, "Approximation algorithms for combinatorial problems," J. Comput. Syst. Sci., vol. 9, pp. 256-278, 1974. for the maximum clique problem," Proc. Int'l Symp. Nonlinear Theory Its Applications, pp. 163-166, Nov. 1999.

[7] N. Funabiki and T. Higashino, "A minimal-state processing search algorithm for graph coloring problems," IEICE Trans. Fundamentals, vol. E83-A, no. 7, pp. 1420-1430, July 2000.