ROUTING ALGORITHM FOR MULTIRATE CIRCUIT SWITCHING IN QUANTIZED CLOS NETWORK

BY

WAI-HUNG KWOK

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF INFORMATION ENGINEERING

THE CHINESE UNIVERSITY OF HONG HONG

JUNE 1997

Acknowledgement

I would like to express my sincere gratitude towards my supervisor, Prof. Tony T. Lee for his advices, supports and encouragements, which make this 2-year research period a fruitful and rewarding experience.

I would also like to thank Irene To and my family for their kindly supports and understanding.

Thanks Cheuk-hung Lam and Soung-Yue Liew for their valuable discussions and comments.

Abstract

A route assignment algorithm for circuit switching in quantized Clos network under multirate traffic is proposed in this thesis. The advantage of using multirate circuit switching in broadband digital network is that it guarantees the Quality of Services (QOS) of each connection as sufficient bandwidth is reserved. However, previous works show that without connection splitting, the Clos network will be non-blocking only if external link utilization β is restricted or otherwise, the number of central module will increase. In addition, routing algorithm has never been proposed because of the unmanageable continuous scale of bandwidth. To tackle this problem, we have transformed this continuous scale to a finite, discrete scale of bit rate levels by bandwidth quantization. With connection splitting and bandwidth quantization, we have derived in this thesis the non-blocking conditions for the Clos network without imposing restriction on β. In this thesis, we have also proposed a splitting and routing algorithm for rearrangeably non-blocking quantized Clos network, based on edge-coloring of a weighted bipartite multigraph. The algorithm is a generalization of the one in the classical circuit switching.

Keywords — Clos network, Circuit switching, Multirate, Bandwidth quantization, Connection splitting, Non-blocking, Routing algorithm.

Contents

List of Figures

Chapter 1

Introduction

The past few decades have experienced the merging of computer and communication technologies due to the various demands of communication services. Wide-area and local-area computer networks have been deployed to interconnect computers distributed throughout the world. In addition to transmission resources, a communication network consists of switching resources. Imagine that there are N terminals in a network without switching facilities, C_2^N unique, independent transmission lines will be required to connect every pair of terminals. When N is large, there will be a huge number of transmission lines. On the other hand, switching makes it possible to change the terminal connectivity dynamically and increase the utilization of the shared resources. As a result, a switching facility and N transmission lines , as shown in Figure 1.1, are sufficient to fulfill any connection request.

Classically, communication network design mainly concerned about circuit switching in the telephone network, or what in this thesis called the *classical circuit switching.* A dedicated circuit will be established between two users

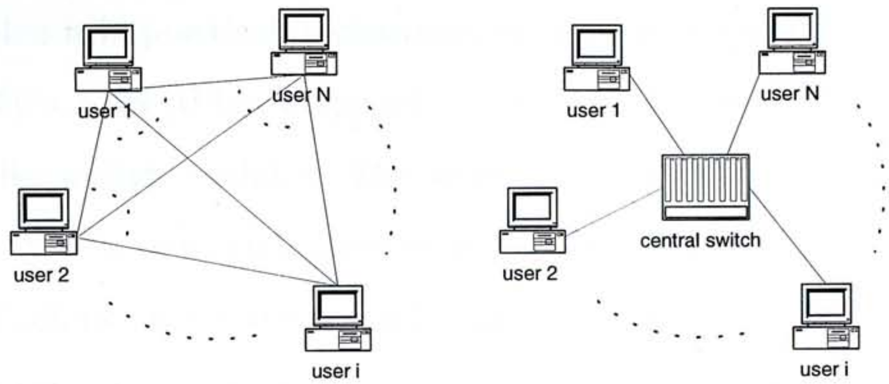


Figure 1.1: Communication networks without and with switching facility

when requested. The transmission rate of each circuit is fixed to the same basic rate. Once established, the circuit will not be shared with other users. If sufficient network resources, either inside and outside the switching facility, are not available to setup a circuit, the corresponding request will be *blocked.* One of the objectives in switch design is to establish non-blocking connections with minimum switch complexity.

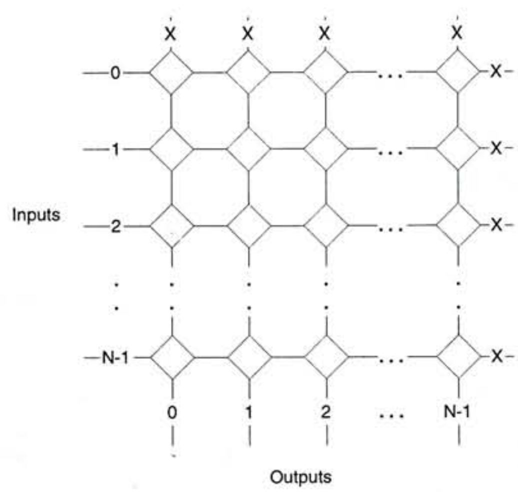


Figure 1.2: $N \times N$ cross-bar switch

Figure 1.2 shows a $N \times N$ internally non-blocking switch called *cross-bar* switch. In this switch structure, a path can always be found to connect an idle input to an idle output. However, the requirement of N^2 2-by-2 switching elements makes it impractical to construct switches with large N.

In early 50's, Clos [1] has proposed to construct a large switch by interconnecting smaller switch modules. The smaller switch modules are arranged in stages and a $N \times N$ symmetric three stage Clos network $C(m, n, p)$ is shown in Figure 1.3. Each module in the network is interconnected with every modules in the adjacent stage via a unique link. There are $p = N/n$ switching modules of sizes $n \times m$ and $m \times n$ in the first and third stage (or, input and output stage) respectively. The m second stage (or, central stage) switching modules are of size $p \times p$.

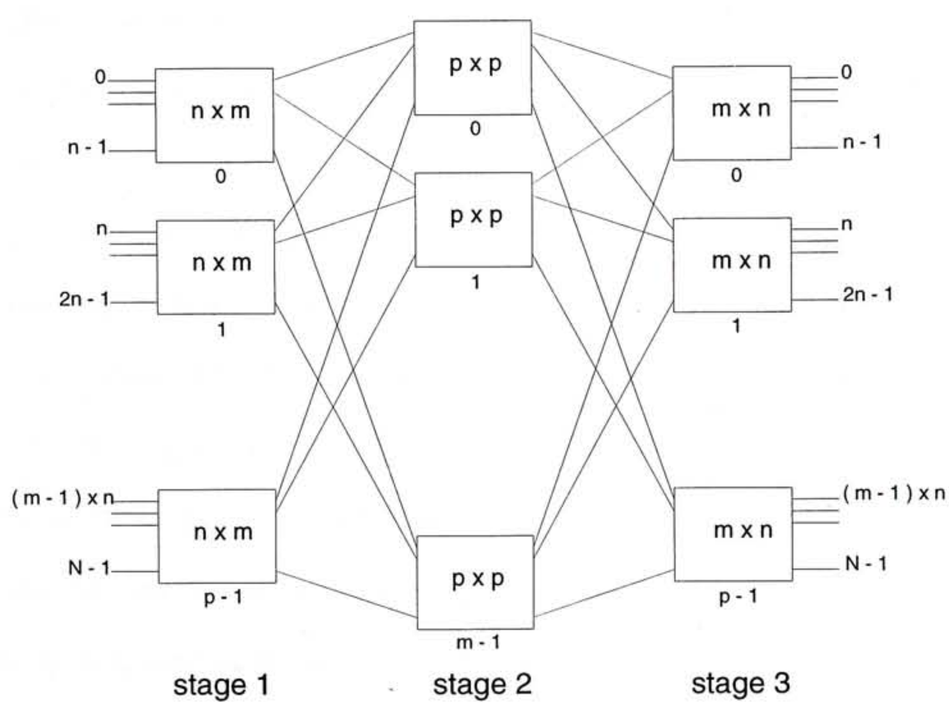


Figure 1.3: $N \times N$ synnetric three-stage Clos network $C(m, n, p)$

Assuming the switching modules are internally non-blocking, routing in Clos network is to choose one of the central module for each connection such that no two connections will share the same internal link. An important parameter of the network is the number of central modules, m. Obviously, the larger the value of m, the more alternative paths will be available to establish a connection. However, network complexity increases with m.

If there are sufficient central modules such that a connection can always be setup between any idle input and output without rearranging the paths of existing connections, the network is said to be *strictly non-blocking*. It has been shown in [1] that the Clos network will be strictly non-blocking if it satisfies the condition $m \geq 2n - 1$.

On the other hand, Benes [2] has proved that if the condition $m \geq n$ is satisfied, the Clos network will be *rearrangeably non-blocking* in which a connection can always be setup, although it may be necessary to rearrange existing connections.

It should be noted that rearranging existing connections is not a trivial task and requires a centralized rearrangement algorithm. A classical implementation of the routing algorithm in Clos network is by Paull's connection matrix [3, 4], based on the edge-coloring problem of a bipartite multigraph. We will briefly present the algorithm in the next chapter.

With the advancement in technology, new services with widely varying characteristics in transmission rates are available. For example, from low speed, bursty file transfer to high speed, non-bursty multimedia services like constant bit-rate video transmission. In contrast to the telephone network, the modern digital network systems are therefore designed to support connections with such a wide range of bandwidth requests.

In this *multirate* environment, each connection can consume an arbitrary fraction of the bandwidth of the link carrying it. In order to improve the efficiency of the shared transmission and switching resources, a number of connections will share one transmission link given that the total data rate of the connections does not exceed the link's capacity. Typically, information is divided into blocks called *packets*. Packets from different connections are carried in multiplexed format for resources sharing.

For the Clos network under such environment, any new connection from input A to output B with bandwidth request ω is defined to be *blocked externally* with respect to the Clos network and will be rejected if

$$\sum_j \lambda_{Aj} + \omega > I_A \quad \text{or} \quad \sum_i \lambda_{iB} + \omega > O_B$$

where λ_{ij} is the total aggregate data rate from input i to output j. I_A and O_B are the available capacities of input A and output B respectively. Once a connection is accepted, the Quality of Service (QOS) can be guaranteed as sufficient bandwidth is reserved.

Many researches studying this multirate network environment have been performed. For example, Melen and Turner in [5], Ross and Chung in [6], Hwang and et. al. in [7], etc. They have derived various conditions for the Clos network to be non-blocking under different traffic conditions and assumptions. Melen and Turner have shown in [5] that the Clos network will be strictly non-blocking if $m \geq 2n - 1$, with the restriction that external link utilization β must be less than or equal to 0.5. External link utilization in Clos network is defined as the fraction of bandwidth consumed in an input or output link. It can also be calculated from Theorem 4.3 of [5] that a Clos network with $N = n^2$ and $\beta \leq 0.5$ will be rearrangeably non-blocking if $m \geq n$ (by putting $N = n^2$ and $\beta/B = 1$

into Theorem 4.3 of [5]). The restriction on β comes from the assumption that connection splitting is *prohibited*. Without connection splitting, a single path inside the network will be used by all packets belonging to the same connection. This will lower the internal link utilization because there may be cases that fragmented bandwidth on internal links is insufficient to accommodate any new connection. As a result, β has to be restricted as shown above to avoid internal blocking or otherwise, the number of central modules will increase. However, limiting the value of β will sharply increase the external blocking probability.

On the other hand, if connection splitting is allowed, the residual bandwidth can be filled up by splitting connections appropriately. The restriction on β can then be released as the internal link utilization increases. However, packets belonging to the same connection may reach their destination in wrong sequence. Packet re-sequencing must be performed to restore the correct packet order.

Another important issue in this multirate environment is route assignment inside Clos network. Although upper bound of the non-blocking conditions have been derived, routing algorithm has never been proposed for rearrangeably non-blocking Clos network under multirate traffic condition. A major difficulty is the continuous scale of bandwidth request of the connections. For an internal link, the optimal arrangement of connections resembles a fractional knapsack problem [8] which is NP-complete. With more than one internal links and connection rearrangements between links, route assignment becomes even more intractable. Lea and Alyatama have also stated in [9] that continuous bandwidth requirement implies an infinite dimensional analytical model for the routing problem and makes routing unmanageable. They have suggested to overcome the problem using bandwidth quantization technique in which only a finite, discrete set of data rates will be supported by the network. They have also shown that the throughput degradation due to quantization is negligible [9].

Motivated by these arguments, we have studied the symmetric three-stage Clos network under multirate environment based on connection splitting and bandwidth quantization. In this thesis, we will derive the non-blocking conditions for the Clos network under these assumptions. In addition, we will propose a route assignment algorithm for multirate circuit switching in the rearrangeably non-blocking quantized Clos network.

For the quantized Clos network $C(m, n, p)$ under multirate traffic condition, we will show that the network will be strictly non-blocking and rearrangeably non-blocking for externally non-blocked connections if the conditions $m \geq \lceil \frac{2Mn-1}{M} \rceil$ and $m \geq n$ are satisfied respectively. M is the number of bandwidth quantization levels. Both conditions impose no restriction on external link utilization β. Consider the special case when $M = 1$, the situation reduces to the classical circuit switching case and the conditions match the well known results. For $M > 1$, only $2M$ central modules will be sufficient for the network to be strictly non-blocking.

The development of the proposed routing algorithm bases on edge-coloring of a *weighted bipartite multigraph* representing the connection configuration in the quantized Clos network. It is a generalization of the algorithm in classical circuit switching. To represent the connection configuration, each vertex in the weighted bipartite multigraph is equivalent to an input or output module of the Clos network. A connection is represented by a weighted edge with the weight equals to the quantized bandwidth request of that connection. According to the rearrangeably non-blocking condition derived, n colors, each representing one central module, will be sufficient to edge-color the graph.

There are three major steps to color an uncolored edge in the weighted bipartite multigraph. The first step tries to color it using a single color and corresponds to route the connection through a single central module. If it fails, the second step will split the edge into several new edges and colors them using distinct colors. It is equivalent to split the connection through several central modules. Otherwise, the final step will color the edge by rearranging existing color assignments, or equivalently, existing connections. In the proposed algorithm, connection rearrangements are performed in a recursive manner. A matrix called *capacity allocation matrix* has been specially designed to implement the routing algorithm based on the edge-color principle.

This thesis is organized as follow: Chapter 2 reviews the routing problem in classical circuit switching Clos network. In Chapter 3, the non-blocking conditions will be derived and the principle of the proposed routing algorithm will be studied. Chapter 4 presents the details of the proposed algorithm. Simulation results will be given in Chapter 5 and Chapter 6 concludes this thesis.

Chapter 2

Preliminaries - Routing in Classical Circuit Switching Clos Network

In classical circuit switching, the route assignment in Clos network is equivalent to the edge-coloring problem of a bipartite multigraph. This formulation was proposed by Lev, Pippenger and Valiant in [10]. In this chapter, we will describe the details of this equivalence relationship and how routing is implemented through Paull's connection matrix [3, 4].

2.1 Formulation of route assignment as bipartite multigraph coloring problem

2.1.1 Definitions

(*The materials given in this subsection are from* [11])

A *graph* G is defined to be a combination of a set of *vertices* $V(G)$ and a set of *edges* $E(G)$. An edge $(i, j) \in E(G)$ joins a pair of vertices $i \in V(G)$ and $j \in V(G)$.

Suppose that the set $V(G)$ can be split into two disjoint sets V_1 and V_2, in such a way that every edges in $E(G)$ join a vertex in V_1 to a vertex in V_2 (see Figure 2.1). G is then defined as a *bipartite graph*, denoted by $G(V_1, V_2)$. As shown in the figure, G can be redrawn in such a way that vertices in V_1 and V_2 are arranged in two separate columns.

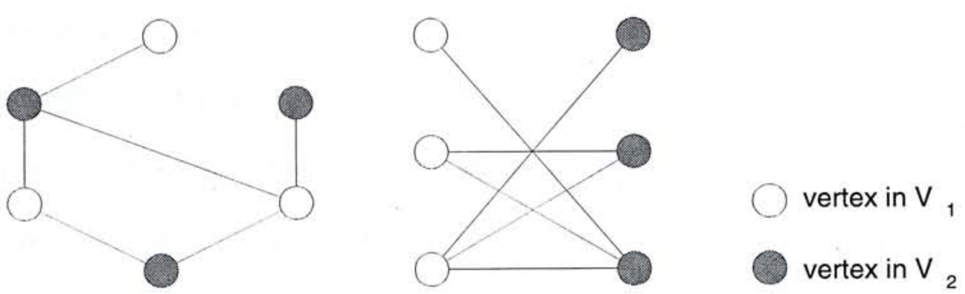


Figure 2.1: Example of bipartite graph

The *degree* of a vertex v of G is the number of edges incident to v. If the maximum vertex-degree of G is d, G is then said to be *with maximum degree d*.

A *subgraph H* of G is simply a graph with vertices $V(H) \subseteq V(G)$ and edges $E(H) \subseteq E(G)$.

A *walk* in G is a finite sequence of edges of the form

$$(v_0, v_1), (v_1, v_2), \ldots, (v_{m-1}, v_m)$$

If the vertices $v_0, v_1, \ldots, v_m$ in a walk are distinct (except possibly, $v_0 = v_m$), then the walk is called a *path*. If in addition $v_0 = v_m$, the path is closed and is called a *cycle*.

Theorem 1 *If $G(V_1, V_2)$ is a bipartite graph, then every cycle has even length.*

Proof : Let $v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_m \rightarrow v_1$ be a cycle and assume that $v_1 \in V_1$. Then since $G(V_1, V_2)$ is bipartite, $v_2 \in V_2$, $v_3 \in V_1$, and so on. It follows that v_m must be in V_2 and hence the cycle has even length. □

A graph is *connected* if, given any pair of vertices v, w of G, there is a path from v to w.

A *component* of G is the subgraph determined by a vertex v, together with the set of vertices which are connected by a path to v.

A graph G is *k-edge-colorable* if its edges can be colored with k distinct colors such that no two edges connecting to the same vertex have the same color. If G is k-edge-colorable but not $(k-1)$-edge-colorable, it is said to be *k-chromatic*.

A *multigraph* is simply a graph in which a pair of vertices may be connected through multiple edges. It retains all properties of a graph described above.

2.1.2 Problem formulation

With reference to Figure 2.2, given a set of connection requests in the three stage Clos network, the connection configuration can be represented using a bipartite multigraph as follow. The input and output modules can be considered as vertices in V_1 and V_2 of $G(V_1, V_2)$, respectively. Each central module can be regarded as one color. An edge connecting vertices $x \in V_1$ and $y \in V_2$ is equivalent to a connection from input module x to output module y. Since each input or output module can accommodate at most n connections, the maximum degree of the bipartite graph equals to n.

For the above connection (edge), one central module (color) must be assigned so that no other connections (edges) in modules (vertices) x or y will share the same central module (color). In the other words, we can obtain a route assignment in the Clos network from edge-coloring of the corresponding bipartite multigraph.

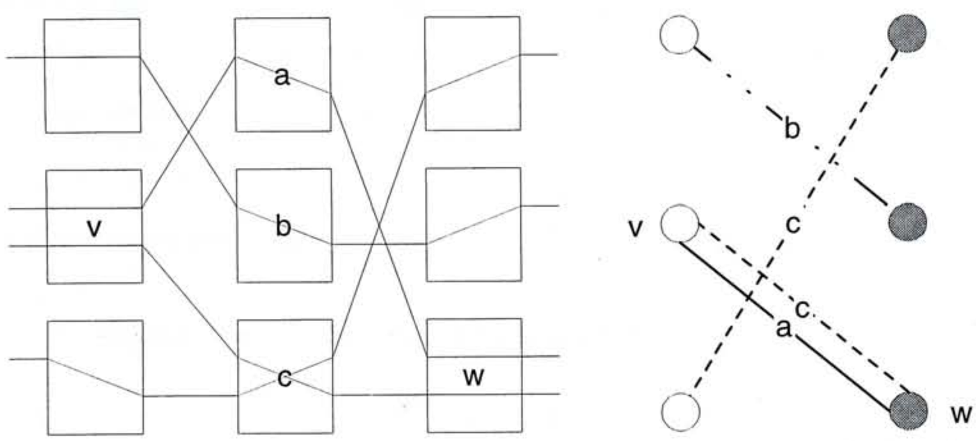


Figure 2.2: Bipartite multigraph representation of the connection configuration in a Clos network

2.2 Edge-coloring of bipartite graph

(*The materials given in this subsection are from* [11] , [12] *and* [13])

Before showing how a bipartite graph can be edge-colored, several theorems have to be introduced.

Define a two-colored subgraph H_{ab} of an edge-colored bipartite graph $G(V_1, V_2)$, consisting of all vertices in $V(G(V_1, V_2))$ and all edges in $E(G(V_1, V_2))$ with colors a or b. H_{ab} will then be a bipartite graph with maximum degree 2. See Figure 2.3(a) for an example. Each component in H_{ab} can be a cycle or a path as shown in Figure 2.3(b). From Theorem 1, the cycles must contain even number of vertices and edges.

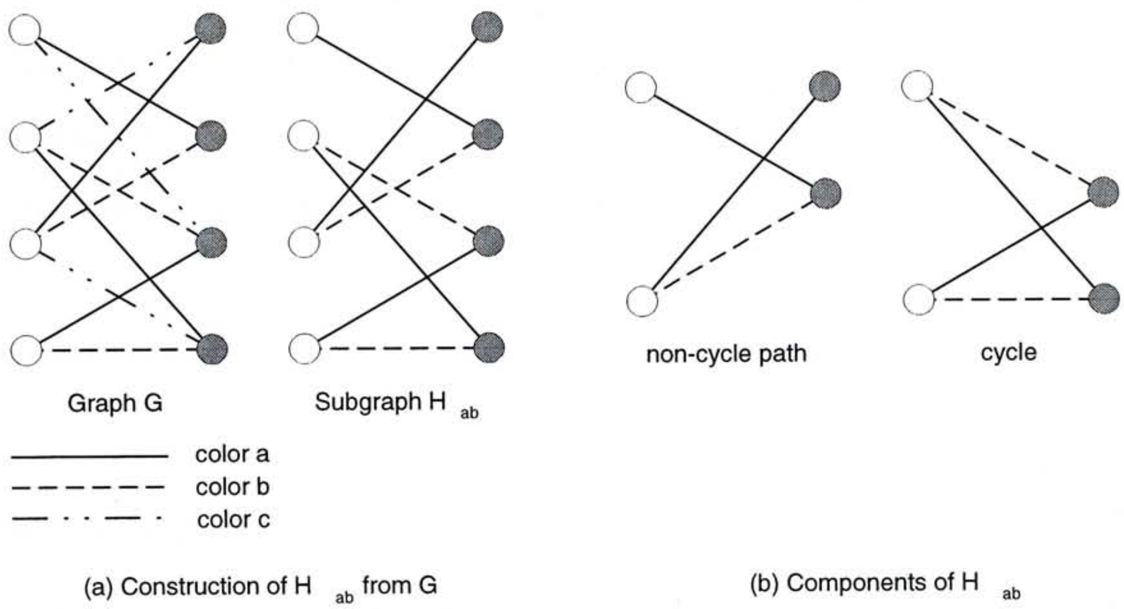


Figure 2.3: A two-colored bipartite subgraph H_{ab}

Theorem 2 *In any component of H_{ab}, each edge must has the same color with edges which are $2k$ vertices away from it, and must has distinct color with edges which are $2k+1$ vertices away from it for $k \geq 1$, if such edges exist.*

Proof: As shown in Figure 2.3(b), adjacent edges must have distinct colors. Therefore, an edge must has the same color with the one that is two vertices away. By transitive closure, the theorem is true. □

Theorem 3 *For any pair of vertices (x, y) of H_{ab} where $x \in V_1$ and $y \in V_2$, if they are incident to edges of different colors, they are disconnected. That is, x and y are on different path.*

Proof: Suppose the two vertices i, j from V_1 and V_2 are connected. Since adjacent vertices in this path must be from V_1 and V_2 respectively, there are even number of vertices between i and j. Hence, the edge incident to j is $2k$ vertices away from the one incident to i, for some $k \geq 0$. From Theorem 2, the two edges must have the same color. In the other words, if vertices i and j are incident to edges of different colors, they must be disconnected. □

Theorem 4 *A bipartite graph $G(V_1, V_2)$ with maximum degree d is d-chromatic.*

Proof: By induction on the number of edges in $G(V_1, V_2)$, it is sufficient to prove that if all but one edges of $G(V_1, V_2)$ have been colored with at most d colors, there exists a d-edge-coloring of $G(V_1, V_2)$.

Suppose that each edge of $G(V_1, V_2)$ has been colored, except edge (v, w). Then there is at least one color missing in both vertices v and w. If the same color is missing, the result follows by coloring (v, w) with this color. Otherwise, suppose colors a and b are missing in v and w respectively. A subgraph H_{ab} can then be obtained. A path starting from w can be found in H_{ab}. By theorem 3, interchanging colors a and b on this path will not affect v. The edge (v, w) can now be colored using a and complete the coloring of $G(V_1, V_2)$. □

Since an input or output module of the Clos network can accommodate at most n connections, the maximum degree of the corresponding bipartite multigraph will be n. From the above theorem, the graph will be n-chromatic. In the other words, the well-known results that "a symmetric three-stage Clos network will be rearrangeably non-blocking with at least n central modules" can also be proved using this theorem.

From the proof of Theorem 4, a bipartite graph with maximum degree d can be edge-colored using d different colors as follow.

Initially, we can color an edge arbitrarily.

Suppose now we want to color an uncolored edge (v, w), there are two possible cases.

1. If there exists a color that has not been used in both vertices v and w, edge-color (v, w) using this missing color.

2. Otherwise, suppose a and b are unused colors in v and w respectively, we can obtain an *alternate path* starting from w, constituting colors a and b. By interchanging a and b on the alternate path, (v, w) can then be colored using color a.

Since a bipartite multigraph is essentially a kind of bipartite graph, it can be edge-colored similarly.

2.3 Routing algorithm - Paull's matrix

A classical implementation of the routing algorithm is the connection matrix devised by Paull [3, 4].

Figure 2.4 shows how the network can be represented by the connection

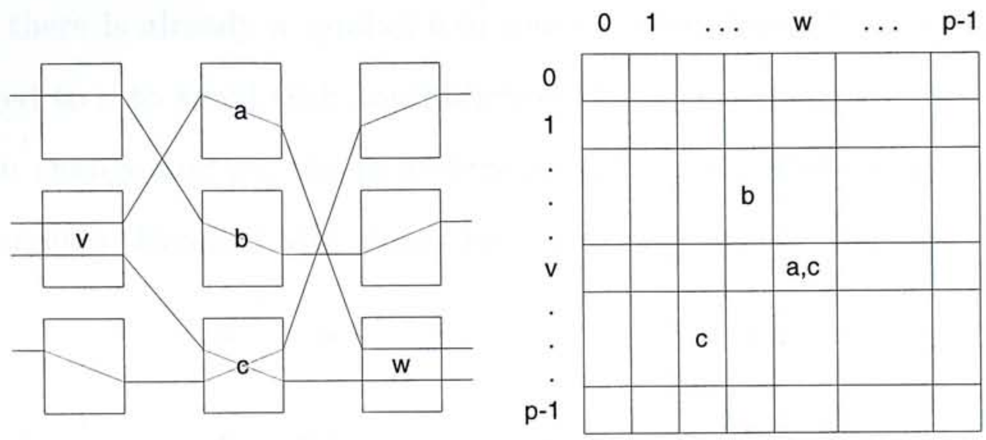


Figure 2.4: Paull's connection matrix of the three-stage Clos network

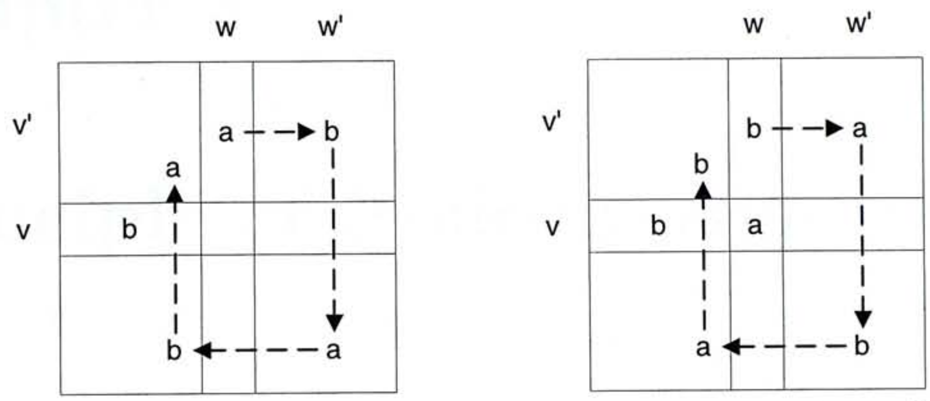
matrix. Row v corresponds to input module v and column w corresponds to output module w. Obviously, the size of the matrix is $p \times p$. Entry (v, w) is associated with the central modules. As shown, if there is connection joining modules v and w via module a, then entry $(v, w) = \{a\}$. Note that each entry may contain multiple symbols.

There are two major constraints in this matrix,

1. Symbols within each row or column must be distinct.

2. Each row (column) can have at most n symbols as each input (output) module contains n input (output) ports.

Suppose now we want to establish a new connection between input module v and output module w. If there exists any symbol which does not appear in both row v and column w, the new connection can be setup directly. However, if it is not the case, rearrangement in the matrix is needed. For example in Figure 2.5(a), all symbols except a occur in row v and all symbols except b occur in column w. If the symbol a in (v', w) is change to b, symbol a can be entered into entry (v, w). However, contradiction to constraint 1 will occur in row v' if there is already a symbol b in row v'. Here, symbol b in (v', w') must be changed to a to avoid such contradiction. Further contradictions may occur after each change and so, the procedure must be performed until constraint 1 can be satisfied. Figure 2.5(b) shows the result after rearrangement.



(a) Search path in Paull's matrix before rearrangement

(b) Connection setup after rearrangement

Figure 2.5: Rearrangement in Paull's connection matrix

Note that this algorithm has implemented the edge-coloring principle described in previous section. The rearrangement chain in the matrix is equivalent to the alternate path in the bipartite multigraph. According to Theorem 4, the rearrangement chain must exist.

In the next chapter, principle of a new routing algorithm in Clos network under multirate traffic condition will be given. The algorithm bases on edge-coloring of a weighted bipartite multigraph and is a generalization of the one in classical circuit switching described in this chapter.

Chapter 3

Principle of Routing Algorithm

With bandwidth quantization and connection splitting, the route assignment problem in the Clos network under multirate environment is equivalent to an edge-coloring problem of a weighted bipartite multigraph with edge-splitting. In this chapter, we will describe the details of this equivalence relationship, which is the basis of the construction of our algorithm.

3.1 Definitions

In this section, the two fundamental concepts, bandwidth quantization and connection splitting, will be introduced.

3.1.1 Bandwidth quantization

As mentioned in chapter 1, the continuous scale of bandwidth in a multirate environment, such as ATM network, can be converted to a finite, discrete set of bit rate levels by bandwidth quantization.

Without loss of generality, suppose the maximum available bandwidth of every links in the network are the same and are all normalized to one. This implies that bandwidth request ω of any connection must satisfy the constraint

$$0 < \omega \leq 1$$

Using *uniform quantization* with M quantization levels, the quantized value $\tilde{\omega}$ of ω is defined as

$$\tilde{\omega} = \frac{i}{M} \quad \text{if} \quad \frac{i-1}{M} < \omega \leq \frac{i}{M}$$

for $1 \leq i \leq M$. Sufficient bandwidth satisfying ω is guaranteed by the condition $\tilde{\omega} \geq \omega$.

By multiplying all link bandwidth by M, a *virtual* quantized Clos network model can be obtained. In the virtual model, the capacities of every input and output links will be equal to M and all connections will request for integer bandwidth $\tilde{\omega} = i$, where $i = 1, 2 \ldots M$. Each central module can supply M units of bandwidth between each pair of input and output modules.

Similar to the classical circuit switching, the connection configuration in the virtual quantized Clos network can be represented by an bipartite multigraph $G(V_1, V_2)$. Again, each vertex in V_1 (V_2) of the graph represents one input (output) module of the network. A connection from input module I to output module O with $\tilde{\omega} = i$, will be represented using i edges connecting vertices I and O. With each of the n ports in an input or output module providing M units of bandwidth, the maximum degree of the graph equals to Mn. The bipartite multigraph representation will be used to prove the non-blocking conditions of the network in section 3.2.

3.1.2 Connection splitting

Connection splitting allows traffic from the same connection to travel through different central modules inside the network. The internal link utilization can be increased since splitting new connections appropriately can fill up the "fragmented" bandwidth on the internal links. With the same number of central modules used in the Clos network, a direct consequence will be the improvement of the external link utilization β. Releasing the restriction on β helps directly in reducing the external blocking probability.

However, connection splitting has its own disadvantages. To store the central module assignments in the routing table, more memory will be needed for a split connection than a non-split one. Therefore, the number of split connections should be kept minimum to minimize the amount of memory required. This needs a specially designed control algorithm. The algorithm proposed in this thesis tries to *reduce* the number of split connections and simulation result in chapter 5 shows a low connection splitting probability.

Connection splitting may also cause the out-of-sequence problem. By providing buffers for each connection in the output ports, the problem can be solved by performing additional works to re-sequence the packets.

3.2 Non-blocking conditions

With bandwidth quantization and connection splitting, we have derived the non-blocking conditions for the Clos network under multirate traffic condition. These conditions will impose *NO* restriction on the external link utilization β.

3.2.1 Rearrangeably non-blocking condition

The following theorem states the rearrangeably non-blocking condition for the Clos network. In the theorem, $\lceil x \rceil$ is the smallest integer greater than or equal to x.

Theorem 5 (Rearranageably Nonblocking) *With bandwidth quantization and connection splitting, a symmetric three-stage $N \times N$ Clos network $C(m, n, p)$ will be rearrangeably nonblocking under multirate traffic if*

$$m \geq n$$

provided that

$$\sum_j \tilde{\lambda}_{ij} \leq M \quad \forall i \qquad \text{and} \qquad \sum_i \tilde{\lambda}_{ij} \leq M \quad \forall j$$

where $\tilde{\lambda}_{ij}$ is the total aggregate quantized traffic from input link i to output link j, and is equal to the sum of all $\tilde{\omega}$'s from i to j.

Proof : Given that the connections are externally non-blocked after bandwidth quantization, that is, the two conditions in the theorem are satisfied, central module assignment is equivalent to edge-coloring of the corresponding bipartite multigraph $G(V_1, V_2)$ in Section 3.1.1. The maximum degree of the graph equals to Mn and according to Theorem 4, Mn colors are sufficient to edge-color the graph. Since each central module can provide M units of bandwidth for each input or output modules, that is, M different colors, $\lceil Mn/M \rceil = n$ central modules will be sufficient for the network to become rearrangeably nonblocking. Note that m must be an integer. To complete the proof, connection splitting must be employed in routing since the $\tilde{\omega}$ edges of a connection may have their edge-colors provided by different central modules. □

With bandwidth quantization and connection splitting introduced, the internal links can be fully utilized and therefore, the above theorem has imposed no restriction on the external link utilization β. In addition, the condition is the same as the one in classical circuit switching for all values of M.

3.2.2 Strictly non-blocking condition

To derive the strictly non-blocking condition for the Clos network, the following theorem is needed.

Theorem 6 *To edge-color a bipartite graph $G(V_1, V_2)$ with maximum degree d without rearranging existing color assignments, a sufficient number of colors is $2d-1$. [14]*

Proof: Suppose that each edge of $G(V_1, V_2)$ has been colored, except (v, w). With maximum degree d, each of the vertices v and w has at most $d-1$ colored edges. In the worst case, colors on these edges are all different from the other and so, $2d-2$ colors would have been used. To edge-color (v, w), another distinct color is needed. The total number of color needed is therefore $2d-1$. □

Theorem 7 (Strictly Nonblocking) *With bandwidth quantization and connection splitting, a symmetric three-stage $N \times N$ Clos network $C(m, n, p)$ will be strictly nonblocking for multirate traffic if*

$$m \geq \lceil \frac{2Mn-1}{M} \rceil$$

provided that the conditions in theorem 5 are satisfied.

Proof: Given that the connections are externally non-blocked after bandwidth quantization, assigning central modules without rearranging existing connections is equivalent to edge-color $G(V_1, V_2)$ without rearranging any colors already assigned. With maximum degree Mn, $2Mn - 1$ colors will be sufficient here according to Theorem 6. Therefore, m must be greater than or equal to $\lceil \frac{2Mn-1}{M} \rceil$ as m must be an integer. Connection splitting is also required to complete the proof. □

Consider the special case when $M = 1$, that is, the traffic environment reduces to classical circuit switching case. The condition becomes $m \geq 2n - 1$ which matches the well known results. For $M > 1$, only $2M$ central modules will be sufficient. Again, no restriction has been imposed on β.

3.3 Formulation of route assignment as weighted bipartite multigraph coloring problem

From the proof of theorem 5, route assignment in the rearrangeably non-blocking Clos network can be performed by edge-coloring of the bipartite multigraph $G(V_1, V_2)$ using colors $0, 1, \ldots, Mn - 1$. The central module assignment, k, of the edge with color i can then be given by

$$k = i \bmod n$$

Although this method can solve the route assignment problem, it has the drawback that the number of split connections is difficult to control. Trying to assign colors such that the $\tilde{\omega}$ edges of the same connection can obtain the same k after the above modulus operation is unmanageable, especially when M is large.

Instead of using the bipartite multigraph, the route assignment problem can be formulated as an edge-coloring problem of a *weighted bipartite multigraph* $WG(V_1, V_2)$ which also represents the connection configuration in the virtual network model. To obtain $WG(V_1, V_2)$ from $G(V_1, V_2)$, simply replace the corresponding $\tilde{\omega}$ edges of a connection by a single edge with weight $\tilde{\omega}$.

Define E_x be the set of edges connected to vertex x, $E_{x,i}$ be the set of edges connected to x using color i and ω_k be the weight on an edge k . The following constraint must be satisfied in each vertex x of $WG(V_1, V_2)$ since the summation in the left hand side is equivalent to the degree of x in $G(V_1, V_2)$.

$$\sum_{k \in E_x} \omega_k \leq Mn \tag{3.1}$$

From theorem 5, n central modules are sufficient for the network to be rearrangeably nonblocking. Combining with constraint 3.1, the route assignment problem can then be formulated as edge-coloring of the weighted bipartite multigraph $WG(V_1, V_2)$ using n colors, each represents one central module, such that

$$\sum_{k \in E_{x,i}} \omega_k \leq M \quad \forall \quad 0 \leq i \leq n-1 \tag{3.2}$$

Constraint 3.2 must be satisfied as each central module can provide at most M units of bandwidth for each input or output modules. Obviously, it requires edge-splitting during the edge-coloring process in order to satisfy constraint 3.2 for all time.

3.4 Edge-coloring of weighted bipartite multigraph with edge splitting

Before describing how to edge-color the weighted bipartite multigraph, some quantities have to be defined. The amount of color i available at vertex z is defined as

$$R_{z,i} = M - \sum_{k \in E_{z,i}} \omega_k$$

which is equal to the total available amount M minus those being used. Let u be an uncolored edge with weight ω_u, connecting vertices $x \in V_1$ and $y \in V_2$. The amount of color i available to edge u is then equal to

$$S_{u,i} = \min(R_{x,i}, R_{y,i})$$

3.4.1 Procedures

In this section, we presents the edge-coloring procedures of the weighted bipartite multigraph. A full example will be given in Section 3.4.2. The coloring procedures of an uncolored edge u involve three major steps.

1. If there exists a color i such that $S_{u,i} \geq \omega_u$, color u using i. Otherwise, goto step 2.

2. For every color $0 \leq i \leq n-1$ with $S_{u,i} > 0$, split an edge u_i from u with $\omega_{u_i} = S_{u,i}$, and color u_i using i until $\omega_u = 0$ or all colors are examined.

3. If all colors are examined and still, $\omega_u > 0$, these remaining ω_u must be colored by rearranging existing color assignments.

Step 1 in the above procedures is equivalent to setup the new connection through a single central module without splitting. Step 2 splits the connection and step 3 setups the remaining unsatisfied bandwidth requests by rearranging existing connections.

The basic idea of rearranging existing color assignment in $WG(V_1, V_2)$ is a direct extension of the alternate path concept in classical circuit switching. Instead of an alternate path, a *weighted alternate tree* will be obtained from the graph. After interchanging the two colors in the weighted alternate tree, sufficient color will be available for u.

Due to the operations in step 2, $S_{u,i}$ will equal to zero for all colors i or equivalently, either $R_{x,i}$ or $R_{y,i}$ will equal to zero. Without loss of generality, suppose $R_{x,a} \geq \omega_u$, $R_{y,a} = 0$ and $R_{x,b} = 0$, $R_{y,b} \geq \omega_u$, a weighted alternate tree starting from vertex y, constituting colors a and b, can be obtained by:

1. Add a set of edges $E'_{y,a} \subset E_{y,a}$ to the root node of the tree such that

   $$\sum_{k \in E'_{y,a}} \omega_k = \min(R_{x,a}, R_{y,b}, \omega_u)$$

   Split any edge in $E_{y,a}$ to satisfied the equality, if necessary.

2. At each "leaf-edge" e,

   (a) Terminate the corresponding branch of the tree if e is using color α, and at the end vertex v of that branch,

   $$R_{v,\gamma} \geq \omega_e \quad \text{where} \quad (\alpha, \gamma) = (a, b) \text{ or } (b, a)$$

   that is, sufficient color γ is available at v for changing e from using color α.

(b) Otherwise, attach a set of edges $E'_{v,\gamma} \subset E_{v,\gamma}$ to e such that

$$\sum_{k \in E'_{v,\gamma}} \omega_k = \omega_e - R_{v,\gamma}$$

that is, edges in $E'_{v,\gamma}$ will be changed to use color α and free out sufficient γ in vertex v for e. Split any edge in $E_{v,\gamma}$ to satisfy the equality, if necessary.

3. Repeat (2) until all branches of the tree are terminated.

Since $R_{x,a} \geq \omega_u$ and $R_{y,b} \geq \omega_u$, then

$$\min(R_{x,a}, R_{y,b}, \omega_u) = \omega_u$$

After interchanging colors a and b in the tree, $R_{y,a}$ will equal to ω_u and therefore, $S_{u,a} \geq \omega_u$ so that edge u can be colored using a.

For the case that no color pair (a, b) satisfies the conditions $R_{x,a} \geq \omega_u$ and $R_{y,b} \geq \omega_u$, the operations are similar and the only different is that several weighted alternate trees will be constructed. After performing color rearrangements in every trees, edge u can be colored with edge-splitting. Details can be found in the algorithm listed in Section 4.2.3.

In the special case with $M = 1$, weights on all edges of the weighted alternate tree will equal to one and the tree will degenerate into the alternate path in the classical circuit switching.

3.4.2 Example

The following example will illustrate the edge-coloring procedures of a weighted bipartite multigraph. Figure 3.1 shows the weighted bipartite multigraph representing a rearrangeably non-blocking Clos network with $m = n = 2$ and $M = 4$.

In the other words, there are two colors available and a maximum of four units of each color can appear in each vertex according to constraint 3.2.

The existing color assignments and an uncolored edge u, connecting vertices I_2 and O_1 with $\omega_u = 1$, are shown in the figure. According to step 1, u can be colored using color 0 since

$$S_{u,0} = \min(R_{I_2,0}, R_{O_1,0}) = 1 \geq \omega_u$$

Parameters $R_{I_2,0}$ and $R_{O_2,0}$ are updated accordingly.

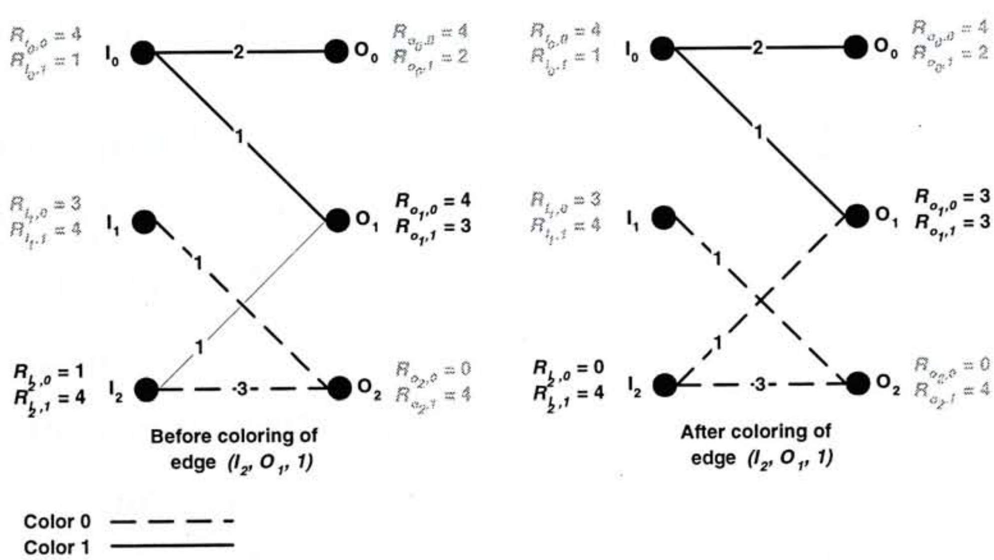


Figure 3.1: Example of edge-coloring without splitting edge

Based on the resulting graph in Figure 3.1, a new uncolored edge u ,connecting I_1 and O_1 with $\omega_u = 4$, is added as shown in Figure 3.2. As $S_{u,0} = S_{u,1} = 3 < \omega_u$, step 1 fails to color u. According to step 2 in the procedures, u is split into two edges, u_0 and u_1, with weights $\omega_{u_0} = 3$ and $\omega_{u_1} = 1$ respectively. The edge-coloring result is shown in the figure.

Figure 3.3 shows an example of color rearrangement using weighted alternate tree. An uncolored edge u, connecting vertices I_0 and O_2 with $\omega_u = 4$,

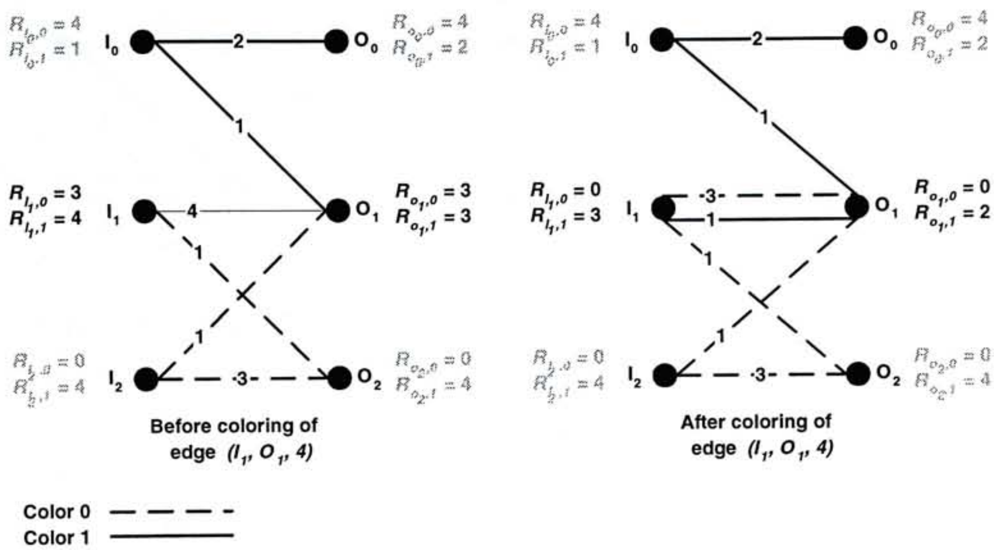


Figure 3.2: Example of edge-coloring with splitting edge

is shown in Figure 3.3(a). After going through steps 1 and 2 of the coloring procedures, color rearrangement is needed in order to color the remaining edge as shown in Figure 3.3(b). Without loss of generality, suppose color 1, which is not available at vertex I_0, will be used for the uncolored edge. A weighted alternate tree starting from I_0 is obtained according to the procedures in Section 3.4.1. Figure 3.3(c) shows the tree before and after interchanging the two colors. The resulting color assignment is shown in Figure 3.3(d) and notes that constraint 3.2 is still satisfied in every vertices.

3.4.3 Validity of the color rearrangement procedure

In the weighted alternate tree, termination of all branches must be guaranteed in order to make the rearrangement procedure valid. However, there are possibilities that infinite branches may occur during the construction of weighted alternate tree. Therefore, pre-caution must be made to avoid such infinite branches.

Infinite branch will be constructed if two edges connecting the same pair of

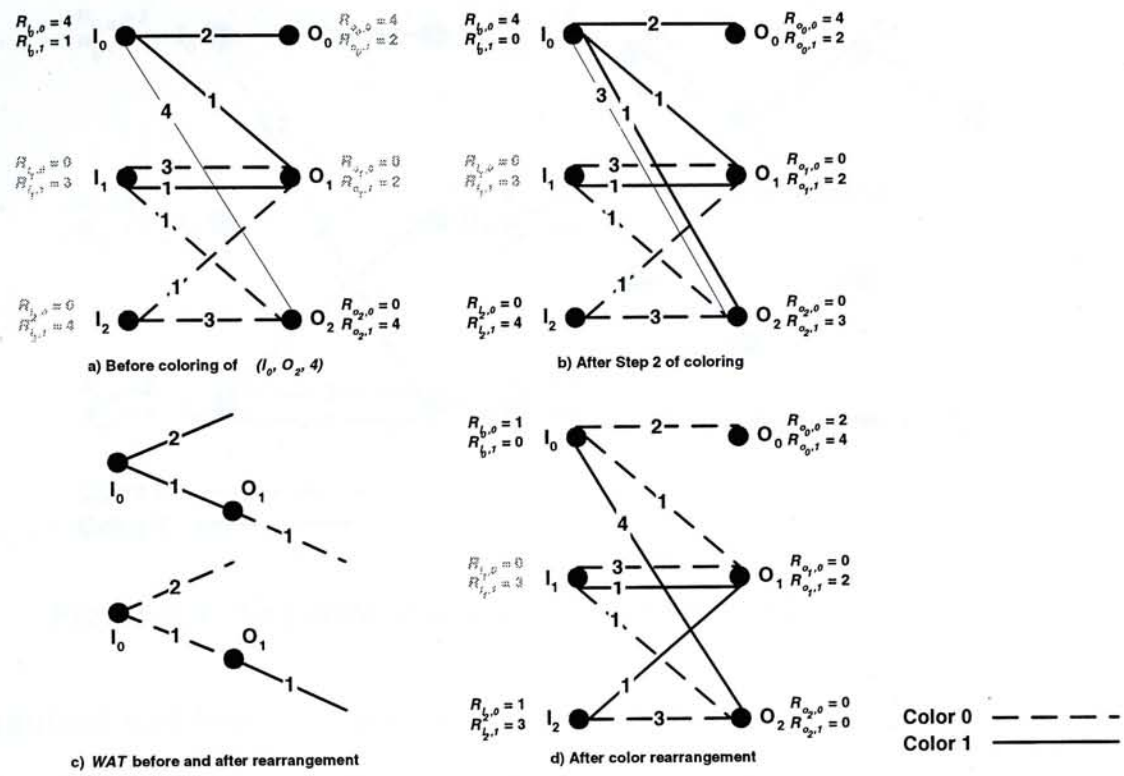


Figure 3.3: Example of edge-coloring using weighted alternate tree

vertices in $WG(V_1, V_2)$ are included in a branch consecutively and repeatedly. In the other words, if any edge k in $E'_{v,\gamma}$ shares the same pair of vertices with the "leaf-edge" e, infinite branch may occur. Figure 3.4 shows an example. Starting from vertex O_2, if the two edges connecting I_2 and O_2 are selected repeatedly, an infinite branch will be resulted. Obviously, this kind of looping can be avoided by excluding from $E'_{v,\gamma}$ those edges that share the same pair of vertices with the "leaf-edge" e.

Observed that if each edge k is decomposed into ω_k unweighted edges, the tree will become an unweighted alternate tree. With the above pre-caution, each branch will then be equivalent to an alternate path in the bipartite multi-graph $G(V_1, V_2)$ described in Section 3.1.1. As all alternate paths will terminate in a bipartite graph, termination of branches in the weighted alternate tree is

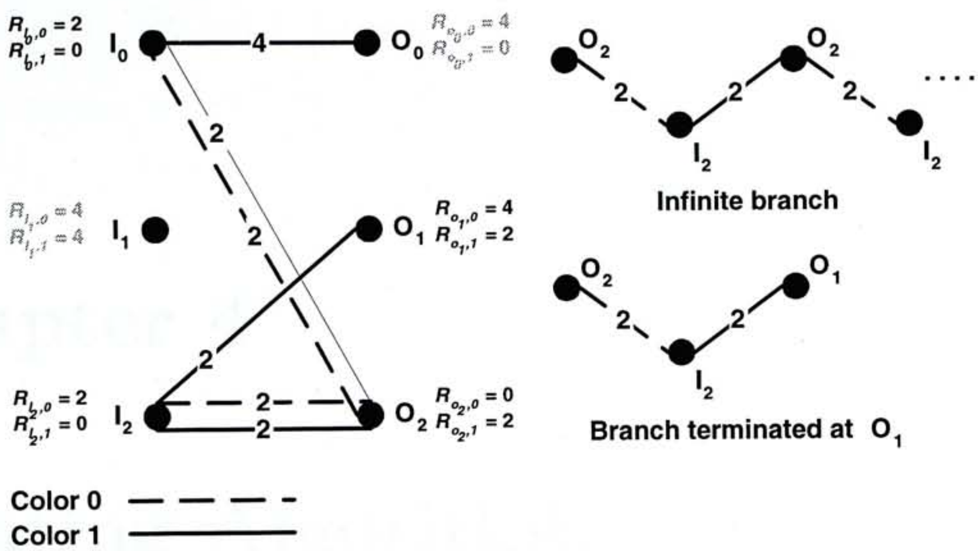


Figure 3.4: Example of infinite branch in weighted alternate tree

guaranteed and hence, makes the rearrangement procedure valid.

In the next chapter, details of the routing algorithm will be given. The proposed algorithm uses a specially designed matrix call *Capacity Allocation Matrix* to implement the edge-coloring principle described in this chapter.

Chapter 4

Routing Algorithm

Based on edge-coloring of the weighted bipartite multigraph representing the connection configuration, a routing algorithm for the quantized Clos network has been implemented using a specially designed matrix called Capacity Allocation Matrix. Instead of minimizing the number of split connections, the proposed algorithm provides a sub-optimal connection splitting control which tries to *reduce* splittings during connection release process. A faster connection setup can be achieved. In this chapter, we will present the details of the routing algorithm.

4.1 Capacity allocation matrix

To implement the edge-coloring principle, a specially designed matrix called *Capacity Allocation Matrix* (CAM) has been employed in the routing algorithm. Figure 4.1 shows a CAM in which each row of the matrix represents an input or output module. The matrix is divided into two parts, the resources matrix

R and the allocation matrix A.

In the resources matrix, each column represents one central module. Central modules are numbered from 0 to $n-1$. Entries $R(I_x, C_y)$ and $R(O_x, C_y)$ are equivalent to R_{I_x,C_y} and R_{O_x,C_y} in the weighted bipartite multigraph model respectively. The value of $S_{u,i}$ can then be calculated from entries of R for any connection u and central module i. Initially, all entries are set to M.

In the allocation matrix, each column represents one connection. In our algorithm, any connection c from input link i to output link o is represented by $c = (p, q, \tilde{\omega})$ where

$$p = \lfloor \frac{i}{n} \rfloor \quad , \quad q = \lfloor \frac{o}{n} \rfloor$$

are the input and output modules of c respectively. $\lfloor \frac{a}{b} \rfloor$ is the integer part of a/b, and $\tilde{\omega}$ is the virtual quantized bandwidth requested by c. Entries $A(I_p, c)$ and $A(O_q, c)$ will be occupied by connection c with value ω. The contents of these two entries must always be the same.

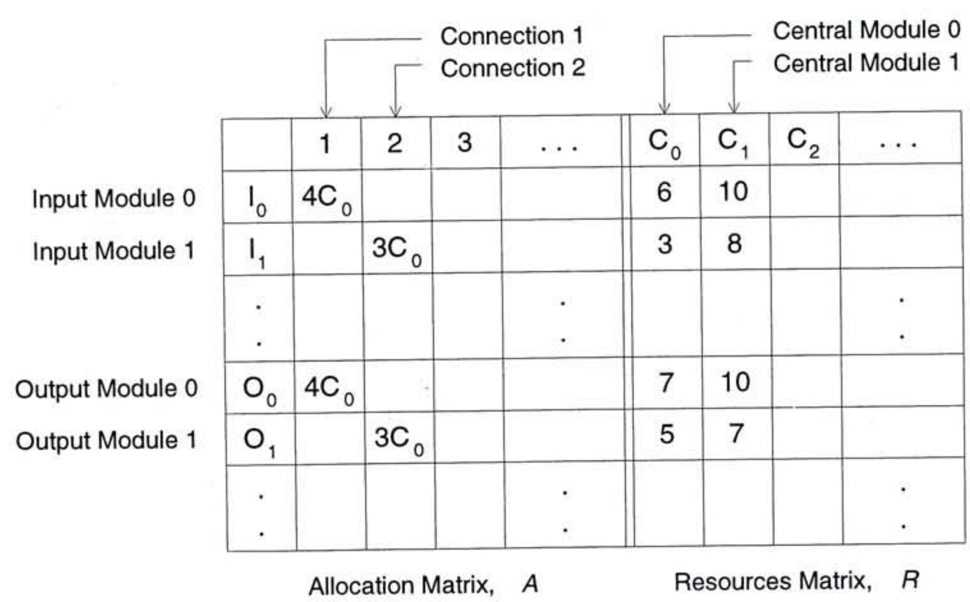


Figure 4.1: Capacity Allocation Matrix

4.2 Connection setup

To setup an externally non-blocked connection $c = (p, q, \tilde{\omega})$, one or more central modules must be assigned to $A(I_p, c)$ and $A(O_q, c)$ to satisfy the capacity request $\tilde{\omega}$. Connection setup consists of three stages: *Non-splitting stage, Splitting stage* and *Recursive rearrangement stage*, according to the three steps in edge-coloring of the weighted bipartite multigraph. Figure 4.2 shows the flow diagram.

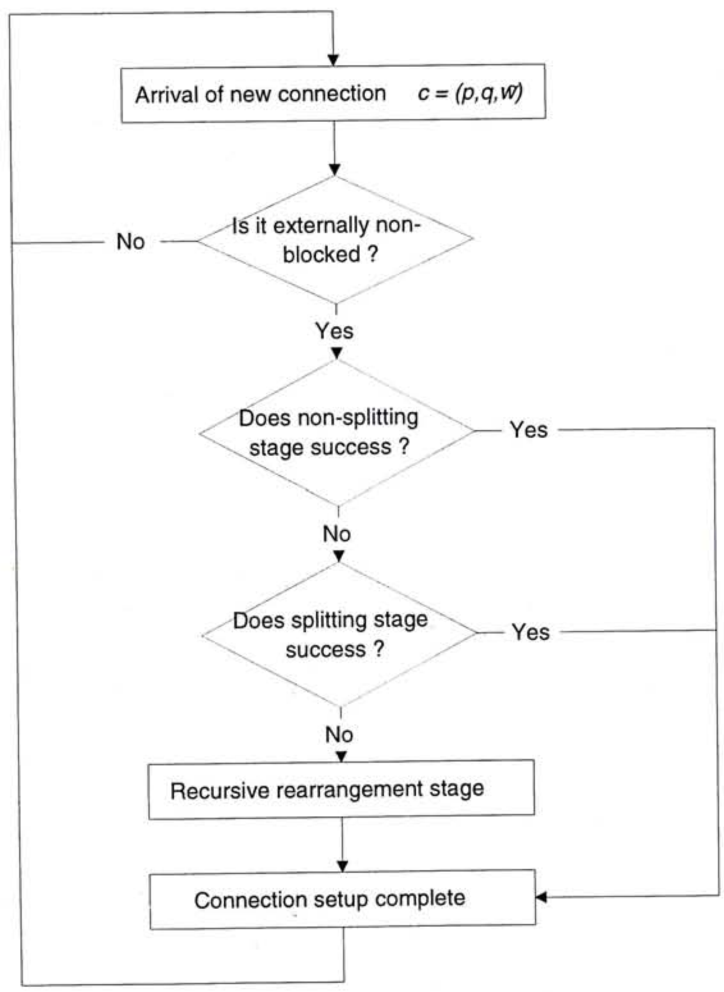


Figure 4.2: Flow diagram for connection setup

4.2.1 Non-splitting stage

The non-splitting stage try to assign exactly one central module to setup connection c and is the counterpart of step 1 in the edge-coloring procedures in last chapter.

Non-splitting Stage

1. $x \leftarrow 0$

2. If $\min(R(I_p, C_x), R(O_q, C_x)) \geq \tilde{\omega}$,
   $A(I_p, c) \leftarrow \tilde{\omega}C_x$
   $A(O_q, c) \leftarrow \tilde{\omega}C_x$
   $R(I_p, C_x) \leftarrow R(I_p, C_x) - \tilde{\omega}$
   $R(O_q, C_x) \leftarrow R(O_q, C_x) - \tilde{\omega}$
   Exit

3. $x \leftarrow x + 1$
   If $x > n - 1$, non-splitting setup fails.
   Else goto step 2.

Figure 4.3 shows an example of non-splitting connection setup. Connections in this example can be represented by the weighted bipartite multigraph in Figure 3.1. Here, $(p, q, \tilde{\omega}) = (2, 1, 1)$ and since

$$\min(R(I_2, C_0), R(O_1, C_0)) = 1 \geq \tilde{\omega},$$

connection 5 will be setup using central module 0.

	1	2	3	4	5	C_0	C_1
I_0	$2C_1$	$1C_1$				4	1
I_1			$1C_0$			3	4
I_2				$3C_0$	1	1	4
O_0	$2C_1$					4	2
O_1		$1C_1$			1	4	3
O_2			$1C_0$	$3C_0$		0	4

Before connection setup

	1	2	3	4	5	C_0	C_1
I_0	$2C_1$	$1C_1$				4	1
I_1			$1C_0$			3	4
I_2				$3C_0$	$1C_0$	0	4
O_0	$2C_1$					4	2
O_1		$1C_1$			$1C_0$	3	3
O_2			$1C_0$	$3C_0$		0	4

After connection setup

Figure 4.3: Example of non-splitting connection setup

4.2.2 Splitting stage

The splitting stage tries to setup connection c using more than one central modules, according to the edge splitting procedure in edge-coloring of the weighted bipartite multigraph.

Splitting Stage

1. $x \leftarrow 0$

2. $A(I_p, c) \leftarrow \min(R(I_p, C_x), R(O_q, C_x), \tilde{\omega})C_x$

   $A(O_q, c) \leftarrow \min(R(I_p, C_x), R(O_q, C_x), \tilde{\omega})C_x$

   $R(I_p, C_x) \leftarrow R(I_p, C_x) - \min(R(I_p, C_x), R(O_q, C_x), \tilde{\omega})$

   $R(O_q, C_x) \leftarrow R(O_q, C_x) - \min(R(I_p, C_x), R(O_q, C_x), \tilde{\omega})$

   $\tilde{\omega} \leftarrow \tilde{\omega} - \min(R(I_p, C_x), R(O_q, C_x), \tilde{\omega})$

   If $\tilde{\omega} = 0$, exit.

3. $x \leftarrow x + 1$

   If $x > n - 1$, unsatisfied bandwidth request remains. (Connection rearrangements are required.)

   Else goto step 2.

Figure 4.4 shows the example of connection setup corresponding to the weighted bipartite multigraph in Figure 3.2. The splitting setup procedure has gone through two passes of step 2 and the details of each pass are also shown in the figure.

	1	2	3	4	5	6	C_0	C_1
I_0	$2C_1$	$1C_1$					4	1
I_1			$1C_0$			4	3	4
I_2				$3C_0$	$1C_0$		0	4
O_0	$2C_1$						4	2
O_1		$1C_1$			$1C_0$	4	3	3
O_2			$1C_0$	$3C_0$			0	4

Before connection setup

→

	1	2	3	4	5	6	C_0	C_1
I_0	$2C_1$	$1C_1$					4	1
I_1			$1C_0$			$3C_0 1C_1$	0	3
I_2				$3C_0$	$1C_0$		0	4
O_0	$2C_1$						4	2
O_1		$1C_1$			$1C_0$	$3C_0 1C_1$	0	2
O_2			$1C_0$	$3C_0$			0	4

After connection setup

Pass	x	min($R(I_p,C_x)$, $R(O_q,C_x)$, ω)	Assignment	ω after step 2
1	0	min(3, 3, 4) = 3	$3C_0$	1
2	1	min(4, 3, 1) = 1	$1C_1$	0

Details of splitting connection setup

Figure 4.4: Example of connection setup with splitting

4.2.3 Recursive rearrangement stage

If the splitting stage cannot satisfy all bandwidth request of c, unsatisfied bandwidth request will be fulfilled by rearranging existing connections. The proposed algorithm takes the recursive approach to traverse the weighted alternate tree and interchange the two colors simultaneously. Recursive algorithm is a classical solution for problems involving tree structure.

Before describing the procedures, two sets have to be defined.

$$S_I = \{i \mid R(I_p, C_i) > 0,\ R(O_q, C_i) = 0, \text{ sorted in decreasing order of } R(I_p, C_i)\}$$
$$= \text{usable central modules for input module } p \text{ but not output module } q$$

$S_O = \{i \mid R(I_p, C_i) = 0,\ R(O_q, C_i) > 0$, sorted in decreasing order of $R(O_q, C_i)\}$

= usable central modules for output module q but not input module p

Recursive Rearrangement Stage

1. $X \leftarrow$ first element in S_I

   $Y \leftarrow$ first element in S_O

   $\omega_{min} \leftarrow \min(R(I_p, C_X), R(O_q, C_Y), \tilde{\omega})$

2. $Rearrange(X, Y, \omega_{min}, I_p, c)$

3. Delete X from S_I if $R(I_p, C_X) = 0$

   Delete Y from S_O if $R(O_q, C_Y) = 0$

4. $\tilde{\omega} \leftarrow \tilde{\omega} - \omega_{min}$

   If $\tilde{\omega} = 0$, exit.

   Else goto step 1.

For the cases that color assignments on several weighted alternate trees are needed to be altered, the way that (X, Y) being chosen minimizes the number of trees involved.

The recursive procedure $Rearrange(x, y, \omega, r, c')$ is shown below.

$Rearrange(x, y, \omega, r, c')$

1. (a) Remove ωC_x from $A(r, c')$ and, $R(r, C_x) \leftarrow R(r, C_x) + \omega$

   (b) $A(r, c') \leftarrow \omega C_y$

   (c) $R(r, C_y) \leftarrow R(r, C_y) - \omega$

2. Search vertically in column c for non-empty entry $A(r', c')$

(a) Remove ωC_x from $A(r', c')$ and, $R(r', C_x) \leftarrow R(r', C_x) + \omega$

(b) $A(r, c') \leftarrow \omega C_y$

(c) $R(r', C_y) \leftarrow R(r', C_y) - \omega$

3. If $R(r', C_y) \geq 0$, return.
   Else $\omega \leftarrow -R(r', C_y)$

4. Search horizontally in row r' for *fewest* entries $A(r', c_i)$ with $\omega_i C_y$ s.t.

$$\sum_i \omega_i = \omega$$

   Exclude c_i's with $\omega_i C_y$ in entry $A(r, c_i)$. Split any c_i if necessary.

5. For each $A(r', c_i)$ in (4), performs *Rearrange*$(y, x, \omega_i, r', c_i)$.

Note that for the first layer of recursion, that is, *Rearrange*$(X, Y, \omega_{min}, I_p, c)$, steps 1(a) and 2(a) must be ignored since nothing has been assigned to $A(I_p, c)$ and $A(O_q, c)$, except those being assigned in non-splitting and splitting stages. Those assignments are not subjected to rearrangements according to the edge-coloring principle.

To obtain an upper bound on the time complexity of the rearrangement algorithm, consider the case that all connections are with $\tilde{\omega} = 1$. In the worst case, there will be $MN - 1$ existing connections in the CAM and connection rearrangement is needed for the remaining new connection. At most $MN - 1$ connection rearrangements will then be performed. Hence, the time complexity of the rearrangement algorithm is upper bounded by $MN - 1$ rearrangements.

Figure 4.5 shows an example and the arrows in the figure indicate the sequences of recursive search in *Rearrange*(x, y, ω, r, c'). Again, the weighted alternate tree representing this example is shown in Figure 3.3.

	1	2	3	4	5	6	7	C_0	C_1
I_0	$2C_1$	$1C_1$					$1C_1$,3	4	0
I_1			$1C_0$			$3C_0 1C_1$		0	3
I_2				$3C_0$	$1C_0$			0	4
O_0	$2C_1$							4	2
O_1		$1C_1$			$1C_0$	$3C_0 1C_1$		0	2
O_2			$1C_0$	$3C_0$			$1C_1$,3	0	3

Before rearranging connections

	1	2	3	4	5	6	7	C_0	C_1
I_0	$2C_0$	$1C_0$					$4C_1$	1	0
I_1			$1C_0$			$3C_0 1C_1$		0	3
I_2				$3C_0$	$1C_1$			1	3
O_0	$2C_0$							2	4
O_1		$1C_0$			$1C_1$	$3C_0 1C_1$		0	2
O_2			$1C_0$	$3C_0$			$4C_1$	0	0

After rearranging connections

Sequence in recursive search

($3C_1$ will be assigned to connection 7)
1. Search vertically for value 3
2. Search horizontally for $3C_1$ and connections 1 ($2C_1$), 2 ($1C_1$) are found
3. Search vertically for $2C_1$
4. Search vertically for $1C_1$
5. Search horizontally for $1C_0$ and connection 5 ($1C_0$) is found
6. Search vertically for $1C_0$

Figure 4.5: Example of connection rearrangements

4.3 Connection release

When any connection $c_R = (s, t, \tilde{\omega}_R)$ is released, the algorithm will examine existing split connections and will re-route them through a single central module to make them non-splitting, if possible. Due to the structure of the Clos network, additional central module capacities are only available through switching modules I_s and O_t. Therefore, the algorithm will only consider those split connections sharing the same input or output modules with c_R.

Before presenting the procedures for connection release, the following items are defined.

$S_R = \{i \mid \text{central module(s) } i \text{ used by } c_R\}$

$\tilde{c}$ = split connection $(u, v, \tilde{\omega})$ with $u = s$ or $v = t$

= split connection sharing the same input or output module with c_R

$S = \{j \mid \text{central modules } j \text{ used by } \tilde{c}, \text{ with capacity } \omega_j \text{ s.t. } \sum_j \omega_j = \tilde{\omega}\}$

$S_{\tilde{c}} = \{\tilde{c}\}$, the set containing all $\tilde{c}$'s

Connection Release

1. Delete column c_R from A. Update R according to c_R.

2. For every $\tilde{c} \in S_{\tilde{c}}$

   (a) Get next i from S_R. If none is available, re-route fails and goto step 2.

   (b) If (1) $i \notin S$ and $\min(R(I_u, C_i), R(O_v, C_i)) \geq \tilde{\omega}$
   or (2) $i = j \in S$ and $\min(R(I_u, C_i), R(O_v, C_i)) \geq (\tilde{\omega} - \omega_j)$,
   Change $A(I_u, \tilde{c})$ and $A(O_v, \tilde{c})$ to $\tilde{\omega}C_i$. Update R according to the changes. Re-route successes and goto step 2.
   Else goto step (a).

Figure 4.6 shows an example of successful re-routing. Connections 5 and 6 in Figure 4.5 are released and re-routed, respectively. In this example,

$$c_R = 5\ ,\quad S_R = \{1\}\ ,\quad c = 6\ ,$$

$$S = \{0, 1\} \text{ with } \omega_0 = 3\ ,\ \omega_1 = 1\ ,\quad S_c = \{6\}$$

	1	2	3	4	5	6	7	C_0	C_1
I_0	$2C_0$	$1C_0$					$4C_1$	1	0
I_1			$1C_0$			$3C_0 1C_1$		0	3
I_2				$3C_0$				1	4
O_0	$2C_0$							2	4
O_1		$1C_0$				$3C_0 1C_1$		0	3
O_2			$1C_0$	$3C_0$			$4C_1$	0	0

After releasing connection 5

	1	2	3	4	6	7	C_0	C_1
I_0	$2C_0$	$1C_0$				$4C_1$	1	0
I_1			$1C_0$		$4C_1$		3	0
I_2				$3C_0$			1	4
O_0	$2C_0$						2	4
O_1		$1C_0$			$4C_1$		3	0
O_2			$1C_0$	$3C_0$		$4C_1$	0	0

After rerouting connection 6

Figure 4.6: Example of re-routing split connection

4.4 Realization of route assignment in packet level

(*Materials given in this section concerning TSI and time-space switching are from* [15].)

Although the routing algorithm is performed in connection base or the so-called call level, the resulting assignments have to be realized in the time-slotted packet level in reality. A combination of time and space switching called *time-space switching* can be employed.

A major component in time-space switching is the *time-slot interchanger* (TSI) shown in Figure 4.7. As the name suggests, a TSI interchanges the time slots occupied by different packets within one packet frame. Switching is performed directly in the time domain. Note that there must be an initial delay equal to one frame time in the TSI. Switching can be performed only after a whole frame of packets is read in. The next frame will be read in while the previous frame is being switched.

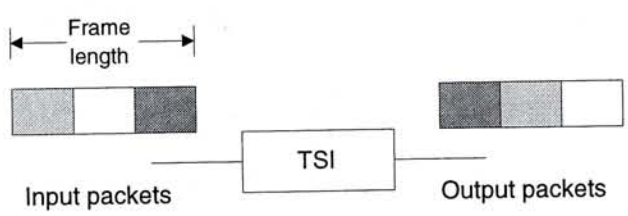


Figure 4.7: Time slot interchanger

By attaching a TSI to each input link of a space-domain switch, packets within a frame can be scheduled by the TSI's such that output contentions can be totally eliminated. Packets can then be switched to their destination by the space-domain switch running different input-output mappings in different time slots. Figure 4.8 shows an example.

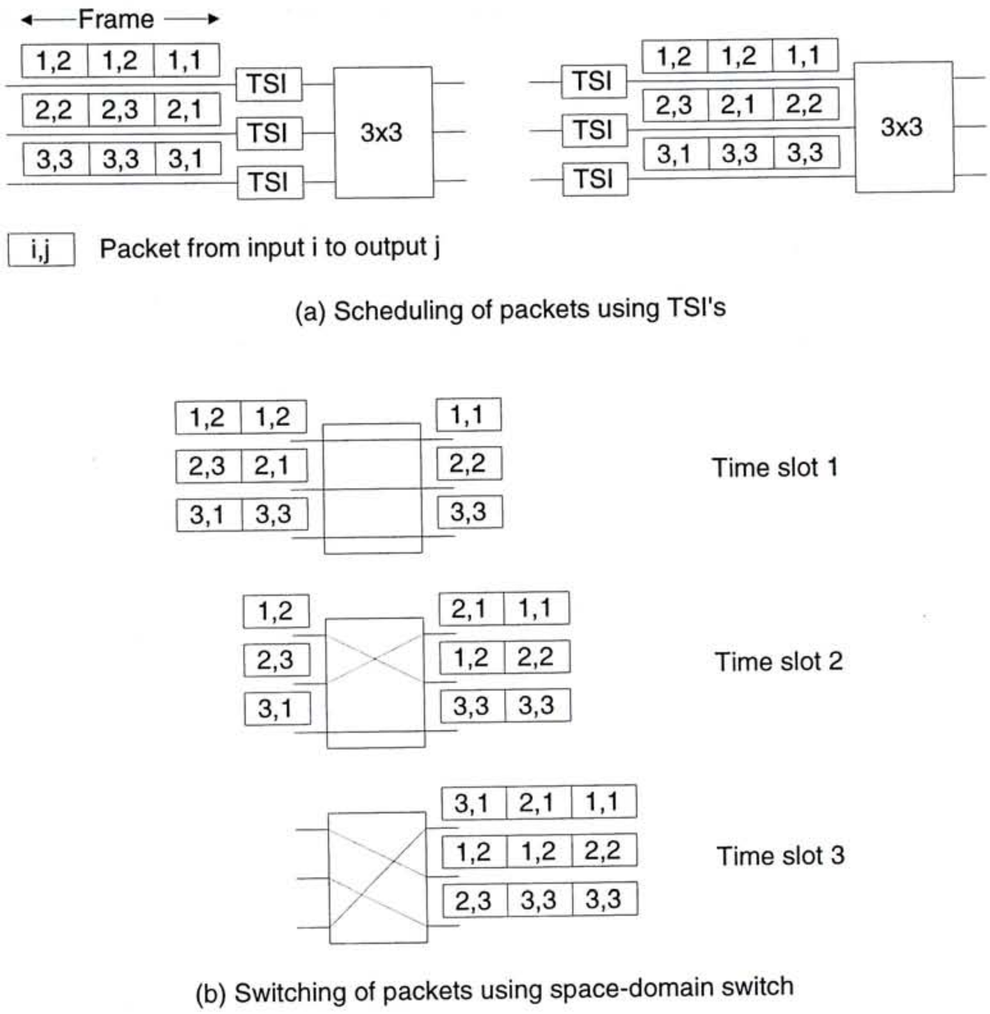


Figure 4.8: Time-space switching

Defining frames with length equals to M time slots, the route assignments of the Clos network can be realized in packet level by replacing all switching modules with time-space switches as shown in Figure 4.9. Any connection with quantized bandwidth request $\tilde{\omega}$ will occupy any $\tilde{\omega}$ time slots in a frame in the corresponding input link. With a *localized* packet scheduling performed in each time-space switching module, packets can be switched through the modules and eventually, the entire network. In the first, second and third stages, packets are scheduled according to their central module assignments, destinated output modules and destinated output links, respectively. Changes in packet schedule will only occur whenever the connection configuration changes.

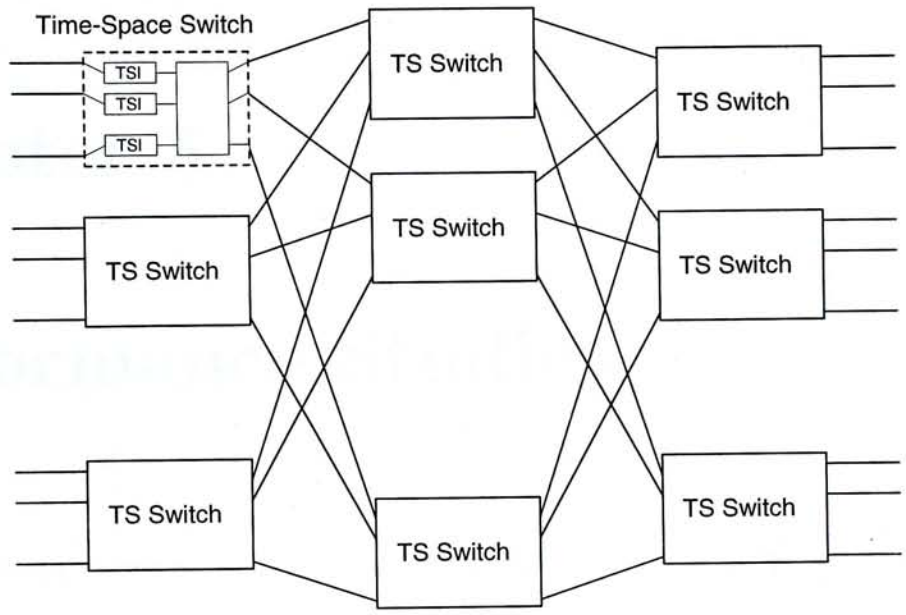


Figure 4.9: Time-space switching Clos network

Note that an initial delay of $3M$ time slots will be experienced due to the three stages of TSI inside the network.

Chapter 5

Performance Studies

The performance of the quantized Clos network under multirate traffic and the proposed routing algorithm have been studied through simulations. In this chapter, we will present the simulation results and will discuss their implications.

In all simulations in this chapter, a 16×16 rearrangeably non-blocking quantized Clos network with $m = n = 4$ is used. Call arrival in any input link is a Poisson process with rate λ. Call duration is exponentially distributed with mean $1/\mu$. Uniform traffic distribution is assumed. It means that each call will destinated for any output link with equal probabilities. Unquantized bandwidth request ω of a call is uniformly distributed in the interval $(0, 1]$. Network loading is defined as λ/μ.

5.1 External blocking probability

The quantized Clos network in this thesis can be considered as a special case of a multi-service circuit switching loss network in which the loss probability can be approximated using *reduced load approximation* [16, 17]. In this section, we will briefly present the approximation and will give simulation results.

5.1.1 Reduced load approximation

(*The materials given in this subsection concerning reduced load approximation are from* [16] *and* [17].)

Consider a network consisting of J links, with link j having capacity of C_j. The network supports K classes of services. Associated with class-k calls is a Poisson arrival with rate λ_k, an exponential holding time with mean $1/\mu_k$, a bandwidth request b_k and a route $R_k \subseteq \{1, \ldots, J\}$. Offered load of class-$k$ calls is $\rho_k = \lambda_k/\mu_k$. A class-k call is admitted if and only if b_k units of bandwidth are free in each link $j \in R_k$. Blocked calls are lost. Such a network is referred as a multi-service circuit switching loss network.

Denote $Q_k(C;\rho)$ be the "blocking probability of a class-k call, with offered load ρ, on a link with capacity C". Also denote L_{jk} be the approximated probability that "a class-k call will be blocked on link j". Intuitively,

$$L_{jk} = Q_k(C_j; \rho_k) \qquad \forall j, k$$

However, *reduced load approximation* states that blocking probability on transmission link j should be approximated by reducing ρ_k so that blocking on the links other than j can be taken into account. With the assumption that blockings are independent from link to link, class-k calls should arrive to link j according to a Poisson process with a *reduced* offered load equals to

$$\rho_k \prod_{i \in R_k - \{j\}} (1 - L_{ik})$$

and hence,

$$L_{jk} = Q_k(C_j; \rho_k \prod_{i \in R_k - \{j\}} (1 - L_{ik})) \qquad \forall j, k$$

This gives rise to a set of fixed-point equations whose solution supplies *approximations* for call blocking probabilities. For a given set of k, C and ρ, Kaufman has proposed an algorithm in [17] to obtain $Q_k(C; \rho)$. Using Kaufman's algorithm, L_{jk} can be solved numerically by repeated substitutions with a set of initial values on L_{jk}. The existence and uniqueness of the solution have been proved in [16]. The overall blocking probability of a class-k call can then be approximated by

$$B_k = 1 - \prod_{j \in R_k} (1 - L_{jk})$$

To approximate the external blocking probability using reduced load approximation, the quantized Clos network can be considered as a special case of such a multi-service network with two transmission links having capacities M and M classes of services. The two transmission links include one of the input links and one of the output links of the Clos network.

In the input link, connections that request for k units of bandwidth are considered as a class-k call. With uniform quantization on the uniformly distributed bandwidth request, a connection will be classified as one of the M classes with equal probabilities, $1/M$. Hence, arrival of class-k calls in the input link will be a Poisson process with rate λ/M for all k. This comes from the fact that splitting a Poisson process of rate λ with probabilities p_i where $\sum_i p_i = 1$ will result in a set of Poisson processes with rates $p_i\lambda$ [18].

In the output link, arrival of class-k calls from a particular input of the $N \times N$ network will be a Poisson process with rate λ/MN due to uniform traffic distribution. Another property of Poisson process states that aggregating a set of Poisson processes with rates λ_i gives another Poisson process with rate $\sum_i \lambda_i$ [18]. Therefore, the overall arrival of class-k calls in the output link will also be a Poisson process with rate λ/M for all k.

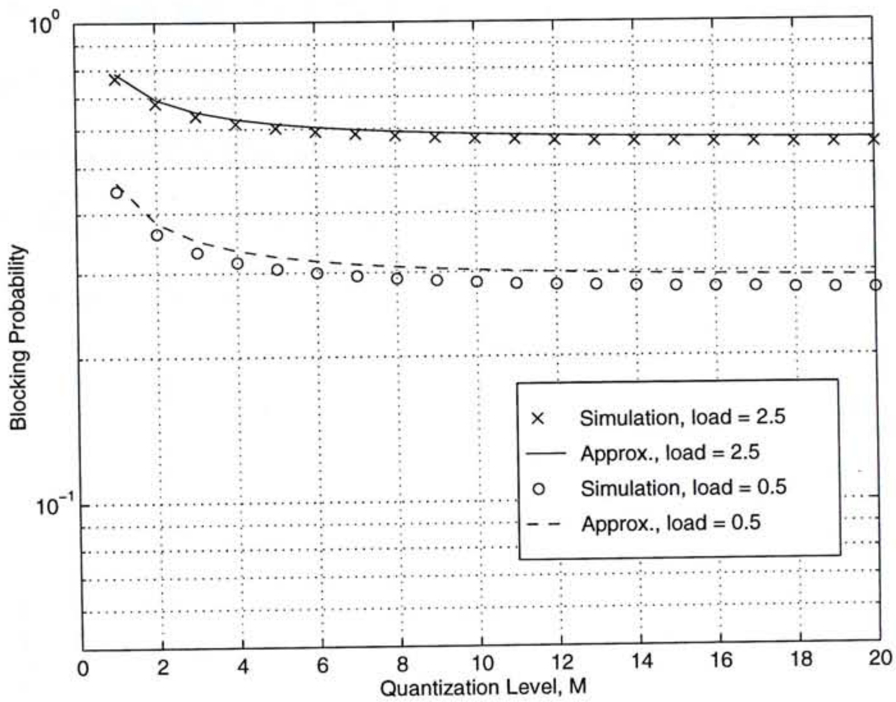


Figure 5.1: External blocking probabilities, approximation and simulation

With this set of parameters, external blocking probability can then be approximated by averaging the set of B_k solved. Figure 5.1 shows the results from both approximation and simulation. It can be observed that the link independence assumption holds well and the approximations always stay closely with the simulation results.

5.1.2 Comparison of external blocking probabilities

In Theorem 5, no restriction has been imposed on the external link utilization β and the network will be rearrangeably non-blocking with $m = n$. Without bandwidth quantization and connection splitting, Clos network with $N = n^2$ and $m = n$ will be rearrangeably non-blocking only if $\beta \leq 0.5$ as mentioned in Chapter 1. Under the same connection arrival process, the external blocking probability will obviously be higher in this case as the usable bandwidth in input and output links are restricted. Figure 5.2 shows the simulation results comparing these blocking probabilities with $m = n = 4$ and $N = n^2 = 16$. Throughput degradation due to bandwidth quantization was negligible when compared with the gain in connection splitting.

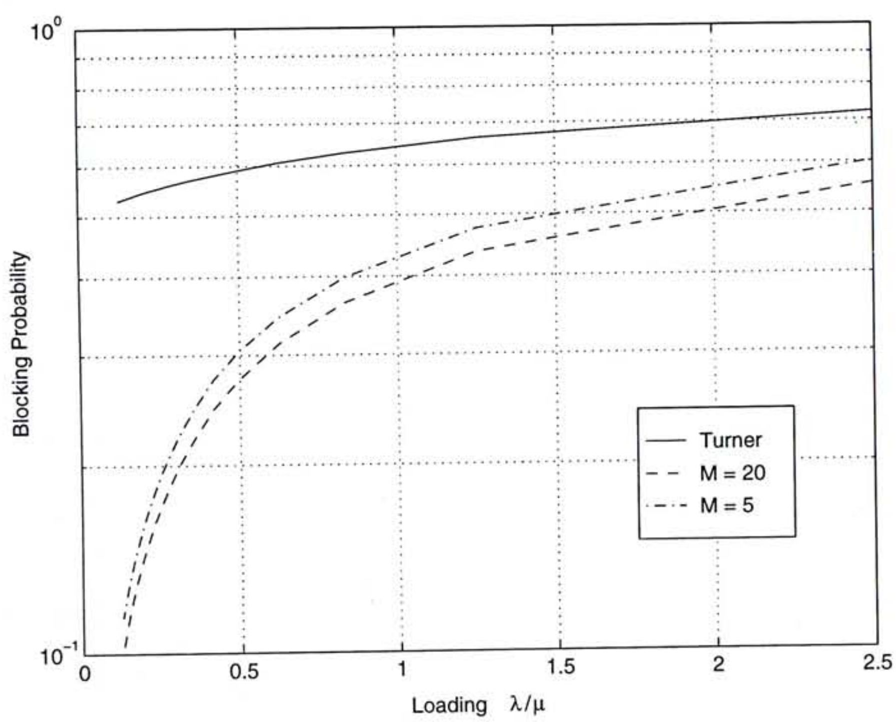


Figure 5.2: Comparison of external blocking probabilities

5.2 Connection splitting probability

As mentioned in Chapter 4, the proposed algorithm only tries to reduce the number of split connections and hence provides a sub-optimal memory requirement in the routing table. Figure 5.3 shows the connection splitting probability against the number of quantization levels M. The low splitting probability implies that the sub-optimal connection splitting control provided by the proposed algorithm can still be acceptable in reducing the memory requirement in the routing table.

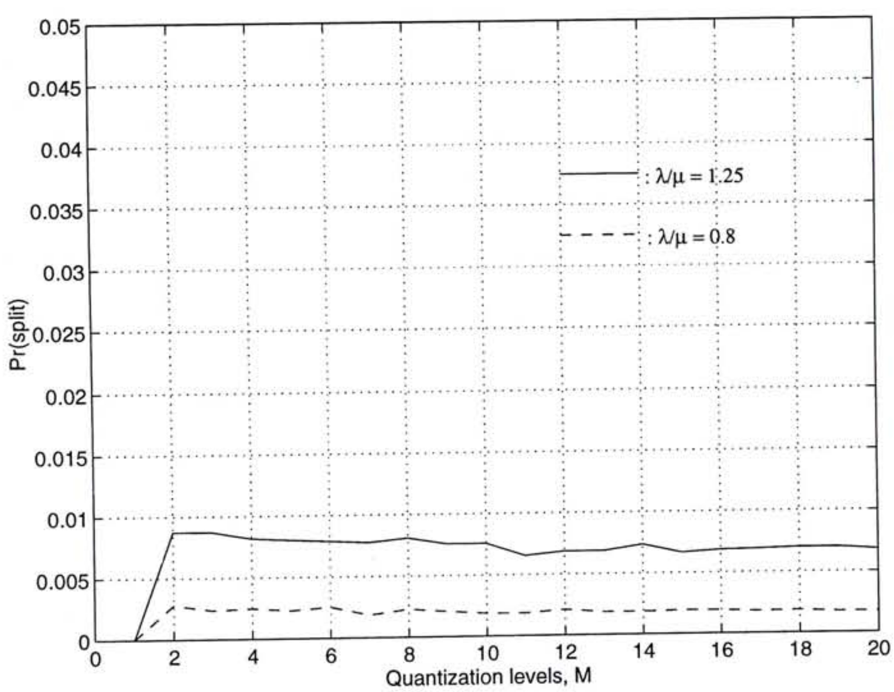


Figure 5.3: Connection splitting probability

5.3 Recursive rearrangement probability

Compared with non-recursive algorithms, recursive algorithms are usually with higher complexity. Instead of analyzing the complexity, we justify the use of a recursive rearrangement procedure in the proposed routing algorithm through simulation. Figure 5.4 shows the probability of connection setup through recursive rearrangement against the number of quantization levels M. From the extremely low probabilities, the recursive rearrangement procedure can still be considered as a good solution in spite of its probably higher complexity.

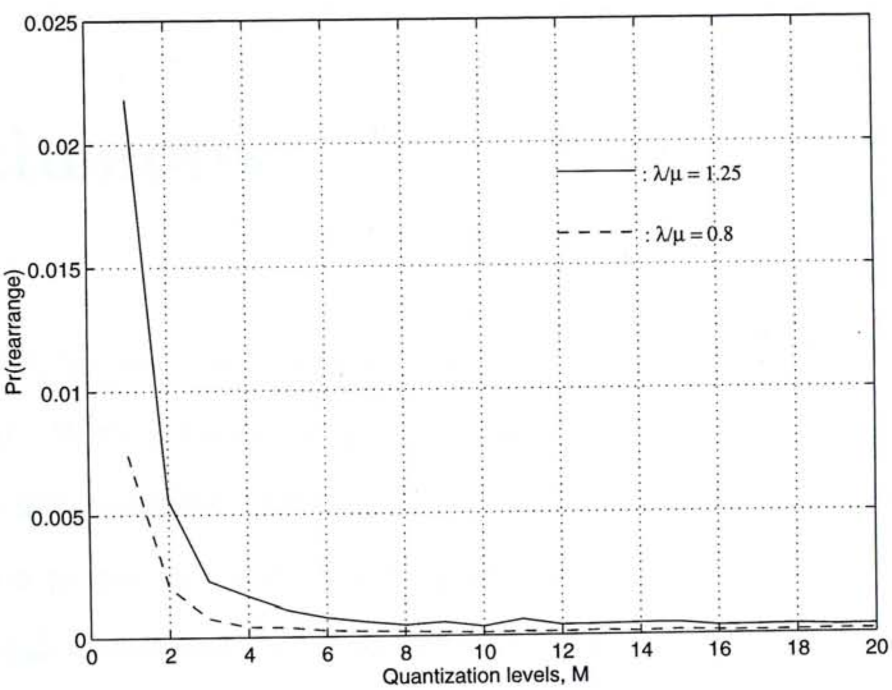


Figure 5.4: Probability of connection setup using recursive rearrangement

Chapter 6

Conclusions

In this thesis, we have investigated the Clos network under multirate traffic environment. With connection splitting and bandwidth quantization, we have derived the non-blocking conditions for the Clos network. A routing algorithm has also been proposed for the rearrangeably non-blocking Clos network.

In classical circuit switching, non-blocking conditions and routing algorithm have been derived for Clos network. With the appearance of services with widely varying bandwidth requests, researches have been performed on Clos network under such multirate environment. Without connection splitting, non-blocking conditions with restricted external link utilization β have been derived. However, this limitation has sharply increased the external blocking probability. In addition, no routing algorithm has been proposed for the Clos network due to the difficulties in handling the continuous scale of bandwidth requests. Thus, it is necessary to study the Clos network under multirate environment with the employment of connection splitting and bandwidth quantization.

In Chapter 2, routing in Clos network in classical circuit switching has been reviewed. By representing the connection configuration using a bipartite multigraph, route assignment can be formulated as an edge-coloring problem. A classical implementation of the routing algorithm based on the graph coloring principle is by Paull's connection matrix.

In Chapter 3, the two major assumptions, connection splitting and bandwidth quantization, have been studied. Connection splitting improves the internal link utilization and thus releases the limitation on β, although it may be necessary to re-sequence the split traffic. Bandwidth quantization converts the continuous scale of bandwidth requests into a discrete, finite scale which makes route assignment manageable. With these assumptions, we have proved that the network will be strictly and rearrangeably non-blocking for the externally non-blocked connections if the conditions $m \geq \lceil \frac{2Mn-1}{M} \rceil$ and $m \geq n$ are satisfied respectively. We have also presented the principle of the proposed routing algorithm in this chapter. By representing the connection configuration with a weighted bipartite multigraph, routing can also be formulated as an edge-coloring problem. The proposed algorithm is a generalization of the one in classical circuit switching.

To implement the proposed algorithm, a specially designed matrix called capacity allocation matrix has been presented in Chapter 4. With this matrix, connection setup in the routing algorithm consists of three stages: non-splitting stage, splitting stage and recursive rearrangement stage, according to the edge-coloring principle. Instead of optimizing the memory requirement in the routing table through minimizing the number of split connections, the algorithm only provided a sub-optimal connection splitting control to achieve a faster connection setup.

As shown in Chapter 5, the external blocking probability of the quantized Clos network can be approximated using reduced load approximation. After releasing the restriction on β through connection splitting, it has been shown from the simulation that external blocking probability has been reduced substantially. Throughput degradation due to bandwidth quantization was comparatively negligible. In addition, the uses of sub-optimal connection splitting control and recursive rearrangement procedure have also been justified by the low connection splitting and rearrangement probabilities respectively.

Finally, some related directions for further research are outlined below. In this thesis, bandwidth requests are assumed to be uniformly distributed. It will be worthwhile to study the determination of optimal quantization schemes for different bandwidth distributions. Another issue that is worthy of study is optimization of connection splitting as the proposed algorithm only provides a sub-optimal splitting control.

Bibliography

[1] C. Clos, "A Study of Non-blocking Switching networks," *Bell System Tech. J.*, Vol. 32 Mar. 1953, pp.406-424

[2] V.E. Benes, "Mathematical Theory of Connection Networks and Telephone Traffic," Academic Press, New York, 1965

[3] M.C. Paull, "Reswitching of Connection Network," *Bell Syst. Tech. J.*, vol 41, 1962, pp.833-855

[4] J.Y. Hui, "Switching and Traffic Theory for Integrated Broadband Networks," pp.60-70, Kluwer Academic Publishers, Boston, 1990

[5] R. Melen and J.S. Turner, "Nonblocking Multirate Distribution Networks," *IEEE Trans. Comm.*, Vol. 41, No. 2 Feb. 1993, pp.362-369

[6] S.P. Chung and K.W. Ross, "On Non-blocking Multirate Interconnection Networks," *SIAM J. Comput.*, Vol. 20, No. 4 Aug. 1991, pp.726-736

[7] F.K. Hwang, J.H. Kim, D.Z. Du and B. Gao, "On Multirate Rearrangeable Clos Networks," To appear in *SIAM J. Applied Math.*

[8] T.H. Cormen, C.E. Leiserson and R.L. Rivest, "Introduction to Algorithms," pp. 335-336, The MIT Press, London, 1994

[9] C.T. Lea and A. Alyatama, "Bandwidth Quantization and States Reduction in the Broadband ISDN," *IEEE/ACM Trans. on Networking*, Vol. 3, No. 3 Jun. 1995, pp. 352-360

[10] G.F. Lev, N. Pippenger and L.G. Valiant, "A Fast Parallel Algorithm for Routing in Permutation Networks," *IEEE Trans. on Computers*, Vol. c-30, No. 2 Feb. 1981, pp.93-100

[11] R.J. Wilson, "Introduction to Graph Theory," Longman, England, 1985

[12] S.Y. Liew, "Parallel Routing Algorithms in Benes-Clos Networks," M.Phil Thesis, 1996

[13] R.J. McEliece, R.B. Ash and C. Ash, "Introduction to Discrete Mathematics," McGraw Hill, New York, 1989

[14] J. Sharony, T.E. Stern and K.W. Cheung, "Extension of Classical Rearrangeable and Nonblocking Networks to The Wavelength Dimension," *Proc. GLOBECOM'92*, Dec. 1992, pp. 1901-1905

[15] S.C. Liew and T.T. Lee, "Principles of Broadband Switching and Networking: Draft 3," pp. 11-15, 116-119

[16] K.W. Ross, "Multiservice Loss Models for Broadband Telecommunication Networks," pp. 17-25, 187-190, Springer, London, 1995

[17] J.S. Kaufman, "Blocking in a Shared Resource Environment," *IEEE Trans. Comm.*, Vol. 29, No. 10 Oct. 1981, pp 1474-1481

[18] S.M Ross, "Introduction to Probability Models: Fifth Edition," Academic Press, San Diego, 1993