# Analysis on Memory-Space-Memory Clos Packet Switching Network

Xiangjie Ma, Yuxiang Hu, Junpeng Mao, Julong Lan, Lian Guan, and Baisheng Zhang

Information Engineering Institute, PLA Information Engineering University
National Digital Switching System Engineering & Technological Research Center
Zhenzhou, Henan, 450002, P. R. China
maxiangjie100@163.com,
{mxj,hyx,mjp,ljl,gla,zbs}@mail.ndsc.com.cn

**Abstract.** Memory-Space-Memory (MSM) Clos packet switching networks are the next step in scaling current crossbar switches to many hundreds or few thousands of ports. Clos networks had been studied and applied quite well in circuit switching system, with much attentions paid to its non-blocking property to decrease call blocking rates. In contrast, for packet switching systems, more care is taken to per-packet based forwarding performance of the switching networks. MSM Clos network has the merit of keeping packet sequence and therefore is quiet adapt to packet switching fabric. By way of buffering architecture, MSM Clos network is quite similar to the CIOQ Crossbar based single stage switching fabric, which promotes us to extend the results of CIOQ matching an OQ switch [1] to MSM Clos networks. Meanwhile, although the CIOQ switch can emulate an OQ switch, it needs cell insertion algorithm and stable matching algorithm with high information complexity and computing complexity. This has prevented its application seriously in new generation of routers with high speed linking rates and large port numbers. So we propose a new method of Per-Input OQ Emulation (PIOE), including both new cell insertion and scheduling algorithm (PVPP-CIP and -CSP) with only per-input local information and new matching algorithm ($S^3$) with computing complexity of O(1), which is more practical in both CIOQ Crossbar and MSM Clos networks.

## 1 Introduction

With the constantly increasing Internet traffic and the development of broadband access technologies, such as DSL, cable modem and gigabit Ethernet, the next generation routers should support a large number of connection ports for the following two reasons [2][5]. (a) increasing number of Internet accessing points leads to increasing number of input ports and output ports; and (b) Optical transmission technologies such as DWDM is making increasing number of transmitting links available in Internet. The current widely used single stage Crossbar switching fabric, however, can not afford to large number of switching ports for surprising high complexity in switching hardware and scheduling algorithms [3][4][5].

M. Xu et al. (Eds.): APPT 2007, LNCS 4847, pp. 209 – 221, 2007.

The Memory Space Memory Clos network, in contrast, is much scalable in switching port number than traditional single Crossbar Fabric, and therefore is causing more and more attention in the next generation of routers. The MSM Clos network itself, however, is not firstly proposed in packet switching domain. In 1953, C. Clos from Bell Systems Labs had proposed the famous Clos network to scale the switching fabric in telephony switches [6]. In circuit switching, more attentions had been paid to blocking property of Clos network to increase call access rates [6][7][8][9]. It is a challenging work to find an efficient and fast scheduling scheme to provide high throughput, starvation-free, acceptable delay, and fairness performance under various traffic conditions for a Clos packet switching network. In [10][11][12], the proposed path-switching scheme and static round-robin (Distro) scheduling algorithm, however, cannot handle various traffic conditions well due to their static nature.

In [1], Shang-Tse Chuang, Ashish Goel etc. studied the speedup problems for CIOQ single Crossbar switch to emulate an OQ switch. They show that a speedup of $2-\frac{1}{N}$ is necessary and a speedup of two is sufficient for this exact emulation. Most interestingly, their result holds for all traffic arrival patterns and is independent with the switching size. The optimal performance of a CIOQ switch urges us to extend the results to MSM Clos network, for the homology in buffering mechanism and the resemblance in architecture between them. We observe and analyze different properties between a single stage Crossbar fabric and a multistage Clos network, and further give the conditions for them to mimic each other. Based on this conditions and non-blocking condition for a reconfigurable Clos network, we provide necessary and sufficient condition for a Clos packet switching network to emulate an OQ switch. Most surprisingly, our result also has the merit of holding for all traffic arrival patterns and being independent with switching size.

However, although the perfect performance of an OQ switch is the target pursued in practical high-speed routers, it has never been achieved in switch with high link speed and large port numbers. This is because the present cell insertion algorithm and matching algorithm have disadvantages of high information complexity and computing complexity in emulating an OQ switch.

We present a method called Per-Input OQ Emulation (PIOE) for MSM CLOS network to emulate an OQ switch based on per-input priority and fairness, which has two merits: (a) without global information exchange among inputs and outputs of the switch, and thus eliminate the information complexity; (b) with an algorithm complexity as low as O(1).

The rest of this paper is organized as follows. In Section II, we introduce some terminology and definitions. In Section III, we describe MSM Clos network Model. In Section IV, we find conditions for the single Crossbar Fabric and the Clos network to mimic each other, and then find the necessary and sufficient condition for Clos network to emulate an OQ switch. In Section V, we put forward a more practical emulation method—the PIOE method, including PVPP-CIP & -CSP and $S^3$ scheduling algorithm. In Section VI, we will have a conclusion of this paper.

## 2 Terminology and Definitions

Before proceeding, it will be useful to define some terms used in our presentation. We adopt fixed-length packet concept and call the packets or segment packets 'cells' afterwards. This is common practice in high performance routers [14].

**Time slot:** Refers to the time taken to transmit or receive a fixed length cell at a link rate of R.

**CIOQ Switch:** A switch in which there are two stages of buffering on input ports and output ports of an N×N switch. Arriving cells are firstly placed in queues at the input, and then switched to the queues at the output.

**OQ Switch:** A switch in which arriving cells are placed immediately in queues at the output, where they contend with other cells destined to the same output. The departure order might be FIFO, in which case we call it an FIFO-OQ switch. Other service disciplines, such as WFQ [15], GPS [16], virtual clock [17], and DRR [18] are widely used to provide QoS guarantees. One characteristic of an OQ switch is that the buffer memory must be able to accept (write) N new cells per time slot where N is the number of ports, and read one cell per cell time. Hence, the memory must operate at N+1 times the line rate.

**Shadow OQ switch:** We will assume that there exists an OQ switch, called the shadow OQ switch, with the same number of input and output ports as the MSM Clos network. The ports on the shadow OQ switch receive identical input traffic patterns and operate at the same line rate as the MSM Clos network.

## 3 Modeling of MSM Clos Networks

The topology architecture of an MSM Clos network is shown in Fig.1. The basic components in Clos network are switching modules, which can be denoted by $X_{nm}$ with $n$ input ports and $m$ output ports. The three stage Clos network (we only study three stage Clos network in this paper, and it is briefly called Clos network in the rest of this paper) is therefore can be denoted as $[X_{nm}, X_{\mathrm{rr}}, X_{\mathrm{mn}}]$ with $r$ first stage $X_{nm}$ (also called input stage, denoted as IM), $m$ second stage $X_{rr}$ (also called central stage, denoted as CM), $r$ third stage $X_{mn}$ (also called output stage, denoted as OM). The $n$ inputs of each first stage $X_{nm}$ are connected to $n$ input memories of Clos network, and $m$ outputs of $X_{nm}$ connecting to one input of the second stage $X_{rr}$; $r$ outputs of $X_{rr}$ are connected to one input of $r$ third stage $X_{mn}$; $n$ outputs of $X_{mn}$ are connected to $n$ output memories of Clos network.

In sense of graph theory, an MSM Clos network can be denoted by a directed graph $C(n,r,m)$, with all switching modules and memories as vertices, and with all connections between all vertices as edges. Then the Clos network can be expressed

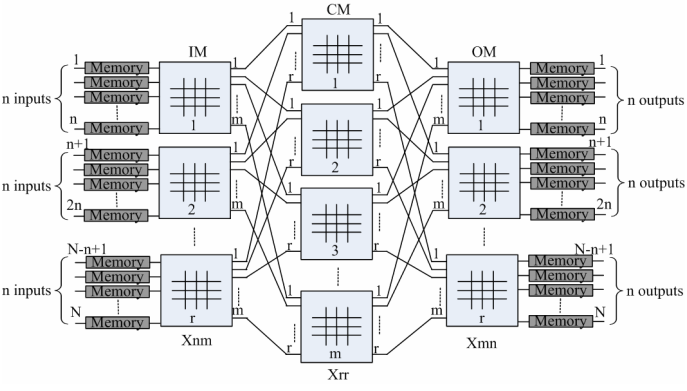


**Fig. 1.** Topology architecture of an MSM Clos network

as $C(n,r,m)=C(V,E)$, in which $V$ is composed of five incompatible vertex sets and four incompatible edge sets. To be precisely, $V=V_0\cup V_1\cup V_2\cup V_3\cup V_4$

$$\text{where, } V_i=\begin{cases}\{v_1^i,v_2^i,...,v_{nr}^i\} & i=0,4\\ \{v_1^i,v_2^i,...,v_r^i\} & i=1,3\\ \{v_1^i,v_2^i,...,v_m^i\} & i=2\end{cases} \tag{1}$$

$E=E_0\cup E_1\cup E_2\cup E_3$, and $E_i$ is edge sets from $V_i$ to $V_{i+1}$ ($i$=0, 1, 2, 3),

$$\text{where, } E_i=\begin{cases}\{\vec{xy}\mid x\in\{v_{(j-1)n+1}^0,v_{(j-1)n+2}^0,...,v_{jn}^0\},y=v_j^1,j=1,2,...,\mathrm{r}\} & i=0\\ \{\vec{xy}\mid x\in V_i,y\in V_{i+1}\} & i=1,\ 2\\ \{\vec{xy}\mid y=v_j^3,x\in\{v_{(j-1)n+1}^4,v_{(j-1)n+2}^4,...,v_{jn}^4\},j=1,2,...,\mathrm{r}\} & i=3\end{cases} \tag{2}$$

To describe Clos network conveniently, we shall give some definitions useful in Clos network.

**Definition 1. Path** — If a directed graph can be tracked from end to end and pass all vertices and edges once, we call the directed graph a Path.

**Definition 2. Stable Path** — Supposing path $p$ is composed of five vertices and four edges, if all five vertices in $p$ belong to five incompatible vertex sets $\{V_0,V_1,V_2,V_3,V_4\}$, and all four edges in $p$ belong to four incompatible edge sets $\{E_0,E_1,E_2,E_3\}$ of $C(n,r,m)$, respectively, we call path $p$ a stable path in Clos network, and denote it as $\hat{p}$. That is to say, stable path $\hat{p}$ can be defined as follows:

$$\hat{p}\overset{\Delta}{=}\{v_i^0v_j^1v_k^2v_l^3v_q^4,\overrightarrow{v_i^0v_j^1}\in E_0,\overrightarrow{v_j^1v_k^2}\in E_1,\overrightarrow{v_k^2v_l^3}\in E_2,\overrightarrow{v_l^3v_q^4}\in E_3\}$$

**Definition 3. Stable Path Set and Stable Path Total Set** — Supposing $P$ to be a stable path set with all paths having no compatible edges mutually, if all left stable

paths in $C(n,r,m)$ are always have compatible edges with one of stable paths in $P$, we call $P$ Stable Path Set in $C(n,r,m)$, and call all possible stable path sets in $C(n,r,m)$ as Stable Path Total Set. We denote stable path set and stable path total set as $S\hat{P}S$ and $S\hat{P}\hat{T}S$, which can be expressed as follows:

$$S\hat{P}S = \{\hat{p}_1, \hat{p}_2, ..., \hat{p}_h, \forall 1 \le i < j \le h, E(\hat{p}_i) \cap E(\hat{p}_j) = \Phi; \forall l > k \ge i \ge 1, E(\hat{p}_i) \cap E(\hat{p}_l) \ne \Phi\}$$

It is very interesting to notice that the stable path set is not unique in $C(n,r,m)$. For example, in stable path set $S\hat{P}S_i$, two stable paths can exchange their vertices in $V_2$ and exchange their edges in $E_1$ and $E_1$, respectively, and then become two new stable paths. Therefore, we get a new path set (i.e. $S\hat{P}S_j$), which is still a stable path set for $S\hat{P}S_j$ shares the same vertices set and edges set with $S\hat{P}S_i$. This property can be explained by architecture of Clos network, in which there are $m$ paths from any input port to any output port. So, there are as many as $(C_m^n)^r \times [(nr)!]$ stable path sets in the stable path total set in $C(n,r,m)$ altogether. Based on the above analysis, we can get two properties of stable path total set by way of graph theory: one is edge exchanging closure, which means a stable path set can become a new stable path set by exchanging any of two stable paths, but this new stable path set is still included by stable path total set. The other is inclusiveness, which means the stable path total set includes all possible stable path sets in $C(n,r,m)$.

**Lemmon 1.** *The number of stable paths* $SIZE(S\hat{P}S)$ *in a three-stage symmetry Clos network* $C(n,r,m)$ *is* $\min(nr, mr)$.

*Proof.* From the definition of stable path and property of Clos network, the number of stable paths in Clos network is equal to the minimum value of all four edge sets in $C(n,r,m)$. This can be expressed as follows:

$$SIZE(S\hat{P}S) = \min\{SIZE(E_0), SIZE(E_1), SIZE(E_2), SIZE(E_3)\}$$

Therefore, $SIZE(S\hat{P}S) = \min\{nr, mr, mr, nr\} = \min(nr, mr) = \begin{cases} nr, m \ge n \\ mr, m < n \end{cases}$ ■

**Lemmon 2.** Supposing the Clos network $C(n,r,m)$ satisfies $m \ge n$, for any integer $k$ ( $1 \le k \le nr$ ) and any vertex pair set $M_f = \{(a_i, b_i) \mid i = 1,2,...,k\}$ from vertex set $A = \{a_1, a_2, ..., a_k\} \subset V_0$ to $B = \{b_1, b_2, ..., b_k\} \subset V_4$, there always stable path set $S\hat{P}S_i \subset S\hat{P}\hat{T}S$ connecting $k$ vertex pairs in $M_f$.

*Proof.* Because integer $k$ satisfies $k \le nr \le mr$, from Lemmon 1 we know that $k \le SIZE(S\hat{P}S)$ and therefore $k \le SIZE(E_0) = SIZE(E_3)$, and $k \le SIZE(E_1) = SIZE(E_2)$ respectively. So we always can find four edge sets composed of $k$ edges

from $E_0$, $E_1$, $E_2$ and $E_3$ to buildup $k$ stable paths to connect $k$ vertex pairs in $M_f$. From property of inclusiveness of stable path total set of $C(n,r,m)$, there always stable path set $S\hat{P}S_i \subset S\hat{P}\hat{T}S$ connecting $k$ vertex pairs in $M_f$. ■

We notice here that Lemmon 2 is equivalent to the reconfigurable non-blocking condition of the Clos network, which had been proved in prior results [6][9].

## 4 Emulating an OQ Switch by MSM Clos Network

The non-blocking condition, in Lemmon 2 in Section III, guarantees any number of vertex pairs connected by a stable path set in the Clos network. Intuitively, this is quite similar to the non-blocking property of a single stage Crossbar Fabric.

**Lemmon 3.** Supposing MSM Clos network $C(n,r,m)$ satisfies $m \geq n$, for single stage CIOQ Crossbar Fabric $C(nr \times nr)$, we say MSM Clos network $C(n,r,m)$ mimics $C(nr \times nr)$.

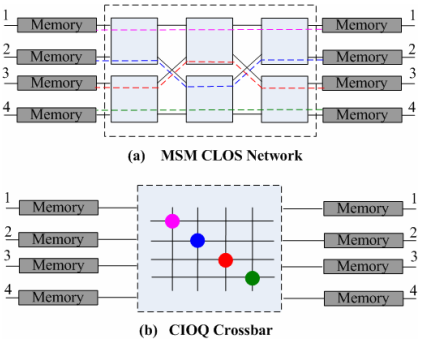


**Fig. 2.** A 4×4 CIOQ Crossbar mimics the MSM Clos network

We do not mean to give strict proof of Lemmon 3 here, but informally provide an analysis by way of an example in Fig.2. There are four input ports and four output ports for both MSM Clos network and CIOQ Crossbar. For the same port pair set $\{(1,1),(2,2),(3,3),(4,4)\}$, we can either find a stable path set $S\hat{P}S_4$ in MSM Clos network, or find matching matrix $\tilde{M}_4$ in CIOQ Crossbar, to find stable paths or configure crosspoints to set up connections between all four port pairs. Intuitively, stable path set $S\hat{P}S_4$ and matching matrix $\tilde{M}_4$ have played the same role in the switching fabrics. In this sense, we say a non-blocking (i.e. $m \geq n$) MSM Clos network and CIOQ Crossbar mimic each other, and call the stable path set is equivalent to the matching matrix $S\hat{P}S \Leftrightarrow \tilde{M}$.

$$S\hat{P}S_4 = \{\hat{p}_1, \hat{p}_2, \hat{p}_3, \hat{p}_4\} = \begin{Bmatrix} \hat{p}_1 = v_1^0 v_1^1 v_1^2 v_1^3 v_1^4 \\ \hat{p}_2 = v_2^0 v_1^1 v_2^2 v_1^3 v_2^4 \\ \hat{p}_3 = v_3^0 v_2^1 v_1^2 v_2^3 v_3^4 \\ \hat{p}_4 = v_4^0 v_2^1 v_2^2 v_2^3 v_4^4 \end{Bmatrix} \quad \tilde{M}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**Theorem 1.** *The necessary and sufficient condition for an MSM Clos network $C(n,r,m)$ to emulate an OQ switch is with speedup of $S$, where $S$ satisfies:*

$$S \geq \begin{cases} \frac{n}{m} \times (2 - \frac{1}{mr}) & m < n \\ 2 - \frac{1}{nr} & m \geq n \end{cases} \tag{3}$$

*Proof:* (a) For the case of $m \geq n$, we can refer to Lemmon 3 and regard the MSM Clos network $C(n,r,m)$ mimics a CIOQ Crossbar fabric with port number $C(nr \times nr)$. So, the results in [1] still hold where $N = nr$, and then we get $S = 2 - \frac{1}{nr}$ for an MSM Clos network to emulate an OQ switch.

(b) For the case of $m < n$, from Lemmon 1 we know that the number of stable paths $SIZE(S\hat{P}S)$ in a stable path set $S\hat{P}S$ is $\min(nr, mr)$. Therefore, there are at most $mr$ stable paths in the MSM Clos network. Fig.3 shows an example of an MSM Clos network with $n = 4$, $m = 2$, $r = 2$ and a CIOQ Crossbar with port number 8×8 and switch fabric of 4×4. Because there is at most 2×2=4 stable paths in the MSM Clos network, regarding the former case of $m \geq n$, four out of eight input ports can emulate an OQ switch with a speed up of $(2 - \frac{1}{4})$. Thus, if we divide the input ports into two groups (i.e. the light colored input ports group and the dark colored input ports group in Fig.3), the emulation can be divided into two phases (i.e. the light colored Phase 1 and dark colored Phase 2) for each group. Therefore, the four input ports in Group 1 can emulate an OQ switch in Phase 1 with speedup of $(2 - \frac{1}{4})$, while the left four input ports in Group 2 can also emulate an OQ switch in Phase 2 with speedup of $(2 - \frac{1}{4})$. Then the total speedup for the MSM Clos network emulating an OQ switch is $(2 - \frac{1}{4}) + (2 - \frac{1}{4}) = 2 \times (2 - \frac{1}{4})$. The proof of any case of $m < n$ is a straight forward extension of the example with $n = 4$, $m = 2$, $r = 2$, where the speedup needed by an MSM Clos network is proportional to $\frac{n}{m}$ and $(2 - \frac{1}{mr})$. According to pigeon hole principle and Constraint Set Principle, with a speedup of $\frac{n}{m} \times (2 - \frac{1}{mr})$, an MSM Clos network can emulate an OQ switch. ■

What deserves more attention here is when the speedup $S$ is not an integer in MSM Clos network. We can use the smallest integer $\overline{S} = \lceil S \rceil$ bigger than $S$ as speedup in some time slots, while using $(\overline{S} - 1)$ as speedup in the other time slots in a circle. To be more precise, supposing $S = \overline{S} - \frac{F}{G} (F < G)$, let a circle length to be $G$ time slots. We adopt a

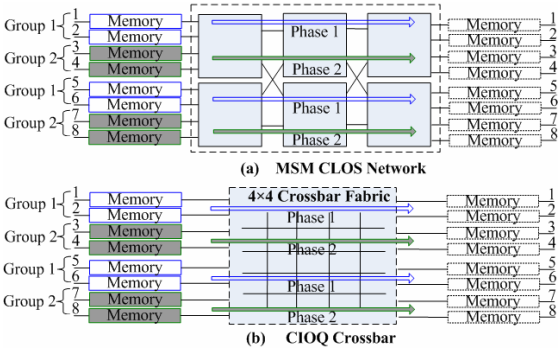


**Fig.3.** Emulating an OQ switch when $n = 4, m = 2, r = 2$

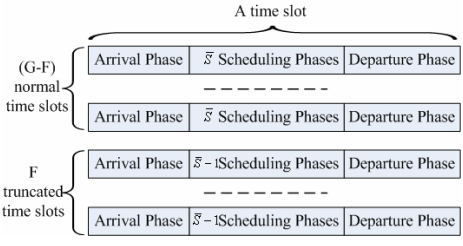


**Fig.4.** Normal time slots and truncated time slots in a circle

speedup of $\overline{S}$ in $G-F$ time slots and a speedup of $(\overline{S}-1)$ in $F$ time slots. Meanwhile, we call these $F$ time slots as truncated time slots. Fig.4 shows the normal time slots and truncated time slots in a circle.

## 5 Per-Input OQ Emulation (PIOE) by MSM Clos Network

The current cell insertion algorithms and matching algorithms have the following disadvantages in emulating an OQ switch:

(a) High Information complexity: In CIOQ emulation with an OQ switch, Shang-Tse Chuang had relied on the stable matching during the two scheduling phases and CCF (Critical Cell First) cell insertion policy during arriving phase in each time slot [1]. The stable matching needs to know the output priority list which relies on information of output order of all cells on the input side. The CCF algorithm needs to know the number of cells with a higher priority on the output side. Meanwhile, both the stable matching algorithm and the CCF cell insertion algorithm need to know the departure time of each cell, which is calculated based on the priority information of all cells in both input side and output side. Thus an input has to communicate with all the other inputs and outputs to obtain the information needed by each cell during each time slot, but this is too difficult to implement not only for real time information exchange, but also for modularization design, which never means to have so many extra communications beyond normal packet forwarding.

(b)High Computing complexity: The stable matching that CIOQ switch needs to find in each scheduling phase can take as many as $O(N^2)$ iterations, which is proved to be equivalent to the number of cells buffering on the input side[Gale and Shapely [13]. Unfortunately, as the increase of link speed as well as the switch port numbers, schedulers will have to make more decisions (i.e. there will be more iterations to guarantee a maximal matching and hence obtain high performance) in a more urgent time slot (i.e. the time will be only a fraction of the time slot in low link speed case). In this case, the matching algorithms available in switch with low link speed and less switch ports just can not be applied to new generation routers.

To overcome these disadvantages, we discuss a method for MSM CLOS switch to emulate an OQ switch with an acceptable or predictable performance degrading (i.e. emulation only based fairness of per input port). For example, the method (a) does not need information exchange among inputs and outputs of the switch, and thus eliminate information complexity; (b) has an algorithm complexity as low as O(1).

We observe that in an OQ switch cells are inserted into and scheduled from input queues based on the global priority principles (i.e. WFQ [15], Strict Priority [18], or FIFO among all cells from all input ports), and this leads to cell insertion algorithm and cell scheduling algorithm requiring global information (cell queueing states among all inputs and outputs). The emphasis on the priority order in an OQ switch, however, is almost meaningless for cells coming from different links and buffered in different input queues, because they have almost no relations with each other. Based on this fact, we present a method for MSM CLOS network to emulate an OQ switch based on per-input priority and fairness, which only uses information locally available on each input. We call this method Per-Input OQ Emulation (PIOE).

In the method of PIOE, we organize cell queueing in a way like Virtual Output Queueing (VOQ), but use a wide class of queueing policies such as WFQ and Strict Priority queueing in each VOQ queue of each separate input, just like an OQ switch does. The PIOE method comprises Per-VOQ Per-Priority based Cell Insertion Policy (PVPP-CIP), Cell Scheduling Policy (PVPP-CSP), and $S^3$ scheduling algorithm.

To describe more precisely, we shall give two definitions as follows:

**Definition 4. VOQ Queue** — *In each input, cells are buffered in different queues according to their output port number, and we denote them as* $Q_{ij}$ *, where* $i, j \in \{1,2,...,nr\}$ *for am MSM Clos networks* $C(n,r,m)$*. It is easy to know that there are* $nr$ *VOQ queues in an input port and* $(nr)^2$ *VOQ queues in all input ports.*

**Definition 5.** VOQ Priority Queue—In each VOQ queue, cells are buffered in different queues according to their priority number, and we denote them as $Q_{p_k}(k = 1,2,...,K)$, where $K$ is the number of priorities supported by routers.

### 5.1 PVPP Cell Insertion Policy (CIP) of PIOE

PVPP Cell Insertion Policy: Supposing that cell $X_{ijk}$ arrives at input port $i$ and is destined for output port $j$ and has a priority number $k$. Upon arrival $X_{ijk}$ is inserted to

the end of VOQ priority queue $Q_{p_k}$ of the VOQ queue $Q_{ij}$. Because cells are inserted into each input port based on Per VOQ and Per Priority, we call this cell insertion policy PVPP-CIP.

### 5.2 PVPP Cell Scheduling Policy (CSP) of PIOE

PVPP Cell Scheduling Policy: Supposing that $X_{ijk}$ represents cells buffered in VOQ priority queue $Q_{p_k}$ of the VOQ queue $Q_{ij}$. Upon departure cell $X_{ijk}$ is exported in sequence of VOQ queues and VOQ priority queues. Because cells are scheduled out of each input port based on Per VOQ and Per Priority, we call this cell scheduling policy PVPP-CSP.

PVPP-CIP & -CSP of PIOE are shown in Fig.5. In the left dashed frame, PVPP-CIP is composed of two phases: one is the VOQ dispatcher, where cells are classified into each VOQ queue according to their output port number; and the other is the Priority dispatchers, where cells are classified into each VOQ priority queue according to their priority number. In the right dashed frame, PVPP-CSP is also composed of two phases: one is the priority schedulers to export cells from different VOQ priority queues; the other is the VOQ scheduler, where cells from different VOQ queues are exported to the output ports.

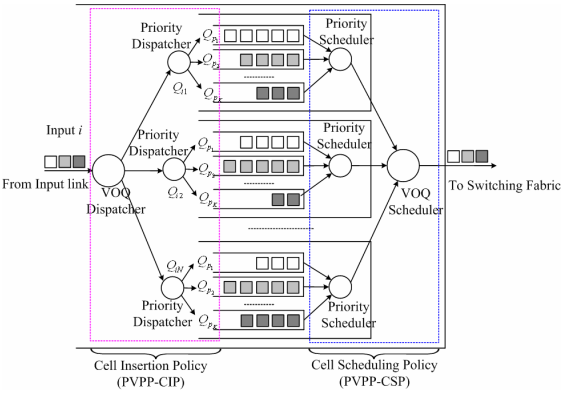


**Fig. 5.** PVPP Cell Insertion Policy and Cell Scheduling Policy of PIOE

### 5.3 Queueing Principle Analysis of PVPP-CIP and –CSP

(a) FIFO queueing principle emulated by PVPP-CIP and –CSP: In FIFO queueing principle, it does not need classify cells based on priorities; therefore the priority dispatche and the priority scheduler only maintain one priority queue (i.e. the highest priority queue $Q_{p_K}$). Thus each priority dispatcher just writes cells to the end of $Q_{p_K}$, while each priority scheduler just reads cells from head of $Q_{p_K}$.

(b) Strict Priority and WFQ queueing principle emulated by PVPP-CIP and –CSP: These two kinds of queueing principles are sensitive to cell's priority. In PVPP-CSP,

cells are written to priority queues by each priority dispatcher, which is similar for both Strict Priority and WFQ. In PVPP-CSP, however, priority schedulers read cells in different ways. Cells in highest priority queues that are not empty are prior to cells in lower priority queues in Strict Priority queueing principle. While in WFQ queueing principle, each priority queue is endowed with a weight value $w_k (k = 1,2,...,K)$, and the scheduling times in each scheduling circle are proportional to each $w_k$.

### 5.4 $S^3$ Matching Algorithmin PIOE

To overcome the high computing complexity of stable matching in [13] (i.e. their solution has a complexity of $O(N^2)$), and overcome the high information complexity of *GBVOQ* in [1] (i.e. it needs global state information), we design a simple and practical matching algorithm based on per input port fairness.

**Definition 6.** Vertex Matching—If a pair of vertices belongs to $V_0$ and $V_4$ of Clos network, respectively, we call the vertex pair a vertex matching, and denote it as $M(v_i^0, v_j^4)$, where $i, j \in (1,2,...,nr)$.

There are $(nr)^2$ pairs of vertex matching altogether in $C(n,r,m)$; we can further divide them into $nr$ incompatible groups, each of which includes all input vertices and all output vertices of $C(n,r,m)$. We call each group a Stable Vertex Matching (*SVM*), and further call all $nr$ groups the *Comple SVM Set* (*CSS*) for including all possible $(nr)^2$ pairs of vertex matching. The definitions of *SVM* and *CSS* are as follows:

$$SVM_i = \{M(v_1^0, v_{(i) \bmod (nr)}^4), M(v_2^0, v_{(1+i) \bmod (nr)}^4), ..., M(v_{nr}^0, v_{(nr-1+i) \bmod (nr)}^4)\}, i = (1,2,...,nr)$$

From Lemmon 2, we can always find a stable path set $S\hat{P}S$ for each *SVM* in *SPTS* of $C(n,r,m)$, and therefore can find $nr$ $SPS_i$ $(i = 1,2,...,nr)$ for all *SVM* in *CSS*.

**Definition 7.** Mapping Table from *SVM* to *SPS*—In non-blocking Clos network, we call each pair of each *SVM* to each *SPS* a mapping table and denote it as $M_{(SVM,SPS)}$.

$$M_{(SVM,SPS)} = \{(SVM_i, SPS_i), i = 1,2,...,nr\}$$

Based on the mapping table our matching algorithm of PIOE is as follows:

*Stable SVM SPS* ($S^3$) **matching algorithm:** During $i^{th}$ of matching of PIOE, we choose $(SVM_{i \bmod nr}, SPS_{i \bmod nr})$ in $M_{(SVM,SPS)}$ as matching of inputs and outputs.

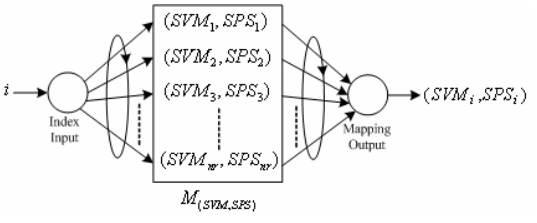


**Fig. 6.** $S^3$ matching algorithm of PIOE

Obviously, we can see there are two properties for $S^3$ matching algorithm: one is low computing complexity of $O(1)$, for just requiring one lookup in mapping table; the other is absolute fairness among all inputs and outputs, for in a round of $nr$ matching, all pairs of input and output appear once and only once. That is to say, all inputs are absolutely fair in $S^3$ matching algorithm.

## 6 Conclusions

We have studied MSM Clos packet switching network in this paper. Firstly, we discussed the modeling method of MSM Clos network based on graph theory, and put forward stable path set and non-blocking property of Clos network (Lemmon 1 and Lemmon 2). Based on the similarity of single stage CIOQ Crossbar and multistage MSM Clos network, we discussed the condition for them to mimic each other (Lemmon 3). Then we extent the results in CIOQ Crossbar emulation an OQ switch to MSM Clos networks (Theorem 1). To overcome the disadvantages of cell insertion algorithm and matching algorithm in CIOQ Crossbar emulating an OQ switch, we provided a PIOE method with PVPP-CIP & -CSP with low information complexity and $S^3$ matching algorithm with computing complexity of $O(1)$.

## Acknowledgements

This work was supported in part by the grants from National Basic Research Program of China (973 Program) with No.2007CB307102 and National Hi-tech Research and Development Program of China (863 program) with No.2005AA121210. We thank several members of Information Engineering Institute for their technical suggestions, including Peng Yi, Yufeng Li and Yang Li, and the anonymous reviewers for their constructive comments and suggestions.

## References

1. Chuang, S.T., Awadallah, A., McKeown, N., Prabhakar, B.: Matching output queueing with a combined input and output queued switch. IEEE Journal on Selected Areas in Communicaions 17, 1030–1039 (1999)
2. Chao, H.J.: Next generation routers. IEEE Proceeding 90(9), 1518–1558 (2002)
3. McKeown, N.: The iSLIP Scheduling Algorithm for Input-Queued Switches. IEEE/ACM Trans. on Networking 7(2) (1999)
4. McKeown, N., Anntharam, V., Walrand, J.: Achieving 100% Throughput in an input-queued switch. In: Infocom 1996 (1996)
5. Wang, F., Hamdi, M.: Analysis on the Central-stage Buffered Clos-network for packet switching. In: IEEE International Conference on Communications (2005)
6. Clos, C.: A Study of Non-Blocking Switching Networks. Bell Systems Technical Journal, 406–424 (1953)
7. Tsai, K.H., wang, D.W.: Lower Bounds for Wide-sense Non-Blocking Clos Network. In: Taipei 1998. Computing and Combinatorics, Springer, Berlin, pp. 213–218 (1998)

8. Lee, T.T., To, P.P.: Non-Blocking Routing Properties of Clos Networks. In: Advances in switching networks, Amer. Math. Soc., Providence, RI, pp. 181–195 (1998)
9. Lin, G.H., Du, D.Z., Wu, W., Yoo, K.: On 3-Rate Rearrangeability of Clos Networks. In: Advances in switching networks, Amer. Math. Soc., Providence, RI, Princeton, NJ, pp. 315–333 (1998)
10. Lee, T.T., Lam, C.H.: Path Switching - A Quasi-Static Routing Scheme for Large-Scale ATM Packet Switches. IEEE J. Select. Areas Communications. 15, 914–924 (2002)
11. Pun, K., Hamdi, M.: Distro: A Distributed Static Round-Robin Scheduling Algorithm for Bufferless Clos-Network Switches. IEEE GLOBECOM (2002)
12. Chao, H.J., Deng, K-L., Jing, Z.: A Petabit Photonic Packet Switch (P3S). IEEE INFOCOM 2003 (2003)
13. Gale, D., Shapley, L.S.: College Admissions and the Stability of Marriage. American Mathematical Monthly 69, 9–15 (1962)
14. Iyer, S., Awadallah, A., McKeown, N.: Analysis of a Packet Switch with Memories Running Slower than the Line Rate. In: IEEE Infocom 2000 (2000)
15. Demers, A., Keshav, S., Shenker, S.: Analysis and Simulation of a Fair Queueing Algorithm. J. Internetworking: Research and Experience, 3–26 (1990)
16. Parekh, A., Gallager, R.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. IEEE/ACM Trans. Networking 1, 344–357 (1993)
17. Zhang, L.: Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks. ACM Trans. Comput. Syst. 9(2), 101–124 (1990)
18. Shreedhar, M., Varghese, G.: Efficient Fair Queueing Using Deficit Round Robin. In: Proc. ACM SIGCOMM, pp. 231–242. ACM Press, New York (1995)