Space-Memory-Memory Architecture for Clos-Network Packet Switches

Xin Li, Zhen Zhou and Mounir Hamdi
Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Hong Kong
Email: {lixin, cszz, hamdi}@cs.ust.hk

***Abstract*— A Clos-network architecture is an attractive alternative for constructing scalable packet switches because of its distributed and modular design. It can be classified according to different buffer (memory) allocation schemes in its switching stages. The most studied architectures are the *Space-Space-Space (S^3)* and the *Memory-Space-Memory (MSM)* architectures. This paper shows that these two architectures cannot achieve efficient throughput with conventional random dispatching schemes even if the fabric itself is non-blocking. Previous research hence has focused on developing intelligent scheduling algorithms for these architectures so as to improve their throughput. However, we face various challenging problems when we try to actually implement the algorithms. The problems include performance degradation under constrained arbitration time, needs of centralized or complex scheduler hardware, etc. To solve these problems and at the same time be practical, this paper proposes a novel *Space-Memory-Memory (SMM)* architecture that does not need any schedulers. The SMM architecture has similar hardware complexity as the MSM architecture has, while been proven to achieve 100% throughput under any admissible traffic. Our queuing analysis demonstrates that only small size buffers are needed in the central stage. The only tradeoff for the proposed SMM architecture is to employ small extra resequencing buffers. As a result, the SMM architecture can achieve very high performance, and is readily implemented using current technology.**

I. Introduction

The Clos network architecture [1] was first proposed by C. Clos in the 1950s, for use in telecommunications networks. The combinatorial properties of this multi-stage interconnected network help to construct strict non-blocking circuit switches with fewer cross-points. With the development of the Internet, data network emerges and flourishes in recent years. The switching technique has also changed to packet switching. The technology evolution, however, does not diminish the practical usage of this architecture. Because of the distributed and modularized property, the three-stage Clos network is still a favored candidate in constructing high performance scalable packet switches.

Clos-netowrk packet switch can be classified based on their buffer (memory) allocation schemes. For example, the simplest Clos-network fabric has no buffers at any stage. Since the switching is done purely in space for all three stages, this architecture is normally quoted as *Space-Space-Space (S^3)* (or *bufferless*) architecture. Chiussi *et al.* [2] proposed a *Memory-Space-Memory (MSM)* architecture. MSM has buffered input and output stages and a bufferless central stage. Although there are other possible buffer allocation schemes (i.e., a fully buffered *Memory-Memory-Memory (MMM)* architecture), most existing researches on Clos-netowrk packet switch are on these two architectures. Their focus is to develop good scheduling algorithms for the architectures. For example, Distro [3] is for S^3 architecture, CRRD/CMSD [4] and MWMD [5] are for MSM architecture.

In fact, the way of allocating the buffers is an important design consideration and it influences the switching performance. S^3 is favored for hardware simplicity. However pure space switching in all three stages increases the cell contention probability. We show that the throughput of S^3 with random dispatching may be as low as 39.7%. MSM tries to enhance the performance of S^3 by adding buffers to input and output stages. The buffers relieves the cell contentions in these two stages, but MSM tends to keep a bufferless central stage in fear the of out-of-sequence problem. This leaves some unsolved contentions. The throughput of MSM with random dispatching is increased but not satisfactory. The worst case throughput is 63% and is still not a satisfied result.

Obviously, MMM architecture does not suffer any throughput degradation since all possible contentions are absorbed by buffers, but it is expensive to be implemented. It is worth asking whether there exists less complex architecture which performs as well as MMM. Observing that the main effect of buffers is to resolve contentions, buffers can be removed if there is no contention at all. This paper proposes a *Space-Memory-Memory (SMM)* architecture to simulate MMM. A desynchronized static round-robin (DSRR) connection pattern is set in bufferless input stage switching modules and guarantees zero cell contention. SMM architecture is proved to be of 100% throughput under any admissible traffic. It needs no scheduler and is practical to be implemented. In response to the out-of-sequence problem, queueing analysis is conducted for output queues of the central stage. The result shows the average queue length is very small (less than 1 with 1.17 expansion ratio under any admissible traffic). This suggests: 1) The packet drop rate can be kept low with small central-stage buffers, and 2) only small resequencing buffers are needed.

The remainder of this paper is organized as follows. Section II describes the Clos-network switch model used

This work was supported in part by a grant from Hong Kong Research Grant Council (Grant Number: RGC HKUST6200/02E).

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

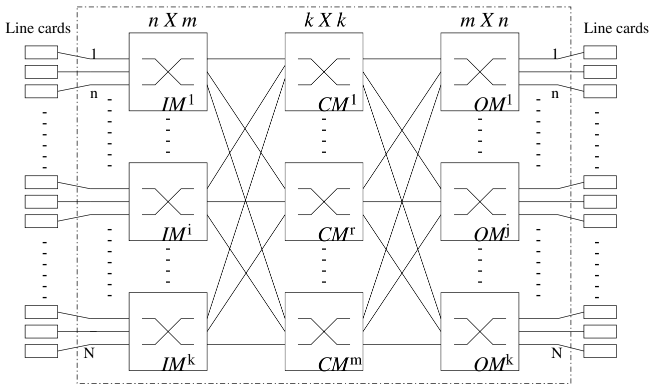

Fig. 1. Three-stage Clos-network packet switch

throughout this paper. Section III compares the throughput of S^3, MSM and MMM architecture under random dispatching. Section IV proposes SMM architecture. Section V investigates the performance of SMM, including its throughput and queue analysis. Section VI concludes the paper.

II. Clos-Network Switch Model

A Clos-network packet switch consists of three stages of switching modules, denoted by $C(n, m, k)$, as shown in Fig. 1. It has k input modules (IM) of size $n \times m$, m central modules (CM) of size $k \times k$, and k output modules (OM) of size $m \times n$. The inputs of the IMs and outputs of the OMs connect to line cards. The switching modules of the neighboring stages are connected to each other by one and only one inter-stage link. As a whole, the switch has $N = nk$ input(output) ports.

When a packet arrives at the switch, it is first segmented into fixed size *cells*. All cells are temporarily stored in the input port cards before they are switched out. A common queuing strategy is virtual output queuing (VOQ). For any input port, it has N queues for cells destined to every output port. There are totally N^2 VOQs equipped for the switch.

The key notations used in this paper are listed as follows.

IM^i	i^{th} input module, where $1 \leq i \leq k$
CM^r	r^{th} central module, where $1 \leq r \leq m$
OM^j	j^{th} output module, where $1 \leq j \leq k$
$IL(i, r)$	inter-stage link connects IM^i and CM^r
$OL(r, j)$	inter-stage link connects CM^r and OM^j
$IP(i, g)$	the $(i * n + g)^{th}$ input port of the switch, where $1 \leq g \leq n$
$OP(j, h)$	$(j * n + h)^{th}$ output port of the switch, where $1 \leq h \leq n$
$IM_I^i(g)$	g^{th} input of IM^i, connects to $IP(i, g)$
$IM_O^i(r)$	r^{th} output of IM^i, connects to $IL(i, r)$
$CM_I^r(i)$	i^{th} input of CM^r, connects to $IL(i, r)$
$CM_O^r(j)$	j^{th} output of CM^r, connects to $OL(r, j)$
$OM_I^j(r)$	r^{th} input of OM^j, connects to $OL(r, j)$
$OM_O^j(h)$	h^{th} output of OM^j, connects to $OP(j, h)$
VOQ_{igjh}	VOQ that holds cells from $IP(i, g)$ to $OP(j, h)$

The inter-stage links work at the same speed as the external line rate, denoted as R. The total switching capacity of $C(n, m, k)$ is kmR. The total traffic load is knR. The fabric speedup S can be expressed as the ratio of the total switching capacity over the total traffic load, as in this case, $S = m/n$. Since this is a spacial speedup (related to the number of CMs), it is also called *expansion ratio*.

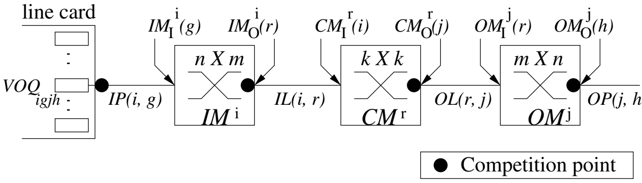

Fig. 2. The possible competition points in S^3 architecture

Under the packet switching environment, we may observe some features of Clos-network fabric, namely,

- *Non-blocking with an expansion ratio no less than 1*: Clos-network fabric are expected to be as non-blocking as crossbars. Because fabric connections are reset in every time slot in packet switching, if Clos-network fabric satisfies the "rearrangeable non-blocking" requirement in circuit switching, it is non-blocking in packet switching too. Benes [6] proved that the non-blocking condition is $m \geq n$, this is equivalent to an expansion ratio no less than 1. This paper always assumes a non-blocking fabric.
- *Self-routing from CM to OM*: A scheduling algorithm can switch a cell out via any central module. However after the cell arrives at a certain CM, it has to follow one determined inter-stage link to its destination. For example, a cell to $OP(j, h)$ is dispatched to CM^r, then it has to go through link $OL(r, j)$ to arrive there. That is, cells are self-routed from central modules to output modules based on their destinations. This property ensures, when buffers exist in the outputs of CMs and OMs, the scheduler can be totally avoided in these modules.

III. Throughput Analysis for Different Clos-network Packet Switch Architectures

The architectural choice of Clos-network packet switch influences the switching performance. This section analyzes the throughput of three Clos-network architectures, namely, *Space-Space-Space (S^3)*, *Memory-Space-Memory (MSM)* and *Memory-Memory-Memory (MMM)* architecture.

A. Space-Space-Space (S^3) Architecture

S^3 architecture is favored in [3] because of the reduced hardware complexity. However, the cost is that cells have to compete drastically to be switched out. As an example, let a cell from $IP(i, g)$ head for $OP(j, h)$. Along its path, it constantly suffers contention, and it must compete against other cells in order to get to the output. Fig. 2 shows all possible competition points along the cell path.

The first competition happens at the input port $IP(i, g)$, for accessing the fabric. There are N VOQs at the input port, but only one cell can be sent to the fabric in each time slot. If the winner is selected randomly, a cell wins this competition at the probability $Pr_1 = \frac{1}{N}$.

After the cell enters IM^i, it may compete with other incoming cells in order to go further to a CM. Suppose a cell randomly selects a CM to go (that is, requests a random output of IM^i), and the output randomly grants one of the received requests, the probability that cell gains access to a particular central module, say CM^r, is $Pr_2 = \frac{1}{n}$.

Now the cell arrives at CM^r. it has to gain the access to output $CM_O^r(j)$, because its destination is the output port $OP(j,h)$ in OM^j, . For other cells at CM^r destined to any output port in OM^j, they compete for $CM_O^r(j)$ too. Things are getting complicated in this competition since the number of competitors varies and depends on the traffic situation. In general, let us assume it is a game of X players (that is, cells). Players are identical and independent. Thus, the probability that another cell competes with the cell for $CM_O^r(j)$ equals to $\frac{1}{X}$. The probability that there exists x other cells competing with the first cell is given by $\left(1-\frac{1}{X}\right)^{X-1-x}\left(\frac{1}{X}\right)^x$, where x takes value from 0 to $X-1$ inclusively. If the winner is selected randomly from among the cell and its competitors, define $f(X)$ the probability of the cell winning which is,

$$f(X) = \sum_{x=0}^{X-1} \binom{X-1}{X-1-x} \left[\left(1-\frac{1}{X}\right)^{X-1-x}\left(\frac{1}{X}\right)^x\left(\frac{1}{x+1}\right)\right].$$

In our switch model, there are in total k cells, one from each input of CM^r. So the probability that the cell can further reach OM^j is $Pr_3 = f(k)$.

Finally, the cell at output module OM^j has to gain the output port $OP(j,h)$, to be switched out. The situation here is similar to what happened in the central modules. All cells from the inputs of OM^j are potential competitors. This time, there are m players, one from each input of OM^j. Therefore, the probability that the cell wins is $Pr_4 = f(m)$.

The overall probability that a cell successfully passes through the arduous journey from input port $IP(i,g)$ to output port $OP(j,h)$ via the central module CM^r, winning all four competitions is,

$$Pr_{pass} = \prod_{i=1}^{4} Pr_i = \frac{1}{N} \cdot \frac{1}{n} \cdot f(k) \cdot f(m). \qquad (1)$$

Equation (1) holds for any cell on any path.

Throughout the above analysis, we assume the cell randomly selects a central module to go through, and the switching modules execute random scheduling. This strategy is normally referred to as *random dispatching (RD)*. It is a default scheduling algorithm used for all analysis in Section III.

1) Throughput of S^3 architecture under RD: It is possible to derive the throughput of output port $OP(j,h)$, $t_{OP(j,h)}$.

$$\begin{aligned} t_{OP(j,h)} &= \sum_i \sum_g \sum_r Pr_{pass} \\ &= \frac{knm}{Nn} f(k) f(m) = \frac{m}{n} f(k) f(m). \qquad (2) \end{aligned}$$

Note that the maximum throughput of the output port $T_{max} = \min\{t_{OP(j,h)}, 1\}$.

We can derive the switch throughput from $t_{OP(j,h)}$ as it is the same for all output ports. When the expansion ratio is 1.0 (that is, $m = n$), the maximum throughput is a function of k and m. Oki *et al.* [4] showed when $X \to \infty$, $f(X)$ approaches $1 - 1/e$. When $k, m \to \infty$, the maximum switch throughput $T_{max} \to (1-1/e)^2 \approx 39.7\%$. To achieve 100% throughput using RD, the expansion ratio should be larger than 2.5.

B. Memory-Space-Memory (MSM) Architecture

The throughput of S^3 architecture is low and not satisfiable. The reason is obvious, pure space switching introduces too much competition between cells and cripples the overall system's performance. A reasonable solution is to introduce buffers (memories) into the switch fabric to resolve the contentions. MSM architecture is one example of this.

1) Buffer allocation scheme: In MSM architecture, both the input and output stages contain buffers. Input module buffers are organized as shared VOQs. A shared VOQ with speedup n in IM accepts all cells from IM to a particular output port. It is called a "shared" VOQ because all inputs of the IM can send cells to it. For an $N \times N$ switch, there are N shared VOQs for each input module. Output module buffers are organized as output queues (OQs). An OQ in OM accepts cells from all inputs of the OM who are destined to a common switch output port. There are n OQs in each OM, each with speedup m.

2) Throughput of MSM architecture under RD: With the help of shared VOQs in IMs and OQs in CMs, contentions in Clos-network fabric are partially relieved. Compared with the four competition points in S^3 architecture, there are only two such points in MSM architecture.

- The N shared VOQs of a input module compete for the access to the central modules. Using the RD scheme, the probability that a shared VOQ can send a cell is $1/N$.
- When the cell arrives the bufferless central module, the situation is the same as in S^3 architecture. The cell can reach the desired output module at probability $f(k)$.

There is no competition in OMs since they are output-queued. Detailed analysis in [4] shows the switch throughput for MSM architecture is $\frac{m}{n} \times f(k)$. When the expansion ratio is 1.0 (that is, $m = n$) and $k \to \infty$, the switch throughput tends to be 63%. 100% throughput can be achieved when the expansion ratio equals to 1.58.

C. Memory-Memory-Memory (MMM) architecture

The existence of memory in switching modules is shown to be helpful in improving the performance of the switch. Let us find out the switch performance when all the switching modules are equipped with memories, i.e., a *Memory-Memory-Memory (MMM)* architecture.

1) Memory allocation scheme: In MMM, the buffer allocation of the input and output modules is the same as that in MSM architecture. MMM has extra buffers in central modules, which are organized as OQs. The OQs work at speed kR (the memory speedup is k). Contentions in central modules are now totally avoided.

2) Throughput of MMM architecture under RD: The contentions now only happen in IMs, and the switch throughput equals to $\min\left\{\frac{m}{n}, 1\right\}$. An expansion ratio of 1.0 is already suffice to ensure 100% throughput in MMM architecture.

TABLE I
COMPARISION OF S^3, MSM, MMM AND SMM ARCHITECTURE

	S^3	MSM	SMM	MMM
Input stage	bufferless	shared VOQs speedup = n	bufferless	shared VOQs
Central stage	bufferless	bufferless	OQ speedup = m	OQ speedup = m
Output stage	bufferless	OQ speedup = m		

IV. SPACE-MEMORY-MEMORY (SMM) ARCHITECTURE

This section presents the Space-Memory-Memory (SMM) architecture. By using desynchronized static round-robin (DSRR) connection pattern in the first stage, SMM can simulate MMM with less hardware complexity.

A. *Desynchronized Static Round-Robin (DSRR) Connection Pattern in IMs*

DSRR is run distributedly and independently by each IM. Any input sequentially connects to all outputs in a round-robin manner; at each time slot, inputs map injectively to outputs. This can be achieved by setting a fully desynchronized initial connection pattern. Take 3×3 switch as an example. $(1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3)$ can be a possible initial connection pattern. At the following time slots, the connection between inputs and outputs are $(1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 1)$, $(1 \mapsto 3, 2 \mapsto 1, 3 \mapsto 2)$, $(1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3)$, $\ldots$ and so on. In fact, DSRR is not a scheduling algorithm in the common sense. No dedicated scheduler hardware are needed for it.

IMs with DSRR connection scheme guarantees no cell contention in input stages when the fabric is non-blocking (i.e., $m \geq n$). This is because at each time slot, there is at most one cell arrives at the IM input, and it is immediately transferred to an IM output according to the connection pattern. Cells arriving at the IM are dispatched to different CMs within a time slot.

B. *Space-Memory-Memory (SMM) Architecture*

Zero contention needs zero buffers. So MMM can be reduced to a *Space-Memory-Memory (SMM)* architecture without performance degradation. SMM architecture consists of a bufferless input stage, output-queued central and output stages. Note that SMM does not actually need a scheduler in any module (static connection pattern in input stages and self-routing in other stages).

Table I summarizes the buffer allocation scheme for S^3, MSM, MMM and SMM architectures. From the hardware perspective, S^3 is the simplest and MMM is the most complex one. MSM and SMM architectures are of medium hardware complexity. When $m = n$, MSM and SMM consist of exactly the same switching modules.

V. PERFORMANCE ANALYSIS OF THE SMM ARCHITECTURE

A. 100% *Throughput Under Any Admissible Traffic*

The traffic to the Clos-network switch can be represented by a traffic matrix $A = [a_{igjh}]_{N \times N}$, where $1 \leq i, j \leq k$, $1 \leq g, h \leq n$. a_{igjh} is the traffic arrival rate from $IP(i,g)$ to $OP(j,h)$. The traffic is *admissible* if it satisfies,

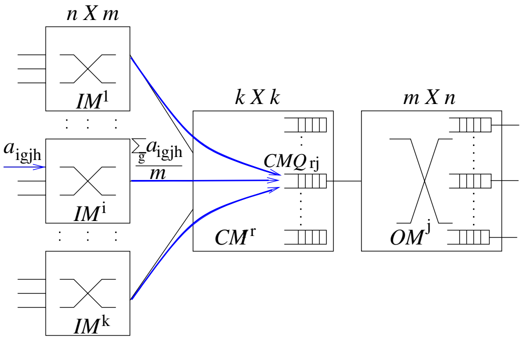

Fig. 3. Traffic arrival rate to CMQ

$$\forall i\ \forall g,\ \sum_j \sum_h a_{igjh} \leq 1 \text{ and } \forall j\ \forall h,\ \sum_i \sum_g a_{igjh} \leq 1.$$

In other words, no input or output is oversubscribed.

100% throughput means, every cell can be switched out within finite time slots. The cell delay equals to the sum of the cell waiting time in the OQs of the CM and the OM. That is to say, if the length of the OQs in both types of modules is finite, the switch is *stable*.

Let us first focus on the OQs in the central modules. For every input port of the Clos-network switch, the DSRR scheme evenly distributes its incoming traffic to the central modules. So any OQ in the CMs receives exactly $1/m$ of the traffic that is heading to its connected output module, as shown in Fig. 3. Take the OQ that resides on the j^{th} output of CM^r as an example, denote it as CMQ_{rj}. The traffic arrival rate to this queue is $\lambda_{CMQ_{rj}} = \sum_i \sum_g \sum_h a_{igjh}/m \leq \frac{n}{m} \leq 1$. Since $OL(r,j)$ is guaranteed to transmit one cell at a time, the service rate $\mu_{CMQ_{rj}} = 1$, which is no less than the arrival rate. It follows that CMQ_{rj} is of finite length for any r or j.

An output module OQ receives traffic from various input ports to one particular output port. The service rate is 1. Admissible traffic condition ensures any OQ in the OMs has finite length. To summarize, SMM with an expansion ratio larger than or equal to 1 achieves 100% throughput under any admissible traffic.

B. *Analysis for Central Module OQ Behavior*

In this section, we take a closer look at the central module OQ's behavior. This is of great importance because we need to determine the required buffer size. In addition, the existence of the buffers in the central modules causes the cell out-of-sequence problem. A behavioral study may also help to determine the size of the resequencing buffer.

The most common approach is to approximate the traffic arrival as independent Poisson process. Cell arrivals from $IP(i,g)$ form a Poisson process of rate λ_{ig} and choose a destination $OP(j,h)$ with probability p_{igjh}.[1] Because the liner combination of independent Poisson process is still Poisson, cell arrivals to a central module input $CM_I^r(i)$ is a Poisson

[1] Poisson traffic is a special case of the general traffic in Section V-A. a_{igjh} now equals to $p_{igjh}\lambda_{ig}$.

process of rate $\left(\sum_g \lambda_{ig}\right)/m$. This further induces that the traffic to CMQ_{rj} (CMQ at the j^{th} output of CM^r) is also a Poisson process of rate

$$\lambda_{CMQ_{rj}} = \sum_i \left(\sum_g \frac{\lambda_{ig}}{m} \cdot \sum_h \sum_g \frac{\lambda_{ig} \cdot p_{igjh}}{m} \right). \qquad (3)$$

The service rate of CMQ_{rj} is $\mu_{CMQ_{rj}} = 1$. This implies that it behaves like an M/D/1 queue. Assume ρ is the ratio of the arrival rate over the service rate, the average number of the cells waiting in the M/D/1 queue is determined by a function $g(\rho) = \rho^2/2(1-\rho)$ given by queuing theory. In our case, since the service rate is 1, it also describes the average length of CMQ_{rj}, denoted by $L_{CMQ_{rj}}$. That is,

$$L_{CMQ_{rj}} = g(\lambda_{CMQ_{rj}}). \qquad (4)$$

Case 1. uniform traffic: Under uniform traffic, $\lambda = \lambda_{ig}$, $p_{igjh} = 1/N$, from Equation (3), we have

$$\lambda_{CMQ_{rj}} = \left(\frac{n}{m}\right)^2 \lambda^2.$$

Case 2. nonuniform traffic: We use the same model as the one given in [4]. $\lambda = \lambda_{ig}$ is the traffic load, and p_{igjh} is given by

$$p_{igjh} = \begin{cases} \omega + \frac{1-\omega}{N} & \text{if } i = j \text{ and } g = h \\ \frac{1-\omega}{N} & \text{otherwise} \end{cases}$$

where $N = nk$ is the switch size, ω is the unbalanced parameter. When $\omega = 0$, the traffic is uniform. On the other hand, traffic with $\omega = 1$ is completed unbalanced. Now, we can calculate the rate of nonuniform traffic

$$\begin{aligned} \lambda_{CMQ_{rj}} &= \sum_i \left(\sum_g \frac{\lambda}{m} \cdot \sum_h \sum_g \frac{\lambda \cdot p_{igjh}}{m} \right) \\ &= \frac{n\lambda^2}{m^2} \sum_i \sum_h \sum_g p_{igjh} \\ &= \frac{n\lambda^2}{m^2} \left[(N-1)n\frac{1-\omega}{N} + n\left(\omega + \frac{1-\omega}{N}\right) \right] \\ &= \left(\frac{n}{m}\right)^2 \lambda^2. \end{aligned}$$

From the above two cases, we observe that the behavior of CMQs is only affected by the system load and the expansion ratio, not the traffic condition. Therefore, CMQs have same average queue length

$$L_{CMQ} = g\left(\left(\frac{n}{m}\right)^2 \lambda^2 \right). \qquad (5)$$

From equation (5), if SMM is of expansion ratio 1, $L_{CMQ} < 1$ for traffic load lower than 0.855. $L_{CMQ} \approx 49$ when the traffic load is 0.99. Alternatively if a small expansion ratio is allowed, i.e. $m/n \geq 1.17$, $L_{CMQ} < 1$ for all admissible traffic loads. The result can be further adjusted for real traffic, but the difference is not much. This suggests that it is sufficient to allocate a small size CMQ in the central modules. In general, cells arrive at the CMQs, wait there and then go to the output modules. In a few cases, some CMQ are full when the cell comes and it simply drops the cell, but this happens with very low probability as shown above. Resequencing buffer is needed at the output port. Its size is upper-bounded by a small number, which is $2m|\text{CMQ}|$.

C. Practical Advantages of SMM Architecture

In Section III, we mentioned that when random dispatching is used, S^3 and MSM with expansion ratio 1 have relatively low throughput, even if they are theoretically non-blocking. Using some smart algorithms, i.e. Distro [3] for S^3 architecture and CRRD [4] for MSM architecture, the throughput can be 100% under uniform traffic. With more complicated algorithms, i.e. MWMD [5], MSM has 100% throughput under any admissible traffic. This implies that good performance of S^3 and MSM needs the help of an intelligent scheduler. However problems may occur when implementing them. Distro is a centralized scheduler which does not quite fit into the distributed architecture. CRRD consists of several iterations when it does the arbitration. If the number of iterations is reduced due to the limitation of the arbitration time, the performance may degenerate. MWMD is too complicated to be practical. MMM architecture does not have this "scheduler" problem, but at the cost of more hardware requirements. In contrast, SMM architecture is proved to achieve 100% throughput under the same expansion ratio, with acceptable hardware complexity. No scheduler is needed, which is suitable for a high speed, scalable system. The only tradeoff is a resequencing buffer, and it is shown to be of small size.

VI. Conclusion

A Clos-network switch architecture is attractive to construct scalable, high performance packet switches. There are several choices of different memory allocation schemes in switching stages of the Clos-network fabric. This paper analyzes the throughput performance for the existing S^3 and MSM architecture under random dispatching. The results show their performance are inefficient when there is no intelligent scheduler to resolve the cell contentions. MMM architecture uses fully buffered stages to achieve better performance. The above move us to propose a SMM architecture, with DSRR connection pattern in the first stage. The SMM is shown to achieve 100% throughput under any admissible traffic. No scheduler is needed in SMM; therefore, it is practical to be implemented. Analysis of the behaviour of the output queues in the central modules is also given. The result shows the actual required buffer size in the central modules is small. This suggests the only extra cost of SMS architecture, resequencing buffer, is small too.

References

[1] C. Clos, *A Study of Non-Blocking Switching Networks*, Bell Sys. Tech. Jour., pp. 406-424, March 1953.

[2] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar, *Low-Cost Scalable Switching Solutions for Broadband Networking: the ATLANTA Architecture and Chipset*, IEEE Commun. Mag., pp. 44-53, vol. 35, issue 12, Dec 1997.

[3] K. Pun and M. Hamdi, *Distro: A Distributed Static Round-Robin Scheduling Algorithm for Bufferless Clos-Network Switches*, GLOBECOM '02, pp. 2298-2302, Vol. 3, Nov. 2002.

[4] E. Oki, Z. Jing, R. Rojas-Cessa, and J. chao, *Concurrent Round-Robin-Based Dispatching Schemes for Clos-Network Switches* , IEEE Trans. on Networking, pp. 830-844, Vol. 10, issue 6, Dec 2002.

[5] R. Rojas-Cessa, E. Oki, and J. chao, *Maximum Weight Matching Dispatching Scheme in Buffered Clos-Network Packet Switches*, ICC'04, pp. 1075-1079, vol. 2, June 2004.

[6] V. E. Benes, Mathematical Theory of Connecting Networks and Telephone Traffic, Academic Press, 1965.