First-Fit Scheduling for Multi-Stage Packet Switching Networks

Joseph Y. Hui
Lingie Li
Department of Electrical Engineering
Arizona State University, Tempe AZ 85287, USA
Email: jhui@asu.edu

***Abstract* - We first propose a First-Fit Scheduling algorithm for a single stage interconnection network. We then classify multi-stage interconnection network in space, such as the 3-stage Clos network, when each spatial stage may allow time domain switching. We examine how scheduling could be performed for these stages for various combinations of space-time switching.**

The First-Fit algorithm is then extended to a three stage Clos network with scheduling in the first and third stage. Similar to other scheduling algorithms, full loading is achievable, while randomization of input selection is replaced by the use of arrival timing. This distributed algorithm applies to both cases of fixed length versus variable length packets. We simulated the performance for both cases. A case is made that for multi-gigabit per second link speed, variable length packets should be used to simplify scheduling and avoid fragmentation, with acceptable increase in delay.

***Index Items* – Switching networks, packet switching, scheduling, Clos Networks.**

I. INTRODUCTION

Packet switching across a single Space (S) switch has been classified as Input Queued (IQ), Output Queued (OQ) [1-4], or Input Scheduled (IS) [5-6]. An IQ switch uses an input queue [2] for each input for which packet arrivals are First-Come-First-Served (FCFS) for each input queue (Fig 1a). An OQ switch (Fig 1c) allows multiple packets for the same output to be transmitted to that output simultaneously [1, 3], while an output queue is set up for transmission on the output. One implementation of an OQ switch uses a large shared memory [4, 19] for all output queues, with packets arriving at all inputs read into the shared memory.

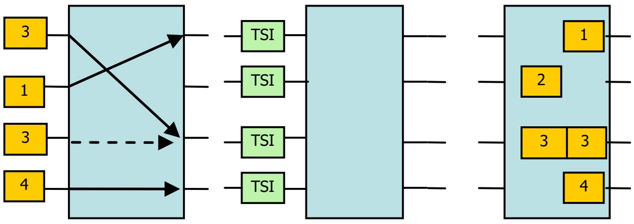

Fig 1a: IQ Switch **Fig 1b:**IS Switch **Fig 1c**:OQ Switch

An IS switch (Fig 1b), similar to an IQ switch, uses a simple crossbar switch and places an input queue at each input to the crossbar switch. Instead of FCFS for IQ switching, which creates the problem of Head-Of-Line (HOL) blocking, a IS switch allows scheduling of transmission of packets in the input queue without the FCFS requirement. In circuit switching terminology, a time slot interchanging (TSI) could be used to rearrange transmission at an input. Time slot interchanging is also used to avoid having multiple inputs transmitting to the same output simultaneously, which is termed an output conflict.

Connecting requests are often expressed as a mapping of $M = \{(i, o)\}$, the set of input-output connection requests. Scheduling reduces a many-to-many mapping (often modeled as a bipartite graph) M into multiple 1-to-1 maps. These 1-to-1 maps each represents the transmission within a time slot from each input in that 1-to-1 map to its corresponding output.

Both IS switching and OQ switching achieves 100% utilization of the inputs and outputs under the assumption of uniformity of traffic load for all inputs and outputs (i.e. traffic on inputs, as well as outputs, are exchangeable). However, OQ is simpler to implement than IS as no scheduling of packet transmission at the input is needed. No output conflict resolution is necessary as in IS as all conflicting requests are stored in a shared memory. Conflict resolution is achieved by the output queue.

Without the FCFS restriction, scheduling avoids HOL blocking and can achieve a throughput approaching 1. Many algorithms have been advocated [5-18], with the objectives of lowered complexity and improved fairness. Fairness is achieved with randomization of selection of inputs contending for the same output.

For IS, we advocate in this paper the use of arrival times for the purpose of selection randomization. Scheduling is performed at the time of arrival of a packet. For unslotted switching system, the arrival times of packets are distinct. We also advocate a first fit algorithm. For slotted switches, a packet request from input i to output j is placed in the first slot that is unscheduled to both the input and the output. For unslotted switch for which variable length packets arrive

Manuscript received April 30, 2009; revised Sept.12, 2009; accepted Nov. 17, 2009.

doi:10.4304/jcm.5.3.205-210

asynchronously, we find the first contiguous unscheduled interval that is large enough for each packet.

The advantages of this First-Fit (FF) algorithm are many. First, the use of packet arrival times for packet selection replaces the use of pseudo-random numbers or pointers, and more importantly, helps to maintain a FCFS discipline. (Nevertheless, larger packets may suffer as smaller packets often can find a first-fit interval earlier.) Second, this FF algorithm is amenable to distributed scheduling with inputs sending requests and outputs granting requests, both done asynchronously and independently for the inputs and outputs. Third, this algorithm allows for scheduling of variable length packets without fragmentation. As we shall argue later, fragmentation of packets into fixed length units such as ATM cells is no longer necessary or desirable for multi-gigabit transmission of large packets over the Internet. We advocate scheduling of large and variable length packets instead of small and fixed length ATM cells, thereby reducing processing and bandwidth overheads.

II. FIRST-FIT SCHEDULING

The input set is I and the output set is J. Consider input $i \in I$ sending a packet schedule request to output $j \in J$. Each input i and output j maintains a set A_i and A_j of available times respectively for the purpose of scheduling.

For unslotted and variable length packets, the availability set is a subset $A_i \subset [t, t+T]$, a continuous scheduling interval of duration T beyond the current time t. We often may assume very large window size to accommodate packet lengths that is very large.

For slotted switching systems, time is discrete and consequently the availability subset $A_i \subset [t, t+T]$ is also discrete. Alternatively, a vector of 0's (representing scheduled slots) and 1's (unscheduled slots) suffices to represent availability in $[t, t+T]$.

For slotted systems, First-Fit (FF) scheduling fits the request from input i to output j in the first jointly unscheduled slot $u = \min A_i \cap A_j$.

For unslotted systems, FF scheduling without fragmentation transmits the request from input i to output j in the first jointly available and contiguous interval for a packet of length p in the interval $U = \min_u [u, u+p] \subset A_i \cap A_j$.

For practical representations, we may record the beginning of the m-th contiguous available subinterval as t_m, with duration τ_m.

Also, the algorithm can be distributed with the availability set A_i kept at the input i and A_j kept at the output j. A request is sent from input i with its A_j to the output j. More than one request may be made to the same output. The output grants input requests in the order of request arrivals from different inputs. The granted requests are acknowledged with their scheduled transmission times sent to the input.

We may use the same switch fabric to communicate requests and acknowledgments in a round robin manner. The incurred overhead could be significant if packet size is small, e.g. for ATM cells. Therefore, we advocate unslotted switching without packet fragmentation.

III. MULTISTAGE CONFIGURATIONS OF IQ, OQ, AND IS SWITCHES.

Single stage S switching (IQ, OQ, or IS) may have limited aggregated throughput due to the limitation of the size of a crossbar switch in terms of number of inputs and outputs. To achieve Terabit or Petabit per second switching, multi-stages of S switches may be interconnected [4].

A well-known interconnection network is the Clos Network (CN) with three S stages, represented in fig 2a as a 3-dimensional SSS network. (The 1st and 3rd stage switches are shown horizontally, with each switch connected to a vertical 2nd stage switch by 1 link.) Clos networks facilitate non-blocking switching by virtue of many choices of the 2nd stage S switch interconnecting the 1st and 3rd stage switches. In classical switching terminologies [see 4], there are r_i switches in stage i (i= 1, 2, 3) with each switch of size $m_i \times n_i$. The network in figure 2a has $r_i = m_i = n_i = 2$ for i = 1, 2, and 3.

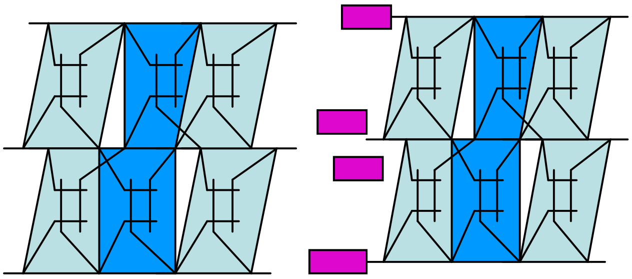

Fig 2a SSS Clos Network **Fig 2b** (IQ)SS Network

When the SSS Clos network is used for packet switching, we may choose IQ, OQ, or IS for each S stage. If input queuing are preferred, we may use the (IQ)SS configuration (fig 2b) for which the 1st stage switches are IQ switches. A multistage 3-phase algorithm for output conflict resolution was proposed in [20] for a 2-stage slotted (IQ)S switch. Randomized routing in the first IQ stage was used. Saturation throughput is reduced to 0.458 from single stage case of 0.586.

In [21], we also proposed a Terabit Ethernet (TbE) with variable length Ethernet frames through a 2-stage asynchronous (IQ)S switch. Assuming packet transmission time as being memoryless, a saturation throughput of 1/3 was obtained. Also, a 3-stage input queued (IQ)SS switch was proposed with alternative routing in the first IQ stage, with saturation throughput of 0.5. The implementation and throughput analysis is described in details in [21].

Many interesting multi-stage switches result from other choices of IQ, OQ, or IS for each stage [12-18]. For example, we may have the 2-stage configuration (IQ)S changed to (IQ)(OQ), thus avoiding output conflicts in the 2nd stage that reduces throughput. Saturation throughput, under uniform traffic, now resembles that of a single stage IQ switch.

Likewise, we may implement the (IQ)S(OQ) configuration. This, compared with the 2-stage (IQ)(OQ) configuration, has the advantage of choosing alternative paths through the first and second stage switches. Saturation through, properly routed, can approach 100%.

As an extreme case, every switch in an SSS network could be an OQ switch, represented as an (OQ)(OQ)(OQ) network (fig 3b). Each OQ switch uses share memory switching as drawn.

While the (OQ)(OQ)(OQ) switch requires no IS, we have to choose a 2nd stage switch to route a packet. Similar to the (IQ)SS switch, we may randomize routing in the first (OQ) stage. However, this randomization may bring about out-of-sequence packet arrivals.

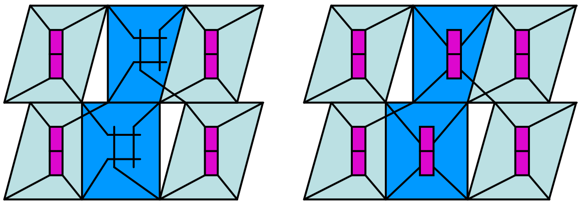

Fig 3a (OQ)S(OQ) Switch **Fig 3b** (OQ)(OQ)(OQ) Switch

This randomization is unnecessary for a 2-stage (OQ)(OQ) switch with unique routing between input and output, but hot spot may result without the randomizing 1st stage.

Studying various switch configurations and associated throughput is the subject of another paper. Here, we apply the First-Fit scheduling to an (OQ)S(OQ) switch shown in figure 3a, which routes packets through a choice of alternative switches in the 2nd stage. It shall be shown in the next section that the FF scheduling algorithm for single stage IS switch (described in the previous section) is applicable to the (OQ)S(OQ) switch configuration.

IV. FIRST FIT SCHEDULING OF THE (OQ)S(OQ) SWITCH

Consider packet arriving at each input to the (OQ)S(OQ) switch. Each 1st stage switch buffers all arriving packets without input queuing. Therefore, the arrival processes for all inputs to a 1st stage switch can be aggregated as a single stream.

The (OQ)(OQ)(OQ) configuration shown in figure 3b for which packets can be randomly routed and buffered in any one of the 2nd stage switches. This is not the case for the (OQ)S(OQ) switch. A packet, when routed through the 2nd stage switch, must be routed without conflict with other packets for the inputs and outputs of that 2nd stage switch. In other words, the input-output connectivity patterns for the different 2nd stage switches are 1-to-1 maps at any time.

For the unslotted case, the aggregate arrival process to a 1st stage switch can be modeled as a Poisson process. For the slotted case, the aggregate number of arrivals per slot can be modeled as Poisson distributed for large 1st stage switches.

We describe with formality the routing process in the 2nd stage. Without memory for IQ or OQ in the 2nd stage, input scheduling for the 2nd stage is required. Let $I = \{1,2,\dots,r_1\}$ be the set of the 1st stage switches. Let $J = \{1,2,\dots,r_3\}$ be the set of the 3rd stage switches. Let $K = \{1,2,\dots,r_2\}$ be the set of the 2nd stage switches. Scheduling for packet transmission from input switch $i \in I$ to output switch $j \in J$ now involves the scheduling of transmission on link (i, k) between the 1st stage switch i and 2nd stage switch $k \in K$ as well as on link (k, j) between 2nd switches k to 3rd stage switch j. We must choose a switch $k \in K$ for which both links (i, k) and (k, j) are free.

Consider the slotted case with a window of slots considered for routing a packet from switch i to switch j. The availability (over time) of the links (i,k) and (k,j) is represented by the sets A_{ik} and A_{kj} respectively.

Thus routing in this Clos network requires the choice of an intermediate switch $k \in K$ such that $A_{ik} \cap A_{jk}$ provides the first fit time slot for the transmitting of a packet from input switch i to output switch j. Hence each input switch $i \in I$ maintains locally a set of $\{A_{ik}\}_{k=1,\dots,n}$, where n=r_2 is the number of intermediate 2nd stage switches. The manner of scheduling is therefore analogous to that for the single stage IS switch.

Similarly, each output switch $j \in J$ maintains locally a set of $\{A_{kj}\}_{k=1,\dots,n}$, where n=r_2 is the number of intermediate 2nd stage switches. The manner of scheduling is therefore analogous to that for the single stage IS switch. The essential difference is that for the single stage IS switch, we are matching the availability of an input i to an output j, while for the three stage (OQ)S(OQ) switch, we are matching the availability of an input switch $i \in I$ to an output switch $j \in J$.

For this (OQ)S(OQ) switch, significant aggregation is achieved for the process of scheduling. Each input switch *i* collects all requests to the same output switch *j*. We assume the input and output switches are paired in a duplex mode. A switch (both 1st and 3rd stage) may send a packet containing both requests and acknowledgments to output switches in a round robin fashion.

V. THROUGPUT AND DELAY – SLOTTED SWITCHING

Fragmentation of packets into fixed length packets or ATM cells has the advantage of reducing queuing delay due to long packets. Also, scheduling can be done slot by slot instead of the entire variable length packet. However, scheduling requirement increases inversely as slot size. Furthermore, Segmentation and Reassembly (SAR) is complex when fragments may arrive out of order.

While analysis of IQ and OQ has been completed for single or multiple stage switching, IS scheduling is difficult to analyze. We simulated the single stage slotted IS switch with FF scheduling. Our first concern is throughput as we increase the scheduling window size *T*. As is the case for most IS switches, scheduling removes the problem of HOL blocking and given sufficient effort of scheduling, saturation throughput approaches 1. Our second concern is delay versus load at each input.

The following table shows simulation results for saturation throughput ρ_{sat} versus window size *T* ranging from *T*=1 (equivalent to IQ with saturation throughput of 0.586) to *T*=128. Packets are assumed to be fixed size occupying exactly 1 slot, with random output addresses from packet to packet. Thus saturation throughput approaches 1 as $T \to \infty$.

Table 1: Saturation Throughput versus Window Size

T	1	2	4	8	16	32	64	128
ρ_{sat}	0.55	0.55	0.61	0.70	0.77	0.85	0.90	0.94

We consider the fragmentation of variable length packets into fixed length slots. Throughput is reduced due to two reasons. First, the last fragment of a variable length packet may not take up the entire slot. Second, fragments of the same packet are destined for the same output, thereby enhancing the correlation of HOL packet addresses. This correlation tends to reduce throughput.

In our simulation, we assume that packet length is exponentially distributed with mean $1/\mu$. The idle time distribution is also exponentially distributed with mean 1/λ. Fig 4 plots saturation throughput versus mean packet length, expressed in multiples of the fixed slot size.

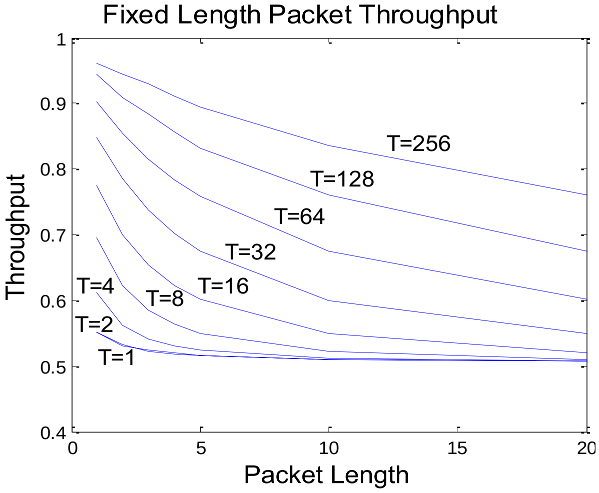

Figure 4 Throughput vs. Mean Packet Length

It is noteworthy that as packet length increases, throughput decreases as explained earlier and if packet length $1/\mu$ is relatively large compared with window size *T*, the throughput drops to 0.5. This 0.5 throughput is the same throughput for IQ for packets of exponentially distributed packet length, which can be derived explicitly. In effect, the HOL packets for the same output form a virtual output M/M/1 queue with mean system time $1/(\mu\text{-}\lambda)$. Thus the IQ is an M/M/1 queue with arrival rate λ and the HOL position serves packets at a rate of $(\mu\text{-}\lambda)$. The IQ is stable if $\lambda < (\mu\text{-}\lambda)$ or equivalently if the load $\lambda/\mu < 0.5$.

Fig 5 shows delay suffered as a function of window size *T* for a fixed packet length of 5. The figure shows a marked increase in delay for large windows. A small window reduces throughput, but with reduced throughput, delay is reduced as well.

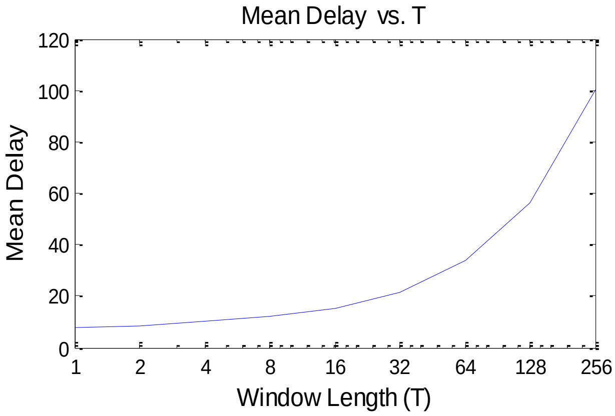

Figure 5 Mean Delay vs. T

While no simulation was performed for the (OQ)S(OQ) switch, such simulation is similar to the single stage case. The window size is now increased by the number of 2nd stage switch. Delay is reduced approximately by the same factor as scheduling transmission in alternative space switches incurs no additional delay.

VI. THROUGPUT AND DELAY – UNSLOTTED SWITCHING

We consider inputs which are alternately busy and idle for packet transmission. We assume that packet length is exponentially distributed with mean $1/\mu$. The idle time distribution is exponentially distributed with mean $1/\lambda$. The scheduling window size is assumed infinite. We plot in fig 6 mean delay versus load $\rho=\lambda/(\lambda+\mu)$. Delay is plotted in terms of mean delay per packet and mean delay per byte, with the latter delay giving weight to packets according to the size of the packet.

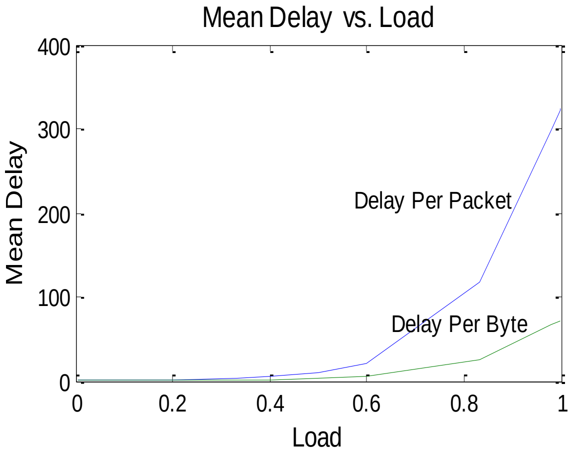

Figure 6 Mean Delay vs. Load (ρ=λ/(λ+μ))

Besides mean delay versus load, we are also interested in how packet size affects delay, i.e. whether large packets may be delayed excessively relative to smaller packets.

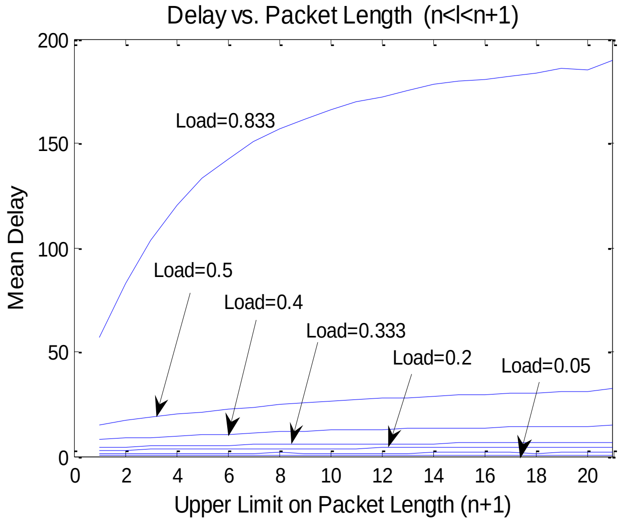

Figure 7 Mean Delay vs. Packet Length $n\leq l\leq n+1$

It is seen that at for load less than 0.5, larger packets do not suffer inordinately compared with smaller packets. As load increases towards 1, it becomes difficult for large packets to find a contiguous slot that is commonly available for both the input and the output. Consequently, we observe a much larger delay suffered for larger packets as shown above for a load of 0.833. Thus relief measure, discussed in the next section, may be necessary for large packets when traffic is heavy.

VII. CONCLUSION: REDUCING DELAY, OVERHEAD

A case is made here that fragmentation of packets for slotted transmission is undesirable. The scheduling and acknowledgment overhead for slotted transmission is too burdensome.

With the Internet fast becoming the network of choice for high bandwidth applications, large (kilo-bytes) packets are becoming quite prevalent. With high multi-megabit per second applications, delay incurred by packetization becomes small and unperceivable. With multi-gigabit per second transmission speed, queuing delay is also reduced for fixed level of utilization. As an example, consider a 1 Mb/s multimedia application transmitting 10kb packets over a 1 Gb/s link. Packetization delay is 10 ms, which is unperceivable to humans. Transmission delay is 10 µs per packet which is small from a network perspective.

The reason for packet fragmentation and reassembly (e.g. the SAR sub-layer of the ATM protocol) is less compelling, as application speed and transmission speed have increased by orders of magnitude. Thus the substantial complexity of SAR protocols could be avoided. More significantly, we avoid per cell scheduling and perform per packet scheduling instead. Since packets are transmitted in relative long microsecond timescale, there is sufficient time for scheduling, particularly when the scheduling can be distributed among the various 3rd stage switches, which acknowledges the 1st stage switches making the schedule requests.

We can reduce further scheduling overhead for short packets, for example by using a 2nd stage hub k_h. An input switch may multiplex all short packets to the hub. The hub collects these short packets and re-multiplexes packets for the same destination switch into the same packet. This would require the use of buffering for the hub. Alternatively, we may use a few 3rd stage switches as hubs which multiplex small packets to the same output through the 3-stage network once again.

Also, priority could be given to larger packets in order to maintain equity of delay of variable length packets. For example, we may reserve some 2nd stage switches for longer packets, while packing the shorter length packets in other 2nd stage switches.

We have demonstrated that a high throughput can be achieved with reasonable delay and scheduling complexity. There is no need for packet fragmentation, and large packets suffer only mildly, particularly at reasonable load.

We also favor the use of the (OQ)S(OQ) switch for transporting variable length packets. Techniques such as segregating packets through the 2nd stage switch according to packet lengths could provide delay relief to larger packets.

REFERENCES

1. M.G. Hluchyj and M.J.Karol, "Queing in High Performance Packet Switching," *IEEE Journal on Selected Area of Comm,* 6(9): 1587-1608, 1988.
2. J.Y.Hui, E.Arthurs,"A Broadband Packet Switch for Integrated Transport," *IEEE J. on Selected Areas of Comm.* Vol 5, pp 1274-1283, Oct. 1987
3. M.J.Karol, M.G. Hluchyj, and S.P.Morgan, "Input versus Output Queuing on a Space Division Packet Switch," *IEEE Trans. Comm.* Vol-35, pp 1347-1356, Dec. 1987
4. J.Y.Hui, *Switching and Traffic Theory for Integrated Broadband Networks,* Kluwer Academic Publishers, 1990.
5. N. McKeown, "The iSLIP Scheduling Algorithm for Input Queued Switches," *IEEE/ACM Trans on Networking,* 7(2):188-201, 1999
6. N. McKeown, A. Mekkittikul,V. Anartharam, and J.Walrand, "Achieving 100% Throughput in an Input Queued Switch," *IEEE Trans. on Comm.*, 47(8): 1260-1267, 1999.
7. C.Y. Tu, C.S. Chang, D.S. Lee, and C. T. Chiu, "Design a Simple and High Performance Switch using a Two-Stage Architecture", *Proceedings of IEEE Globecom*, 2005.
8. E. Oki, R. Rojas-Cessa and H.J. Chao, "PMM: a pipelined maximal-sized matching scheduling approach for input-buffered switches", *Global Telecommunications Conference*, vol. 1, 25-29 Nov. 2001, pp. 35-39.
9. H.J.Chao, C.H. Lam and X. Guo, "A Fast Arbitration Scheme for Terabit Packet Switches", *Global Telecommunications Conference*, vol. 2, 1999, pp. 1236-1243.
10. H.J.Chao, "Saturn: A Terabit Packet Switch using Dual Round-Robin", *Global Telecom Conference*, vol.1, 27 Nov.-1 Dec.2000, pp. 487-495.
11. K.Yoohwan and H.J. Chao, "Performance of Exhaustive Matching for Input-Queued Switches", *IEEE International Conference on Communications*, vol. 3, 11-15 May 2003, pp. 1817 - 1822.
12. H.J. Chao, S.Y. Liew and Z. Jing, "A Dual-level Matching Algorithm for 3-stage Clos-Network Packet switches", *High Performance Interconnects*, 20-22 Aug.2003, pp. 38-43.
13. H.J. Chao, Jing Zhigang and S.Y. Liew, "Matching Algorithms for Three-Stage Bufferless Clos Network Switches", *IEEE Communications Magazine*, vol. 41, no.10, Oct. 2003, pp. 46-54.
14. L. Chuanjun, S.Q. Zheng and Y. Mei, "Scalable Schedulers for High-Performance Switches", *IEEE workshop on High Performance Switching and Routing*, 2004, pp. 198-202.
15. G. Shabtai, I. Cidon and M. Sidi, "Two Priority Buffered Multistage Interconnection Networks", *IEEE workshop on High Performance Switching and Routing*, 2004, pp. 75-79.
16. J. Wu, H.P. Shiang, K. T.Chen and H. W. Tsao, "Delay and Throughput Analysis of the High Speed Variable Length Self-Routing Packet Switch", *IEEE workshop on High Performance Switching and Routing*, 2002, pp. 314-318.
17. A. Bianco, M. Franceschinis, S. Ghisolfi, A.M. Hill, E. Leonardi, F. Neri and R. Webb, "Frame-Based Matching Algorithms for Input-Queued Switches", *IEEE workshop on High Performance Switching and Routing*, 2002, pp. 69-76.
18. C.S. Chang, D.S. Lee and Y.S. Jou, "Load Balanced Birkhoff-Von Neumann Switches", *IEEE workshop on High Performance Switching and Routing*, 2001, pp. 276-280.
19. F.M. Chiussi and F. Tobagi, "A Hybrid Shared-Memory/Space-Division Architecture for Large Fast Packet Switches", *IEEE International Conference on Comm.* vol. 2, 14-18 June 1992, pp.905 -911.
20. J.Y. Hui, T.H.Lee, "A Large Scale ATM Switching Network With Sort-Banyan Switch Modules," *Proc. of IEEE Globecom, 1992*
21. J.Y. Hui, D.A. Daniel, "Terabit Ethernet: A Time-Space Carrier Sense Multiple Access Method," *Proceedings of the 2008 IEEE Globecom,* pp. 1-6, New Orleans 2008.