Matching Algorithms for Three-Stage Bufferless Clos Network Switches

H. Jonathan Chao, Zhigang Jing, and Soung Y. Liew, Polytechnic University

ABSTRACT

Three-stage Clos network switches is an attractive solution to future broadband packet routers due to their modularity and scalability. Most of the three-stage Clos network switches assume either all modules are space switches without memory (bufferless), or employ shared memory modules in the first and third stages (buffered). The former is also referred to as the space-space-space (S^3) Clos network switch, while the latter is referred to as the memory-space-memory (MSM) Clos network switch. In this article we provide a survey of recent literature concerning switching schemes in the S^3 Clos network switch. The switching problem in the S^3 Clos network switch can be divided into two major parts, namely port-to-port matching (scheduling) and route assignment between the first and third stages. Traditionally, researchers have proposed algorithms to solve these issues separately. Recently, a new class of switching algorithms, called Matching Algorithms for Clos (MAC), has been proposed to solve the scheduling and route assignment simultaneously. We focus on the MAC schemes and show that the new class of algorithms can achieve high performance and maintain good scalability.

INTRODUCTION

The last decade has witnessed rapid growth of the Internet. The number of users is increasing exponentially and applications that demand more bandwidth are emerging. Meanwhile, the price of optical transport has dropped tremendously with the advent of dense wavelength-division multiplexing (DWDM) technologies. The demands for increased Internet bandwidth and development on optical transmission technologies have put a great challenge on the design of high-speed packet routers. Current router technologies cannot provide switching capacity large enough to meet future Internet bandwidth demands. Because the number of switching elements of single-stage switches is proportional to the square of the number of switch ports, they become unattractive as the switches become larger. On the other hand, the multistage switch architecture, such as the three-stage Clos network, provides much higher switch capacity, e.g., up to a few hundred terabits per second or even petabits per second [1].

Without loss of generality, a three-stage Clos network switch has three stages of switch modules: k input modules (IMs), m center modules (CMs), and k output modules (OMs). Each IM (OM) has n input (output) ports, where $N = nk$ is the switch size. All the switch modules are nonblocking. Their dimensions are $n \times m$, $k \times k$, and $m \times n$, respectively.

Most of the packet switching algorithms in three-stage Clos network switches are based on one of the following assumptions:

- All modules (i.e., IMs, CMs, and OMs) are bufferless space (e.g., crossbar) switches; this kind of architecture is known as the space-space-space (S^3) Clos network switch.
- IMs and OMs are shared memory switches, while CMs are space switches; this kind of architecture is known as the memory-space-memory (MSM) Clos network switch.

In this article we address the scheduling problems in the S^3 bufferless Clos network switch. Interested readers can refer to [2] for the packet dispatching schemes in the MSM Clos network switch.

To schedule cells from input ports to output ports in the S^3 bufferless Clos network switch, there are two major issues: cell scheduling and route assignment. The former is necessary for determining a set of input-output matches for cell switching, while the latter is necessary for assigning a set of internally conflict-free paths for the above matches.

It is very challenging 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 multistage bufferless Clos network switch. In [3], a path switching scheme was proposed for scheduling in the three-stage bufferless Clos network packet switch. In path switching, a periodic IM-to-OM route assignment is predetermined prior to cell arrivals in accordance with a given traffic loading pattern. With this predetermined assignment, cells can be transferred when there are routes

This work was supported in part by the National Science Foundation under Grant 9814856 and Grant 9906673, and in part by the New York State Center for Advanced Technology in Telecommunications.

0163-6804/03/$17.00 © 2003 IEEE

(or tokens) available for the corresponding IM-OM pairs. However, the path switching scheme assumes the traffic loading pattern is known in advance, which is not applicable to the Internet because most Internet traffic is connectionless. In [4] the authors proposed a distributed static round-robin (Distro) scheduling algorithm to route cells in a round-robin manner. However, the Distro algorithm cannot handle various traffic conditions well due to its static nature.

Recently, a new class of algorithms, called Matching Algorithms for Clos (**MAC**), has been proposed to provide efficient scheduling solutions for the three-stage bufferless Clos network switch [5–7]. To relax the strict arbitration time constraint, MAC operates based on a frame of r cells ($r > 1$). It is highly distributed such that the input-output matching and routing path finding are simultaneously performed by scheduling modules. MAC is capable of providing high performance while maintaining good scalability.

In this article we will first formulate the scheduling problems in the three-stage bufferless Clos network switch. Then a new class of matching algorithms for Clos network switches, f-MAC, c-MAC, and d-MAC, are presented to resolve the scheduling problems in the three-stage bufferless Clos network switch.

Problem Formulation

The scheduling problems in the three-stage bufferless Clos network switch can be formulated as two major issues: cell scheduling and route assignment. The cell scheduling problem can be formulated as the bipartite graph (port-to-port) matching problem, while the route assignment can be formulated as the bipartite graph (module-to-module) edge-coloring problem.

Port-to-port Matching

In general, virtual output queuing (VOQ) structure can be used to alleviate the head-of-line (HOL) blocking phenomenon that occurs in input-buffered switches. In this buffering scheme, each input port maintains a separate queue for each output port in such a way that cells in a VOQ do not block cells in any other VOQs except for contention.

The VOQ status of the switch can be represented by a port-to-port bipartite graph, in which input port g is connected to output port h with a unique edge if and only if the corresponding VOQ, say $VOQ(g, h)$, is not empty, for $1 \le g, h \le N$. In each time slot, a scheduling algorithm determines a set of nonconflicting cells (no common input port or output port) for switching. This is equivalent to finding a matching from the port-to-port bipartite graph mentioned above.

The matching found in a given bipartite graph is not unique. Basically, the larger the cardinality of the matching found, the more cells can be switched in a time slot and the better the performance of the switch. However, other criteria (e.g., fairness, time complexity) also need to be considered. A number of scheduling algorithms have been proposed to solve the matching problem based on different criteria. Interested readers can refer to [2] for some of them.

Route Assignment

Given a set of input-output matches, the route assignment problem in the S^3 Clos network switch can be formulated as the edge-coloring problem in the corresponding module-to-module bipartite graph. We illustrate this with the following example.

We assume $m = n = 3$, $k = 2$. Suppose that the port-to-port matching found for the switch can be represented by a matching matrix as shown in Fig. 1a, where entry (g, h) is 1 if input port g and output port h are matched; 0 otherwise. With reference to Fig. 1b, this port-to-port matching matrix can be transformed into an IM-OM connection matrix, in which entry (i, j) denotes the number of central routes needed to assign for cells from IMi to OMj, where $1 \le i, j \le k$. Note that any row sum or column sum in the IM-OM connection matrix is no more than n. With reference to Fig. 1c, the IM-OM connection matrix is logically equivalent to a bipartite multigraph.

Suppose that each CM is labeled with a different color; for example, CM1 is labeled with blue, CM2 green, and CM3 red. Assigning central routes for the input-output matches is equivalent to edge-coloring the corresponding IM-OM bipartite multigraph with the three colors in such a way that no two edges incident on the same node are assigned the same color. Figure 1c gives an edge-color assignment of the IM-OM bipartite multigraph; Fig. 1d shows the corresponding route assignment.

Note that the connection pattern in each CM is actually an IM-OM matching. Thus, the route assignment (edge-coloring) problem can also be formulated as finding m (number of CMs) sets of matching from the IM-OM bipartite multigraph. In other words, it can be regarded as decomposing the IM-OM connection matrix into m IM-OM matching matrices as illustrated bellow:

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \xrightarrow{\text{matrix decomposition}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

where any row column sum is either 0 or 1 in each IM-OM matching matrix. Consider $m = n$ (which is known as the rearrangeably nonblocking condition) in the Clos network switch, there have been several route assignment (edge-coloring) algorithms proposed in literature. They can be classified into two categories:

- Optimized algorithms, which can provide guaranteed results for all matches but with a higher complexity in time [16] or in implementation [8]
- Heuristic algorithms (e.g., a parallel matching scheme [9]), which can provide all or part of the matches with routes in much lower time complexity

Here we briefly show how a parallel matching scheme [9] can be adopted to perform route assignment. Let us label the set of output links of each IM_i by A_i ($1 \le i \le k$) and the set of input links of each OM_j by B_j ($1 \le j \le k$), as shown in

The cell-scheduling problem can be formulated as the bipartite-graph (port-to-port) matching problem, while the route assignment can be formulated as the bipartite-graph (module-to-module) edge-coloring problem.

Fig. 1e. Each A_i and B_j contains exactly m elements denoting the state of each physical link, that is, $A_i = \{a_{i,h}, 1 \le h \le m\}$, $B_j = \{b_{j,h}, 1 \le h \le m\}$. Note that, $a_{i,h}$ $(b_{j,h}) = 0$ means that this link has not been matched; otherwise, $a_{i,h}$ $(b_{j,h}) = 1$. To find a routing path between IM_i and OM_j, one just needs to find a vertical pair of zeros in A_i and B_j. For instance, Fig. 1e shows two available paths between A_i and B_j.

Matching Algorithms for Clos Network Switches (MAC)

The matching algorithms must be able to scale gracefully because the scheduling time for resolving port-to-port matching and route assignment becomes more stringent with the increases of switch size and port speed. When building a large switching system, different line cards, the packet scheduler, and the switch fabric can be put into different racks that are separated by distances of up to a few tens of meters. The round-trip propagation delay between the line cards and the packet scheduler is very likely to be more than one time slot in this case. For instance, with 64-byte fixed length cells at the port speed of 10 Gb/s, a 30-m distance accounts for about 6 cell slots with fiber lightspeed.

One way to cope with the situation is to use the pipeline mechanism [10], in which multiple packet schedulers are employed. In every time slot, requests are sent to one of the packet schedulers in a round-robin fashion without waiting for the previous scheduling results. Suppose that it takes r time slots for a packet scheduler to schedule cells; then the pipeline mechanism needs r packet schedulers, hence increasing the hardware complexity.

Another solution is to schedule cells in batch so as to relax the scheduling cycle [11–13]. The enlarged scheduling cycle is often called a *frame*, which consists of r time slots. By switching frames instead of cells, some optical switches with slower reconfiguration speeds can be good candidates for constructing the switch fabric. This is because their high transmission speeds (e.g., 160 Gb/s) can significantly reduce optical interconnections between racks [5]. A frame-based scheduling scheme proposed in [11] achieves attractive delay bounds. However, it requires not only more than N^2 schedulers (N is the switch size) in the system, but also each scheduler to perform sophisticated packet scheduling schemes, such as weighted fair queuing or shaped virtual clock, which are prohibitively costly. In [12], contentions are resolved on a frame basis and r sets of matching are determined in the beginning of each frame. In other words, in each frame, the packet scheduler determines the switching schedule (r sets of matching) for the next frame. In [13] a batch of requests is accumulated and a corresponding schedule for a constrained switch is generated. However, to achieve high performance, all of these schemes have very high time complexity, which limits the switch scalability.

In the following, we present a new class of matching algorithms, called MAC, for resolving scheduling problems in the three-stage Clos network switches. To relax the strict arbitration time constraint, MAC operates based on a frame of r cells ($r > 1$).

Figure 2 shows the structure of a MAC packet switch, which consists of a packet scheduler (PS) and a three-stage Clos network switch fabric. The PS consists of k scheduling input modules (SIMs) and k scheduling output modules (SOMs), each of which corresponds to an input switch module (or output switch module) in the three-stage Clos network switch. A crosspoint switch with a reconfigured pattern is used to interconnect these SIMs and SOMs.

All incoming packets are first terminated at ingress line cards (ILCs), where they are segmented into cells (fixed length data units) and stored in a memory. The packet headers are extracted for IP address lookup, classification,

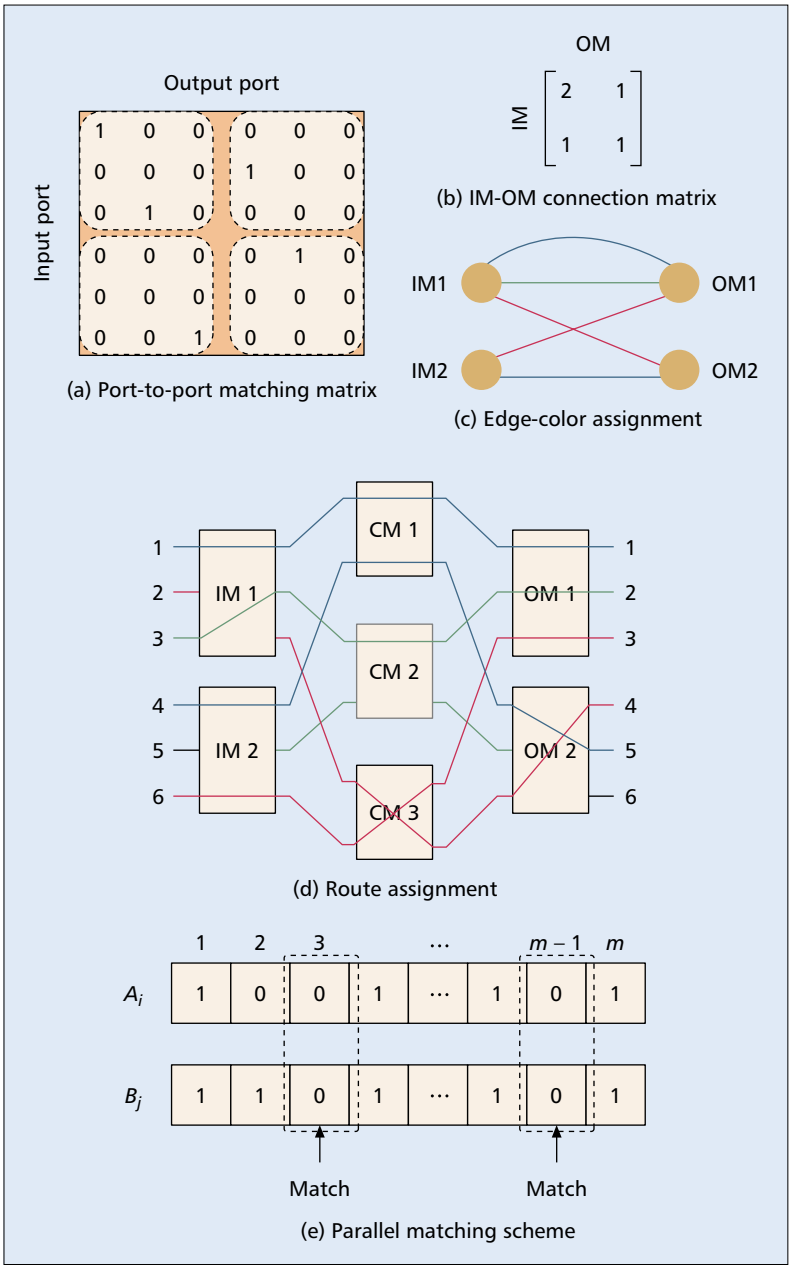

Figure 1. *Edge-coloring and route assignment: a) port-to-port matching matrix; b) IM-OM connection matrix; c) edge-color assignment; d) route assignment; e) parallel matching scheme.*

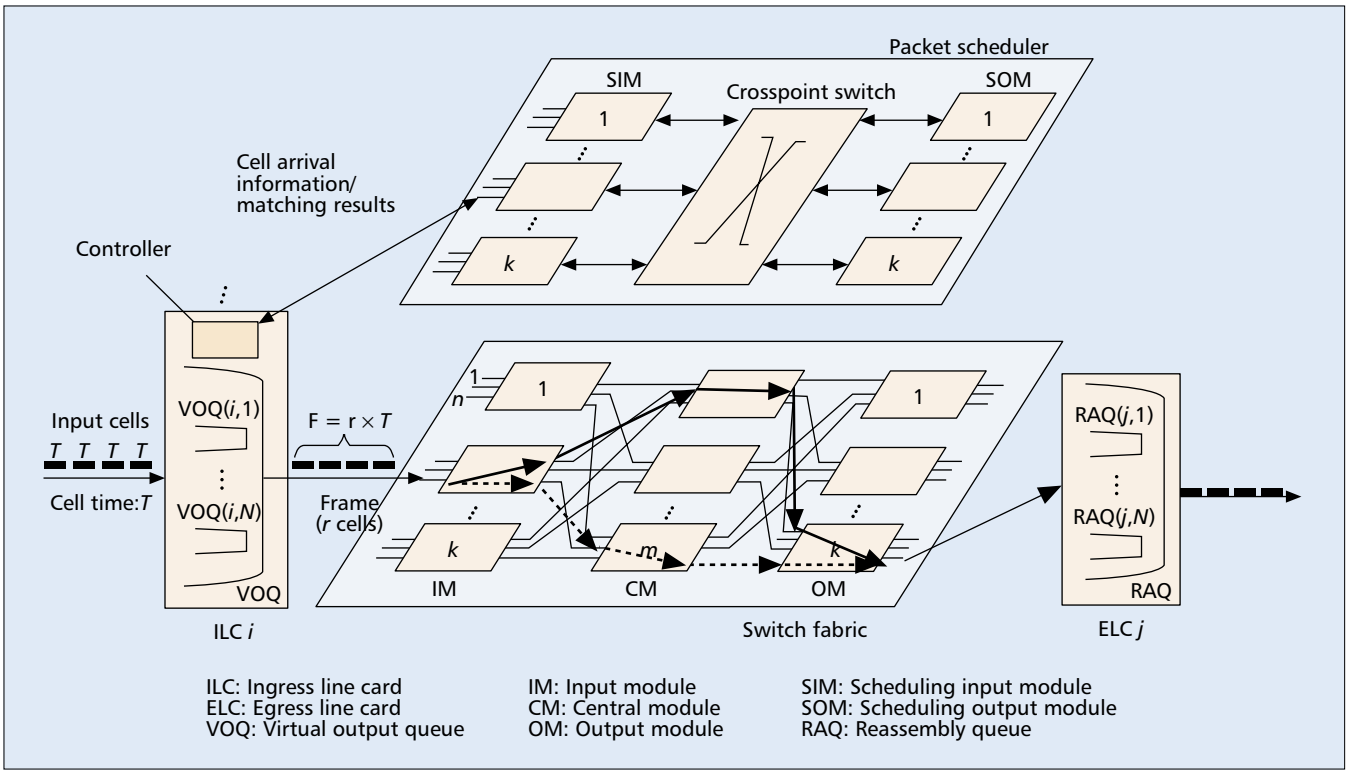

Figure 2. *The structure of a three-stage Clos network switch and a packet scheduler.*

and other functions such as traffic shaping/policing. Packets destined for the same output port are stored in the same VOQs in the ILCs. Multiple cells (e.g., r cells) in the same VOQ form a frame that is sent to an output port once a grant signal is given by the PS. Let us define a cell time slot to be T, and the frame period $F = r \times T$, where r is the frame size in number of cells. The ILCs send cell arrival information to the PS. The PS then resolves contention, finds routing paths in the center stage, and sends grant signals to the ILCs in each frame period F. A large buffer with a reassembly queue (RAQ) structure is used in the egress line card (ELC) to store the frames from the switch fabric and reassemble them into packets.

F-MAC

This subsection presents the frame-based MAC switch (**f-MAC**) [5]. To relax the strict arbitration time constraint, f-MAC operates based on a frame of r cells ($r > 1$). f-MAC includes two phases to solve the scheduling problems. It first resolves the contention of the frames from different input ports that are destined for the same output port. It then determines a routing path through the center stage (i.e., chooses a CM) for each matched input-output pair. Since there can be multiple possible paths (determined by the number of CMs) for each matched I/O, how to choose a CM to reduce internal blocking and thus increase the throughput further complicates the scheduling algorithm design.

In the first phase, f-MAC is an extension of exhaustive dual round-robin matching (EDRRM) scheme [14] by including the frame concept. Most iterative matching schemes, such as *i*SLIP [15] and DRRM [2], suffer from the problem of throughput degradation under unbalanced traffic distribution. The EDRRM scheme improves throughput by maintaining the existing matched pairs between the inputs and outputs so that the number of unmatched inputs and outputs is drastically reduced (especially at high load), thus reducing the inefficiency caused by not being able to find matches among those unmatched inputs and outputs. The f-MAC also modifies EDRRM slightly to further improve the throughput. One of the major problems of the exhaustive matching is that it may cause starvation in some inputs. One way to overcome this problem is to set a timer for each HOL frame. When the timer expires, the request from the "expired" frame has the highest preference to be granted.

Phase 1: Find I/O Matches

In phase 1, f-MAC consists of three steps.

Step 1 (Request): Each unmatched input sends a request to every output port arbiter for which it has cell(s) in the corresponding VOQ. The request is set at high priority if the queue length is above (or equal to) the threshold r; otherwise, the request is set at low priority. Each matched input only sends a high-priority request to its matched output.

Step 2 (Grant): If each output port arbiter receives one or more high-priority requests, it chooses the first request to grant starting from the current position of the high-priority pointer. Otherwise, it grants the first low-priority request starting from the current position of the low-priority pointer.

Step 3 (Accept): If each input port arbiter receives one or more high-priority grants, it accepts the first starting from the current position of the high-priority pointer. Otherwise, it

With the increase of switch sizes and port speeds, the hardware and interconnection complexity between input and output arbiters makes it very difficult to design the packet scheduler in the centralized way.

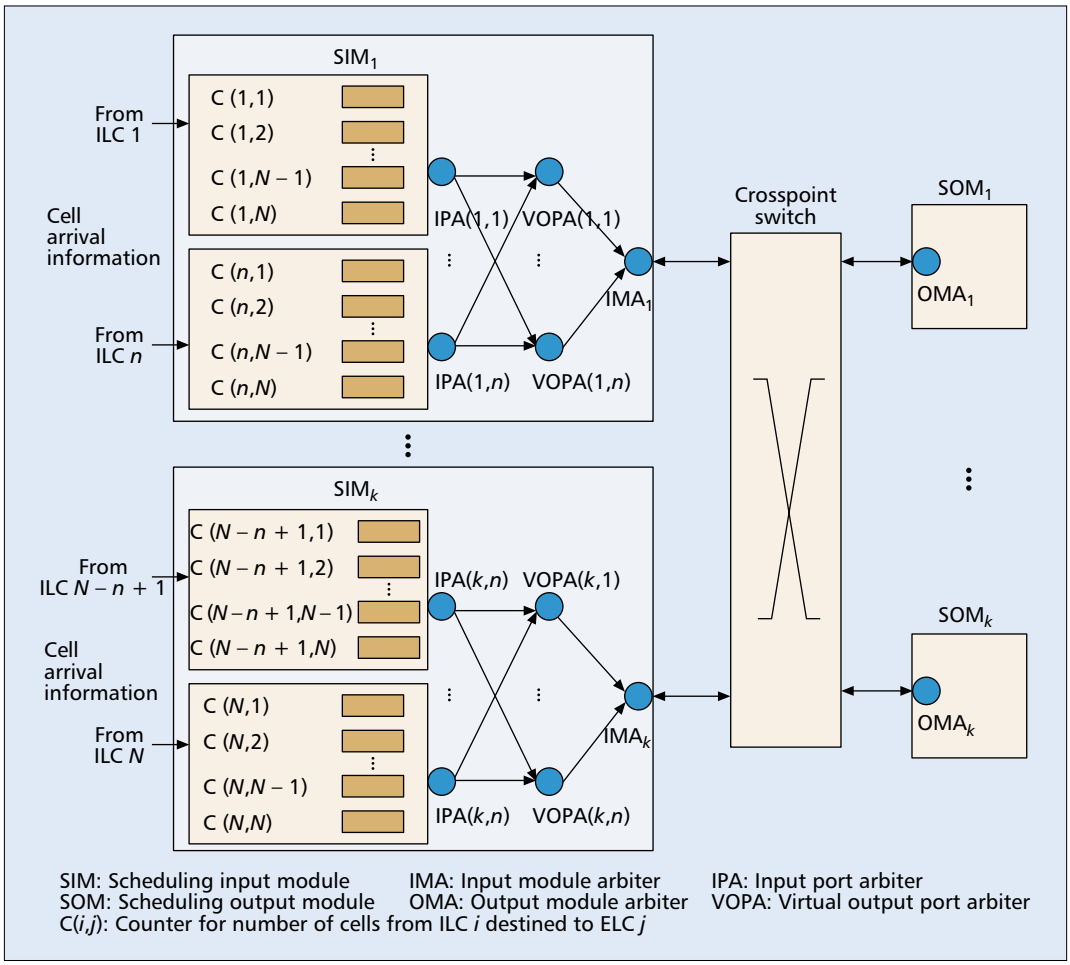

Figure 3. *The schematic of the packet scheduler.*

accepts the first starting from the current position of the low-priority pointer.

The pointers of the input and output port arbiters are updated to the chosen position only if the grant is accepted in step 3.

Phase 2: Find Routing Paths for the Matched I/O Pairs

After input-output matching is completed in phase 1, f-MAC finds a routing path for each matched input-output pair through the three-stage bufferless Clos network switch. To reduce computation complexity, a simple parallel matching scheme [9] is adopted. That is, f-MAC includes k matching cycles. In each matching cycle, each SIM is matched with one of k SOMs, and the parallel matching scheme described in previous section is adopted to find the vertical pairs of zeros between A_i and B_j.

c-MAC

With the increase of switch sizes and port speeds, the hardware and interconnection complexity between input and output arbiters makes it very difficult to design the packet scheduler in a centralized way. This subsection presents a more scalable concurrent MAC, **c-MAC** [6]. It is highly distributed such that input-output matching and routing path finding are *concurrently* performed by scheduling modules.

Figure 3 shows the architecture of the packet scheduler. It consists of k SIMs and k SOMs, each of which corresponds to an input switch module (or output switch module) in the three-stage Clos network switch (Fig. 2). There are n input port arbiters (IPAs) in each SIM. Each SIM consists of n virtual output port arbiters (VOPAs), each of which corresponds to an output port in the corresponding OM. Each SIM has an input module arbiter (IMA), and each SOM has an output module arbiter (OMA). A crosspoint switch with a predetermined reconfigured pattern is used to interconnect these SIMs and SOMs. As shown in Fig. 2, each ILC has N VOQs, each corresponding to an ELC. A counter $C(i,j)$ in PS is used to record the number of cells in the corresponding $VOQ(i,j)$.

The c-MAC scheme divides one frame period f into k matching cycles, as shown in Fig. 4. In each matching cycle, each SIM is matched with one of k SOMs. During each cycle, c-MAC includes two phases to find the input-output matches and routing paths, respectively.

At the beginning of each matching cycle, each SOM passes $m + 2n$ bits to the corresponding SIM, where m bit corresponds to the state of m input links of the corresponding OM; $2n$ bits corresponds to the state of n output ports of the corresponding OM. There are four possible states for each output port: 00 when the output port is unmatched, 01 when the output port is

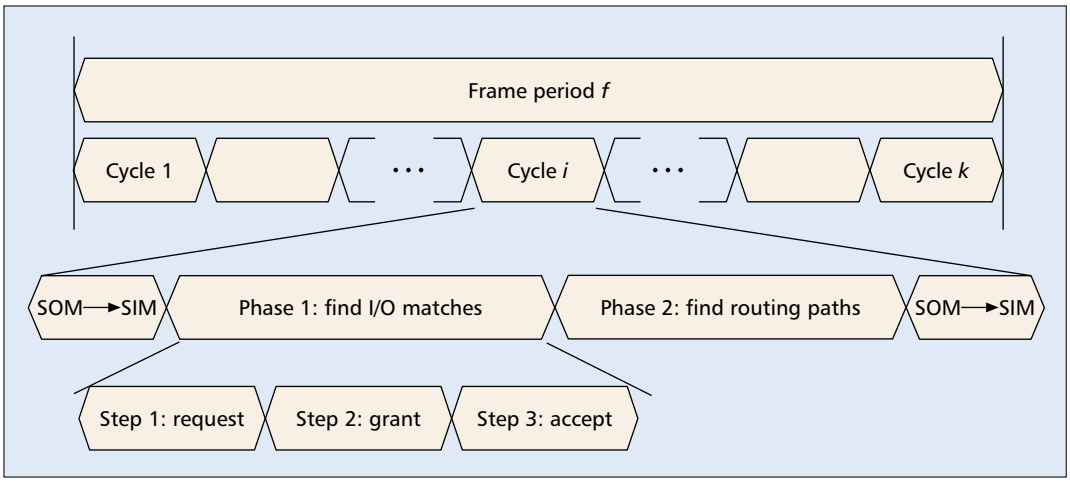

Figure 4. *The timing schematic of the c-MAC scheme.*

The c-MAC scheme divides one frame period f *into* k *matching cycles. In each matching cycle, each SIM is matched with one of* k *SOMs. During each cycle, c-MAC includes two phases to find the input-output matches and routing paths, respectively.*

matched with low priority in the last frame period, 10 when the output port is matched with high priority in the last frame period, and 11 when the output port is matched in this frame period.

It is assumed that the matching sequence between SIMs and SOMs is predetermined. For instance, in a cycle, SIM_i is matched with SOM_j, where $1 \le i \le k$; $1 \le j \le k$. In the next cycle, SIMi is matched with SOM$(j + 1)$ mod k. The procedure is repeated k times. To achieve matching uniformity for all the SIMs, the beginning matching sequence between SIMs and SOMs is skewed one position at the beginning of each frame period.

Phase 1: Find I/O Matches

Phase 1 consists of three steps.

Step 1: Request — Each matched IPA only sends a high-priority request to its matched VOPA; each unmatched IPA (including the currently matched IPA but whose matched VOQ's queue length is less than a threshold r) sends a 2-bit request to every VOPA for which it has queued cells in the corresponding VOQ. (00 means no request; 01 means low-priority request because queue length is less than r; 10 means high-priority request because queue length is larger than r; 11 means the highest priority because the waiting time of the HOL frame is larger than a threshold, T_w.) Note that using the waiting time mechanism for the HOL frames prevents the starvation problem.

Step 2: Grant — Only the "available" VOPA performs the grant operation. A VOPA is defined as available if its corresponding output port is:

- A. Unmatched
- B. Matched in the last frame period with low priority (the VOPA receives at least one high-priority request at this frame period)
- C. Matched in the last frame period with high priority, but receives the request from the matched IPA and its priority is becoming low in this frame period

If a VOPA is available and receives one or more high-priority requests, it grants the one that appears next in a fixed round-robin schedule starting from the current position of the high-priority pointer. If there are no high-priority requests, the output port arbiter grants one low-priority request in a fixed round-robin schedule starting from the current position of the low-priority pointer. The VOPA notifies each requesting IPA whether or not its request is granted.

Step 3: Accept — If the IPA receives one or more high-priority grants, it accepts the one that appears next in a fixed round-robin schedule starting from the current position of the high-priority pointer. If there are no high-priority grants, the input port arbiter accepts one low-priority request in a fixed round-robin schedule starting from the current position of the low-priority pointer. The input port arbiter notifies each VOPA whether or not its grant is accepted.

Update of the pointers: The pointer of IPA and VOPA is updated to the chosen position only if the grant is accepted in step 3 of phase 1 and also accepted in phase 2.

Phase 2: Find Routing Paths for the Matched I/O Pairs

In phase 2, c-MAC adopts the parallel matching scheme [9] to find the routing paths for the matched I/O pairs as described in the previous section.

D-MAC

In c-MAC, each SIM-SOM pair needs to be matched once regardless of the queuing status of the switch, yielding k matching cycles for the arbitration in one frame period. This, however, results in a high time complexity that prevents one from selecting a small frame size. To further reduce the scheduling time complexity and relax the arbitration time constraint, we have proposed a new dual-level MAC, call **d-MAC** [7], to determine the matching sequence between the SIMs and SOMs according to the queuing information of the switch.

The d-MAC scheme consists of two levels of matching, module- and port-level matching. The former is responsible for determining the SIM-SOM matching pattern according to the queuing status of the switch. The latter is responsible for determining the port-to-port matching and find-

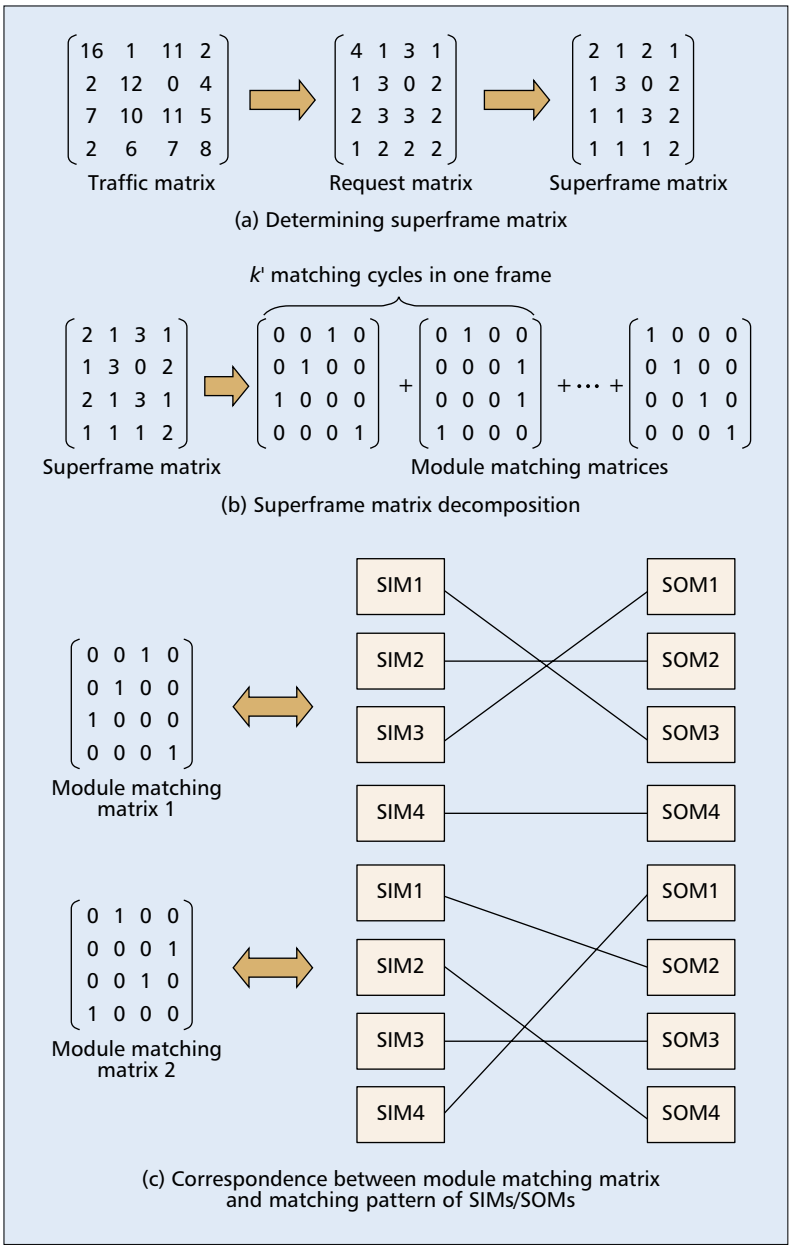

Figure 5. *An illustration of module-level matching of d-MAC* (k = *4,* k′ = *2,* F = *3).*

ing the internal routing path for the matched input-output pair, like the task of matching cycle in c-MAC.

For module-level matching, the switching patterns of a number, say F, of frames can be determined simultaneously. These F frames constitute a superframe. We use the example shown in Fig. 5 to illustrate the module-level matching steps of d-MAC as follows. With reference to Fig. 5a, a traffic matrix is used to represent the queuing status of the switch, in which entry (i, j) denotes the number of buffered cells that desire to be transmitted from IMi to OMj. According to this traffic matrix, a request matrix can be obtained. Note that the request matrix gives the number of requests that can be sent from each SIM to each SOM. Then a scheduling algorithm can be employed to do arbitration among these requests, and this process produces a superframe matrix, in which each entry represents the matching opportunities between each SIM and each SOM in one superframe. With reference to Fig. 5b, the superframe matrix is further decomposed into the module-level matching matrices, where a module-level matching matrix represents the matching pattern of SIMs and SOMs in one matching cycle, as shown in Fig. 5c. The module-level matching is done after the module-level matching matrices are determined. With each of these matrices, the port-level matching can thereafter perform the task of matching cycle (i.e., port-to-port matching and route assignment) for the given SIM-SOM pairs.

The module-level matching and port-level matching assignment can be performed in a pipelined manner, as shown in Fig. 6. As described above, each superframe is composed of F frames, and each frame consists of r cells. Suppose that the number of matching cycles in each frame is set to be k', where $k' \le k$. The transmission time of r cells must be greater than or equal to the arbitration time of k' matching cycles. This is the constraint against selecting a small r. In each super-frame, there are totally $F \times k'$ matching cycles. Thus, F sets of k' module-matching matrices are to be determined for the next superframe, while the scheduler is doing the port-level matching for the current superframe.

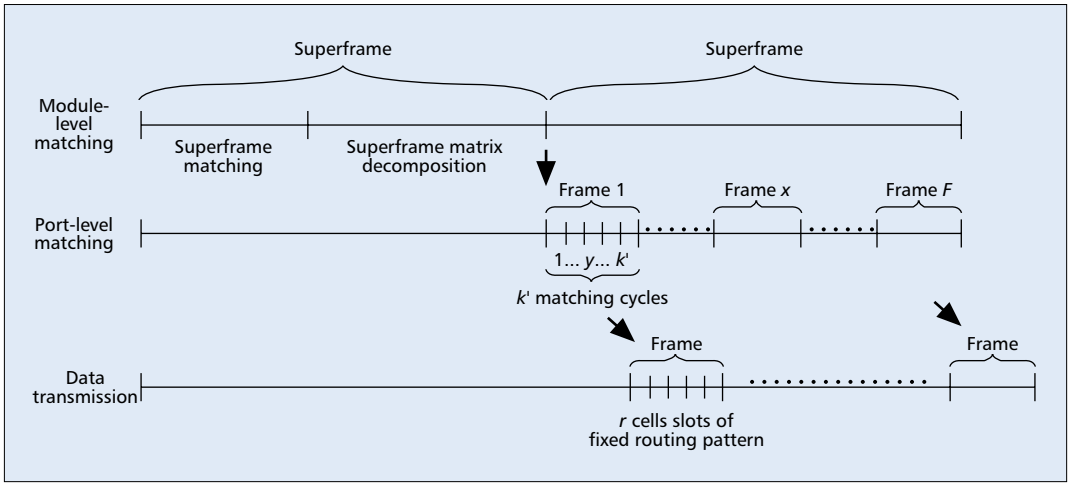

Figure 6. *The pipelined process of the d-MAC matching scheme.*

The Module-Level Matching Algorithm

The module-level matching algorithm is composed of two phases: superframe matching and superframe decomposition.

The superframe matching is to determine a superframe matrix in accordance with the queuing status between the switch modules. Each entry in the superframe matrix represents the matching opportunities between each SIM and each SOM in one superframe. A superframe matrix is a $k \times k$ matrix with

- No row/column sum greater than $F \times k'$
- No entry greater than F

To determine the superframe matrix, the d-MAC scheme adopts an iterative request/grant/accept algorithm, modified from the *i*SLIP scheme.

The superframe decomposition is to decompose the superframe matrix into $F \times k'$ module-matching matrices. Each module-matching matrix records the matching status between the SIMs and SOMs in one matching cycle of the next superframe. The matrix decomposition problems can be solved by edge-coloring algorithms. However, optimal edge-coloring algorithms are not preferable here because of their high time complexity [8]. Instead, we use the parallel matching heuristic [9] to decompose the superframe matrix. This algorithm contains k rounds. In each round, each SIM is communicating with one of k SOMs. Each SIM/SOM maintains a two-tuple array that contains $F \times k'$ zero-one variables. Let $W_i(Z_j)$ be the array of SIM_i (SOM_j)

$$W_i(x,y) = \begin{cases} 0, & \text{if SIM } i \text{ is unmatched in cycle } y \text{ of frame } x, \\ 1, & \text{if it has been matched in cycle } y \text{ of frame } x; \end{cases}$$

$$Z_i(x,y) = \begin{cases} 0, & \text{if SOM } j \text{ is unmatched in cycle } y \text{ of frame } x, \\ 1, & \text{if it has been matched in cycle } y \text{ of frame } x; \end{cases}$$

where $1 \le x \le F$ and $1 \le y \le k'$.

When SIM_i is communicating with SOM_j, the d-MAC scheme tries to find as many common zero entries in W_i and Z_j as possible to meet the number given by entry (i, j) in the superframe matrix. Note that no more than one 0 entry in the same frame can be assigned to the same SIM-SOM pair.

The Port-Level Matching Algorithm

When the module-level matching is completed, the matching sequence between SIMs and SOMs (recorded in the module matching matrices) is determined for the next superframe. The port-level matching algorithm consists of k' matching cycles in a frame. In each matching cycle, the port-level matching algorithm includes two steps:

- The port-to-port matching assignment
- The central module assignment

To find the port-to-port matching for the corresponding pair of IM-OM, the d-MAC scheme adopts an iterative request/grant/accept algorithm (e.g., iSLIP). To improve the matching efficiency, the d-MAC scheme also introduces high- and low-priority arbiters in the SIMs. Under the priority mechanism, the VOQs with queue lengths of more than r cells will send out high-priority requests and have higher priority than the unfilled ones. To determine an internal routing path for each matched input-output pair, the d-MAC scheme adopts a heuristic parallel matching algorithm [9] described in the previous section.

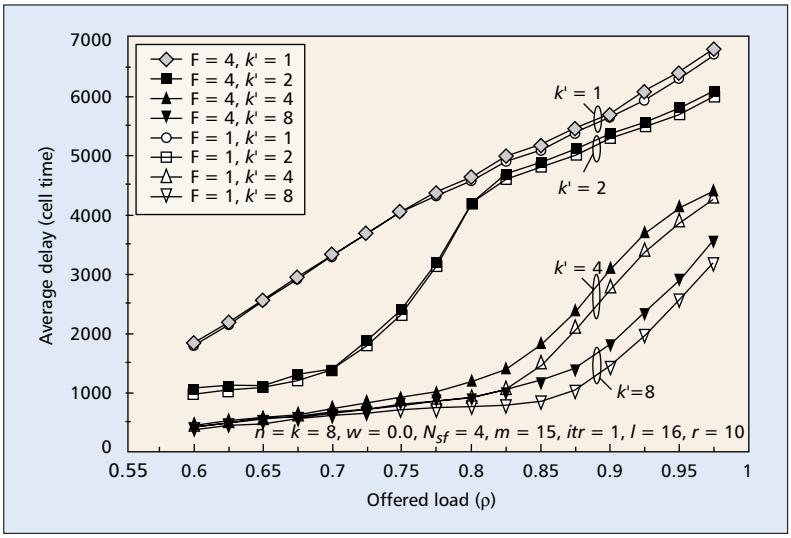

Figure 7. *Average cell delay of the d-MAC scheme under uniform traffic with different superframe size* F *and matching cycle* k′.

Figure 7 shows the average cell delay of the d-MAC scheme under uniform traffic with different superframe sizes (F) and numbers of matching cycles (k') for a switch size ($N = nk$) of 64. We assume the frame size (r) to be 10 cells, and the CM number (m) to be 15. The number of iterations (N_{sf}) to find the superframe matrix is 4, and *itr* to find port-level matches in each matching cycle is 1. In the superframe matching phase, we assume that each SOM cyclically grants the requests by choosing one request from each SIM in turn. Traffic is assumed to be bursty with an average burst length l of 16 cells. The average delay is not sensitive to F, but very sensitive to k'. The larger k' is, the more module-to-module matching combination we can have in each frame, hence the more port-to-port matches can be found, which leads to better delay performance. However, smaller k' can relax the time complexity so that the scheme is more implementable with existing hardware technologies. This allows d-MAC to provide flexibility to trade off timing complexity with delay performance.

Conclusions

In this article we address the packet scheduling problems in three-stage bufferless Clos network switches. The scheduling problems can be basically divided into two issues: port-to-port matching and IM-OM route assignment. Traditional approaches perform these two assignments separately; hence, they either have high time complexity or are difficult to implement. Recently, a new class of switching algorithms, called Matching Algorithms for Clos (MAC), has been proposed to resolve port-to-port matching and route

To find the port-to-port matching for the corresponding pair of IM-OM, the d-MAC scheme adopts an iterative request/grant/accept algorithm, e.g., iSLIP. To improve the matching efficiency, the d-MAC scheme also introduces high-priority and low-priority arbiters in the SIMs.

assignment simultaneously in the three-stage bufferless Clos network switch. We have also shown that the new class of matching algorithms, including f-MAC, c-MAC, and d-MAC, can achieve high performance and maintain good scalability.

References

[1] H. J. Chao, "Next Generation Routers," *Proc. IEEE*, vol. 90, no. 9, Sept. 2002, pp. 1518–58.
[2] H. J. Chao, C. Lam, and E. Oki, *Broadband Packet Switching Technologies — A Practical Guide to ATM Switches and IP Routers*, Wiley, 2001.
[3] T. T. Lee and C. H. Lam, "Path Switching — A Quasi-Static Routing Scheme for Large-Scale ATM Packet Switches," *IEEE JSAC*, vol. 15, June 1997, pp. 914–24.
[4] K. Pun and M. Hamdi, "Distro: A Distributed Static Round-Robin Scheduling Algorithm for Bufferless Clos-Network Switches," *IEEE GLOBECOM '02*, Taipei, Taiwan, Nov 17–21, 2002.
[5] H. J. Chao, K-L. Deng, and Z. Jing, "A Petabit Photonic Packet Switch (P3S)," *IEEE INFOCOM '03*, San Francisco, CA, Apr. 1–3, 2003.
[6] H. J. Chao, K-L. Deng, and Z. Jing, "PetaStar: A Petabit Photonic Packet Switch," *IEEE JSAC* Special Issue on High-Performance Optical/Electronic Switches/Routers for High-Speed Internet, vol. 21, no. 7, Sept. 2003.
[7] H. J. Chao, S. Y. Liew, and Z. Jing, "A Dual-Level Matching Algorithm for 3-stage Clos-Network Packet Switches," *Proc. Hot Interconnects 11*, Stanford Univ., CA, Aug. 2003.
[8] T. T. Lee and S. Y. Liew, "Parallel Routing Algorithms in Benes-Clos Network," *IEEE Trans. Commun.*, vol. 50, no. 11, Nov. 2002, pp. 1841–47.
[9] M. Karol and C-L. I, "Performance Analysis of a Growable Architecture for Broadband Packet (ATM) Switching," *GLOBECOM '89*, 1989, pp. 1173–80.
[10] E. Oki, R. Rojas-Cessa, and H. J. Chao, "A Pipeline-Based Approach for Maximal-Sized Matching Scheduling in Input-Buffered Switches," *IEEE Commun. Letts*, vol. 5, no. 6, June 2001, pp. 263–65.
[11] K. Kar *et al.*, "Reduced Complexity Input Buffered Switches," *Proc. Hot Interconnects 8*, 2000.
[12] A. Bianco *et al.*, "Frame-based Matching algorithms for Input-Queued switches," *Proc. High-Perf. Switching and Routing '02*, pp. 69–76.
[13] B. Towles and W. Dally, "Guaranteed Scheduling for Switches with Configuration Overhead," *IEEE INFOCOM '02*, vol. 1, pp. 342–51.
[14] Y. Li, S. S. Panwar, and H. J. Chao, "The Dual Round Robin Matching Switch with Exhaustive Service," *Proc. High-Perf. Switching and Routing, 2002*.
[15] N. McKeown, "The iSLIP Scheduling Algorithm for Input-Queues Switches," *IEEE/ACM Trans. Net.*, Apr. 1999, pp. 188–200.
[16] R. Cole, K. Ost, and S. Schirra, "Edge-Coloring Bipartite Multigraphs in O(*E* log *D*) Time," *Combinatorica 21*, 2001, pp. 5–12.

Biographies

H. Jonathan Chao [F] (chao@poly.edu) is a professor of electrical and computer engineering at Polytechnic University, Brooklyn, New York, which he joined in January 1992. He has been doing research in the areas of terabit switches/routers, quality of service control, optical networking, and network security. He holds more than 20 patents and has published over 100 journal and conference papers in the above areas. During 2000–2001 he was co-founder and CTO of Coree Networks, New Jersey, where he led a team to implement a multiterabit MPLS switch router with carrier-class reliability. From 1985 to 1992 he was a member of technical staff at Telcordia, where he was involved in transport and switching system architecture designs and ASIC implementations. He received the Telcordia Excellence Award in 1987. He is a co-recipient of the 2001 Best Paper Award from *IEEE Transaction on Circuits and Systems for Video Technology*. He co-authored two networking books, B*roadband Packet Switching Technologies* (Wiley, 2001) and *Quality of Service Control in High-Speed Networks* (Wiley, 2001).

Zhigang Jing [M'01] received B.S., M.S., and Ph.D. degrees from the University of Electronic Science and Technology of China in 1993, 1996, and 1999, respectively, all in electrical engineering. He then joined the Department of Electrical Engineering, Tsinghua University, Beijing, China, where he was a post-doctoral fellow. Since March 2000 he has been with the Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, New York, as a post-doctoral fellow. His current research interests are high-speed networking, terabit IP routers, multimedia communication, Internet QoS, Diffserv, and MPLS. He is a member of the Association for Computing Machinery (ACM).

Soung Y. Liew received his B.S. degree in electrical engineering from National Taiwan University in 1993, and his M.Phil. and Ph.D. degrees in information engineering from the Chinese University of Hong Kong (CUHK) in 1996 and 1999, respectively. He joined CUHK as an assistant professor in 1999. In 2002–2003 he was a research associate at Polytechnic University, New York. Currently he is an assistant professor at University Tunku Abdul Rahman, Malaysia. His research interests are in packet switching systems, routing and scheduling algorithms, and optical networks.