Scheduling Algorithms for Shared Fiber-Delay-Line Optical Packet Switches Part II: The 3-Stage Clos-Network Case

Shi Jiang Depart. of ECE Polytechnic University Brooklyn, NY 11201 sjiang01@utopia.poly.edu Gang Hu Depart. of ECE Polytechnic University Brooklyn, NY 11201 hgang01@utopia.poly.edu

Abstract— In all-optical packet switching, packets may arrive at an optical switch in an uncoordinated fashion. To prevent packet loss in the switch, fiber delay lines (FDLs) are used as optical buffer to store optical packets. However, assigning FDLs to the arrival packets to achieve high throughput, low delay, and low loss rate is not a trivial task. In our companion paper, we have proposed several efficient scheduling algorithms for singlestage shared-FDL optical packet switches. To further enhance the switch's scalability, we have extended our work to a multistage case. Here, we propose two scheduling algorithms: (1) sequential FDL assignment and (2) multi-cell FDL assignment algorithms for a 3-stage optical Clos-Network switch (OCNS). We show by simulation that a 3-stage OCNS with these FDL assignment algorithms can achieve satisfactory performance.

Keywords—all-optical network, optical cell switching; fiber delay line; scheduling algorithm; Clos-network switch

I. INTRODUCTION

Today, the processing speed of electronic devices has become the bottleneck of optical networks. It can be foreseen that such capacity mismatch is getting worse in the near future because the growth of optical fiber transmission capacity exceeds the improvement of electronic devices' processing power. Thus, it is generally recognized that all-optical switching is the key to the success of the next-generation optical network.

To flexibly use the tremendous capacity of optical fiber, several all-optical switching paradigms have been proposed and under intensive study. In this paper, we focus on the switch architectures and the corresponding scheduling algorithms in the time-slotted all-optical switching schemes, such as optical packet switching (OPS) [1-4], time sliced optical burst switching (TSOBS) [5], and optical cell switching (OCS) [6]. In these schemes, time is divided into slots of fixed size, and each timeslot is referred to as an optical cell, or a cell. It should be noted that the terms "cell" and "packet" are used interchangeably in this paper. In a cellswitched network, cells may arrive at a switch in an uncoordinated fashion. That is, cells from different inputs may be destined for the same output port in the same timeslot. Therefore, fiber delay lines (FDLs) are needed to buffer (delay) cells when contention occurs. The architecture of opticalbuffered switches and the corresponding scheduling algorithms are the most challenging issues to the all-optical packet switching network.

In our previous work [6-8], we have studied the cell scheduling algorithms for single-stage shared-FDL switches. The structure of a single-stage shared-FDL switch is given in

Soung Y. Liew Faculty of ICT Univ. Tunku Abdul Rahman Selangor Malaysia syliew@mail.utar.edu.my H. Jonathan Chao Depart. of ECE Polytechnic University Brooklyn, NY 11201 chao@poly.edu

Figure 1. The switch has a number of feedback FDLs that are shared by all input ports. Suppose that there are Z feedback FDLs, N input ports, and N output ports. Each FDL delays cells by a fixed number of timeslots, and any two FDLs may have the same or different delay values. The outputs (inputs) of FDLs and the inputs (outputs) of the switch are collectively called the inlets (outlets) of the switch fabric, yielding N+Z inlets and N+Z outlets.



Figure 1. A shared-FDL switch

In the shared-FDL switch, when two or more cells are destined for the same output port in the same timeslot, only one of them gets access to the output port and the remaining are routed to the FDLs (if available), waiting to be transferred to the output port in future timeslots. Scheduling cells to avoid output port and FDL conflict is usually called the FDL assignment.

We have proposed in [7, 8] several reservation FDL scheduling algorithms for the single-stage shared-FDL switch, such as the sequential FDL assignment (SEFA) algorithm, which searches available FDL routes to delay the cells in a cell-by-cell basis; and the multi-cell FDL assignment (MUFA) algorithm, which uses sequential search to find available FDL routes for multiple cells simultaneously. Details about SEFA and MUFA can be found in our companion paper [8].

However, the scalability of the single-stage shared-FDL switch is greatly limited by the number of required cross points, which is $(N+Z)^2$. To further enhance the scalability of the optical-buffered switches, it is common to consider the multi-stage modular switch architecture due to its high scalability and low complexity nature. Among all multi-stage modular switch architectures the Clos-Network is the most practical and frequently used scheme, which gives the balance of the switch performance and the hardware complexity. Thus we investigated the FDL assignment for a 3-stage optical Clos-Network switch, A 3-stage OCNS consists of $K N \times M$ input

modules (IMs), $K M \times N$ output modules (OMs) and $M K \times K$ center modules (CMs). Note that the size of the switch is $NK \times NK$.

To buffer cells when contention occurs, FDLs can be placed at IMs, CMs, and/or OMs. However, different FDL placements can result in different scheduling complexity and performance. If the FDLs are only placed at OMs, cells are forced to be routed to the last stage as soon as they arrive at the switch. Since each OM can only accept up to M cells at any given timeslot, excess cells will be discarded, which resulting a high cell-loss rate. If the FDLs are only placed at CMs, global availability information are needed when scheduling a batch of incoming cells and thus this placement discourages distributed FDL scheduling schemes. In this paper, we are focused on the OCNS in which FDLs are only placed at IMs, as shown in Figure 2. We call this switch structure the 3-stage shared-FDL-IM OCNS (SFI-OCNS). In the SFI-OCNS, cells can be delayed only at the first stage, while the second and third stages are used only for routing purpose, hence cells can be scheduled in a distributed manner. To give a fair comparison, we have studied the performance of different FDL placements. Our simulation results confirm that the SFI-OCNS has the lowest cell loss rate.



Figure 2. A 3-stage Shared-FDL-IM Optical Clos-Network Switch

In addition to the FDL assignment, there is another important issue in the SFI-OCNS: the central-route assignment. It is well-known that the number of CMs (i.e., M) in a 3-stage Clos-network switch determines the non-blocking characteristic of the switch. If $M \ge 2N-1$ [9], the switch is said to be strictly nonblocking because central routes can be arbitrarily assigned for the existing connections, yet none of the future connections will be blocked. However, given an M smaller than 2N-1, the central routes must be assigned carefully; otherwise rearrangement may be necessary or internal blocking may occur. There are two kinds of centralroute assignment algorithms for Clos-network switches: the

optimized and the heuristic. Although the optimized algorithms can always find the optimal solution to the centralroute assignment, they have a very high time complexity. Therefore, in practice, heuristic algorithms are preferable for scalability with a cost of slight performance degradation.

Extended from our previous work in [8], here we propose two FDL assignment algorithms for the OCNS: (i) sequential FDL assignment for Clos-network switches (SEFAC), which assigns departure times for each arriving cell, searches FDL routes, and determines central-module routes at a cell-by-cell basis; and (ii) multi-cell FDL assignment for Clos-Network switches (MUFAC), which assigns departure times and FDL routes for multiple cells simultaneously, and then assigns central-module routes for these cells in a heuristic manner. MUFAC is a practical algorithm to perform cell scheduling for the SFI-OCNSs due to its graceful scalability and distributed nature.

The remainder of this paper is organized as follows. Details on SEFAC and MUFAC are discussed in section II, and III, respectively. In section IV, we evaluate the performances of SEFAC and MUFAC. Conclusions will be given in section V.

II. SEQUENTIAL FDL ASSIGNMENT FOR OPTICAL CLOS-NETWORK SWITCH (SEFAC)

A. SEFAC

With reference to Figure 2, since each input module is a single-stage shared-FDL switch, it maintains its own slot transition diagram. In addition, the whole system has a bigger configuration table that keeps track of the availability of all outputs in each timeslot. Therefore, the output port and centerroute availabilities are accessible by all input modules for perform scheduling algorithm. The FDL assignment and cell departure schedule are described below. Each input port takes turn to search for the earliest timeslot that satisfies the following three conditions: (i) the destined output port is available in the timeslot, (ii) there exists an FDL route on the corresponding input module that can move the cell from the current timeslot to that timeslot, (iii) a route between the IM and the destined OM is available at that timeslot. When all three conditions are met, input port assigns the FDL routes, departure time; and randomly selects center-route among all available center-routes. This searching process is performed one input port after another. In order to achieve fairness among all input ports, round-robin mechanism can be included in the SEFAC algorithm in such a way that the priory of searching is rotated among all input modules

The time complexity of SEFAC is a function of the size of the SFI-OCNS. Since SEFAC has a similar operation as SEFA, SEFAC has a time complexity of $K \times N \times (Q \times T)$, where *K* is the number of OMs, *N* is the number of output ports for each OM, *Q* is the number of nodes in the transition diagram *G*, and *T* is the time for each input request to search one node in the transition diagram *G* for output port and FDL availability. For instance, in a 1024×1024 SFI-OCNS, which has 32 IM, 32 CM, and 32 OM, each module has size of 32×32, and each IM has 32 FDLs, then *K*=32, *N*=32. If we limit the maximum number of delay operations to 2, then *Q*, the total number of nodes in transition diagram *G* is 36. Let us assume T=10ns. Thus, the total complexity of SEFAC is $32\times32\times(36\times10ns) = 369\mu s$.

B. FDL Distribution Study in 3-Stage OCNS

The cell-loss performances of different FDL placements with SEFAC for the OCNS are studied. We assume that the OCNS has 32 IMs, 32 OMs, 32 CMs, and each IM (OM) has 32 input ports (output ports). The overall switch size is 1024×1024. We consider five different cases of FDL placement and compare their performances under uniform traffic. With reference to Figure 3, let Zin be the number of FDLs that are attached on each input module, Zout be the number of FDLs that are attached on each output module. To give a fair comparison, we let Zin+Zout=32. The cases studied are as follows: (a) Zin=32, Zout=0; (b) Zin=24, Zout=8; (c) Zin=16, Zout=16; (d) Zin=8, Zout=24; and (e) Zin=0, Zout=32.



Figure 3. FDL distribution in 3-stage OCNS

As shown in Figure 4, placing all FDLs (buffers) at the input modules achieves the best performance; while placing all of them at the output modules performs the worst. To explain this, let us assume that there is no blocking in the middle stage and the entire switch is logically equivalent to a set of K independent concentrator-knockout switches [10], each having the structure as shown in Figure 5. Since each input modules has no buffer, incoming cells in the input modules are forced to go through the center stage to the output modules immediately upon their arrival at the switch. In the worst case scenario, all $K \times N$ input ports could have cells destining to a same output module. However, at any given timeslot, only up to M cells can arrive at a given OM and the excess cells will be discarded by the CMs even before cells reach the OM, this is a so-called knockout phenomenon. Therefore, the loss rate is the highest when all buffers are placed at OMs. On the contrast, when FDLs are located at input modules, cells can be buffered at input stage and directed to the corresponding output modules center-route and output ports are available. Therefore, the performance is the best among all cases.







Figure 5. Knockout principle at the OM of the OCNS

III. MULTI_CELL FDL ASSIGNMENT FOR OPTICAL CLOS-NETWORK SWITCH (MUFAC)

MUFAC is modified from MUFA so that it can schedule FDL routes and departure times for multiple cells simultaneously in a distributed manner. Based on the simulation result given in Section II, we only consider the MUFAC algorithm for the SFI-OCNS. In addition, we assume M < 2N-1. In this case, central-module contention is unavoidable and thus MUFAC must take into the consideration of central-route assignment.

There are three tasks for the MUFAC algorithm: (1) assign FDL routes in the IMs, (2) schedule cell departure times according to the output port availability, and (3) assign central-routes between IMs and OMs for multiple cells simultaneously. In order to accomplish all three tasks, the original single-stage MUFA algorithm is enhanced and further combined with Karol's matching algorithm [11].

A. Karol's matching algorithm

Karol's matching algorithm assigns central routes for cells in a heuristic manner, yet achieves a good assignment result. Referring to Figure 2, let *i* be the index of IMs, where $i \in \{0, 1, 2, 3 \dots K-1\}$; and let *j* be the index of OMs, where $j \in \{0, 1, 2, 3 \dots K-1\}$. In Karol's matching algorithm, each timeslot is divided into *K* cycles. For $0 \le t \le K-1$, in cycle *t*, IM *i* is scheduled to communicate with OM *j*, where $j = [(t + i) \mod K]$, in order to perform the central-route assignment for cells that are from IM *i* to OM *j*. After *K* cycles, each IM-OM pair is paired up once and the entire central-module assignment can then be done in a distributed manner. To achieve fairness among all traffic loads, the matching sequence of Karol's matching algorithm can be modified and done in a round-robin fashion in such a way that for each set of traffic request, the order of modules matching can be skewed among all matching pairs. For example, suppose that in the current timeslot, IM *i* starts its cycle matching procedure from OM *j*, then in the next timeslot, it may start from OM $[(j + 1) \mod K]$, and so on and so forth.

An example of the matching sequence in Karol's matching algorithm in the OCNS is given as follows. Let us consider a 9×9 SFI-OCNS, which has 3 input modules, 3 center modules and 3 output modules. It takes 3 cycles for all IMs to perform Karol's matching with all OMs. Figures 6(a), 6(b), and 6(c) illustrate such a matching sequence.



(a). First cycle



(b). Second cycle



(c). Third cycle

Figure 6. Karol's matching algorithm in the OCNS

To find an available center-route in Karol's algorithm is quiet simple. Each IM-OM pair can be connected via M CMs, one can use a vector for each input and output module to record the availability of the central modules. With reference to Figure 7, A*i* vector records the available route from IM *i* to all CMs. Similarly, B*j* vector records the available route between all CMs and OM*j*. Each element in those vectors corresponding to each CM; and a '0' means available and '1' represents unavailable. For those pairs of modules which have a cell to dispatch between them, the two vectors will be compared to locate an available central module if any.



Figure 7. Vector representation of center route availability in Karol's algorithm

B. MUFAC

In MUFAC, each IM maintains its own transition diagram, and each level-*k* node $T_k(t)$ (as shown in Figure 12 in [8]) keeps the FDL availabilities of that IM for timeslot *t*. In addition, each OM keeps the corresponding output-port availabilities (OPAs), and each of the IMs and OMs keeps the corresponding central-route availabilities (CRAs). With the transition diagram, nodes take turn to be the parent node from level-0 node to each of the level-(*L*-1) nodes. Each of these turns is called an iteration.

Based on Karol's matching algorithm, an iteration is further divided into K cycles. In each cycle, each IM is paired up with a particular OM, yielding K IM-OM pairs, and only the cell requests for these IM-OM pairs will be handled. This is done by means of four phases, namely Request, Grant, Accept, and Update.

In the Request phase, each IM works independently from the others, in which the parent node sends the unfulfilled requests to its child nodes so that they can execute the Grant phase independently. At the same time, each child node also collects the OPAs from the paired OM for the corresponding timeslot so that it can grant the unfulfilled requests with the available output ports in the Grant phase. After granting the unfulfilled requests, the child nodes pass their grant decisions back to the parent node. At the same time, the parent node collects the CRAs from its home IM and the paired OM. In the Accept phase, the parent node makes the accept decision based on the following four criteria: (1) unfulfilled input requests; (2) availability of FDL on that IM for the corresponding timeslots; (3) availability of center-routes from that IM to the paired OM in the corresponding timeslots; and, (4) when multiple grants occur, the parent accepts the grant with the earliest departure time. After parent node makes the accept decision, it passes the decision to its child nodes for updating. In the Update phase, the parent node updates CRAs and FDL availabilities; while the child nodes update OPAs on the paired OM. This completes a cycle of MUFAC. Note that all these phases can be executed in a distributed manner. After K cycles, a node is done with the role of the parent node, and the next node will take the role and run again the K cycles. This process continues until the last parent node (a level-(L-1) node) is done with the iteration.

We illustrate MUFAC with an example as follows. With reference to Figure 2, suppose that the switch just gets reset, so all output ports are available. We also assume the incoming cell requests from input 1 to input 9 are output ports 1, 4, 7, 1, 4, 7, 1, 4, 7, 1, 4, 7, respectively. Each IM has transition diagram as shown in Figure 8.



Figure 8. Transition diagram for a 9 by 9 OCNS

In the first iteration, MUFAC tries to assign direct connections for the cell requests. In this 9×9 OCNS, it requires 3 cycles to finish this task. Within each cycle, each IM consults with a different OM for current OPAs. At first cycle, IM1 gets OPAs and center-routes information from OM1. It finds out output port 1 is available and assigns the direct connection. Similarly, IM2 and IM3 resolve output requests 4, and 7, respectively. At second and third cycle, no more requests can be resolved because all desired output ports are being assigned at first cycle. After 3 cycles, the

assignment diagram for T0 nodes at each IMs is shown in Figure 9(a).

In the second iteration, T0 becomes the parent of T1, T2, and T4. Four-step assignment process namely request, grant, accept, and update is performed 3 times in 3 cycles. At first cycle, each IM does not have unfulfilled request heading to the matched OM, so no assignment is made. At second cycle, IM1 has a match with OM2 for output port 4 at T1; IM2 resolves output port 7 with OM3 at T1; and IM3 finds matching for output port 1 with OM1 at T1. At third cycle, all remaining unfulfilled requests are resolved at T2. Figure 9(b), 9(c) shows the assignment diagram at T1 and T2 for each IM, respectively.



(a) Assignment diagram for T0



(b) Assignment diagram for T1



(c) Assignment diagram for T2Figure 9. Assignment diagrams for a 9 by 9 OCNS

To find the time complexity of MUFAC, let us consider the complexity of MUFAC at each cycle first. Suppose the time needed for a parent node to send out unfulfilled request is Tr; the time need for child nodes to make grant decision is Tg, which includes a step of parallel AND operations to match the unfulfilled requests with the available output ports; the time needed to find available center-routes is Tc; the time needed for parent nodes to make accepting decision is Ta, where Ta consists of $\log_2 F$ sequential steps of bit comparison (to grant the matches for each child node); and the time needed for all processing nodes to update information is Tu. Although requesting and updating are two different procedures in MUFAC algorithm, these two tasks consists of only the register accessing; so, they can be performed in parallel. Therefore, the time needed for these two tasks can be counted as one called Tr/u. Then the time for one cycle process is Tg+Ta+Tr/u. Moreover, let K be the number of cycles in each process, and let P be the number of nodes that act as parent nodes during the MUFAC process. The time complexity of MUFAC is P×K×(Tg+Ta+Tc+Tr/u). For example, for a 1024 by 1024 OCNS which has 32 IMs, 32 CMs, and 32 OMs, each module has size of 32 by 32, then P=1+7=8 with limited delay operation of 2 and K=32. Assume Tc=Tg=Tr/u=5ns, then $Ta=(1+log_2F)\times 5ns=40ns$. Therefore, the total time complexity for MUFAC is $8 \times 32 \times 55$ ns $\approx 14 \mu$ s.

IV. PERFORMANCE EVALUATION OF SEFAC AND MUFAC

In our performance evaluation, we considered a 1024×1024 SFI-OCNS for both SEFAC and MUFAC. The SFI-OCNS consists of 32 IMs, 32 CMs, and 32 OMs, each module has 32 inputs and 32 outputs; we assume 32 FDLs are employed at each IMs, and there are 5, 5, 5, 5, 4, 4, and 4 FDLs with delay values 1, 2, 4, 8, 16, 32, and 64 cell times, respectively. Furthermore, we limited the delay operation for each cell to 2 in both scheduling algorithm. In addition, we use a single stage 32×32 SEFA as a benchmark.

As shown in Figure 10, both FDL assignment algorithms for the SFI-OCNS can achieve $\sim 10^{-7}$ loss rate at 0.87 loads. There are two possible phenomena that make MUFAC and SEFAC perform differently. (1) In MUFAC, for a particular

output port, we guarantee that the FDL routes with the fewer delay operations are assigned the earlier. However, considering two FDL routes with the same number of delay operations, it is possible that the route with the larger delay is selected rather than the route with the smaller delay. This occurs when the former's parent node has a smaller index than that of the latter's parent node. Such a phenomenon doesn't occur in SEFAC. (2) In SEFAC, since cells that could be destined for different outputs are scheduled sequentially, it is possible that FDL routes with the more delay operations are assigned to cells in the early time in such a way that they occupy the FDL resources and prevents the subsequent cells from finding FDL routes with the fewer delay operations. In this case, FDL resources are less efficiently used in SEFAC than MUFAC. From Figure 9, phenomenon (1) makes SEFAC performs better at a load below 0.94; phenomenon (2) makes MUFAC performs better at a load above 0.94. Overall the performances of SEFAC and MUFAC for the SFI-OCNS are compatible.

Figure 11 shows the delay comparison of SEFAC and MUFAC, with SEFA as a benchmark. The plot gives that SEFAC and MUFAC have identical delay performance, and have expected disadvantage as compared to SEFA at load 0.9 and above. This delay disadvantage is mainly the result of center-route limitation in the Clos-Network switch architecture. Under light traffic loading, limited center-routes are more than the system's need, thus the Clos-Network switch architecture is transparent to FDL assignment. Therefore, SEFAC and MUFAC have compatible delay performance as SEFA at light load. While under heavy traffic loading, center-route availabilities in the Clos-Network switch architecture become a resource limitation; hence, SEFAC and MUFAC shows delay disadvantage over SEFA at load 0.9 and above. On the other hand, as the offered load approaches 1, their difference becomes smaller. This may be due to the fact that the congestion mainly occurs at the FDL assignment in each switch module, rather than the route limitation through the CMs.

With similar loss and delay performance, MUFAC is more feasible than SEFAC due to its low time complexity. Additionally, SEFAC has scalability limitation, because its time complexity is linearly proportion to the switch size. On the other hand, MUFAC is capable of handling multiple packets at the same time, resulting in greater scalability.



Figure 10. Cell loss rate comparison of SEFAC and MUFAC



Figure 11. Delay performance comparison of SEFAC, MUFAC, and SEFA

Comparing single-stage switch and 3-stage Clos-network switch, single-stage architecture does take a slight lead over SFI-OCNS in terms of system performance. However, singlestage is non-scalable due to its hardware and scheduling complexity. To give a fair comparison, let us compare the hardware complexities between a 1024×1024 single-stage shared-FDL optical switch and a 1024×1024 3-stage SFI-OCNS. A 1024×1024 3-stage SFI-OCNS consists of 32 IMs, 32 CMs, and 32 OMs, each module has 32 inputs and 32 outputs; and there are 32 FDLs are employed at each IMs. Let us assume the each switch module is just simply cross-bar switch. As a result, there is total about 200 thousands crosspoints in a 1024×1024 3-stage SFI-OCNS. On the contrast, to build a 1024×1024 single stage shared-FDL optical switch with 1024 FDLs, require about 1 trillion cross-points. In addition, in a 1024×1024 3-stage SFI-OCNS, each IM only need to manage a transition diagram with 32 FDLs; in a single-stage shared-FDL optical switch, the system requires a huge transition diagram consists of all 1024 FDLs to process either SEFA or MUFA algorithm, which is just impractical. Thus, SFI-OCNS has compatible system performance to single-stage shared-FDL optical switch, and it enjoys low implementation complexity.

V. CONCLUSIONS

In this paper, we have proposed two fiber-delay-line (FDL) assignment algorithms for a 3-stage shared-FDL-IM optical Clos-Network switch: the sequential FDL assignment for SFI-OCNS (SEFAC) algorithm and the multi-cell FDL assignment for SFI-OCNS (MUFAC) algorithm. Overall, both algorithms can achieve low loss rate and low average delay. MUFAC is a more practical and scalable scheduling scheme than SEFAC for the SFI-OCNS.

REFERENCE

- [1]. M. C. Chia, et al., "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide gratings-based optical packet switches with WDM inputs-outputs," *IEEE J. Lightwave Technol.*, vol. 19, no. 9, pp. 1241-1254, September 2001.
- [2]. F. S. Choa and H. J. Chao, "All-optical packet routing architecture and implementation," *J. Photonic Network Commun.*, vol. 1, no. 4, pp. 303-311, 1999.

- [3]. M. J. Karol, "Shared-Memory Optical Packet (ATM) Switch," SPIE Vol. 2024: Multigigabit Fiber Communications Systems (1993), July 1993.
- [4]. S. Yao, B. Mukherjee, and S. Dixit, "Advances in photonic packet switching: an overview," *IEEE Communications Magazine*, vol. 38, no. 2, pp. 84-94, February 2000.
- [5]. J. Ramamirtham and J. Turner, "Time sliced optical burst switching," in *Proc. IEEE INFOCOM 2003*, San Francisco, April 2003.
- [6]. H. J. Chao and S. Y. Liew, "A New Optical Cell Switching Paradigm", International Workshop on Optical Burst Switching, Dallas, TX, Oct. 2003.
- [7]. S. Y. Liew and H. J. Chao, "Scheduling Algorithms for Shared-Fiber-Delay-Line Optical Cell Switches," *Optical Fiber Communications* (*OFC*), Los Angles, Feb. 2004.
- [8]. S. Y. Liew, H. J. Chao, and G. Hu "Scheduling Algorithms for Shared-Fiber-Delay-Line Optical Packet Switches, Part I: The Single-Stage Case" submitted to *IEEE J. Lightwave Technol.*.
- [9]. C. Clos, "A Study of Non-Blocking Switching Networks," Bell Sys. Tech. Jour., pp. 406-424, March 1953.
- [10]. Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The knockout switch: a simple, modular architecture for high-performance switching," *IEEE J. Select. Areas Communications*, vol. 5, no. 8, pp. 1274-1283, Oct. 1987.
- [11]. M. Karol, and C-L. I, "Performance Analysis of a Growable Architecture for Broadband Packet (ATM) Switching," *Globecom'89*, pp.1173-1180, 1989