Scheduling Algorithm and Evaluating Performance of a Novel 3D-VOQ switch

Ding-Jyh Tsaur[†], *Hsuan-Kuei Cheng*[††], *Chia-Lung Liu*[††], *and* *Woei Lin* [††]

[†]*Chin Min Institute of Technology, Mioali, 351 Taiwan, R.O.C.*
[††]*National Chung-Hsing University, Taichung, 402 Taiwan, R.O.C.*

Summary
This paper studies scheduling algorithms and evaluates the performance of high-speed switching systems. A novel architecture for three-dimensional Virtual Output Queue (3D-VOQ) switches is proposed with a suitable scheduling algorithm to improve the competitive transfer of service. This 3D-VOQ switch, which exactly emulates an output-queued switch with a broad class of service scheduling algorithms, requires no speedup, independently of its incoming traffic pattern and switch size. First, an N×N 3D-VOQ switch is proposed. In this contention-free architecture, the head-of-line problems are eliminated using a few virtual output queues (VOQ) from input ports and the output sides were arranged using sufficient separate queues. Next, a Small Time-to-leave Cell First (STCF) algorithm is proposed to generate a stable many-to-many assignment. Finally, analysis and simulation confirm the performance of 3D-VOQ and its satisfying high/low QoS requirements.

Key words:
QoS guarantees, Virtual Output Queue (VOQ), switching system, packet switching, 3D-VOQ.

1. Introduction

Network operators require high-capacity routers that give guaranteed performance. Many modern commercial switches and routers adopt output queueing. However, this approach is impractical for switches with high internal-line rates. If the speedup of the input queuing switch is increased by 1, then the speedup of output queuing switch requires increase by as much as N times [1]. By contrast, the required speedup of a combined input/output queued switch (CIOQ) is between 1 and N. The CIOQ switch via the MUFCA [2], CCF [3] and JPM algorithms [4] can emulate its speedup by 2 or 4 times to be simulated as an OQ switch.

Conversely, the required speed of the fabric and memory need be only as fast as the line rate for an input queued (IQ) switch, making input queueing very appealing for switches with fast line rates or many ports. However, although many scheduling algorithms [5]-[9] proposed for the input-queued architecture can achieve 100% asymptotic throughput, none of these algorithms perform as well as an output-queued switch. The main problem of IQ switching is head-of-line (HOL) blocking, which can severely affect the throughput. If each input maintains a single FIFO, then HOL blocking can reduce the throughput to only 58.6% [10]. Restated, HOL blocking can be eliminated entirely using a method known as virtual output queueing in which each input maintains a separate queue for each output. The throughput of an IQ switch can be increased up to 100% for independent arrivals [9].

Input buffered switches with VOQ can achieve 100% throughput [9,11], thus specifying the relationship of proper scheduling with high speed. Existing scheduling algorithms, such as PIM[11], DRR[12] and iSLIP[6], are based on matching parallel and iterative request-grant-accept cycles. However, these algorithms are impractical and extremely time-intensive. Even $\mathbf{O(N^2)}$ iterations may be possible in the worst case. Mei Yang [13] presented a CIOQ switch with Space-division multiplexing expansion by grouping input/output ports (SDMG CIOQ switch) that use extra hardware architecture to reduce the transfer pressure and competition for internal cells based on a space-division concept. This development trend is therefore considered feasible.

These studies were surveyed to discuss how many speedups are needed to emulate OQ switching and how to achieve 100% throughput. The research data indicates that at least two speedups are needed to emulate the OQ switch completely [3,4]. Therefore, the CIOQ switch can be used to emulate an OQ switch precisely. However, this method is not practical because for two reasons. First, speedup becomes necessary. However, handling a large number of packets by internal speedup of switch is impractical due to the ever-increasing speed of the Internet. Second, since the speedup is impractical, the decrease in switch speed can enable emulation of the OQ switching performance. The internal switch architecture requires the information of each packet needs to be updated. Additionally, the computation is very complex. In summary, the major difficulty with speedup is a hardware restraint. That is, high performance output switches are required to maintain the capability of transmitting a cell from an input port to an output port as expected. However, due to the physical line problem, VOQ must compete for ownership of the physical line, causing the possibility of contention.

Manuscript revised January 10, 2006.

Consequently, this study presents an access of using speedup method to avoid contention. Conversely, reducing contention also decreases the required number of speedup times. Therefore, to avoid contention without increasing the physical line, the transferring method of each cell requires more information to judge the method of transfer with respect to its algorithm, increasing the algorithm complexity.

This study proposes a novel three-dimensional Virtual Output Queue (3D-VOQ) switch architecture is ever proposed in our previous paper [14]. The physical lines between the input port and output port are increased without using additional hardware. A new algorithm, called Small Time-to-leave Cell First (STCF) Algorithm in this work, which is similar to that proposed by MUCFA [2] is proposed. Consequently, the proposed architecture reduces the amount of information that must be judged for transferring cells, and improves the calculation complexity. Furthermore, the proposed architecture eliminates the drawbacks resulting from the algorithm designed by using emulative OQ switch are resolved.

The next section presents the new switch architecture and composition with 3D drawing. Section 3 outlines the architecture's operational schemes and the operation of the scheduling/matching algorithms. Section 4 provides the system analysis and evaluation of simulation results. Section 5 summarizes the performance evaluation.

2. 3D-VOQ Switch Architecture

A 3D-VOQ switch architecture of size N consists of N input buffers at each input, M output buffers at each output, and an N-layer N×M crossbars switch fabric along with a scheduler for each layer. Figure 1 shows the N×N 3D-VOQ switch architecture.

This architecture, adopts N input buffers at each input port for virtual output queueing (VOQ) to eliminate a well-known head-of-line blocking. An N-Layer N×M crossbars switch fabric, abbreviated to a Layer-j-N×M crossbar, provides *N* ingress lines for the j^{th} queues of all inputs in all VOQs and *M* egress lines to output-j's *M* queues. The output buffer on some output port *j* is divided into Multi Output Queues, which are called M-OQ(*j*,*k*) where *j* is in the range 1–N, and *k* is in the range 1–M. Each queue is accessed only once in each time slot without speedup. Each input/output queue is pushed into an arbitrary-out (PIAO) queue, in which cells are removed from the queue in an arbitrary order. Additionally, as illustrated Fig.1, input ports are arranged horizontally from left to right, and output ports are arranged vertically from top to down. The switch fabric based on N-layer N×M crossbars is shown in layers.

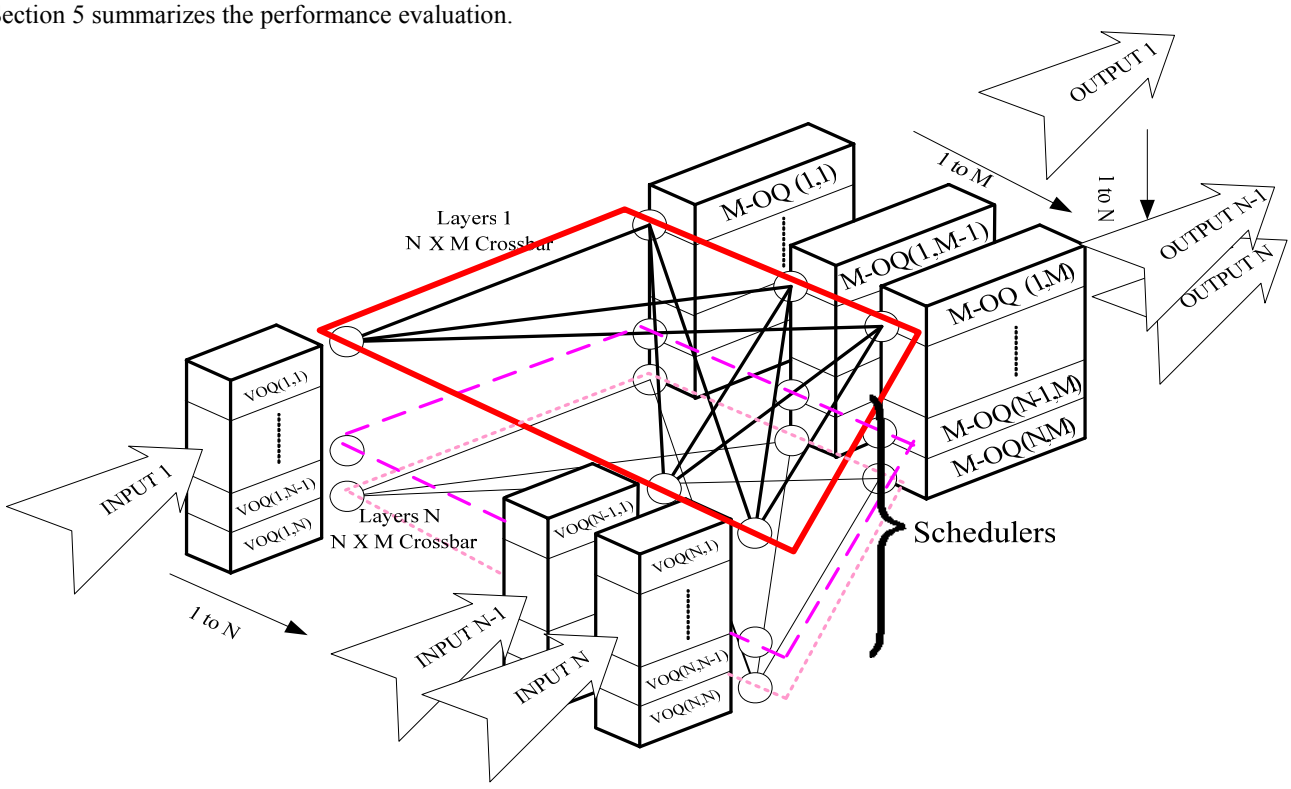


Fig. 1 3D-VOQ Architecture.

As previously noted, cells entering input ports can successfully reach output ports after completing two competitions: transferring VOQ to the switching fabric with internal lines (input contention), and competing for output ports with other input ports (output contention). Contention is described in terms of output *j*. Based on the switch fabric, among all input's VOQs, only those cells of the j^{th} queue are transmitted to (M-OQ(*j*,*k*), $1 \le k \le M$) via the Layer-*j*-N×M crossbar switch, without competing with those with different destinations. Only a single competition was required for the output ports. For example, all first-layer VOQs were transferred through the first-layer transfer line from VOQ(1,1) and VOQ(2,1) to VOQ(N,1), instead of competing with the other layer cells. An independent small switch was developed after the cells finished entering VOQ at every layer. A 3D-VOQ still cannot avoid output contention, so an efficient algorithm is required above the schedulers.

The schedulers employ a Central Control Unit (CCU), which is called i_CCU, located between the input port and the output port. This CCU decides the scheduling of cells transferring from the input port to the output port. An additional CCU, called o_CCU is placed on the output side to decide the sequence of cells leaving the switch. The next section describes the operating mechanism of the 3D-VOQ switch in detail.

3. Operation Schemes for 3D-VOQ

This section proposes a novel scheduling algorithm, called Smallest Time-to-leave Cell First (STCF), which enables a 3D-VOQ switch to emulate an OQ switch precisely without speedup for any input traffic.

3.1 STCF Algorithm

The proposed algorithm adopts centralized control to assign the switching to cells. By doing so, the algorithm can perform maximum matching with only one iteration. The centralized control unit is divided into two parts, *i*_CCU and *o*_CCU. First, the main aim of *i*_CCU is to determine which cells must be transferred between the input and output ports. *i*_CCU maintains three tables, *i_table*, *o_table* and *g_table*. Tables *i_table* and *g_table* are matrices with size N×N, where N represents the number of input/output ports. Each row in the matrix denotes an output port, while each column denotes an input port. The *o_table* is the N × (M+1) matrix, where rows denote output ports, and each column denotes the M-OQ(*j*,*k*) status. The column of M+1's is the sum of each row of o_table.

The steps in the internal operation of *i*_CCU and *o*_CCU are explained below. First, three steps are described below in response to the operating steps of *i*_CCU:

- **Initialize:** Set all elements of *i_table* and *g_table* to ∞ before the 3D-VOQ switch working. The N × M matrix in the *o_table* is initialized to one, indicating that the queue is currently available. Column M+1's is initialized to the sum of each row of *o_table*(value at *M*), i.e., the available size of each output. Figure 2 showes the initialize of a 3×3 3D-VOQ switch.

$$\begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} X & Y & Z \end{array} \\ \begin{bmatrix} \infty & \infty & \infty \\ \infty & \infty & \infty \\ \infty & \infty & \infty \end{bmatrix} \end{array} \quad \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{cccc} X & Y & Z & Sum \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \\ 1 & 1 & 1 & 3 \end{bmatrix} \end{array} \quad \begin{array}{c} \\ A \\ B \\ C \end{array}\begin{array}{c} \begin{array}{ccc} X & Y & Z \end{array} \\ \begin{bmatrix} \infty & \infty & \infty \\ \infty & \infty & \infty \\ \infty & \infty & \infty \end{bmatrix} \end{array}$$

(a) i_table (b) o_table (c) g_table

Fig. 2. the initialize i_CCU of a 3×3 3D-VOQ switch (t=0)

(1) **Request:** Each input sends requests of HOL cell of VOQ TL (Time to Leave) value to *i*_CCU. If the HOL cell of VOQ is empty, then the request is not sent to *i*_CCU. The corresponding element of *i_table* is recorded as ∞ in the matrix.

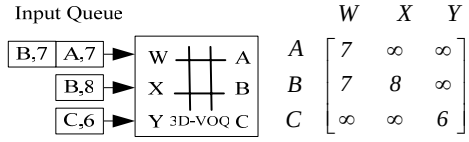
(2) **Grant:** Decide whether time slot *t* of input port's cell can be transferred to M-OQ(*j*,*k*) after receiving the requests from input. First, check all elements of *o_table*. If *o_table*(*j*,*k*) = 1, then M-OQ(*j*,*k*) corresponding to output port *j* its k^{th} *queue* is available (ON). By contrast, if *o_table*(*j*,*k*) = 0, then the corresponding queue of M-OQ(*j*,*k*) is unavailable (OFF). The value of sum(M+1 col.) in *o_table* indicates the number of cells to be transferred simultaneously. Second, sort each row in *i_table*. The results of this sorting are matched with *o_table* to determine: (1) how many cells can be transferred simultaneously by summing *o_table*, and (2) which input cells can be sent to M-OQ(*j*,*k*). Sorting sequences of transferring to M-OQ(*j*,*k*) by value (1 or 0) of *o_table* generally result in a level of complexity given by ***O(nlogn)***, where *n* denotes the switch size. Third, the grant value in *g_table* must be filled when *i*_CCU finishes its computation results. However, to avoid overlap with the input port grant value, the subsequent reply can not send the duplicate to the input port of each layer. This method is applied to avoid collision. Finally, *g_table* sends the grant back to the input port. If the grant value (1≤grant value≤N) is valid, then the cell is transmitted to M-OQ(*j, grant value*).

(3) **Update:** Each M-OQ(*j*,*k*) sends a feedback to *i*_CCU at the next time slot. The feedback represents the state of each M-OQ(*j*,*k*), and can initialize *i_table* and *g_table*. The feedback can be used to make a correct decision for next run.

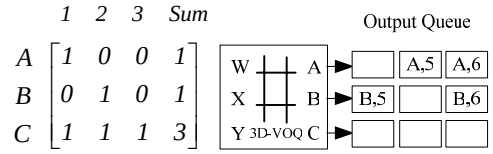
Figure 3 illustrates a snapshot of a 3×3 3D-VOQ switch at time slot *t*. The entire hardware design needs to be

Manuscript revised January 10, 2006.

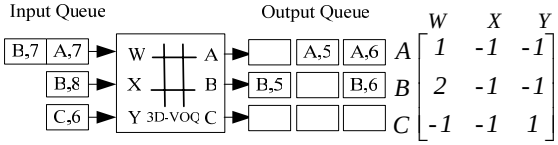
explained before any details can be explained for Fig. 2. The hardware architecture of *input queue* is a virtual output queue with a queue length of one, while the *output queue* is M-OQ(j,k), $1 \le j \le 3$, $1 \le k \le 3$, with a queue length of 1. Additionally, output ports 1, 2 and 3 denote A, B and C, and input ports 1, 2 and 3 denote W, X and Y, respectively. Cell (**P**,**T**) is destined to output port **P**, which depart in time slot **T** according to the emulated OQ switch. Figure 3(a) illustrates that *i_table* receives a request from input. Since the input queue structure is VOQ, all four cells are HOL. Hence, four cells must be delivered from input to output. The *i*_CCU received requests from HOL cell of each VOQ, which can be recorded as TL in the *i_table* matrix.



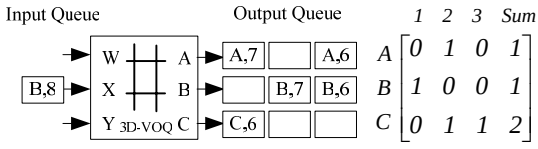
(a) i_table snapshot at time slot t (*request state*)



(b) o_table snapshot at time slot t (*grant state*)



(c) g_table snapshot at time slot t (*grant state*)



(d) o_table snapshot at time slot t+1(*update state*)

Fig.3 a snapshot of a 3×3 3D-VOQ switch (t=5)

Figure 3(b) demonstrates that *o_table* records the ON/OFF state of each M-OQ(j,k). Column (M+1) contains the sum of each row. Here, output A and output B can receive only one cell, and output C can receive three cells at time slot t (i.e., M-OQ(1,1), M-OQ(2,2) and M-OQ(3,1~3) are available).

Figure 3(c), the row of *i_table* was sorted and used to compare with the *o_table*. We can obtain *g_table* to respond to each VOQ of input. For example, (B,7) and (B,8) are located in the same row in *i_table*, but output B only receives one cell (row B of *o_table* has only one queue in ON state, or the sum of *o_table* is 1). Cell (B,7) is chosen to transmit to output after sorting *i_table*, because cell (B,7) is a smaller TL than cell (B,8). The *g_table* is calculated after comparing two tables, and *i*_CCU responds with grant to input. The HOL cell (A,7) receives grant value 1; cell (B,7) receives grant value 2, and cell (C,6) receives grant value 1. The grant value is the number of the k^{th} output queue. Therefore, Cell (A,7) must be delivered to output A at the 1st output queue A (M-OQ(1,1)); Cell (B,7) must be delivered to output B at the 2nd output queue B (M-OQ(2,2)), and cell (C,6) must be delivered to output C at the 1st output queue C (M-OQ(3,1)). If the input and output do not match, then *i*_CCU responds with grant value −1.

In Fig. 3 (d), after matching cells, *o_table* requires the latest state of M-OQ(j,k) at time slot t. The M-OQ(j,k) feeds back the state (ON/OFF) at the next time slot to update the *o_table* of *i*_CCU in order to make a correct decision for the next run. The graph of Fig. 3 plots the result after running by the 3D-VOQ switch.

The operation mechanism of *o*_CCU, which is much simplified than that of *i*_CCU. Only one N × N table, called *d_table* representing departure, is required. The mechanism has two steps: (1) **request:** each HOL cell of M-OQ(j,k) sends a request to *d_table* of the *o*_CCU, and (2) **grant:** after receiving requests from each M-OQ(j,k), *o*_CCU only sends one grant to each output port. Only one cell for each output port can be departed from a switch in one time slot. To do it by each output port, the minimum entries in all rows can be searched in parallel. With parallel search, the complexity of computing the *d_table* of the *o*_CCU is ***O(n)***.

3.2 Analyze algorithm and Proof

A 3D-VOQ switch with our proposed STCF algorithm can be shown to emulate OQ switch exactly without speedup.

***Definition* 1:** Input Priority List (**IPL**): Each VOQ maintains an ordered list of all queued cells, which can be ordered according to various input ordering schemes.

***Definition* 2:** Output Priority List (**OPL**): Each M-OQ(j,k) also maintains an ordered list. Each queue has an associated OPL, which sets the departure order of cells from these queues.

***Definition* 3:** The "time to leave" for cell c, **TL(*c*)**, is the time slot at which c leaves the shadow OQ switch.

***Definition* 4:** The "output cushion of a cell c", given by **OC(*c*)**, is the number of cells waiting in the output buffer at cell c's output port with less time to leave than cell c.

***Definition* 5:** The "input thread of cell c", given by **IT(c)**, is the number of cells of smaller time-to-leave than cell c in the input side.

***Definition 6*:** The "slackness of cell *c*", **L(c)**, equals the difference between the output cushion and input thread of cell *c*, i.e., L(*c*) = OC(*c*) − IT(*c*)

The proceeding definitions are similar to those presented in [11]. The following analysis proves that a 3D-VOQ switch based on the proposed STCF algorithm can emulate an OQ switch exactly without speedup.

***Lemma* 1**: No input/output contention occurs in the 3D-VOQ switch when $N = M$, where N denotes the switch size, and M is the number of (M-OQ(j,k), $1 \le j \le N$, $1 \le k \le M$) in STCF algorithm with any traffic.

Proof: The input contention occurs when more than two cells transmit simultaneously from the same input port. Fortunately, the 3D-VOQ architecture means that the transmission of cell becomes an independent operation, since each VOQ has its own dedicated physical line and each VOQ can correspond to each different output port independently. Consequently, the remaining cells are not affected by other cells in the same input port. Therefore, as long as cell need to be transmitted from the input port in VOQ, the transmission can certainly be executed successfully. That is, the probability of giving contention with other cells in the same input port is zero.

Although the hardware architecture can be applied to resolve input contention, a more effective algorithm is required to resolve the issue of output contention.

***Lemma 2*:** A cell with smaller time-to-leave value is transmitted from a 3D-VOQ (N=M) switch much faster than a cell of greatly time-to-live value using the STCF algorithm.

Proof: Assume that a 3D-VOQ switch has two cells under STCF algorithm. Cell p is transferred from input i to output j and has a lower TL value than Cell q, in that it delivers from input k to m.

The proof is provided by contradiction. Cell p of TL is assumed to be less than Cell q, which however leaves the 3D-VOQ switch before cell p.

***Theorem 1*:** A 3D-VOQ switch that uses STCF algorithm should have identical behavior to an OQ switch under any traffic.

Proof: *Lemma 1* and *Lemma 2* show that the time needed for a cell to enter and depart 3D-VOQ can be found if the designated TL has the same sequence and time as the sequence and time of the OQ switch. Thus, cells entering the 3D-VOQ and OQ switches always have identical behavior.

***Lemma 3*:** The slackness L of Cell c waiting on the input is never less than zero in any time slot.

Proof: Set the slackness of c to L at the start of a time slot. The three phases are separated to obvert the change of IT(c) and OC(c).

***Theorem 2*:** Regardless of the incoming traffic pattern, a 3D-VOQ switch using STCF without speedup can emulate an OQ switch exactly.

Proof: Assume that the 3D-VOQ switch has successfully emulated the OQ switch up until time slot t−1. Consider the beginning of time slot t (arrival phase). We must indicate that any cell reaching its time to leave is either (i) already at the output side of the switch, or (ii) to be transferred to the output during time slot t. *Lemma* 3 shows that the slackness L of cells waiting on the input is never non-negative. Consequently, if a cell has reached its time to leave (i.e., its output cushion and input thread both equal zero), then either (i) it is already at its output, and may depart on time, or (ii) it is at the HOL and has the smallest TL. In case (ii), the STCF algorithm is guaranteed to transfer the cell to its output during the time slot, so the cell departs on time.

4. System Simulation and Performance Analysis

The 3D-VOQ cell delay waiting in an input queue was first analyzed. Then, the performance of the OQ, CIOQ and 3D-VOQ switches were analyzed under different traffic models. Finally, a simulation was performed to show the exactly emulative OQ switch and comparison of the 3D-VOQ with different Algorithm and PPVOQ switches. Additionally, a 3D-VOQ switch has the same output buffer size as an OQ switch. Finally, the 3D-VOQ switch can support quality of service (high/low priority).

4.1 Analysis of 3D-VOQ Delay

The switch interconnection is an architecture of N layers and N×M crossbar switches. The model and switch analysis are based on the following assumptions:

- The switch operates synchronously.
- Cells arrive at the beginning of a time slot, and depart only at the end of a time slot.
- The arrival of cells follows the independent and identically distributed (i.i.d.) Bernoulli process, and cell destinations are uniformly distributed over all outputs.
- Every VOQ in an input has the same buffer size B.
- The M-OQ queues of output ports are assumed to consist of M queues with a fixed size L.

Under these assumptions, the probability of a request is first derived from a VOQ, which is denoted as Ps and which can be successfully serviced by the scheduler. The request is created from a VOQ at a rate φ. As cell competition on 3D-VOQ switch occurs at input ports, the calculation of theoretical values becomes focused on input

ports. Equations (1)–(6) show the theoretical derivation of various states within the switch, and their convergence values are obtained by iterative calculation. The following are lists of notations and their corresponding formulae.

(1) **λ**: offered load for every VOQ.

(2) **ρ**: occupancy of VOQ, namely, the probability of cells staying at VOQ. The calculation formula is as follows:

$$\rho = \frac{\lambda(1-Ps)}{Ps(1-\lambda)} \quad (1)$$

(3) **φ**: probability required by VOQ to transfer the cells, namely, the probability of possessing cells within VOQ.

$$\varphi = \lambda + \rho - \lambda\rho \quad (2)$$

(4) Blocking rate: the probability that the cells are blocked at VOQ when they enter the switch.

$$B_{(N*M\_switch)} = \frac{1}{N\varphi}\sum_{i=m+1}^{N}(i-m)\frac{N!}{i!(N-i)!}\varphi^{i}(1-\varphi)^{N-i} \quad (3)$$

(5) Service rate: the rate of unblocking that cells successfully send out from the input ports. The formula is as follows:

$$Ps_{(VOQ\_service)} = 1 - B_{(N*M\_switch)} \quad (4)$$

(6) VOQ delay: B denotes the length of a queue. According to the well-known M/M/1/B model [16], the VOQ delay results are as follows:

$$D_{(Delay\_VOQ)} = \frac{\rho\left[1-(B+1)\rho^{B}+B\rho^{B+1}\right]}{(1-\rho^{B+1})(1-\rho)} \quad (5)$$

(7) Throughput: the throughput per input port is the total number of N queues sent out with request rate φ plus a successful service rate Ps, i.e., throughput of every input port can be given as follows:

$$T = N * \varphi * Ps \quad (6)$$

Calculations for throughput and VOQ delay proceed as follows:

(1) At the beginning of the first calculation, λ=φ.
(2) The blocking rate B(N×M_switch) is derived from Eq. (3)
(3) The service rate Ps is derived from Eq. (4)
(4) The occupancy ρ of the Virtual Output Queue is obtained from Eq. (1)
(5) The required probability φ at next time slot is derived from Eq. (2)
(6) Repeat procedures (2) ~ (5) until all data are converged.
(7) The VOQ delay and throughput is achieved by the occupancy ρ of convergence.

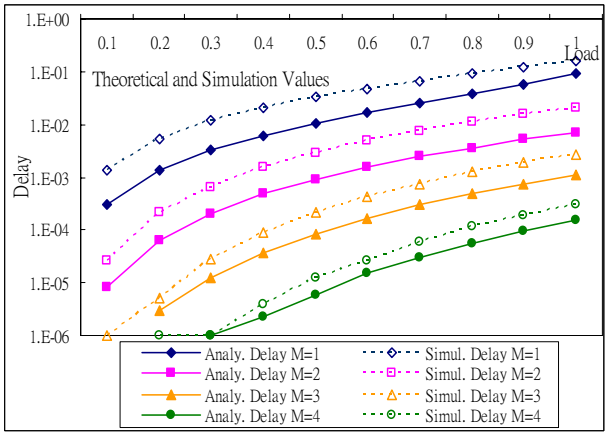


Fig. 4 Theoretical values and Simulation Values

Figure 4 compares the numerical results obtained from the proposed analysis model in comparison with simulation results under switch size of N=16 and length of VOQ=10, and shows that the average delay in simulations has a higher value than that derived from Eqs. (1) to (5), because theoretical values are calculated under optimum state. In this graph, the queueing delay in input buffers decreases as the M value of M-OQ rises. Significantly, the average delay of a cell in input buffers is less than 10^{-4} in a 3D-VOQ switch where M-OQ = 4. That is, only one in every 10^4 cells is delayed in input buffers.

4.2 Traffic model

To distinguish the effects among the OQ, 3D-VOQ and CIOQ switches under different traffic, the simulations used three traffic models: Bernoulli arrivals traffic, on-off traffic and polarized traffic.

Bernoulli arrivals traffic and on-off traffic

Each input in the on-off traffic model can alternate between active and idle periods of geometrically distributed durations. During an active period, cells destined for the same output arrive continuously in consecutive time slots. The probability that an active or an idle period ends at a time slot is fixed. Let p and q denote the respective probabilities for an active and an idle period. The duration of an active or an idle period is geometrically distributed and expressed as follows:

$$\Pr[\text{An active period} = i \text{ slot}] = p(1-p)^{i-1}, i \geq 1,$$
$$\Pr[\text{An idle period} = j \text{ slot}] = q(1-q)^{j}, j \geq 0.$$

Notably, an active period, called a burst, is assumed to contain at least one cell. The mean burst length is given by

$$b = \sum_{i=1}^{\infty} ip(1-p)^{i-1} = \frac{1}{p} \quad (7)$$

and the offered load ρ denotes the portion of time that a time slot is active:

$$\rho = \frac{\frac{1}{p}}{\frac{1}{p} + \sum_{j=0}^{\infty} jq(1-q)^{j}} = \frac{q}{q+p-pq} \quad (8)$$

Significantly, the Bernoulli arrival process is a special case of the bursty geometric process where $p+q=1$ [17], since the destination addresses of bursty traffic are distributed randomly.

Polarized traffic

Polarized traffic is a non-uniform, locally unbalanced but globally balanced traffic pattern, defined as follows [18]. Let $d_{i,j}$ denote the proportion of traffic received by $VOQ_{i,j}$, and define q as the polarization factor with

$$d_{i,j} = \frac{q^{(i+j) \bmod N/g} \cdot (q-1)}{q^{N/g} - 1} \quad (9)$$

such that

$$\forall i \in [1..N/g], \sum_{j'=1}^{N/g} d_{i,j'} = 1, \forall j \in [1...N/g], \sum_{i'=1}^{N/g} d_{i'j} = 1 \quad (10)$$

where $q \geq 1.00$. Polarized traffic with $q=1.00$ is uniform traffic. Both uniform and polarized traffic can easily be shown to satisfy the SLLN condition and no input/output is oversubscribed.

4.3 Performance Evaluation of OQ, CIOQ and 3D-VOQ switches

Figure 5 depicts Bernoulli arrival traffic, on-off traffic (with b=10 in Eq.(7)) and polarized traffic (with q=2 in Eq.(9)). The figure shows that the throughput of 3D-VOQ and OQ switches better than those of the CIOQ switch.

Figure 6 illustrates the average cell delay under different traffic load. OQ switch and 3D-VOQ switch signify the highest value regarding the average cell delay. This result arises from the fact that when input cells need to be transmitted to a specific output port and only one cell is allowed to leave the switch output port at one timeslot. The average cell delay of CIOQ switch is smallest. But, the average cell delay under Bernoulli traffic is greater than that under busty traffic for CIOQ switch. This phenomenon is opposite to that observed from both OQ switch and 3D-VOQ switch, which can be attributed to the drop rate. In Fig. 7, it can be seen that CIOQ switch has the highest drop rate. This finding results from the fact that VOQ of input sides can only provide limited memory space. So, cells will be dropped before they enter the input sides, and then average cell drop rate will be increased as well.

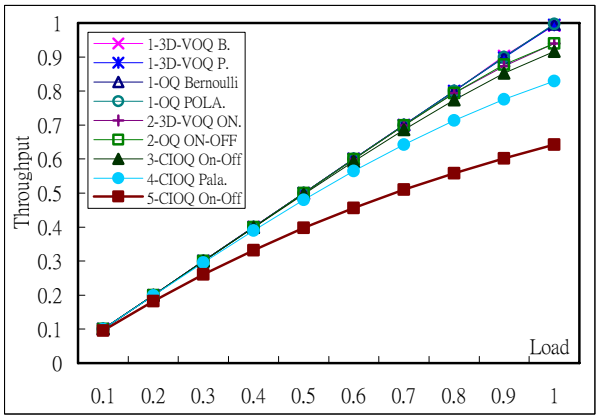


Fig. 5 Comparison of throughput among OQ, 3D-VOQ and CIOQ switch

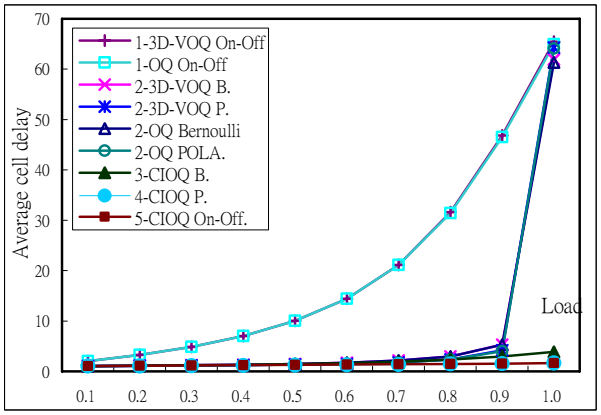


Fig. 6 Comparison of delay among OQ, 3D-VOQ and CIOQ switch

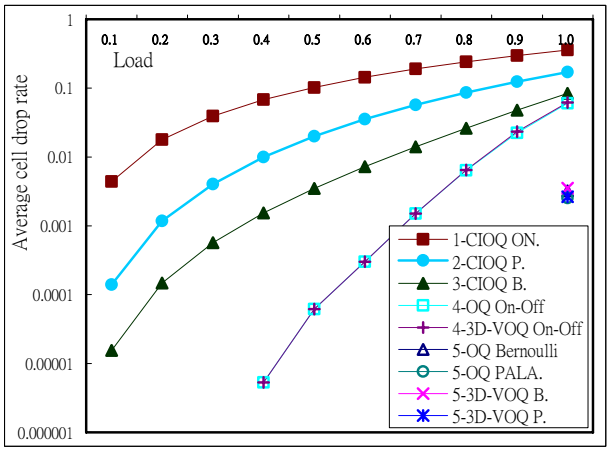


Fig. 7 Comparison of drop rate among OQ, 3D-VOQ and CIOQ switch

4.4 3D-VOQ Emulate OQ Switch by Simulation

Section 3 analyzes the STCF algorithm, which can emulate an OQ switch via a 3D-VOQ switch. This section simulator was used to show cells entering the 3D-VOQ and OQ switches. Each point in Fig.7, denotes one cell. Figure 8 reveals that the cell with ordering will has identical delay times in both the OQ and 3D-VOQ switches in every case. In conclusion, for the same traffic to enter into 3D-VOQ switch and OQ switch their behaviors are identical.

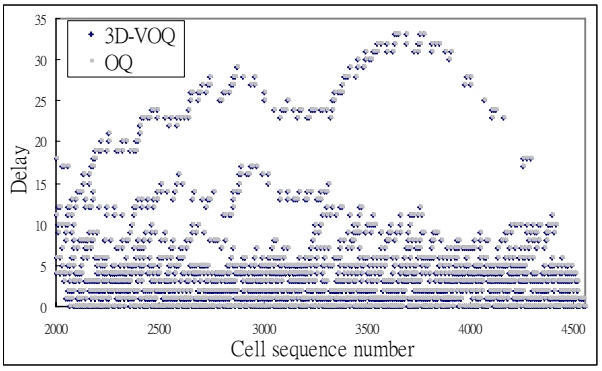


Fig . 8 Comparison of conditions of OQ and 3D-VOQ cell when entering and departing switch

4.5 Comparison of 3D-VOQ and PP-VOQ Performance

This section evaluates the performance of 3D-VOQ and PPVOQ [1][15]. The STCF algorithm and Parallel-Polled algorithm were also considered for 3D-VOQ switches. As shown in Fig.9, the 3D-VOQ switch achieves smaller average cell delay than the PPVOQ switch with sufficient memory space and without drop-offs. As the load approaches 1.0, the average cell delay of the 3D-VOQ is about half that of the PPVOQ switch. This finding confirms that the performance of the 3D-VOQ switch is better than that of PPVOQ switch. Additionally, all work-conserving scheduling algorithms were found to achieve the same level of performance on the 3D-VOQ switch.

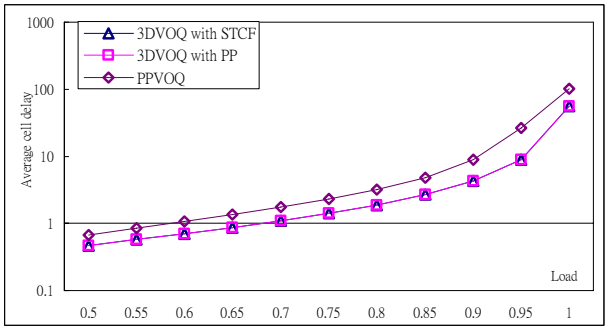


Fig. 9 Comparison of 3D-VOQ and PPVOQ

4.6 Estimating Performance with Different Output Queue

The number M of MOQ influence the cells that are simultaneously sending the output port at the same time, affecting the performance of the 3D-VOQ switch.
The effect of the M value of M-OQ(j,k) on the performance of the 3D-VOQ switch is studied. Both the OQ switch and the 3D-VOQ are assumed to have the same amount of memory associated with output port. When M equals one, a 3D-VOQ switch is actually a CIOQ switch. Fig. 10 and 11 show the performance of a 32×32 3D-VOQ switch under bursty on-off traffic. When the number of M exceeds four, the 3D-VOQ switch performs as well as the OQ switch in both throughput and average cell delay. The CIOQ switch has a higher drop rate than the OQ switch under low and medium traffic load. However, the drop rate equals that of the OQ switch under heavy traffic.

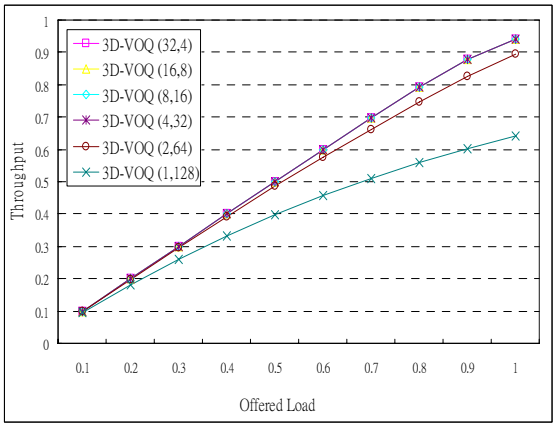


Fig. 10 Comparison throughput for var. M

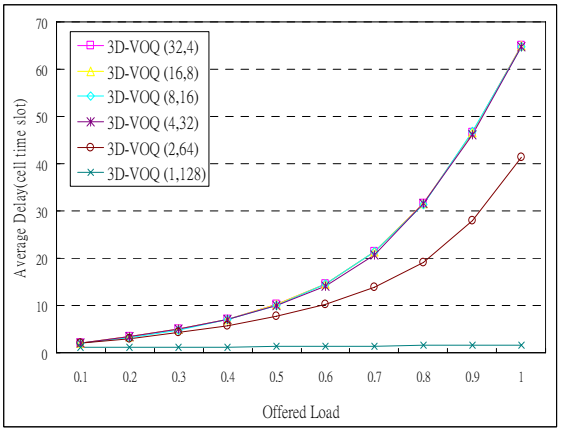


Fig. 11 Comparison average delay for var. M

4.7 Support QoS of 3D-VOQ

The concept of QoS was incorporated into 3D-VOQ to enable the switch to offer primary QoS. The QoS scheme employed in this study was dependent on strict priority, namely, the high-priority cells have lower delay times than low-priority cells in every case. Figure 12 depicts the conceptual diagram of 3D-VOQs of high/low priority, where the colored queues denote low priority and uncolored queues denote high priority.

The strict priority mechanism specified above reveals that 3D-VOQ switches are capable of screening different priorities. Figure 13, shows the delay results of two different cells simulated with this mechanism. The delay time for different cells of high/low priority varies significantly because the delay time of high- priority cells was shorter than that of low-priority cells.

Class of service is also easily implemented using o_CCU, when M-OQs employs the PIAO queues. Fig. 14 shows the delay results of four various cells simulated with this mechanism. Class 1 comprises the highest-priority cells with the shortest delay, and class 4 comprises the lowest-priority cells with the longest delay.

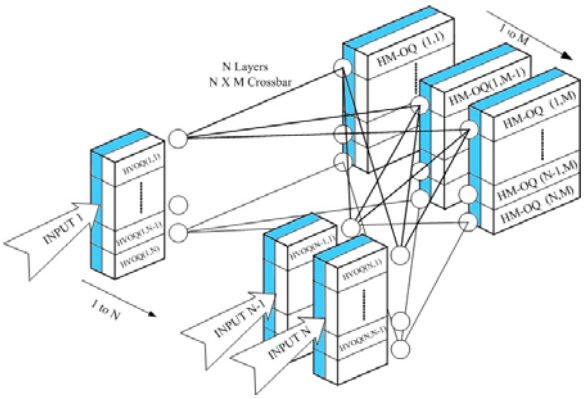


Fig. 12 Conceptual diagram of 3D-VOQ of High/Low Priority

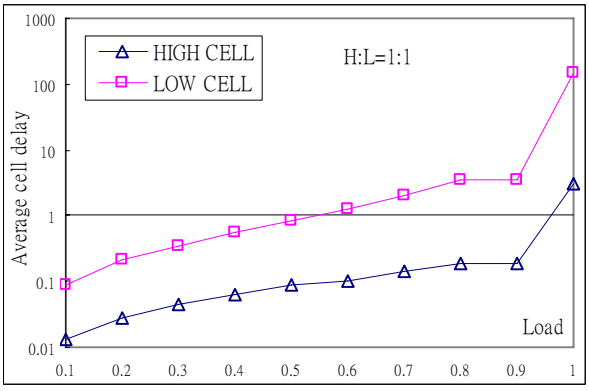


Fig. 13 Comparison of Delay of High/Low Priority

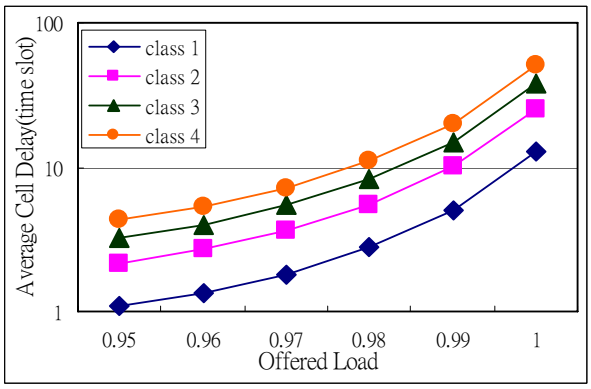


Fig . 14 Comparison of Delays associated with Class Priorities

5. Conclusions

The 3D-Virtual Output Queue switch adopts the 3D concept to improve the original switches of the plane structure. Moreover, packet switching within the switches can be operated efficiently, thus reducing competition within original switch architecture.

This study shows that the 3D-VOQ switch can emulate exactly an output queued switch with no speedup. This result holds for all arriving traffic patterns. That is, any size of switches and a broad class of service scheduling algorithms including FIFO, WFQ and strict priority queueing are applicable using this design. An N×N 3D-VOQ switch with sufficient separate output queues was found to make switching an input/output contention-free architecture. This study also proposes the STCF algorithm that can produce a stable many-to-many assignment. Additionally, the 3D-VOQ switch was found to be able to emulate an exact OQ switch.

The performance of 3D-VOQ was verified by analysis and simulation. Furthermore, the primary concept of QoS, of applying cells of different priorities and strict priority mechanism to achieve the desired packet switching, was incorporated into 3D-VOQ.

References

[1] Hyoung-Il Lee and Seung-Woo Seo, “Matching Output Queueing with a Multiple Input/Output-Queued Switch” INFOCOM 2004. IEEE International Conference on Hong-Kong, Volume:1 , 07-11 March 2004

[2] B. Prabhakar, N. Mckeown, “On the speedup required for combined input and output queued switching”, Automatica, Vol. 35, 1999.

[3] S. T. Chuang, A. Goel, N. Mckeown, B. Prabhakar, “Matching output queueing with a combined input output queued switch”, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 6, pp. 1030-1039, June 1999.

[4] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch", Proc. 6th IEEE/IFIP IWQoS'98, Napa Valley, CA, pp. 218-224, May 1998.
[5] N. Mckeown, "Scheduling Algorithms for Input-queued Cell Switches," Ph. D. dissertation, Univ. California at Berkeley, 1995.
[6] N. Mckeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches," IEEE/ACM Transactions on Networking, Vol. 7, No. 2, pp. 188-201, April 1999.
[7] R. O. LaMaire and D. N. Serpanos, "Two Dimensional Round-Robin Schedulers for Packet Switches with Multiple Input Queues," IEEE/ACM Trans. Networking, Vol. 2, pp. 471-482, Oct. 1994.
[8] H. J. Chao, "Saturn: A Terabit Packet Switch using Dual Round-Robin," IEEE Comunnications Magazine, Vol. 38, pp. 78-84, Dec. 2000.
[9] N. McKeown, V. Anantharam, J.Walrand, Achieving 100% Throughput in an Input-Queued Switch", INFOCOM '96, pp.296-302.
[10] M. Karol, M. Hluchyj, S. Morgan, Input Versus Output Queueing on a Space Division Switch", IEEE Trans. Comm, vol.35, no.12, pp.1347-1356, Dec. 1987.
[11] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High-Speed Switch Scheduling for Local-Area Networks," ACM Transactions on Computer Systems, Vol. 11, No. 4, pp. 319-352, November 1993.
[12] 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 Commun., vol. 5, no. 8, pp. 1274-1283, Oct. 1987.
[13] Mei Yang and S.Q. Zheng, "An Efficient Scheduling Algorithm for CIOQ Switches with Space-Division Multiplexing Expansion. " INFOCOM 2003. IEEE , Volume: 3 , 30 March-3 April 2003 pp:1643 - 1650
[14] Ding-Jyh Tsaur, Xian-Yang Lu, Chin-Chi Wu, Woei Lin "3D-VOQ Switch Design and Evaluation" IEEE 19th International Conference on Advanced Information Networking and Applications, vol.2, pp359-362, 2005.
[15] K. Yoshigoe and K. J. Christensen, "A Parallel-Polled Virtual Output Queued Switch with a Buffered Crossbar." 2001 IEEE Workshop,29-31 May 2001 pp:271– 275, 2001
[16] D. Gross and C. M. Harris, Fundamentals of Queueing theory, 3rd Edition. New York Wiley, 1998.
[17] G.D. Stamoulis, M.E. Anagnostou, and A.D. Georgantas, "Traffic models for ATM networks: a survey," Computer Commun., Vol. 17, No. 6, pp. 428-438, June 1994
[18] J. Blanton, H. Badt, G. Damm, and P. Golla, "Impact of polarized traffic on scheduling algorithms for high speed optical switches", ITCom2001, Denver, August 2001

Ding-Jyh Tsaur received the M.S. and Ph.D. degrees in Computer Science from National Chung-Hsing University in 1995 and 2005, respectively. He joined the Department of Information Management, Chin Min Institute of Technology, Miao-Li ,Taiwan, in 1995, where he is now an associate Professor. His research interests include computer network, switching system and interconnection network.

Hsuan-Kuei Cheng received the B.S. degree from National Defense University, Chung Cheng Institute of Technology in 2003, and M.S. degree in Computer Science from National Chung-Hsing University in 2005. His research interests include computer network, switching system and Parallel/Distributed System.

Chia-Lung Liu received the B.S. degree from TamKang University, Computer Science & Information Engineering in 2001, and M.S. degree in Computer Science from National Chung-Hsing University in 2003. Currently, he is working toward the Ph.D. degree in Computer Science at the same university. His research interests include computer network, switching system and Parallel/Distributed System.

Woei Lin received the B.S. degree from National Chiao-Tung University, Hsing-Chu, Taiwan in 1978, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from University of Texas of Austin, USA in 1982 and 1985, respectively. He is now a Professor in Institute of Computer Science, National Chung-Hsing University, During 1992-2005. His research interests include Network Switching System, Network QoS, System Performance Evaluation and Parallel/Distributed System.