CHAPTER 6

PROPOSED SCHEDULING ALGORITHMS FOR SYMMETRIC CROSSBAR SWITCH

6

PROPOSED SCHEDULING ALGORITHMS FOR SYMMETRIC CROSSBAR SWITCH

6.1 INTRODUCTION

In this chapter, we propose two new scheduling algorithms namely a modified Diagonal Propagation Arbiter (m-DPA) and Dynamic Scheduling Algorithm (DSA). In m-DPA, instead of using round robin priority scheme as DPA [51], we have changed the priority rotation dynamically based on queue occupancy to efficiently utilize the buffers. In addition to the standard modular cells, our approach requires extra hardware for priority order selection. Dynamic Scheduling Algorithm (DSA) modifies the priority rotation dynamically based on the two parameters: queue occupancy and quality of service requirement of input and output instead of three in Glimpse [19][20]. In DSA, normally queue occupancy and QoS weight are equal but as soon as individual queue in VOQs is full, its weight is doubled and it enjoys a maximum weight. DSA efficiently utilizes the buffers and at the same time gives better service to the selected inputs and outputs.

We present the simulation of 4x4, 8x8, 16x16, and 32x32 crossbar switches with m-DPA, DSA and other scheduling algorithms. The simulation results show that m-DPA saves loss of the cells due to buffer overflow and thus increases the throughput by 2% to 4% compared to DPA in case of different traffic distribution. The simulation results show that DSA saves loss of the cells due to buffer overflow and thus increases the throughput by 2% to 5% compared to DPA. The proposed DSA reduces the average latency for a given Quality of Service class input output and increases the average latency of other inputs and outputs.

We also present the design and implementation of 4x4 and 8x8 crossbar switches with m-DPA and DSA scheduling algorithms and compare it with DPA algorithm in terms of area requirement.

6.2 NEED OF m-DPA and DSA

Symmetric scheduling algorithm, named Diagonal Propagation Arbiter (DPA) is used for configuring the crossbar in input-queued switches that support Virtual Output Queuing (VOQ). Round robin priority rotation scheme in DPA gives good fairness to all the cells, but performs average under bursty arrivals and non-uniformly distributed traffic as discussed in chapter 5. Assume that there are three VOQ buffers are full in diagonal 3 as shown in figure 6.1. Now if it has lowest priority than other diagonal then diagonal 3 will lost the new incoming cells. So in bursty traffic arrival case there is no point to rotate the priority in round robin.

Quality of Service has been one of the important areas in networking and DSA supports QoS along with buffer occupancy. So high QoS input output always have an edge over other input output. Low priority input output data can be taken care by using buffer occupancy status.

6.3 m-DPA ARCHITECTURE

Suppose buffer capacity in each queue is ***k***. If buffer size of any VOQ reaches ***k*** then any packet now enters into the cell will lost. So we convert that cell into red-zone as shown in figure 6.1. During each time slot, numbers of cells whose buffers are in the red-zone for each diagonal group are calculated and priority is given to the group with maximum number of cells having red-zone buffers .

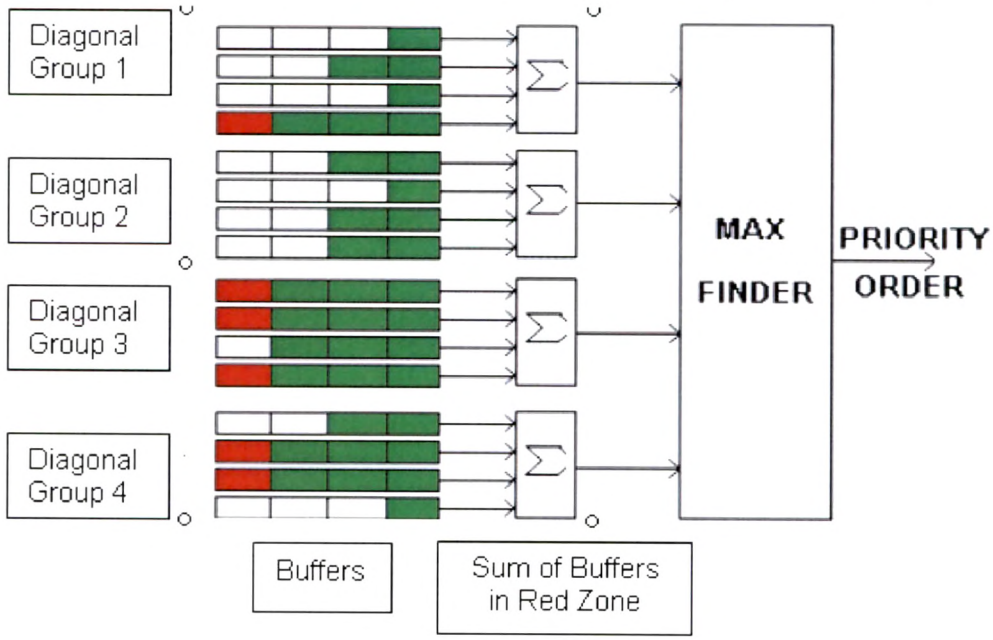

Figure 6.1 m-DPA architecture

In case of diagonal groups having equal number of red-zone cells are found, priority vectors will rotate as usual in round robin fashion. Thus, the most urgent diagonal group which is likely to loss the packet is routed through the switch first, and so on. This way, our scheme for priority order selection has a propensity towards saving cell loss in case of buffer overflows. Intuitively, benefit of the scheme will be puffed-up in cases of unevenly distributed and bursty traffics. Hence, throughput of the crossbar switch will be increased.

6.4 DYNAMIC SCHEDULING ALGORITHM ARCHITECTURE

In DSA the priority order is varied dynamically with respect to following corresponding parameters:

1. Queue occupancy (buffer size of each VOQ)
2. Quality of Service (QoS)

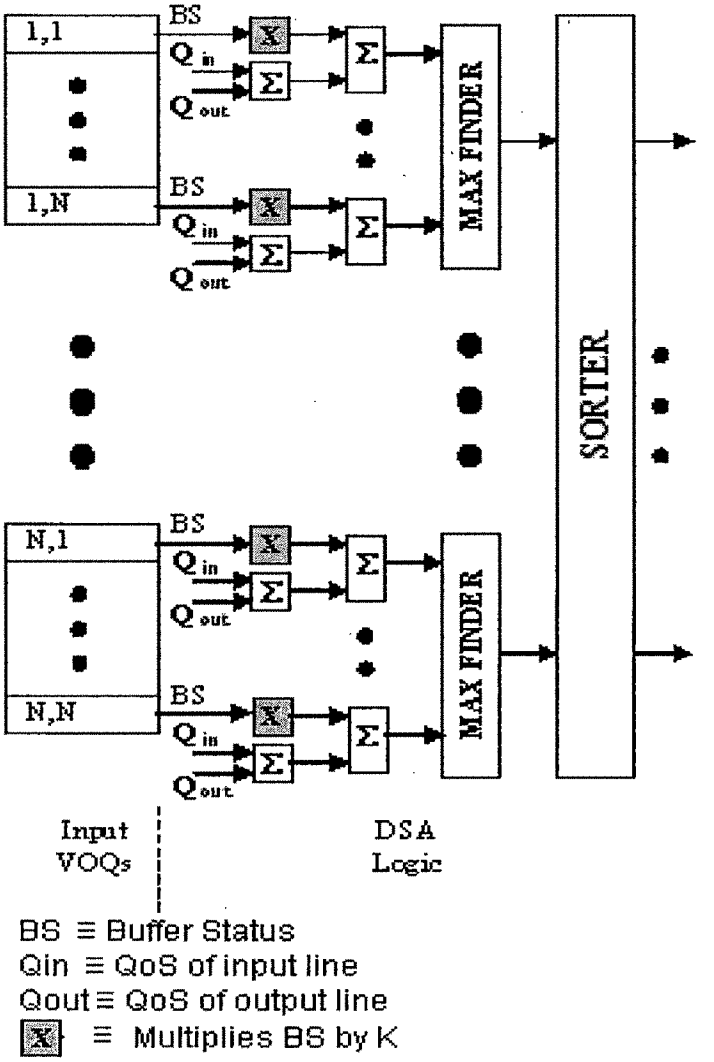

Figure 6.2 DSA Scheduler Architecture

Figure 6.2 depicts the architecture of DSA scheduler. QoS of a cross-point is obtained by summing QoS of corresponding input (Q_{in})and output(Q_{out}) lines. Buffer Status (BS) signal of a VOQ indicates the present percentage occupancy of the VOQ in terms of number of packets. It is multiplied by a variable factor (say K). This feature imparts dual behavior to the DSA. There are two cases; viz. BS < 100% and BS = 100%, for which K may assume one of the values given in table-6.1.

Table 6.1: Values of K

Case	K	Behavior
BS < 100 %	< 2	Strong capitalistic
	= 2	Fair
	> 2	Socialistic
BS = 100 %	> 2	Socialistic and Cautious

For simulation, analysis and implementation purposes, it is assumed that K = 2 when BS is less than 100% and K=4 when BS is equal to 100%. That is, when queues are not overflowing, QoS and queue occupancy enjoy equal weight (K=2). Thus, DSA decides priority in a dynamic order determined by QoS and queue occupancy. In this case QoS has considerable weightage hence it is a capitalistic approach. When any of the queues is fully occupied, i.e. it has reached into red zone [100] and is likely to lose packets, then; weight of queue occupancy for that particular queue is doubled (k=4) compared to weight of QoS. In this case QoS has less weightage than queue occupancy, hence it is a socialistic approach. After deciding weight of each VOQ in input port, we sort the VOQs in input port based on weight and arrange them in descending order. VOQs with the highest weight, from all the input ports are again sorted by second sorter. Based on the results, internal clocks are swapped and given to the grant block of each input, which decides grants. That is, DSA takes up a cautious step to avoid packet loss and to decrease packet latency by discouraging QoS and select socialistic approach. This dual behavior (normally capitalistic and when buffer is full socialistic) of DSA makes it more dynamic and robust to different traffic models.

6.5 MATLAB SIMULATION OF m-DPA AND DSA WITH OTHER SCHEDULING ALGORITHMS

We have simulated m-DPA and DSA with other algorithms for 4x4, 8x8, 16x16, and 32x32 crossbar switches with four different traffic models (A,B,C and D) using MATLAB 7.0 as we have done in chapter 5. Algorithms are simulated for 10000 time slots and the results are taken by averaging the outcomes for 100 simulations for 4x4 and 8x8 switches. Algorithms are simulated for 1000 time slots and the results are taken by averaging the outcomes for 10 simulations for 16x16 and 4 simulations for 32x32 switches and various parameters like throughput (efficiency), average latency and delay variance have been measured for variation in offered load as well as variation in buffer size. For variation in offered load the buffer size is 2 in 4x4, 3 in 8x8 and 16x16 switches, while buffer size is 4 in 32x32 switches. For variation in buffer size the offered load is 90% for 4x4 switch and 80% for 8x8, 16x16 and 32x32 switches.

6.5.1 4x4 Switch Comparison

A. With i.i.d. Bernoulli arrivals and uniformly distributed destinations:

Simulation results from 6.3 to 6.8 show that for this traffic model, throughput of m-DPA increases by 0.5% to 1.5% compared to DPA. Average latency and delay variance for m-DPA are at par with DPA. Throughput of DSA increases by 2.0% to 6.0% compared to DPA. Average latency and delay variance for DSA are at par with DPA.

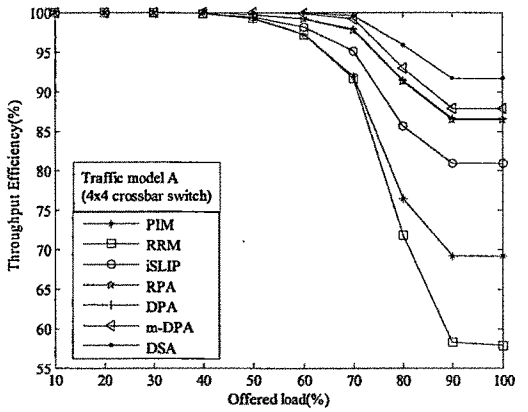

Figure 6.3 Throughput (%) v/s offered load

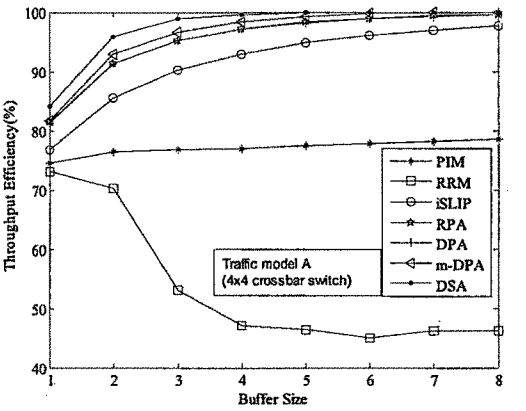

Figure 6.4 Throughput (%) v/s Buffer size

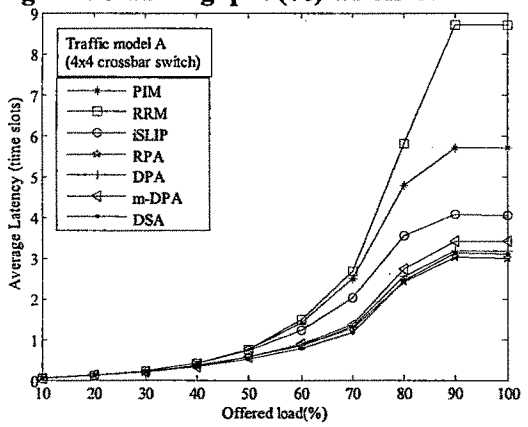

Figure 6.5 Average Latency v/s offered load

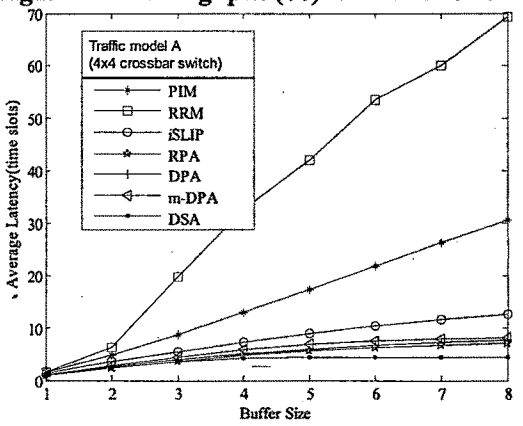

Figure 6.6 Average Latency v/s Buffer size.

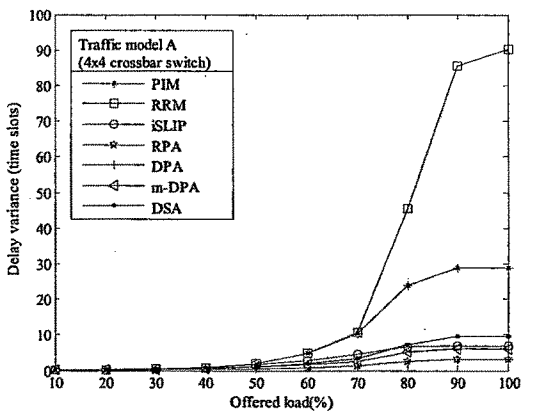

Figure 6.7 Delay variance v/s offered load

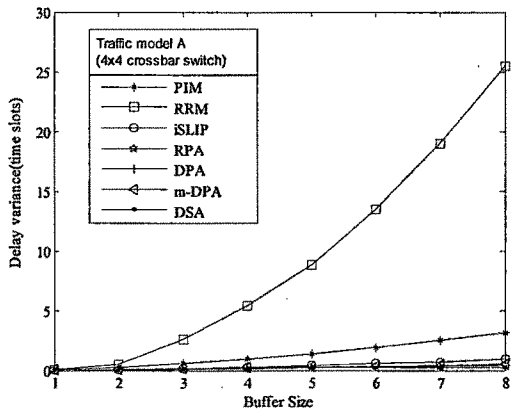

Figure 6.8 Delay variance v/s Buffer size.

B. *With i.i.d. Bernoulli arrivals and non-uniformly distributed destinations:*

Simulations are done for normally distributed destinations. Results are plotted from figure 6.9 to 6.14, throughput (efficiency) of m-DPA increases by 0.5% to 2.0% compared to DPA. Average latency increases by 1 to 2 timeslots in m-DPA compared to DPA for figure 6.11 and 6.12. Delay variance increases by 2 to 4 timeslots in m-DPA compared to DPA for offered load variation as shown in figure 6.13, but it increases by 10 to 140 timeslots for buffer size variation as shown in figure 6.14.

Throughput (efficiency) of DSA increases by 1.0% to 3.0% compared to DPA. Average latency increases by 1 to 3 timeslots in DSA compared to DPA for figure 6.11 and 6.12. Delay variance increases by 5 to 65 timeslots in DSA compared to DPA for offered load variation as shown in figure 6.13, but it increases by 20 to 150 timeslots for buffer size variation as shown in figure 6.14.

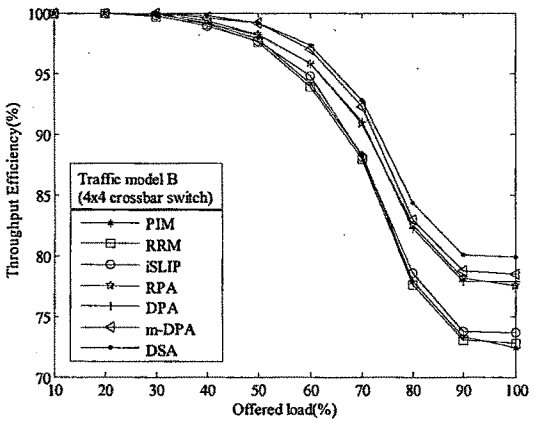

Figure 6.9 Throughput (%) v/s offered load

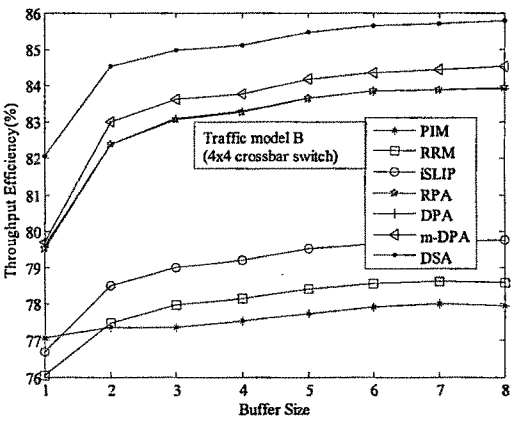

Figure 6.10 Throughput (%) v/s Buffer size

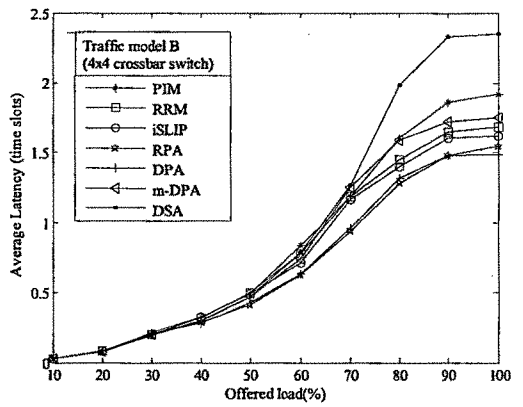

Figure 6.11 Average Latency v/s offered load

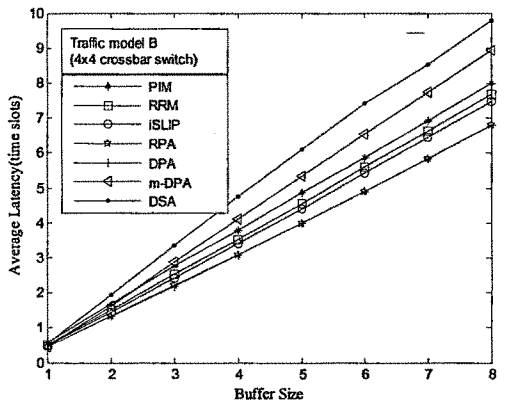

Figure 6.12 Average Latency v/s Buffer size.

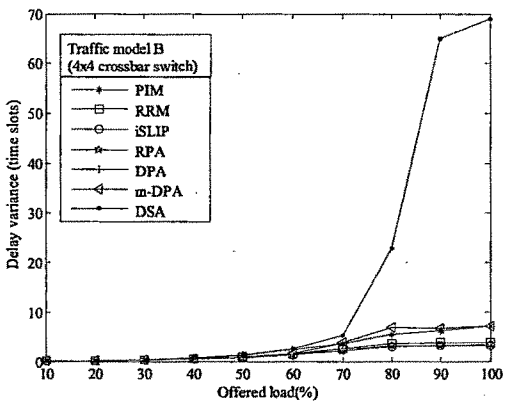

Figure 6.13 Delay variance v/s offered load

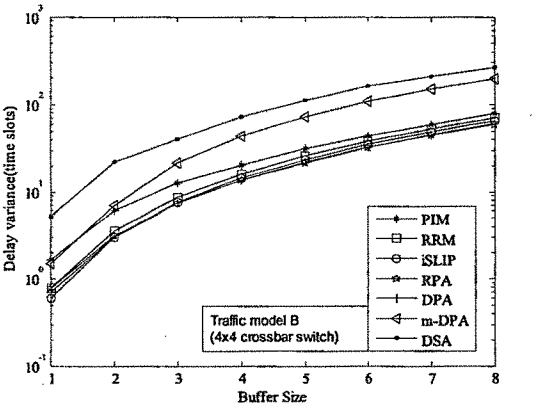

Figure 6.14 Delay variance v/s Buffer size.

C. With bursty arrivals and uniformly distributed destinations:

We illustrate the effect of burstiness on m-DPA and DSA using an on-off arrival process. Simulation results from figure 6.15 to 6.20 are shown below.

Results show that throughput (efficiency) of m-DPA increases by 0.5% to 1.5% compared to DPA for offered load variation as shown in figure 6.15. Average latency as well as delay variance of m-DPA are at par with DPA.

Throughput (efficiency) of DSA increases by 1.0% to 2.5% compared to DPA for offered load variation as shown in figure 6.15. Average latency as well as delay variance of DSA are at par with DPA.

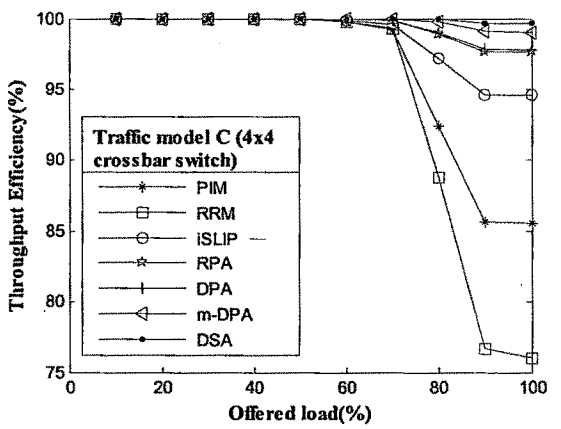

Figure 6.15 Throughput (%) v/s offered load

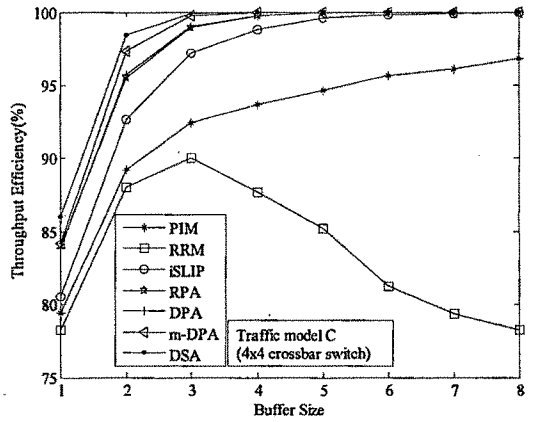

Figure 6.16 Throughput (%) v/s Buffer size

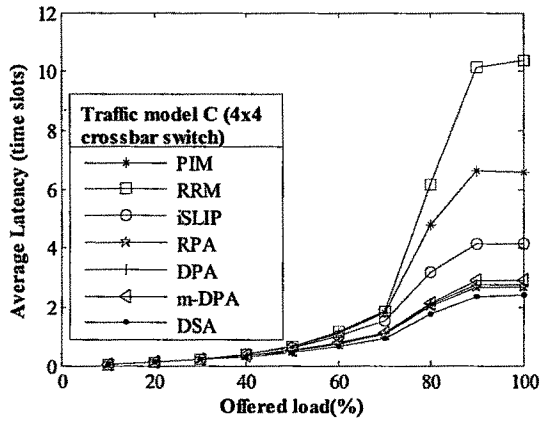

Figure 6.17 Average Latency v/s offered load

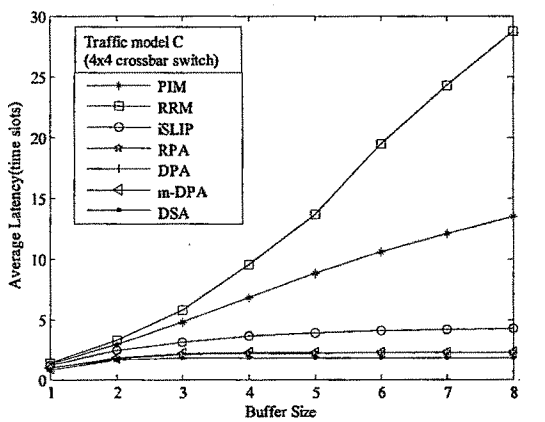

Figure 6.18 Average Latency v/s Buffer size.

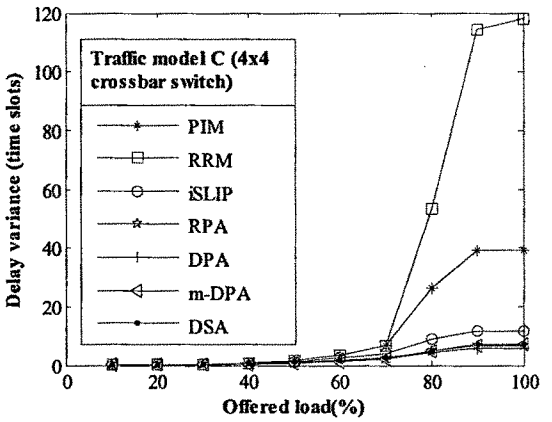

Figure 6.19 Delay variance v/s offered load

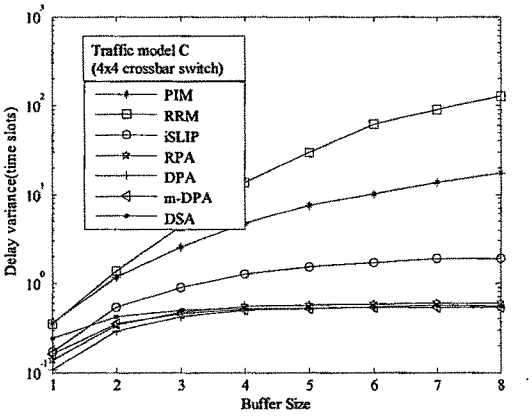

Figure 6.20 Delay variance v/s Buffer size.

D. *With bursty arrivals and non-uniformly distributed destinations*:

We illustrate the effect of burstiness as well as non-uniform distribution for output on DSA in this traffic model. Simulation results from figure 6.21 to 6.26 are shown below.

Throughput (efficiency) of m-DPA increases by 0.5% to 2.0% compared to DPA for offered load variation as shown in figure 6.21. Average latency as well as delay variance of m-DPA are at par with DPA.

Throughput (efficiency) of DSA increases by 0.5% to 2.5% compared to DPA for offered load variation as shown in figure 6.21. Average latency is at par with DPA but delay variance increases by 1 to 3 time slots.

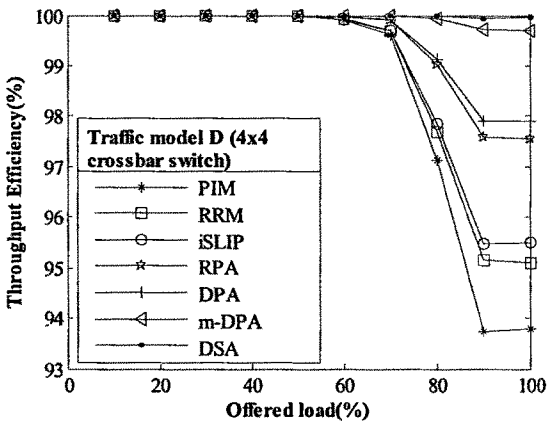

Figure 6.21 Throughput (%) v/s offered load

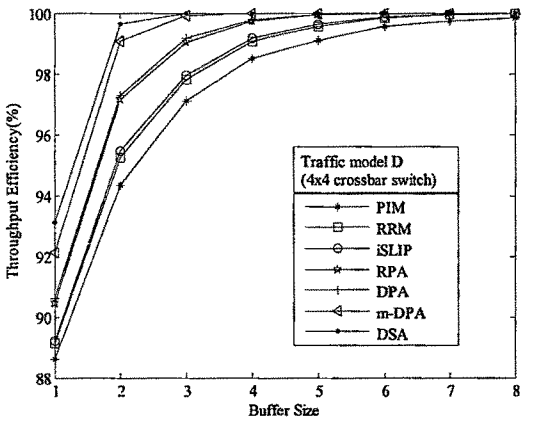

Figure 6.22 Throughput (%) v/s Buffer size

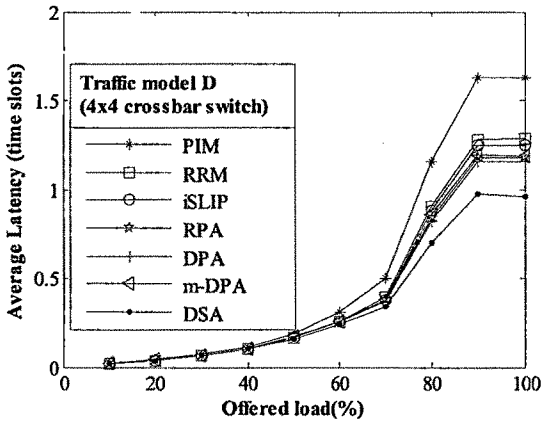

Figure 6.23 Average Latency v/s offered load

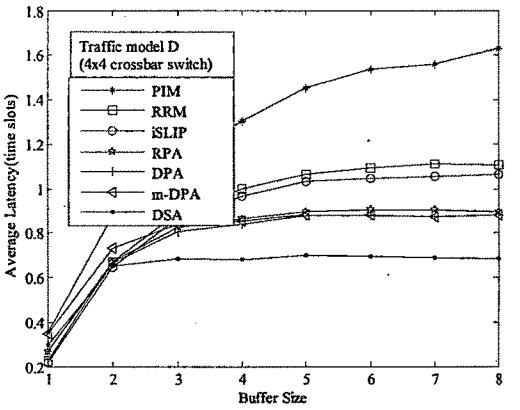

Figure 6.24 Average Latency v/s Buffer size

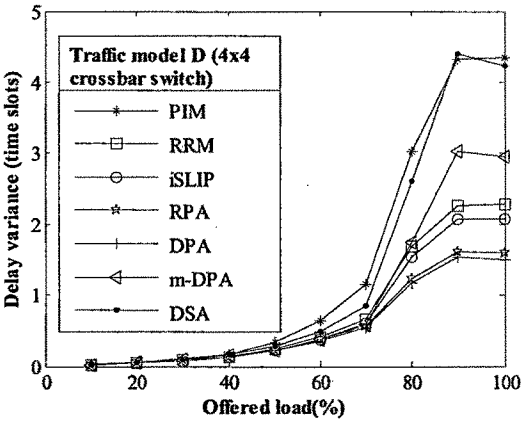

Figure 6.25 Delay variance v/s offered load

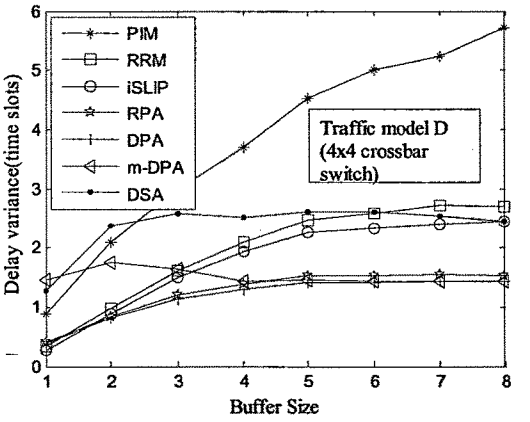

Figure 6.26 Delay variance v/s Buffer size.

6.5.2 8x8 Switch Comparison

A. With i.i.d. Bernoulli arrivals and uniformly distributed destinations:

Simulation results from figures 6.27 to 6.32 show that for this traffic model, throughput of m-DPA increases by 0.5% to 1.5% compared to DPA. Average latency and delay variance for m-DPA are at par with DPA.

Throughput of DSA increases by 1.0% to 4.0% compared to DPA. Average latency for DSA is at par with DPA. Delay variance in DSA is at par with DPA. In case of buffer size variation delay variance of DPA increase with increase in buffer size and it remains constant after buffer size 4 in DSA.

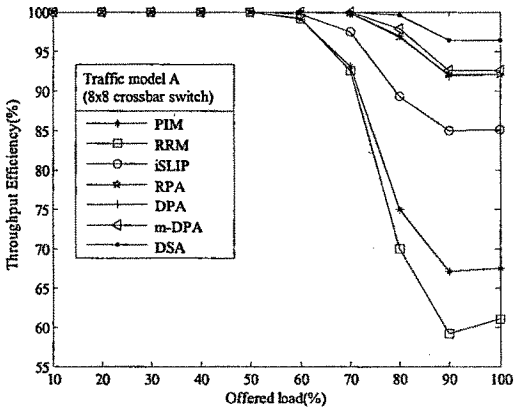

Figure 6.27 Throughput (%) v/s offered load

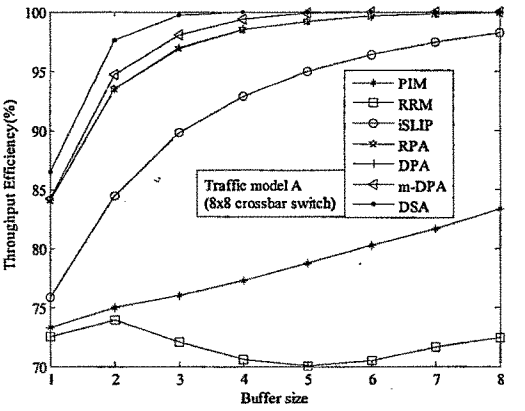

Figure 6.28 Throughput (%) v/s Buffer size

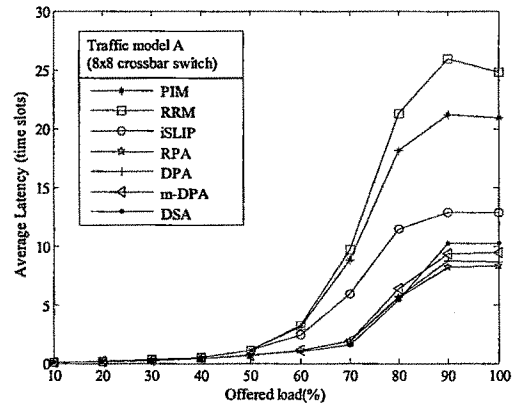

Figure 6.29 Average Latency v/s offered load

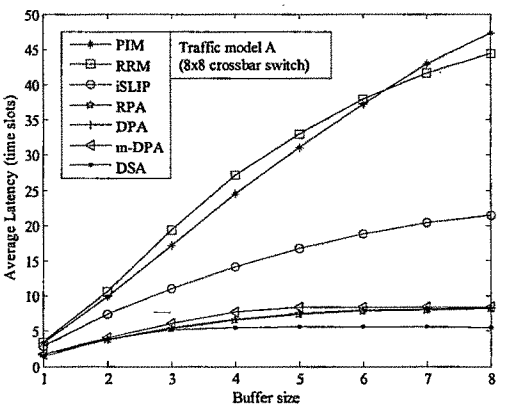

Figure 6.30 Average Latency v/s Buffer size

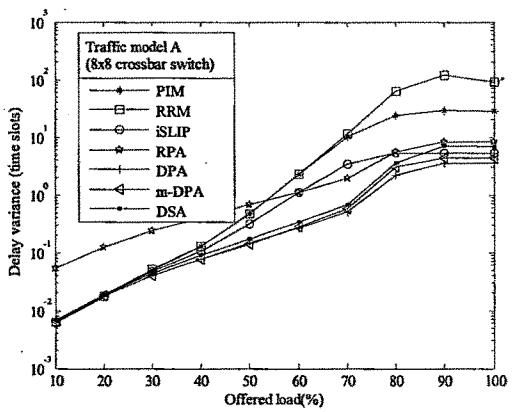

Figure 6.31 Delay variance v/s offered load

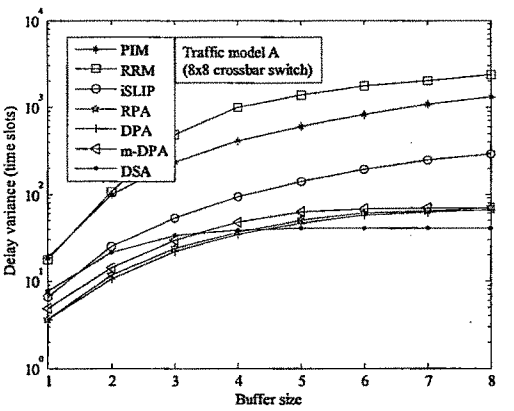

Figure 6.32 Delay variance v/s Buffer size.

B. *With i.i.d. Bernoulli arrivals and non-uniformly distributed destinations:*

Simulations are done for normally distributed destinations. Results are plotted from figure 6.33 to 6.38. Results show that for this traffic model, throughput (efficiency) of m-DPA increases by 1.0% to 2.0% compared to DPA. Average latency increases by 1 to 1.5 timeslots in m-DPA compared to DPA for figure 6.35 and 6.36. Delay variance increases by 5 to 27 timeslots in m-DPA compared to DPA

for offered load variation as shown in figure 6.37 but it increases by 10 to 70 timeslots for buffer size variation as shown in figure 6.38.

Throughput (efficiency) of DSA increases by 0.5% to 2.5% compared to DPA. Average latency increases by 1 timeslots in DSA compared to DPA for figure 6.35 and 6.36. Delay variance increases by 10 to 80 timeslots in DSA compared to DPA as shown in figure 6.37 and figure 6.38.

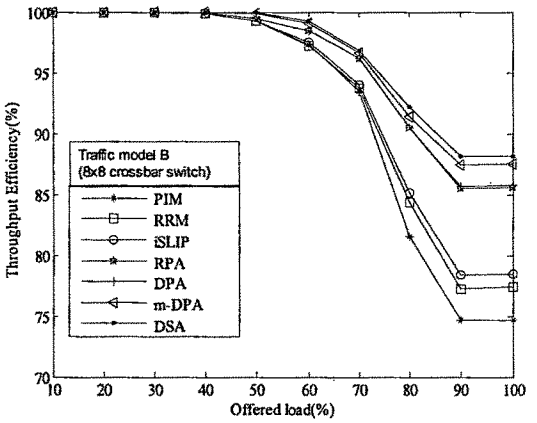

Figure 6.33 Throughput (%) v/s offered load

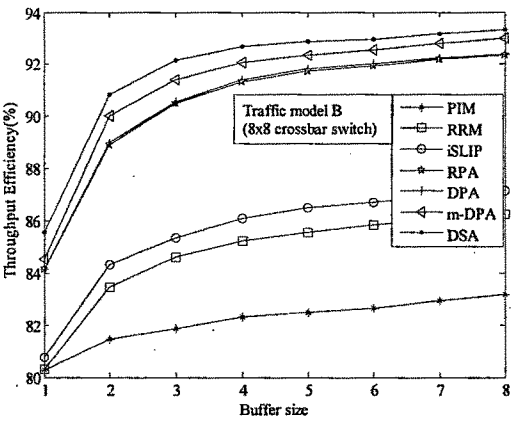

Figure 6.34 Throughput (%) v/s Buffer size

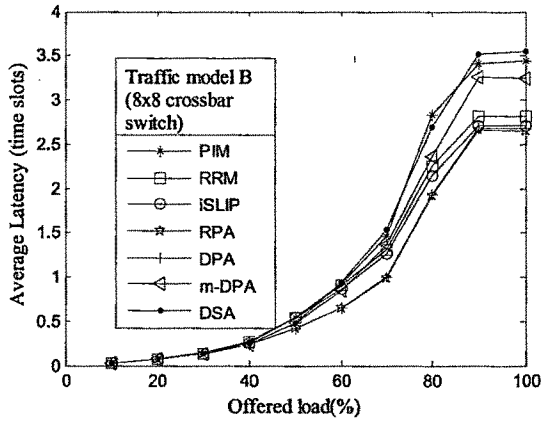

Figure 6.35 Average Latency v/s offered load

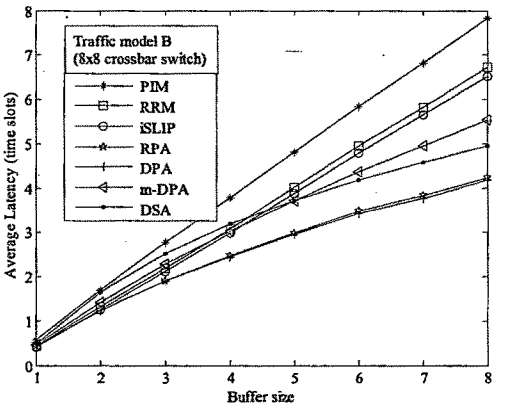

Figure 6.36 Average Latency v/s Buffer size

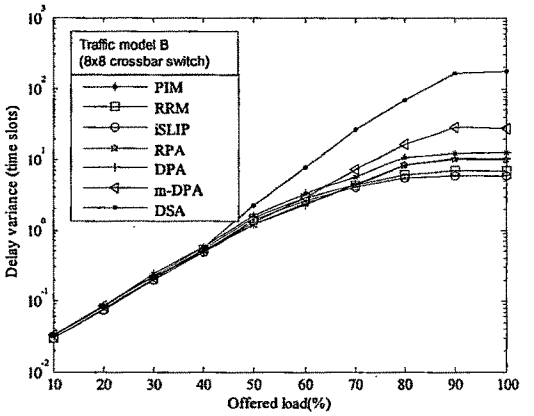

Figure 6.37 Delay variance v/s offered load

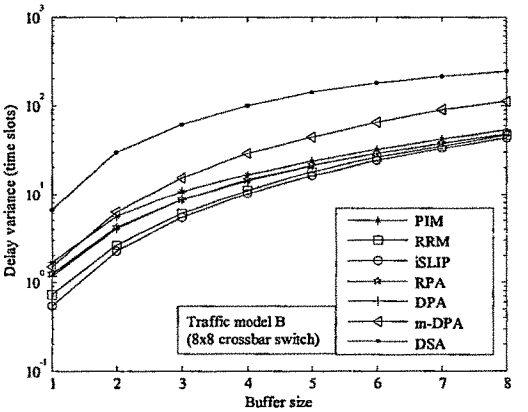

Figure 6.38 Delay variance v/s Buffer size.

C. With bursty arrivals and uniformly distributed destinations:

We illustrate the effect of burstiness on m-DPA using an on-off arrival process. Simulation results from figure 6.39 to 6.44 are shown below. They show that throughput (efficiency) of m-DPA increases by 0.5% to 1.5% compared to DPA for offered load variation as shown in figure 6.39. Average latency as well as delay variance of m-DPA is at par with DPA.

Throughput (efficiency) of DSA increases by 0.5% to 2.0% compared to DPA for offered load variation as shown in figure 6.39. Average latency of DSA is at par with DPA. Delay variance for DSA in case of offered load increases byl to 10 time slots while in case of buffer size variation it is at par with DPA.

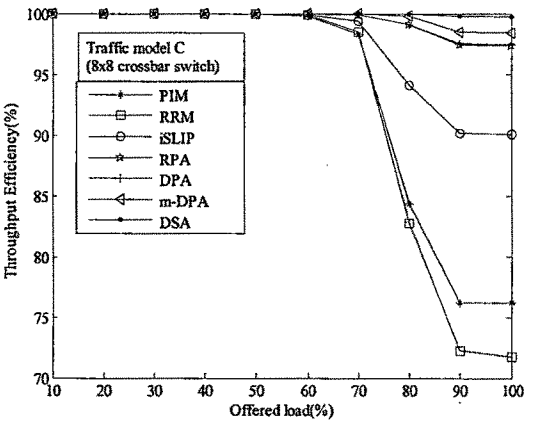

Figure 6.39 Throughput (%) v/s offered load

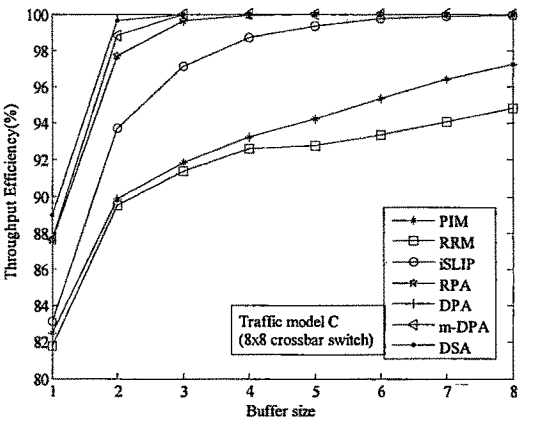

Figure 6.40 Throughput (%) v/s Buffer size

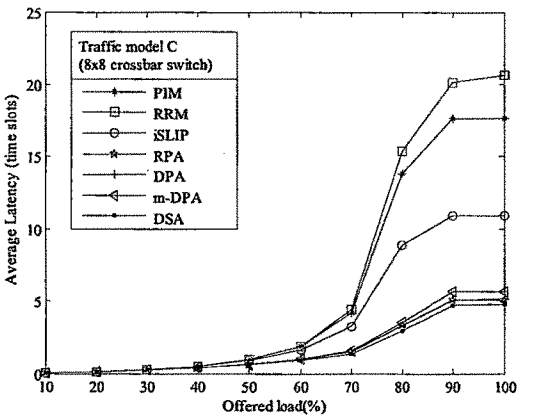

Figure 6.41 Average Latency v/s offered load

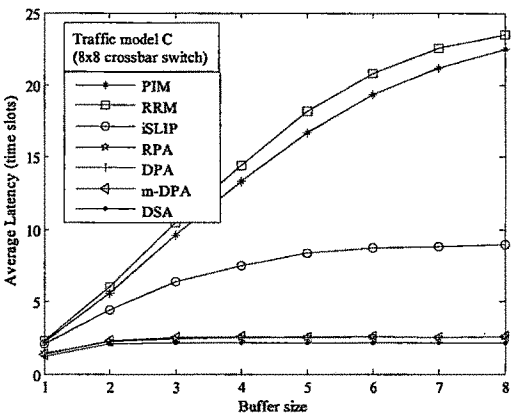

Figure 6.42 Average Latency v/s Buffer size

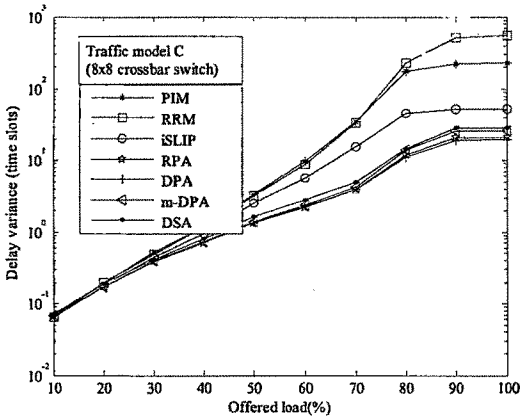

Figure 6.43 Delay variance v/s offered load

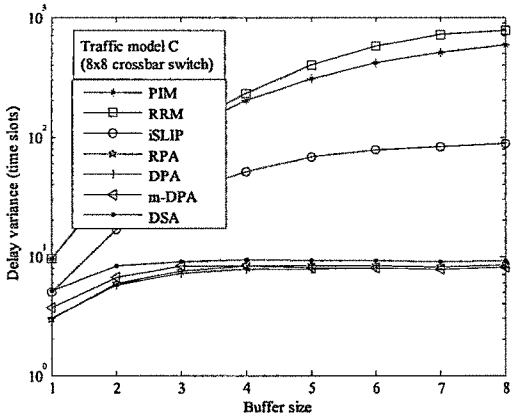

Figure 6.44 Delay variance v/s Buffer size.

D. *With bursty arrivals and non-uniformly distributed destinations*:

We illustrate the effect of burstiness as well as non-uniform distribution for output on m-DPA in this traffic model. Simulation results from figure 6.45 to 6.50 are shown below. They show that throughput (efficiency) of m-DPA increases by 0.5% to 2.0% compared to DPA for offered load variation as shown in figure 6.45. Average latency of m-DPA is at par with DPA while delay variance of m-DPA increases 1 to 2 time slots compared to DPA as shown in figures 6.49 and 6.50.

Throughput (efficiency) of DSA increases by 0.5% to 2.5% compared to DPA for offered load variation as shown in figure 6.45. Average latency of DSA is at par with DPA while delay variance of DSA increases 1 to 6 time slots compared to DPA as shown in figures 6.49 in offered load and at par in buffer size variation as shown in figure 6.50.

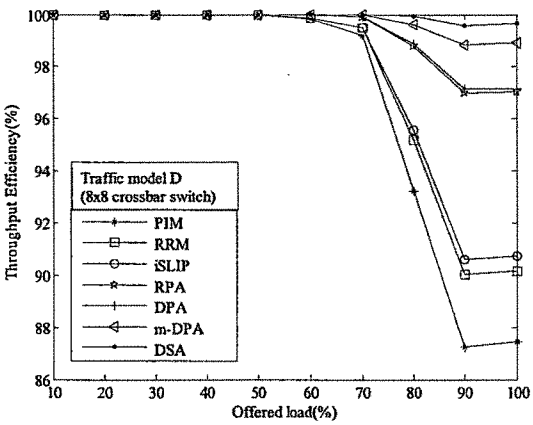

Figure 6.45 Throughput (%) v/s offered load

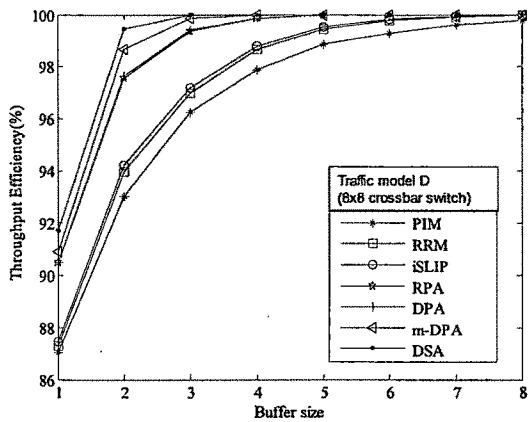

Figure 6.46 Throughput (%) v/s Buffer size

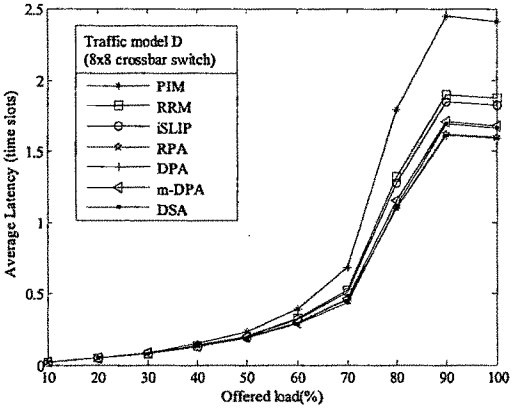

Figure 6.47 Average Latency v/s offered load

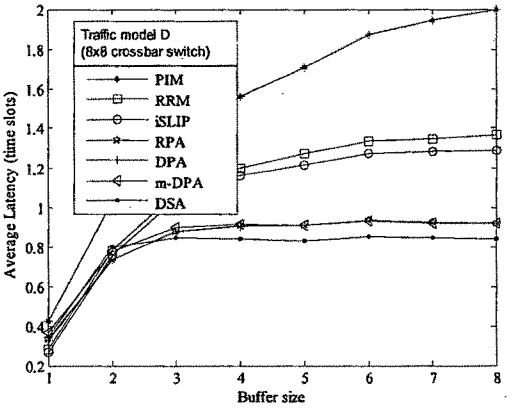

Figure 6.48 Average Latency v/s Buffer size

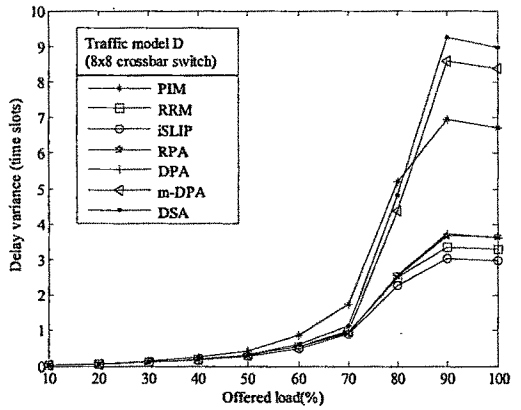

Figure 6.49 Delay variance v/s offered load

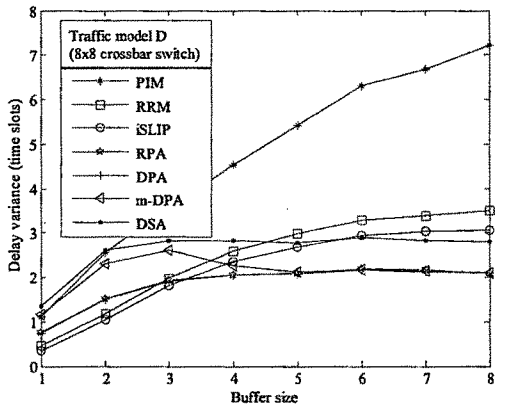

Figure 6.50 Delay variance v/s Buffer size.

6.5.3 16x16 switch Comparison

A. With i.i.d. Bernoulli arrivals and uniformly distributed destinations:

Simulation results from figures 6.51 to 6.56 show that for this traffic model, throughput of m-DPA increases by 0.5% to 1.0% compared to DPA. Average latency for m-DPA is at par with DPA and delay variance in case of offered load is at par with DPA but in case of buffer size delay variance increases by 5 to 30 time slots.

Throughput of DSA increases by 1.0% to 4.0% compared to DPA. Average latency for DSA increases by 1 to 4 time slots for offered load variation and at par with DPA for buffer size variation. Delay variance in case of offered load increases by 10 to 150 time slots for offered load variation and at par with DPA in case of buffer size variation

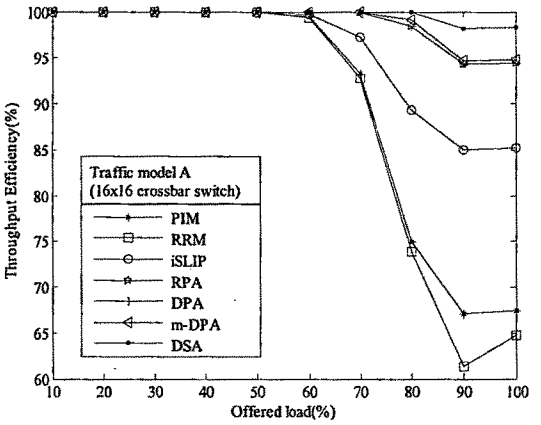

Figure 6.51 Throughput (%) v/s offered load

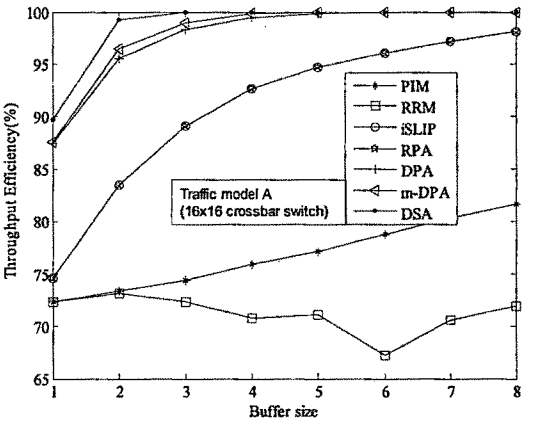

Figure 6.52 Throughput (%) v/s Buffer size

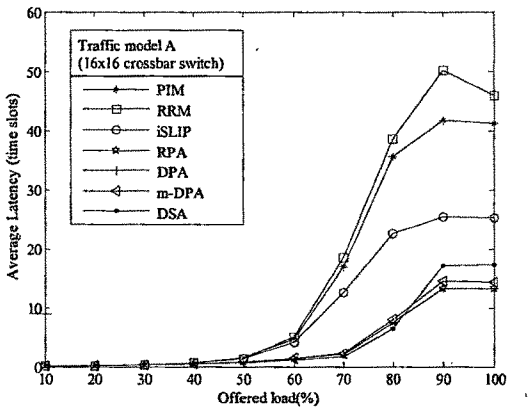

Figure 6.53 Average Latency v/s offered load

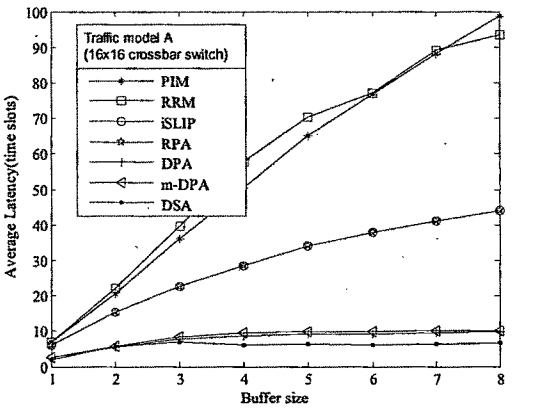

Figure 6.54 Average Latency v/s Buffer size

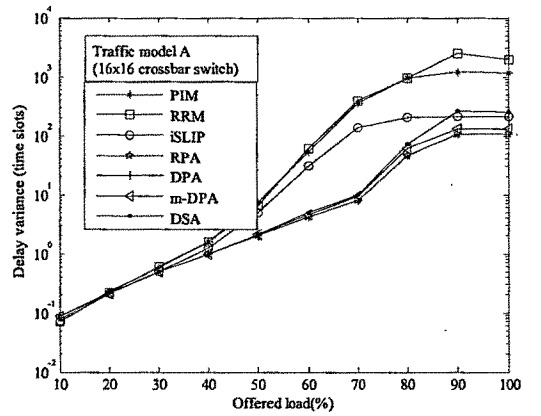

Figure 6.55 Delay variance v/s offered load

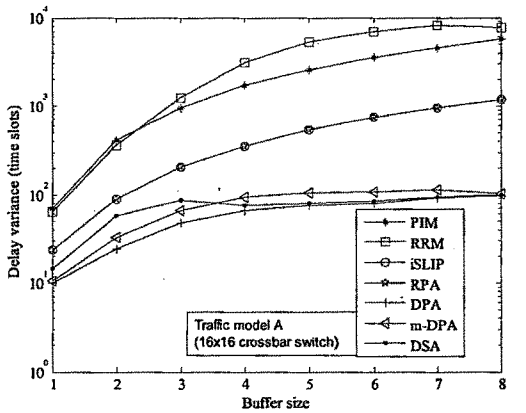

Figure 6.56 Delay variance v/s Buffer size.

B. *With i.i.d. Bernoulli arrivals and non-uniformly distributed destinations:*

Simulations are done for normally distributed destinations. Results are plotted from figure 6.57 to 6.62. Results show that for this traffic model, throughput (efficiency) of m-DPA increases by 0.5% to 1.0% compared to DPA. Average latency is at par with DPA as shown in figures 6.59 and 6.60. Delay variance increases by 2 to 4 timeslots in m-DPA compared to DPA for offered load variation as

shown in figure 6.61 but it increases by 5 to 60 timeslots for buffer size variation as shown in figure 6.62.

Throughput (efficiency) of DSA increases by 0.5% to 2.5% compared to DPA. Average latency is at par with DPA as shown in figures 6.59 and 6.60. Delay variance increases by 5 to 45 time slots for offered load variation and 10 to 75 time slots for buffer size variation in DSA compared to DPA as shown in figure 6.61 and figure 6.62.

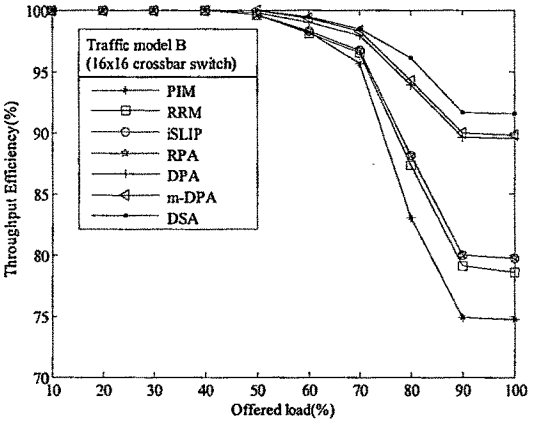

Figure 6.57 Throughput (%) v/s offered load

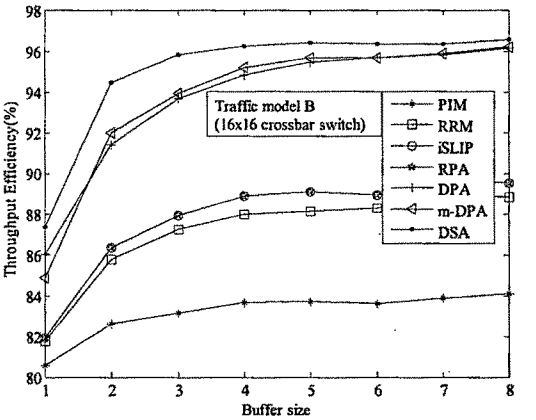

Figure 6.58 Throughput (%) v/s Buffer size

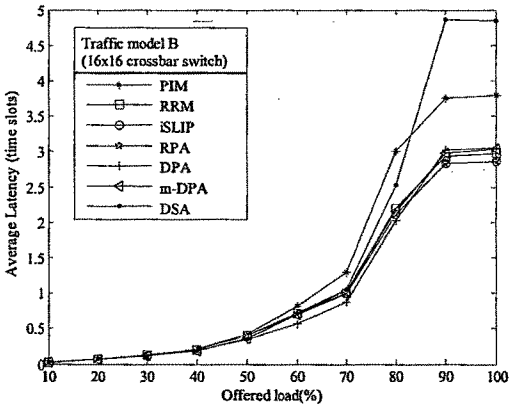

Figure 6.59 Average Latency v/s offered load

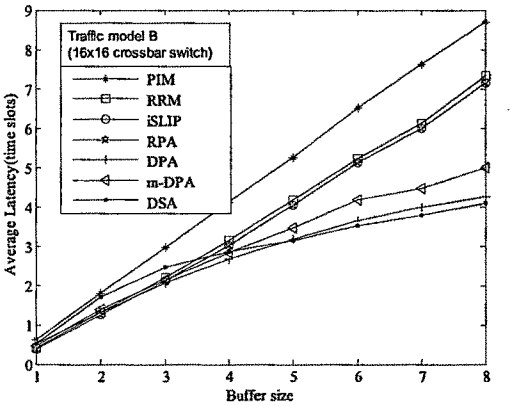

Figure 6.60 Average Latency v/s Buffer size

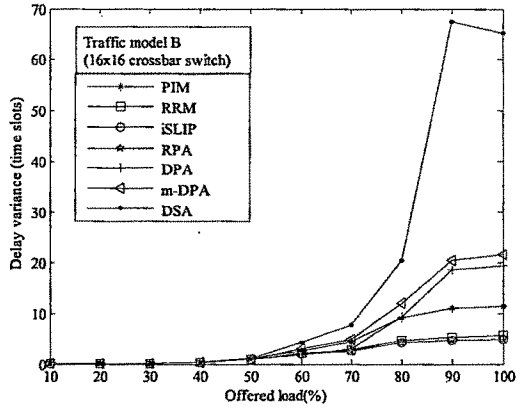

Figure 6.61 Delay variance v/s offered load

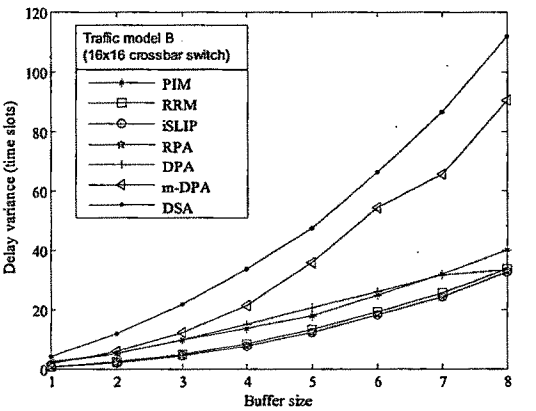

Figure 6.62 Delay variance v/s Buffer size.

C. With bursty arrivals and uniformly distributed destinations:

We illustrate the effect of burstiness on m-DPA using an on-off arrival process. Simulation results from figure 6.63 to 6.68 are shown below. They show that throughput (efficiency) of m-DPA increases by 0.5% to 1.0% compared to DPA for offered load variation as shown in figure 6.63. Average latency of m-DPA is at par with DPA. Delay variance increases by 2 to 5 time slots for m-DPA in offered load and it increases by 1 to 2 time slots in case of buffer size variation as shown in figure 6.67 and 6.68.

Throughput (efficiency) of DSA increases by 0.5% to 2.0% compared to DPA for offered load variation as shown in figure 6.63. Average latency of DSA is at par with DPA. Delay variance increases by 2 to 7 time slots for DSA in offered load and it increases by 1 to 2 time slots in case of buffer size variation as shown in figure 6.67 and 6.68.

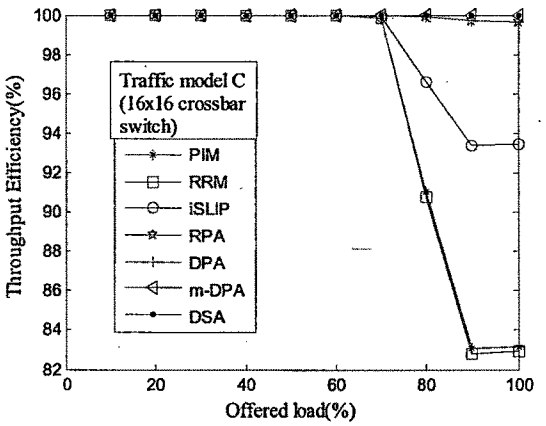

Figure 6.63 Throughput (%) v/s offered load

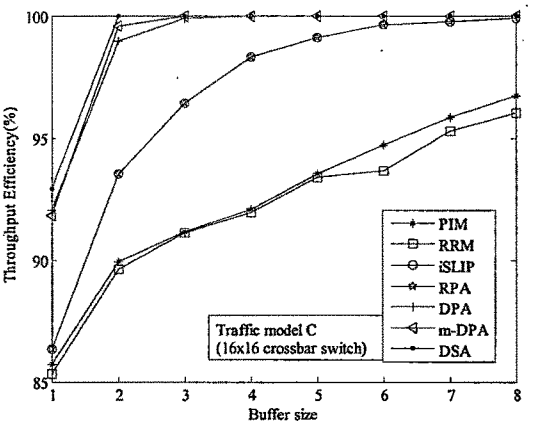

Figure 6.64 Throughput (%) v/s Buffer size

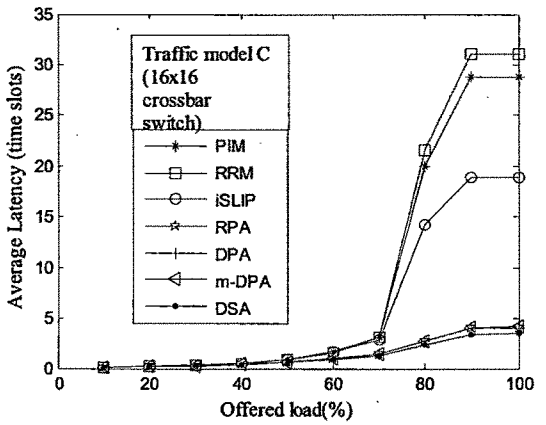

Figure 6.65 Average Latency v/s offered load

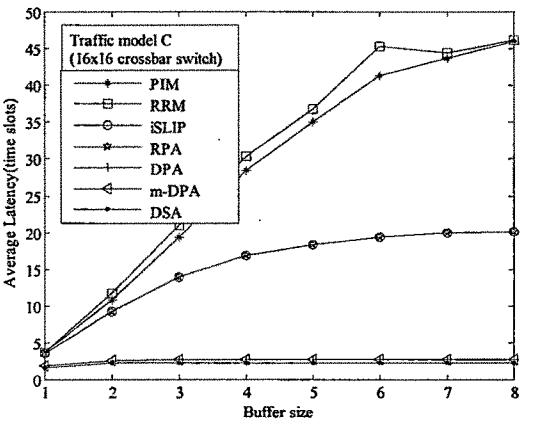

Figure 6.66 Average Latency v/s Buffer size

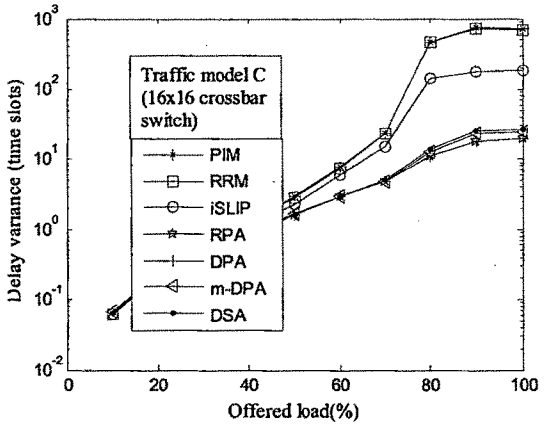

Figure 6.67 Delay variance v/s offered load

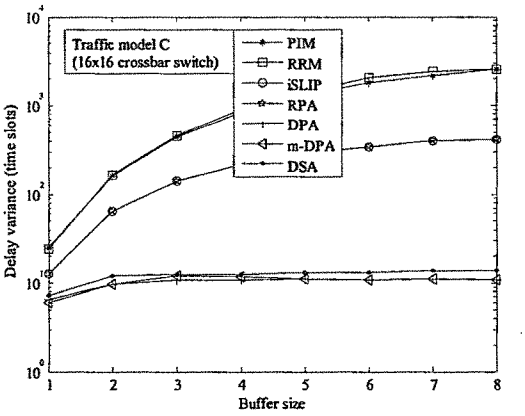

Figure 6.68 Delay variance v/s Buffer size.

D. *With bursty arrivals and non-uniformly distributed destinations*:

We illustrate the effect of burstiness as well as non-uniform distribution for output on m-DPA in this traffic model. Simulation results from figure 6.69 to 6.74 are shown below. They show that throughput (efficiency) of m-DPA increases by 0.5% compared to DPA for offered load variation as shown in figure 6.69. Average latency and delay variance of m-DPA are at par with DPA as shown in figures 6.73 and 6.74.

Throughput (efficiency) of DSA increases by 0.5% to 2.0% compared to DPA. Average latency and delay variance of DSA are at par with DPA.

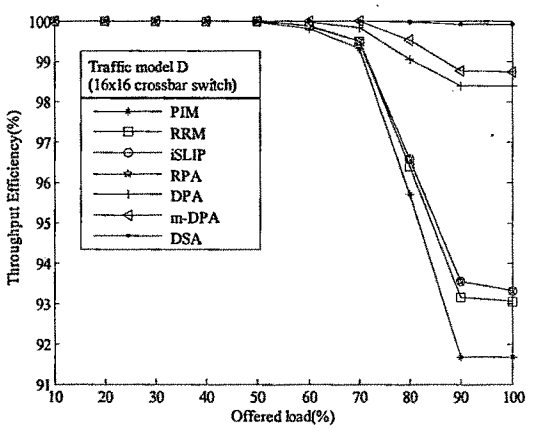

Figure 6.69 Throughput (%) v/s offered load

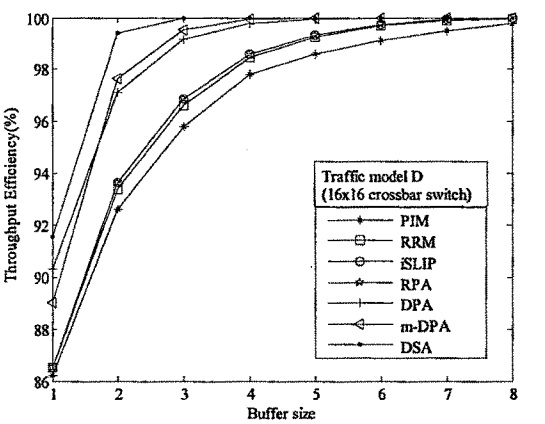

Figure 6.70 Throughput (%) v/s Buffer size

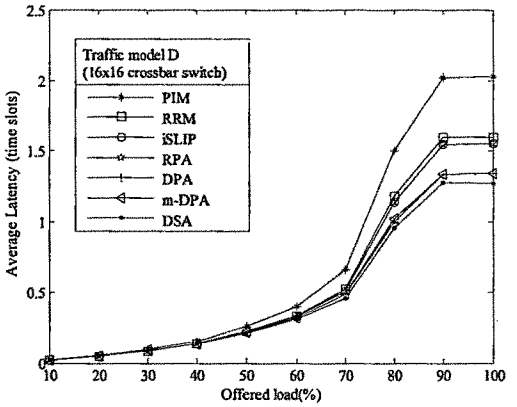

Figure 6.71 Average Latency v/s offered load

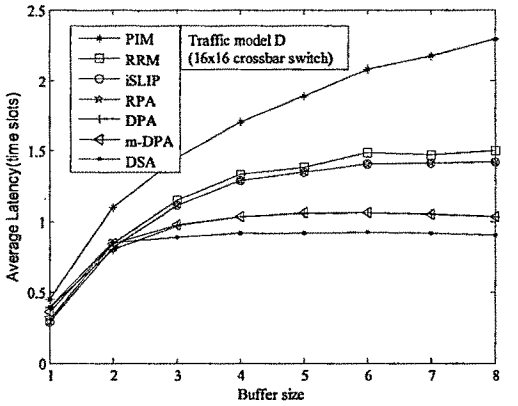

Figure 6.72 Average Latency v/s Buffer size

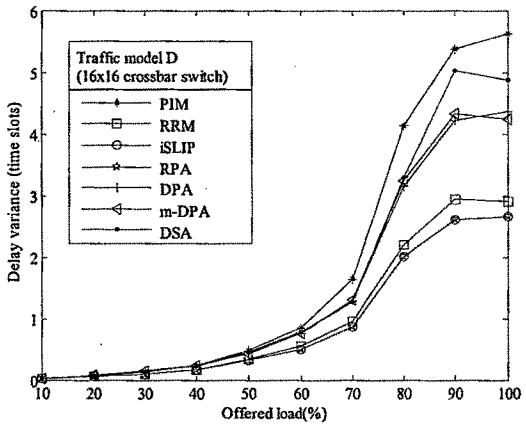

Figure 6.73 Delay variance v/s offered load

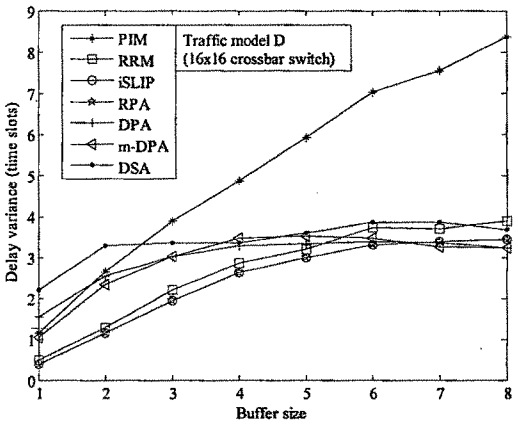

Figure 6.74 Delay variance v/s Buffer size.

6.5.4 32x32 Switch Comparison

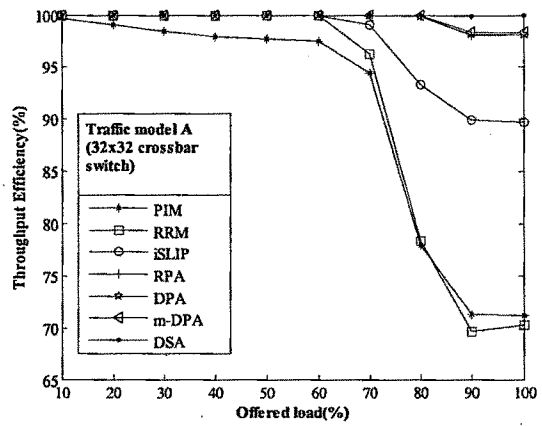

Figure 6.75 Throughput (%) v/s offered load

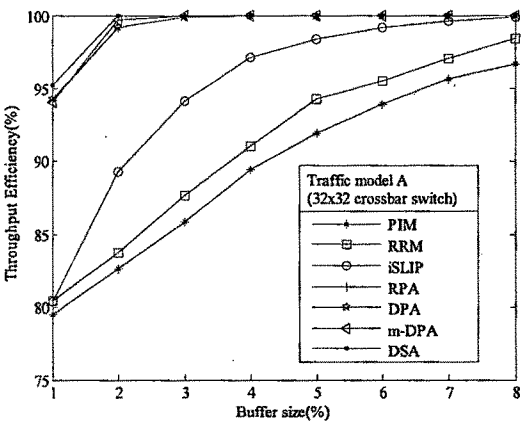

Figure 6.76 Throughput (%) v/s Buffer size

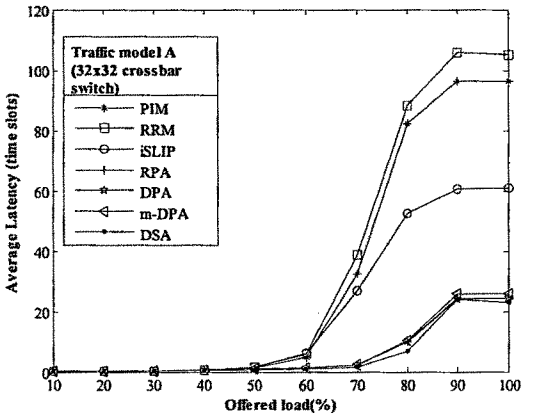

Figure 6.77 Average Latency v/s offered load

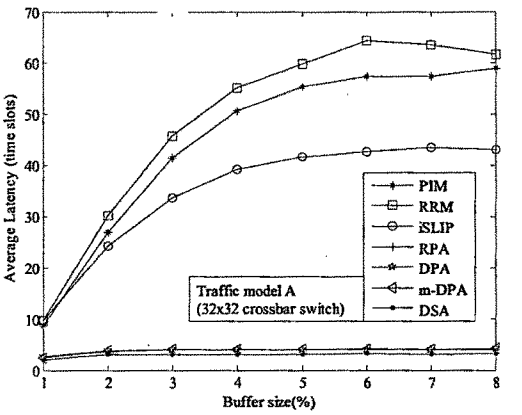

Figure 6.78 Average Latency v/s Buffer size

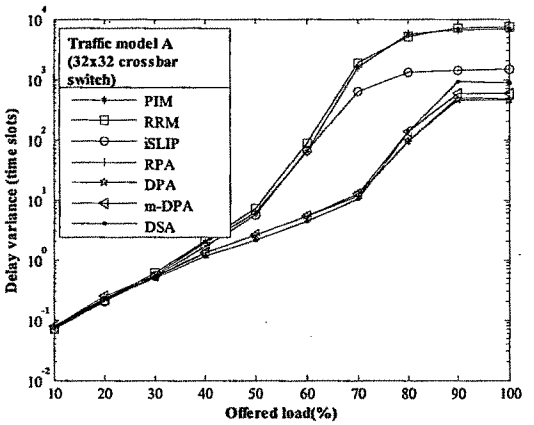

Figure 6.79 Delay variance v/s offered load

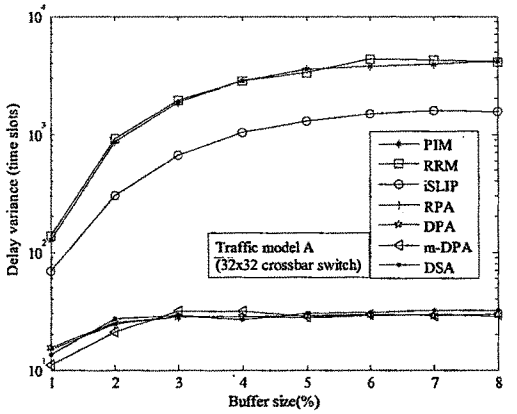

Figure 6.80 Delay variance v/s Buffer size

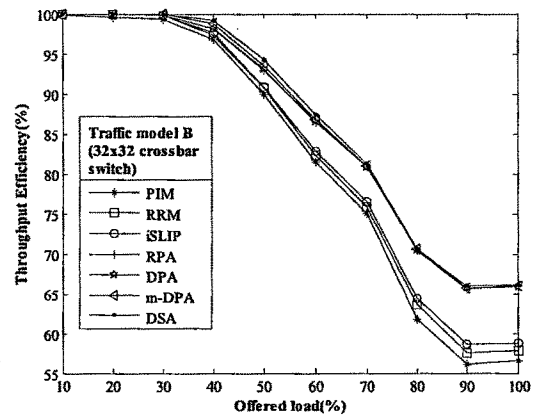

Figure 6.81 Throughput (%) v/s offered load

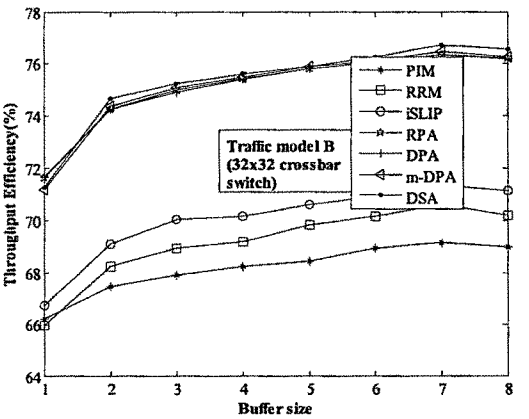

Figure 6.82 Throughput (%) v/s Buffer size

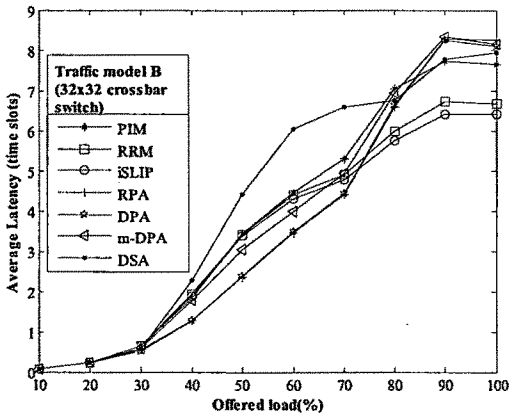

Figure 6.83 Average Latency v/s offered load

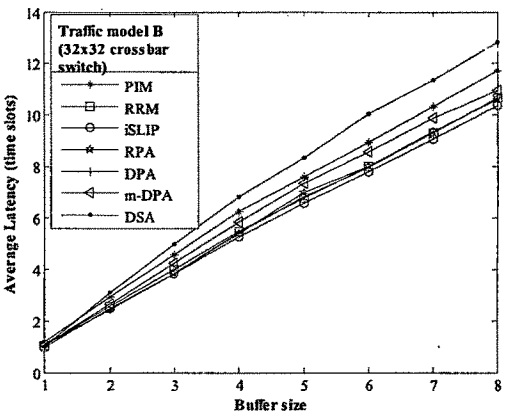

Figure 6.84 Average Latency v/s Buffer size

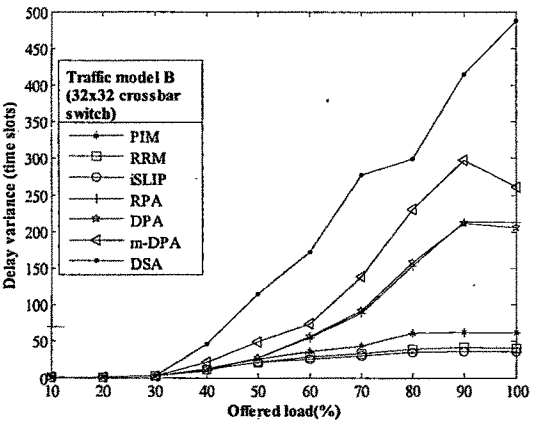

Figure 6.85 Delay variance v/s offered load

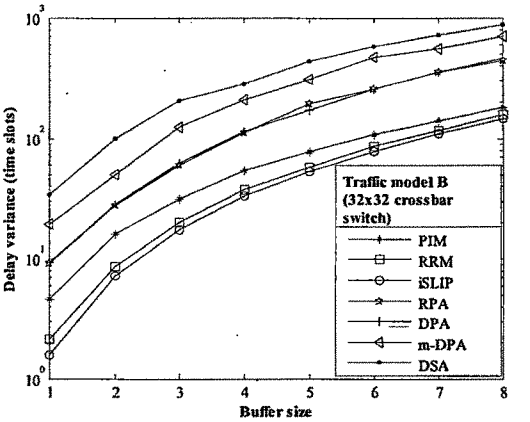

Figure 6.86 Delay variance v/s Buffer size.

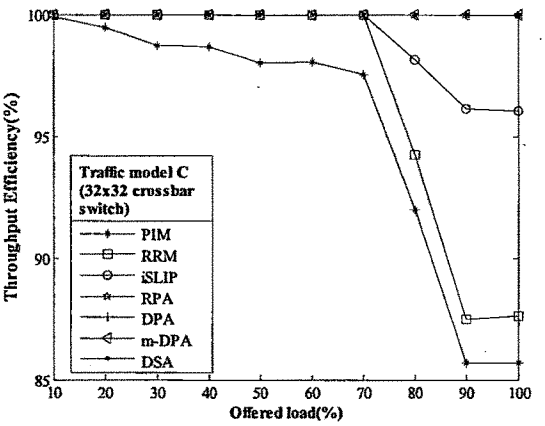

Figure 6.87 Throughput (%) v/s offered load

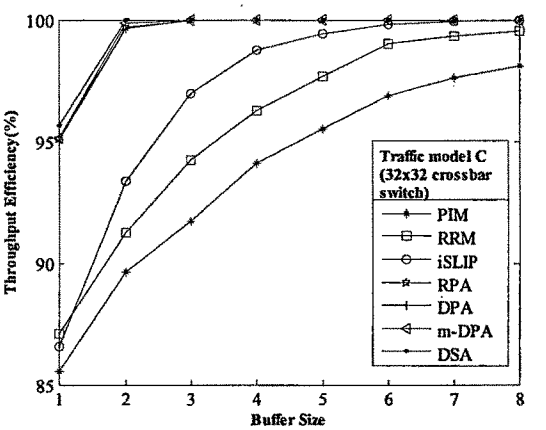

Figure 6.88 Throughput (%) v/s Buffer size

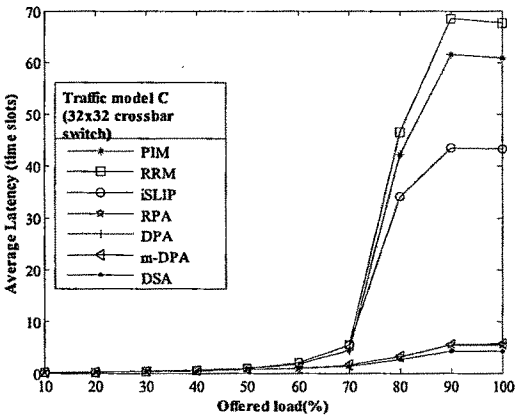

Figure 6.89 Average Latency v/s offered load

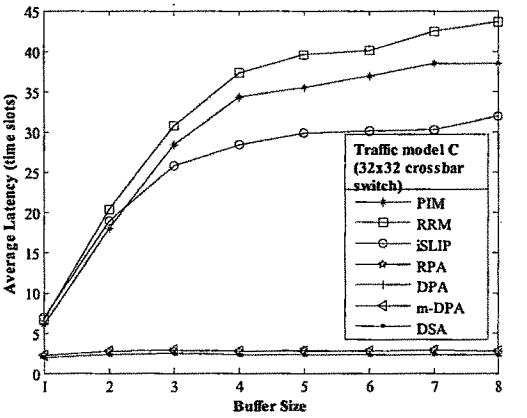

Figure 6. 90 Average Latency v/s Buffer size

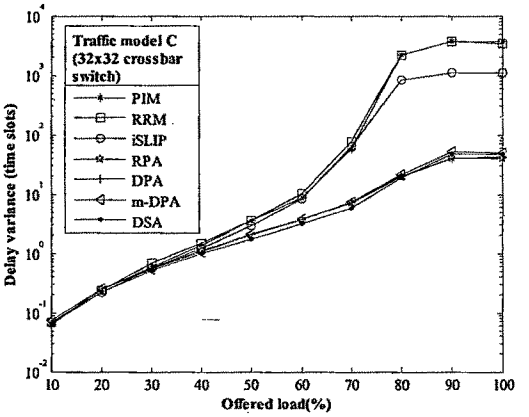

Figure 6.91 Delay variance v/s offered load

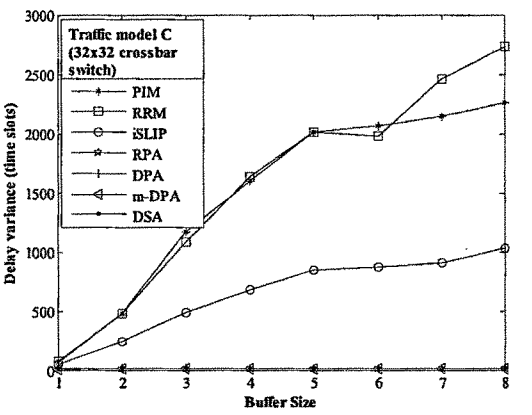

Figure 6.92 Delay variance v/s Buffer size

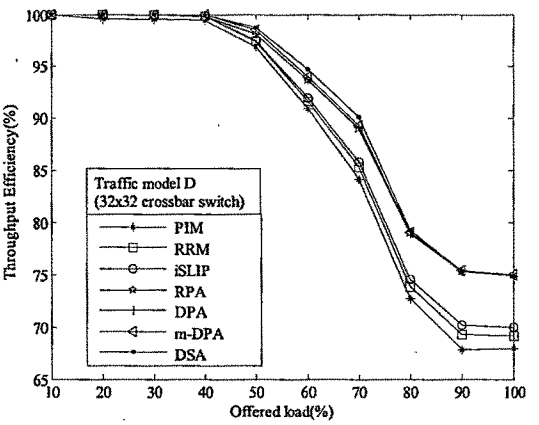

Figure 6.93 Throughput (%) v/s offered load

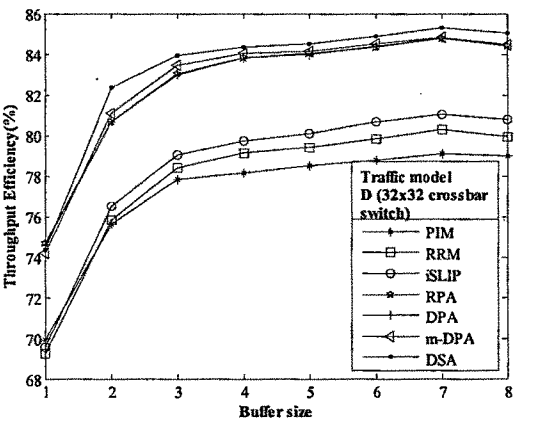

Figure 6.94 Throughput (%) v/s Buffer size

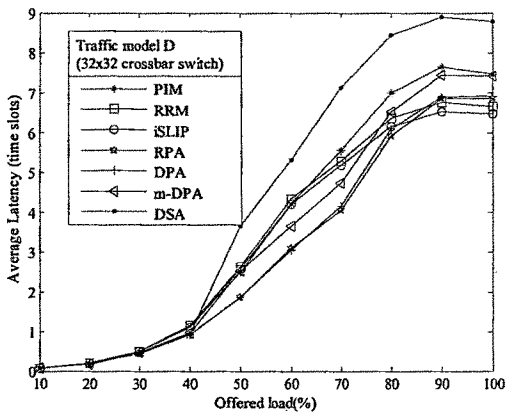

Figure 6.95 Average Latency v/s offered load

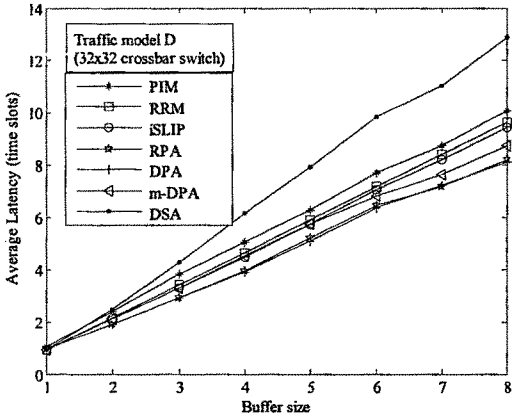

Figure 6.96 Average Latency v/s Buffer size

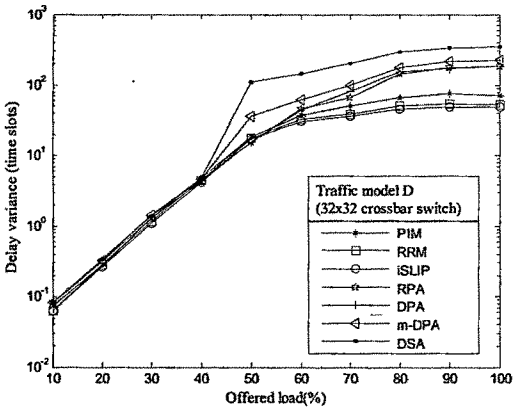

Figure 6.97 Delay variance v/s offered load

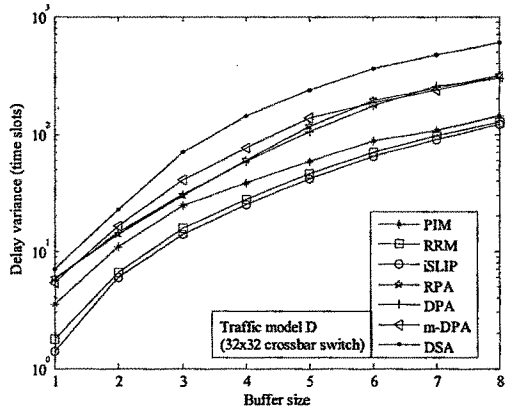

Figure 6.98 Delay variance v/s Buffer size.

Simulation results from figures 6.79 to 6.98, shows 32x32 switch comparison. The analysis of 32x32 switch results resembles that of 16x16 switch result.

6.5.5 High QoS DSA Simulation Results

We have separated and simulated High QoS input output within a DSA with other algorithms for 4x4, crossbar switches with four different traffic models (A,B,C and D) using MATLAB 7.0. Algorithms are simulated for 1000 time slots and results are taken by averaging the outcomes for 100 simulations. Various parameters like throughput (efficiency), average latency and delay variance have been measured for variation in offered load as well as variation in buffer size. For variation in offered load the buffer size is 3, and for variation in buffer size the offered load is 90%.

From all the simulation result as shown in figure 6.99 to 6.122 high QoS input output in DSA always get the highest throughput (efficiency), lowest average latency and lowest delay variance.

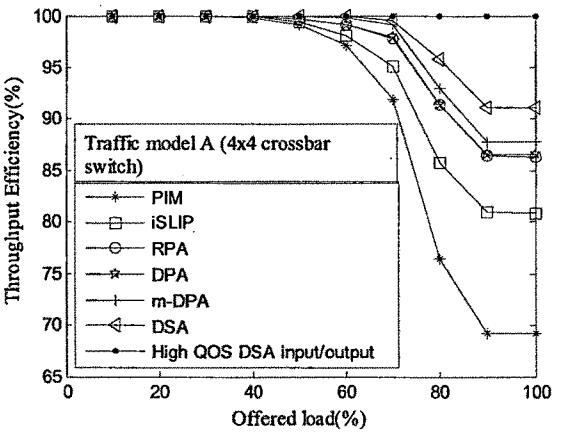

Figure 6.99 Throughput v/s offered load

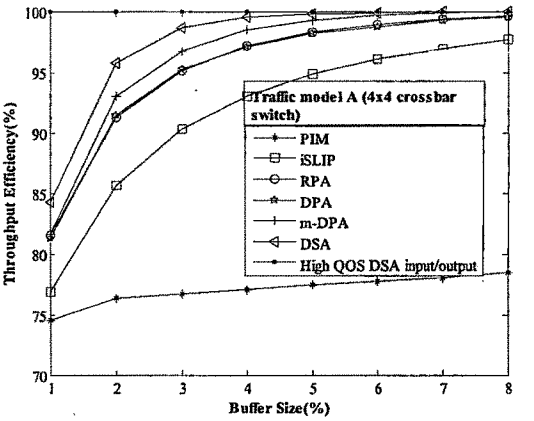

Figure 6.100 Throughput v/s Buffer size

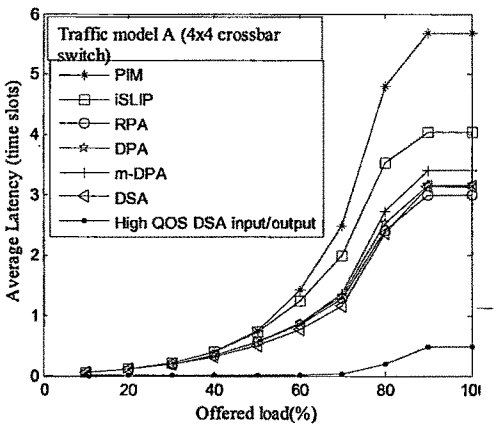

Figure 6.101 Average Latency v/s offered load

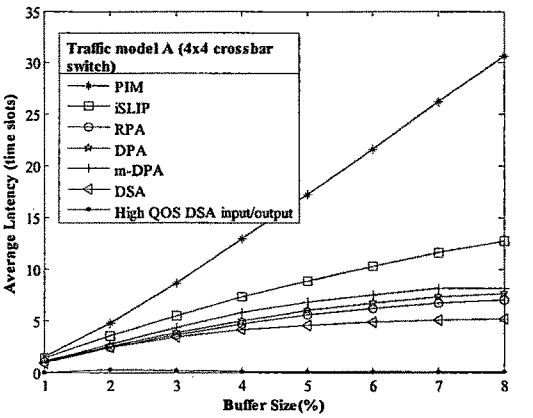

Figure 6.102 Average Latency v/s Buffer size

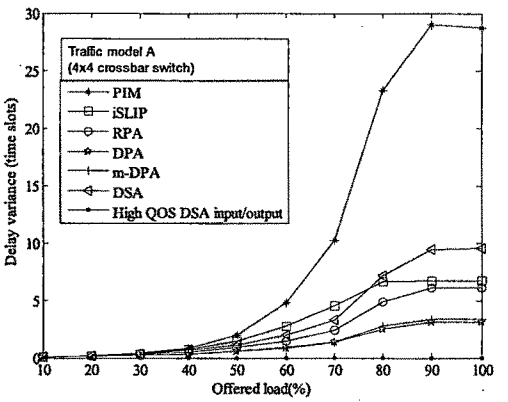

Figure 6.103 Delay variance v/s offered load

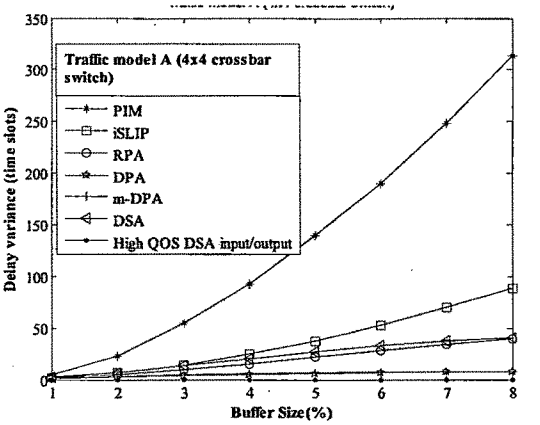

Figure 6.104 Delay variance v/s Buffer size.

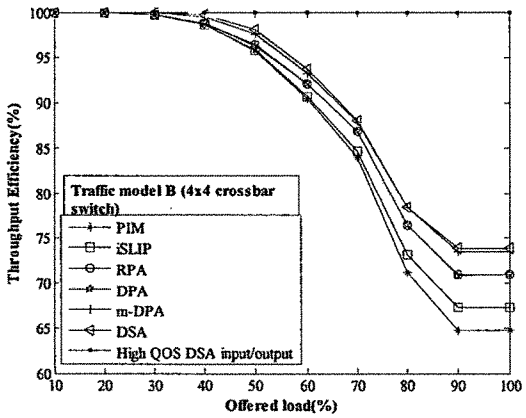

Figure 6.105 Throughput v/s offered load

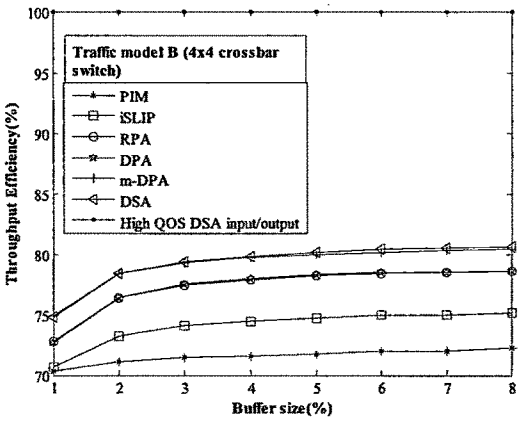

Figure 6.106 Throughput v/s Buffer size

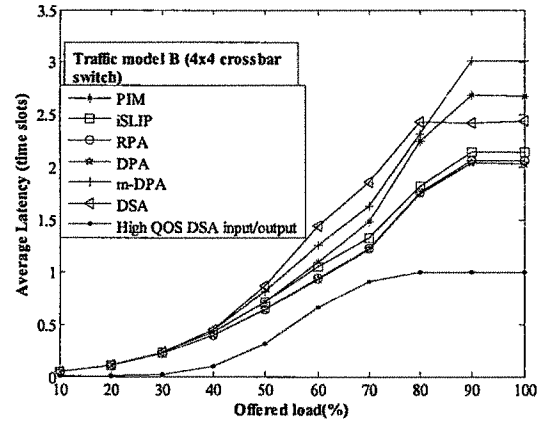

Figure 6.107 Average Latency v/s offered load

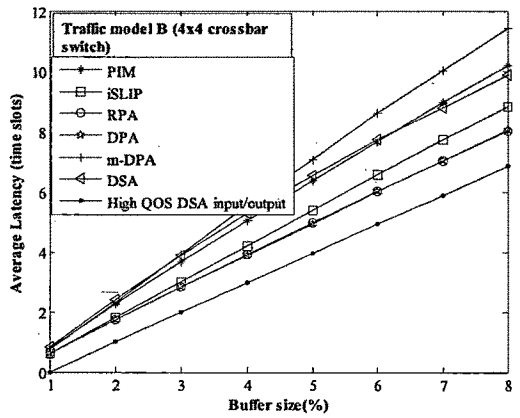

Figure 6.108 Average Latency v/s Buffer size

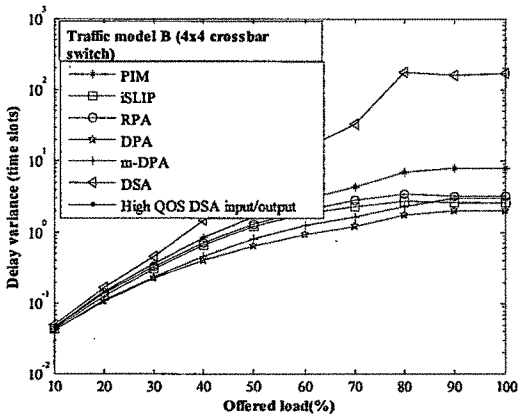

Figure 6.109 Delay variance v/s offered load

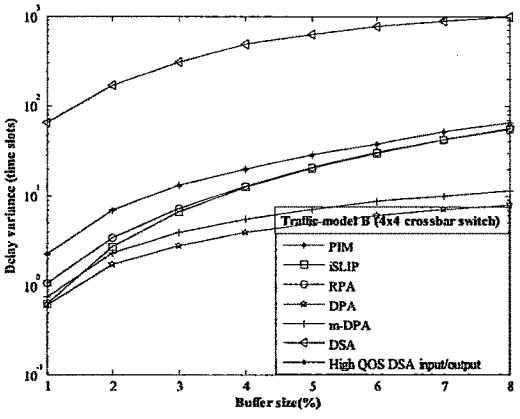

Figure 6.110 Delay variance v/s Buffer size.

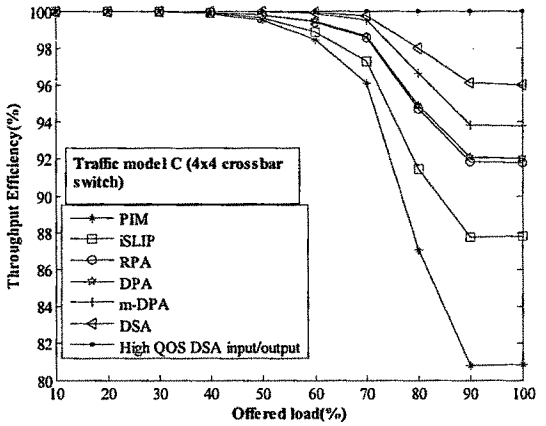

Figure 6.111 Throughput v/s offered load

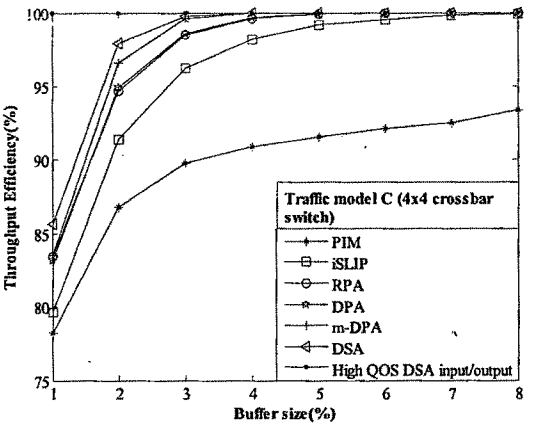

Figure 6.112 Throughput v/s Buffer size

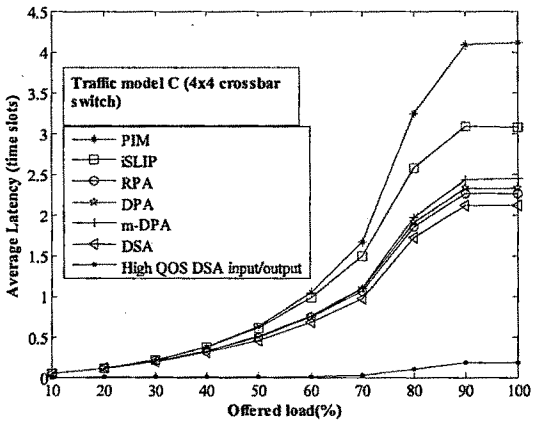

Figure 6.113 Average Latency v/s offered load

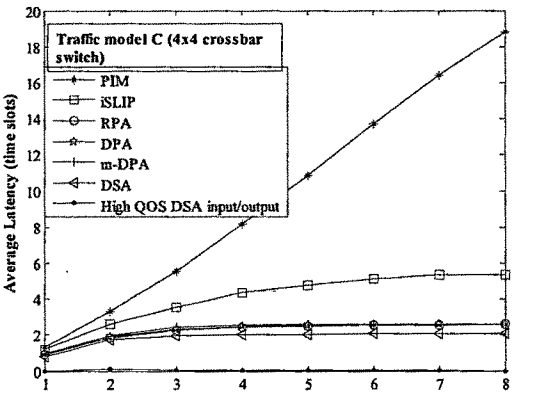

Figure 6.114 Average Latency v/s Buffer size

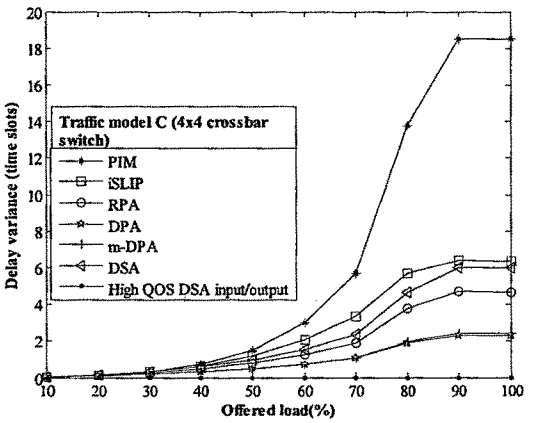

Figure 6.115 Delay variance v/s offered load

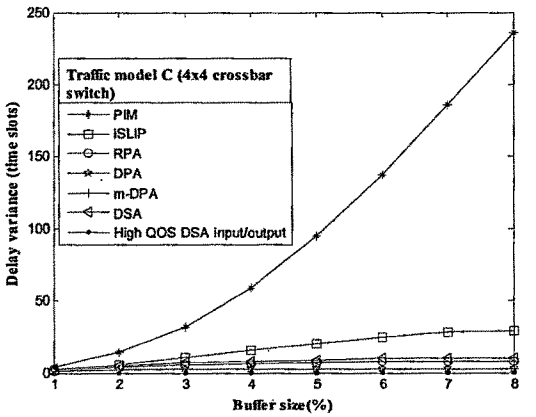

Figure 6.116 Delay variance v/s Buffer size.

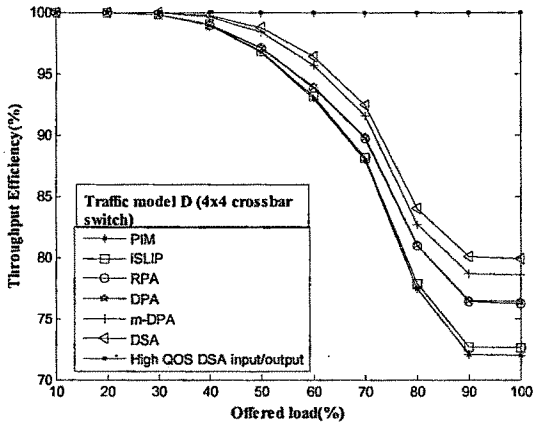

Figure 6.117 Throughput v/s offered load

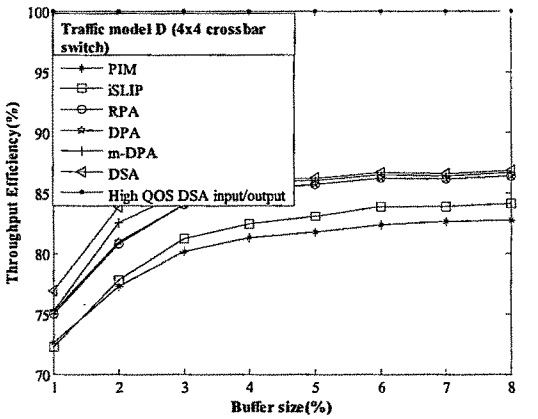

Figure 6.118 Throughput v/s Buffer size

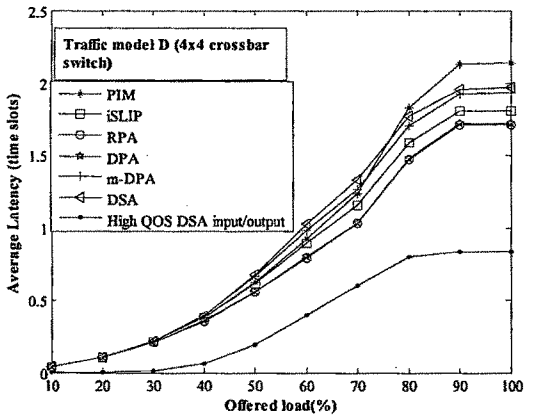

Figure 6.119 Average Latency v/s offered load

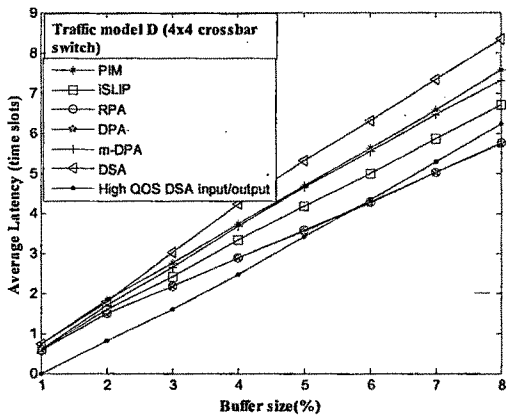

Figure 6.120 Average Latency v/s Buffer size

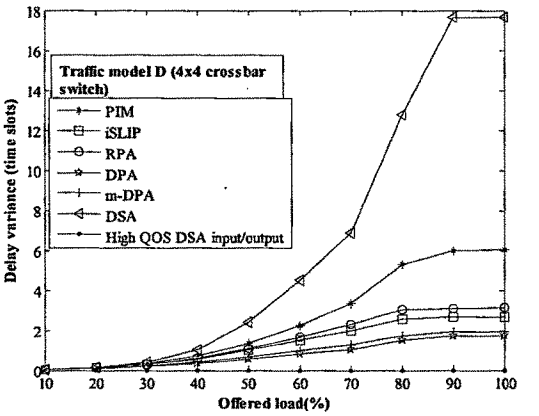

Figure 6.121 Delay variance v/s offered load

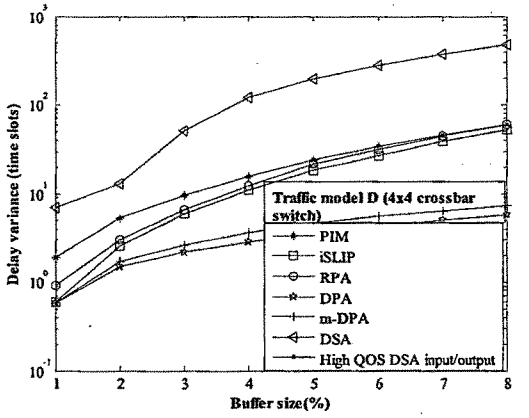

Figure 6.122 Delay variance v/s Buffer size

6.6 VLSI IMPLEMENTATION OF m-DPA

We have implemented 4x4 and 8x8 ATM crossbar switches with m-DPA algorithm using ALTERA'S QUARTUS II tool. Scheduler of 4x4 m-DPA consists of following sub blocks as shown in figure 6.123. 1. Req_allip 2. schedular44m 3. Grt_for_ip 4. M_group44m 5. Gr_arng44M. Out of five blocks req_all_ip and Grt_for_ip sub blocks are same as RRM's and other scheduler's subblocks.

Block schedular44m is same as schedualr subblock of DPA with one modification. One more input signal gr_arrng[1..4] is given to schedular44m subblock, which decides the starting point of priority vector in m-DPA. Depending on the current VOQ buffer status, indicated by the usedwxx output of input_port, M-group44M block calculate no of red-zone buffer in the particular group of m-DPA. as shown in figure 6.124. Subblock GR_ARNG44M set the corresponding group bit high in gr_arng[1..4] output, as shown in figure 6.125. Signal gr_arng[1..4] is input to the schedular44m subblock.

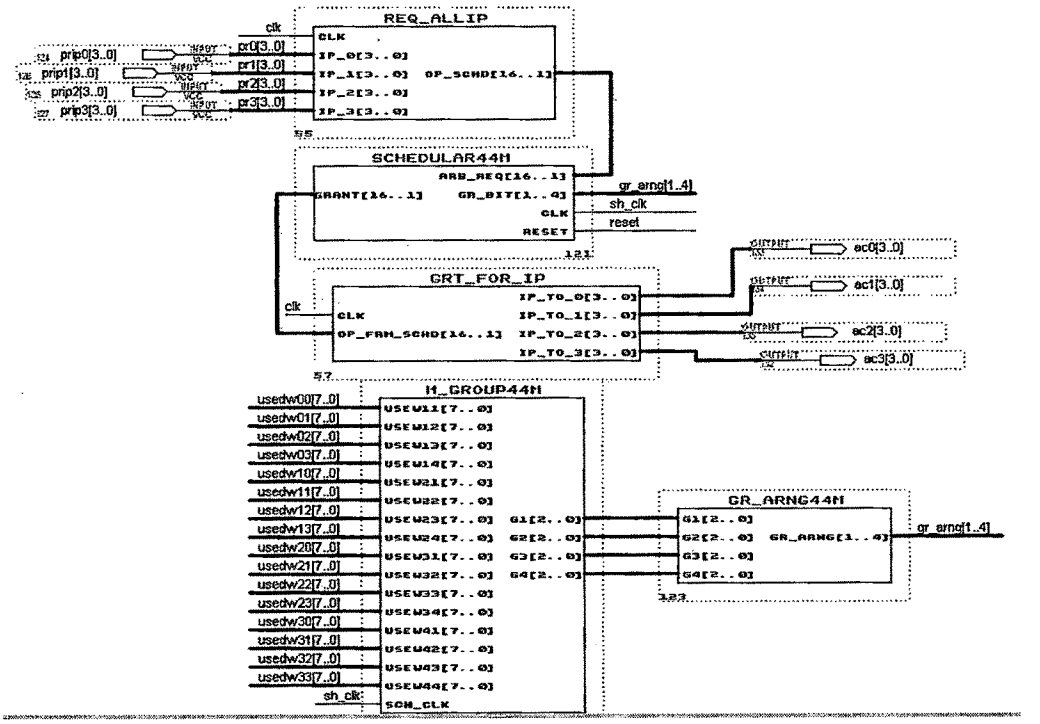

Figure 6.123 Implementation details of m-DPA

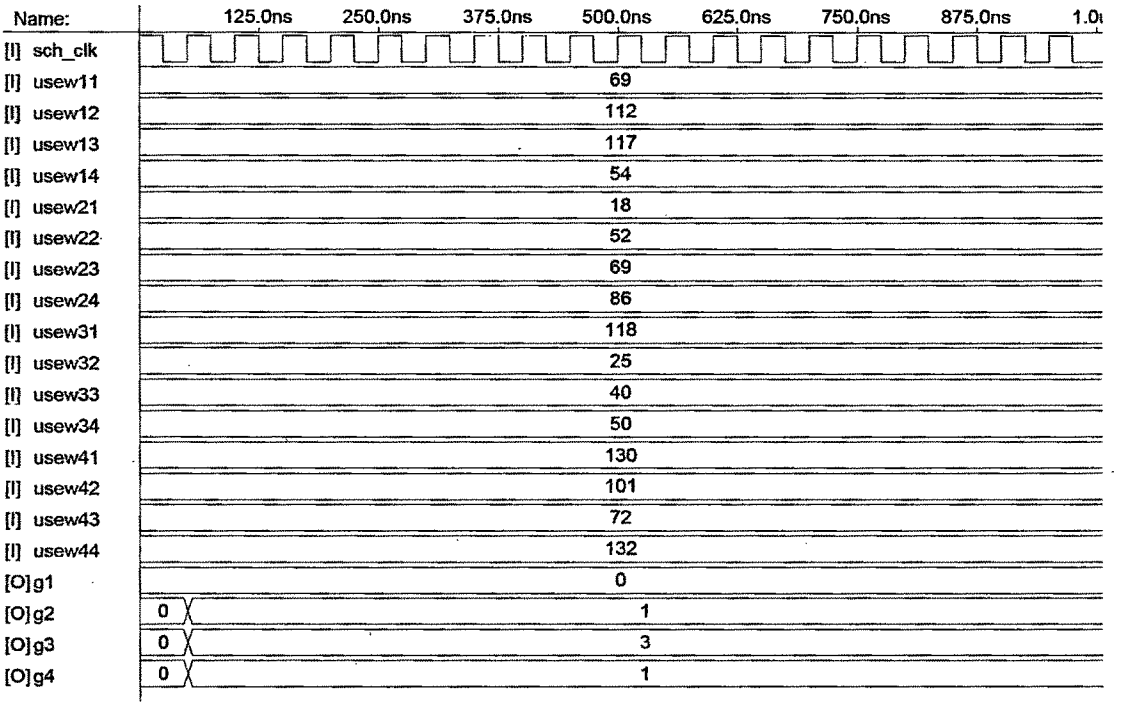

Figure 6.124 Simulation waveform of m-group44m

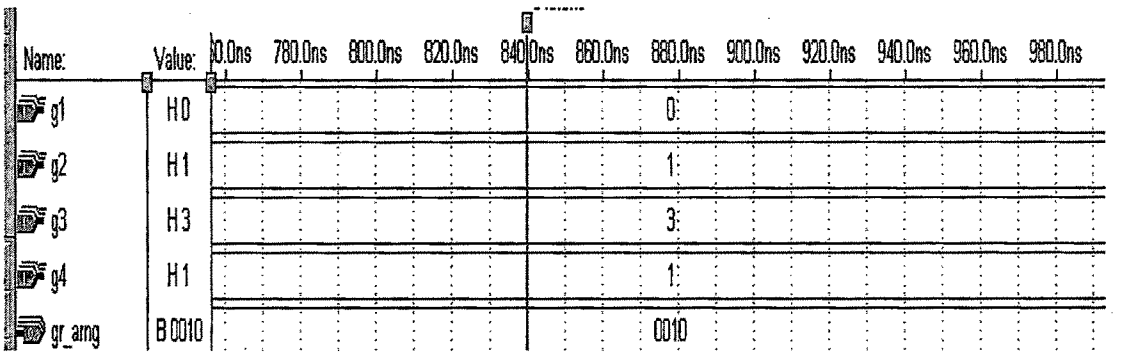

Figure 6.125 Simulation waveform of gr_arng44m

Based on the implementations, we present the area analyses of DPA and m-DPA in Table 6.3 and 6.4.

6.7 VLSI IMPLEMENTATION OF DSA

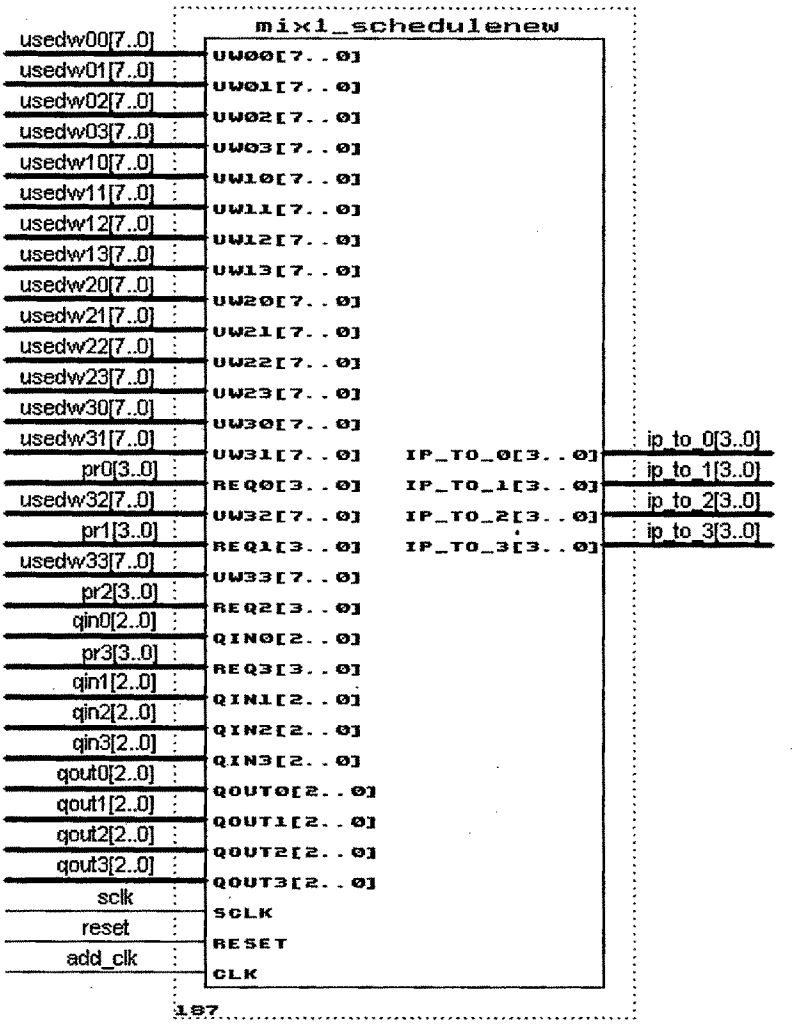

Figure 6.126 Scheduler block of DSA

Overall architecture of 4x4 switch using DSA scheduling algorithm is same as architecture of 4x4 switch using other scheduling algorithms as shown and described in figure 5.108. Scheduler block of DSA is different, which is described in figure 6.126. The inputs usedw00---- usedw33 are used to describe the status of buffer size in input port. The inputs qin0...qin3 and qout0...qout3 are used to indicate the QoS of input/output. The inputs pr0 to pr3 are used to indicate the request status from each VOQ.

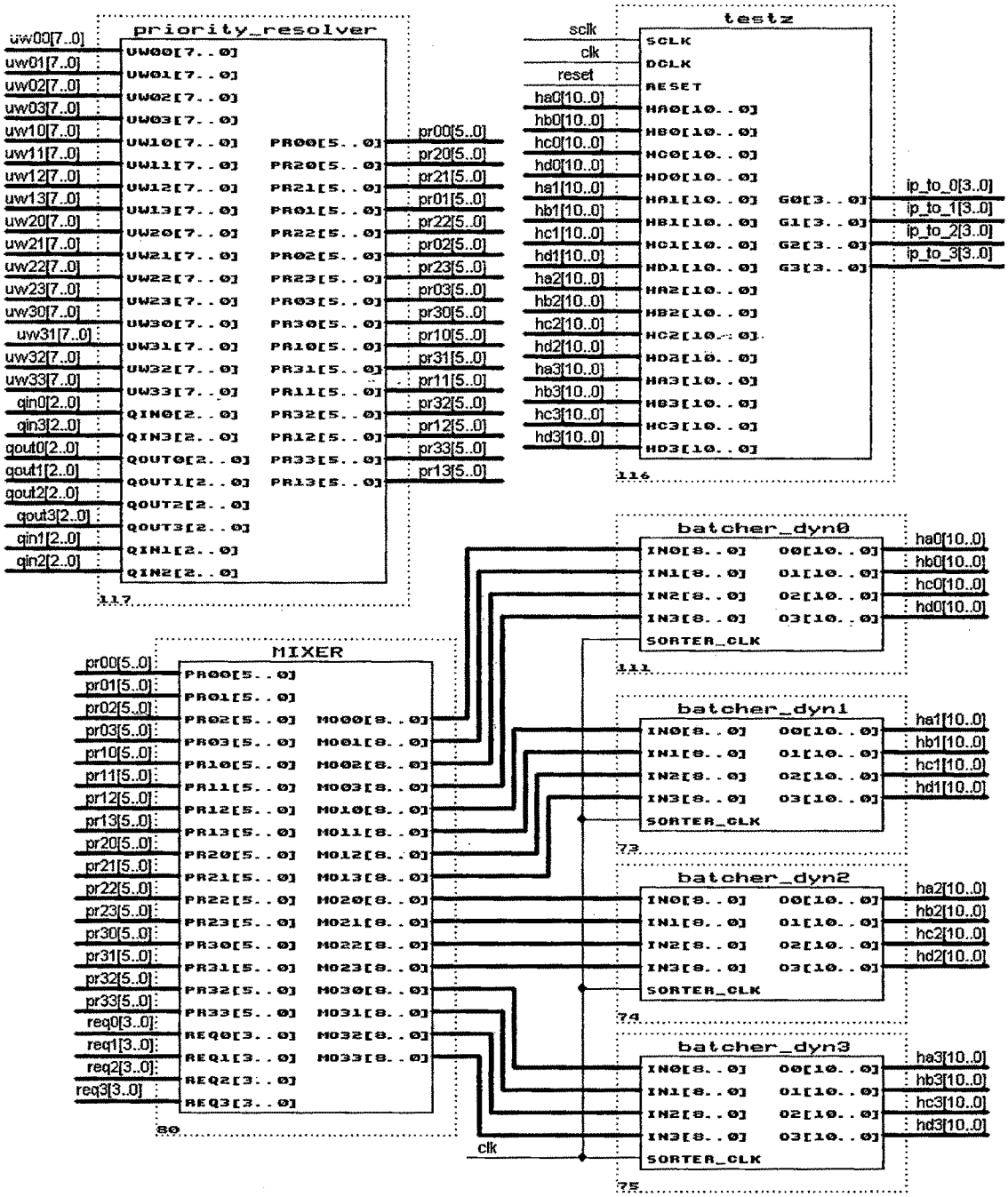

Figure 6.127 Internal Architecture of Scheduler (Mix1_schedulenew)

The internal architecture of DSA scheduler is described in figure 6.127. Priority resolver subblock internal architecture is described in figure 6.128. Priority resolver subblock takes usedword status and QoS status of each individual input/output and generates the priority for the input/output connection. The usew_converte converts each VOQ buffer status into 3 bits. In our implementation the buffer size in the FIFO is 4. We map the buffer size from 0 to 4 in 3 output bits in usew_converte subblock as shown in table 6.2.

Table 6.2: Mapping of ATM_Packet in Buffer

Size of the buffer indicated by used word	ATM packet	Output bit mapping
used_word = 0	0	0 0 0
0 < used_word ≤ 53	1	0 1 0
53 < used_word ≤ 106	2	1 0 0
106 < used_word ≤ 159	3	1 1 0
159 < used_word ≤ 212	4	1 1 1

In usew_converte sub block for each input-output pair this process is done by ATM_cell_size_3 as shown in figure 6.129 and simulation of ATM_cell_size_3 is as shown in figure 6.130. QoS of input and output are added as shown in figure 6.128 and simulation is shown in figure 6.131. Mult24 block adjust the k factor if input is less than 7 output is twice the input but if input is equal to 7 (buffer size =4 = full) then output is four times the input as shown in figure 6.132. As shown in figure 6.128 Add_4_5_6 sub block adds 5 bit buffer size(Normally 4 bit except buffer size is maximum) with 4 bit QoS. Mixer sub block combines the priority status, destination address and request as shown in simulation waveform 6.133. Let us take m032 output waveform, which is req bit (1) appended with destination number(10), and pr32(001011). Batcher-dyn0, to Batcher-dyn3 are batcher sorter used to sort the priority status at each input and arrange in descending order from O_0 to O_3, become hax to hdx as shown in internal architecture figure 6.134. These blocks also append the source number as shown in simulation waveform figure 6.135.One more batcher sorter is used in testz subblock as shown in figure 6.136 to sort the highest priority among the highest priority of all input ports. These sorted signals are used to swap the clocks. Based on main clock dclk and packet clock sclk, Clkipol subblock generates 6 different clocks; Srclk, (To reset the last time slot status), clkl(To latch the data) and four other clocks for four input scheduler blocks Gloabalschedipxnox, as shown in simulation waveform figure 6.137. Clk_swp subblock in figure 6.138 swap the clocks based on the sorted inputs (in0 to in3 in descending order) and source number (bits 10 and 9 of in0 to in3). Pri_latch block latches the data with clkl clock. Four ORG41 subblocks are 4 input OR gate to inhibit that particular output for other inputs. Globalschedipxnx block generates 4 bit grant for that x input port based on the hax to hdx ,inhibit inputs ix0 to ix3 and the clock (which was already swapped and arranged in priority order.) Simulation waveforms are shown in figure 6.139.

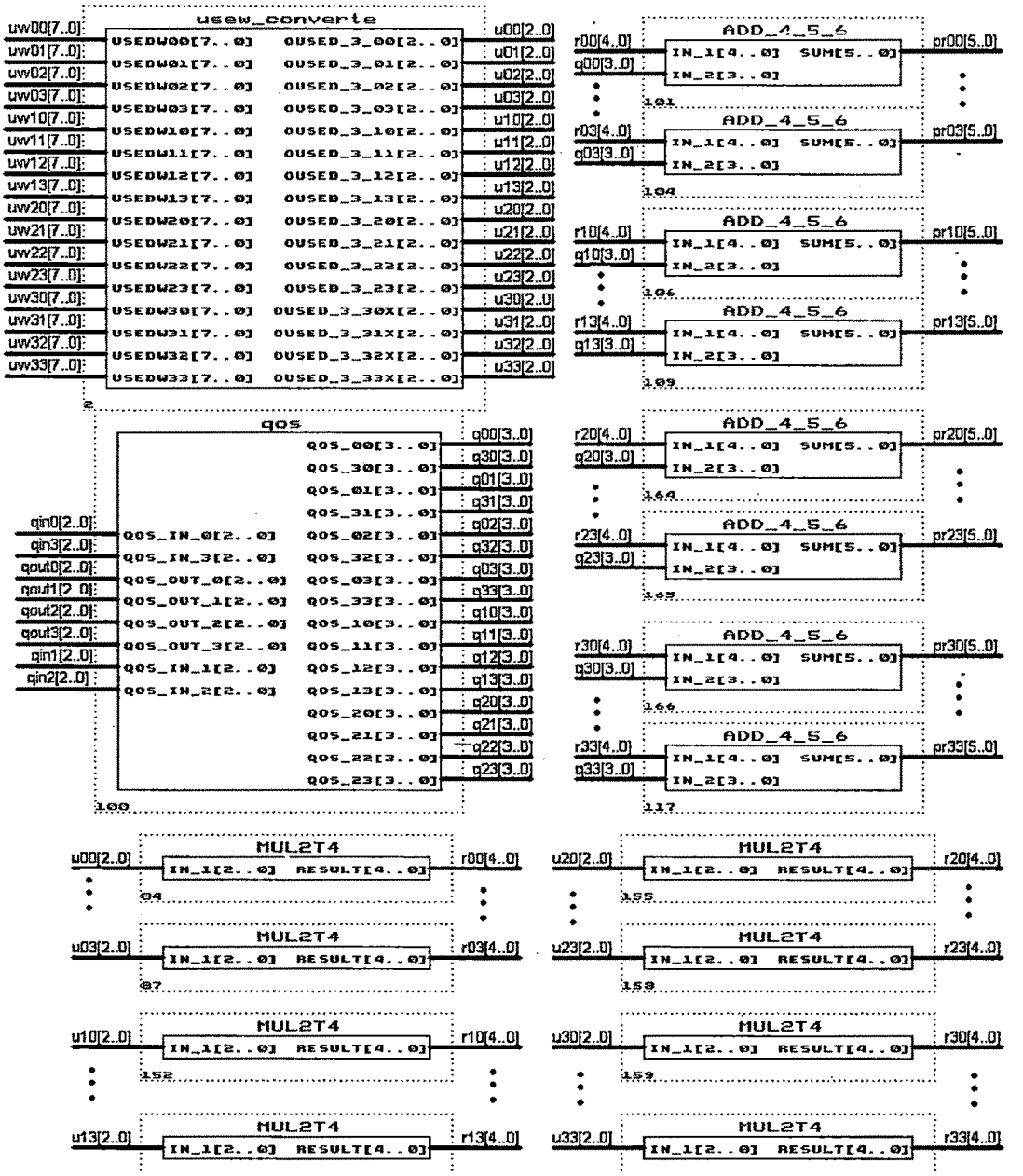

Figure 6.128 Internal Architecture of Priority Resolver

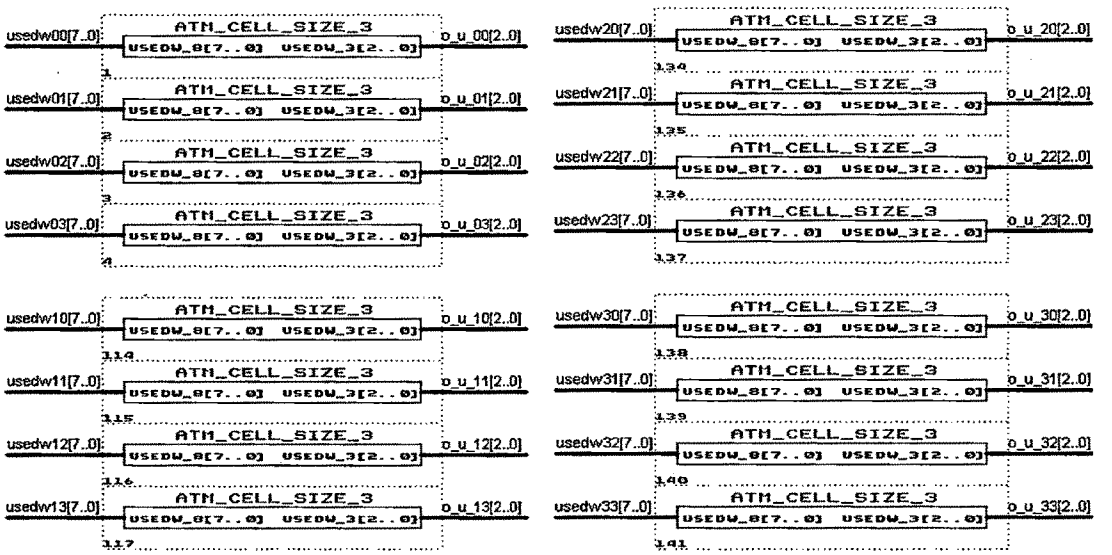

Figure 6.129 Internal Architecture of usew_converte

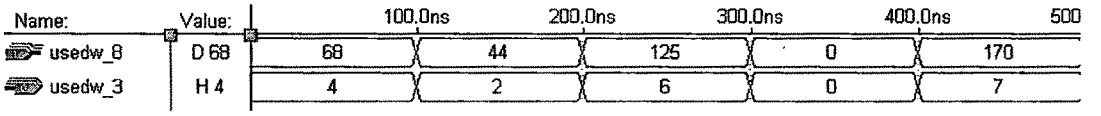

Figure 6.130 Simulation waveform for ATM_CELL_SIZE_3

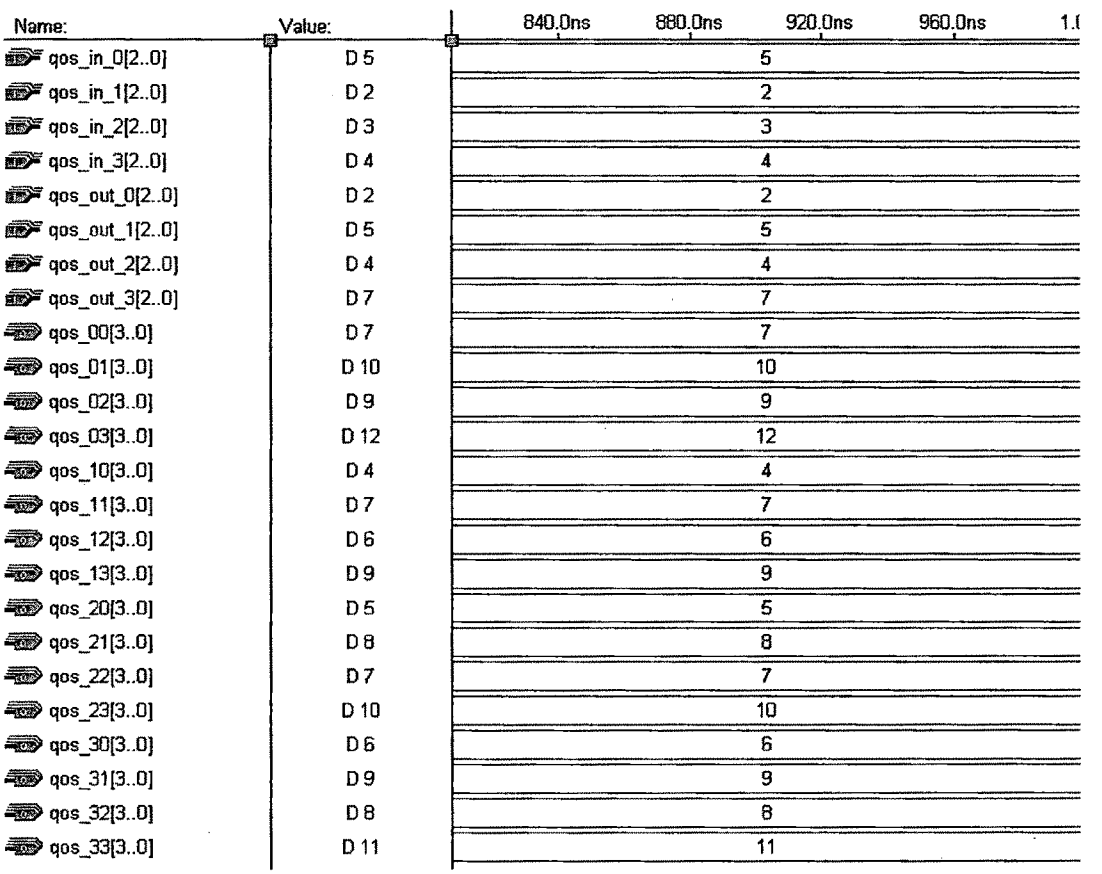

Figure 6.131 Simulation waveform for QoS

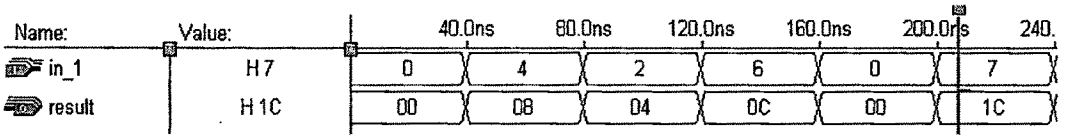

Figure 6.132 Simulation waveform for mult24

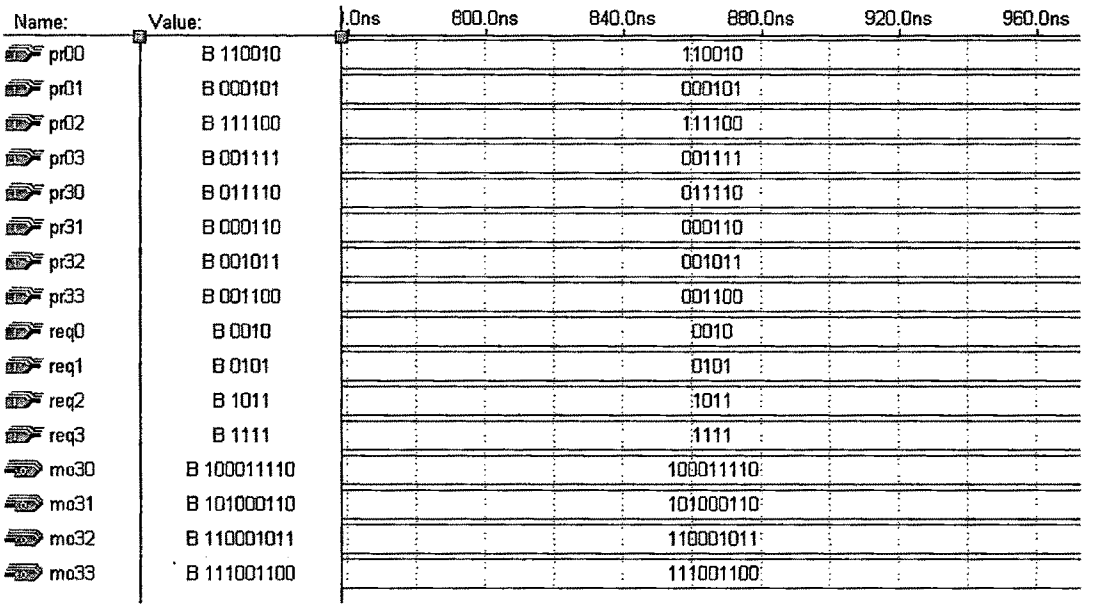

Figure 6.133 Simulation waveform for mixer

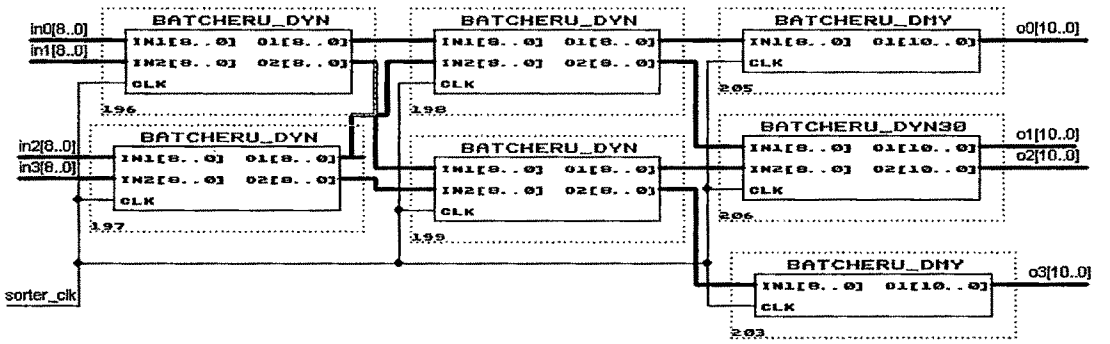

Figure 6.134 Internal Architecture of basic Batcher sorter

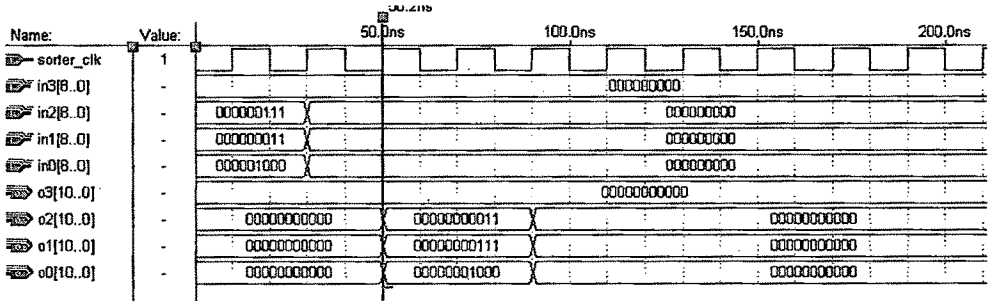

Figure 6.135 Simulation waveform for batcher_dyn0

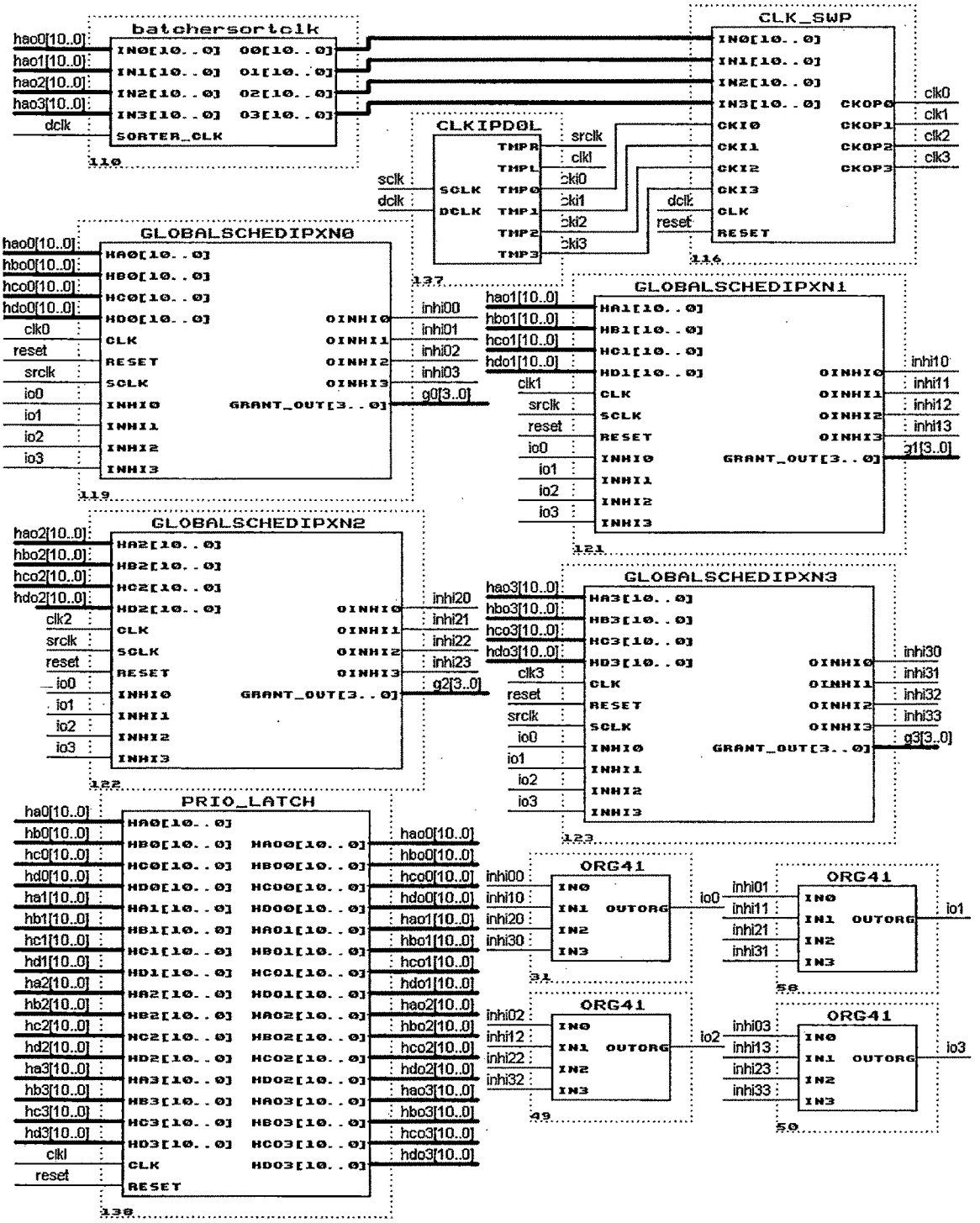

Figure 6.136 Internal Architecture of TestZ block

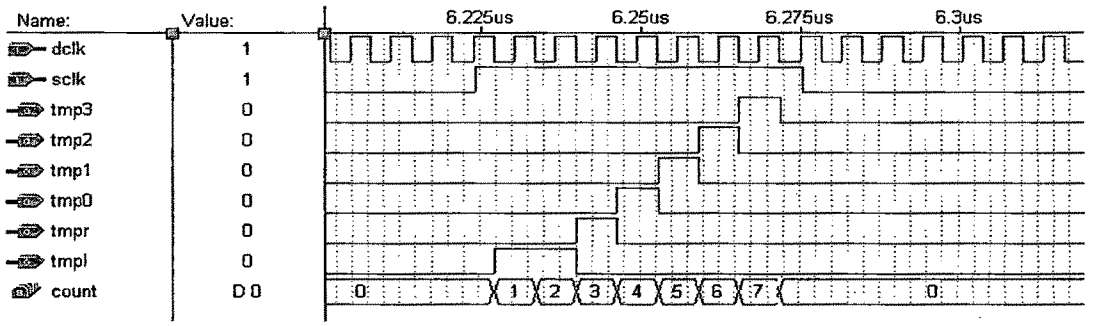

Figure 6.137 Simulation waveform for clkipol

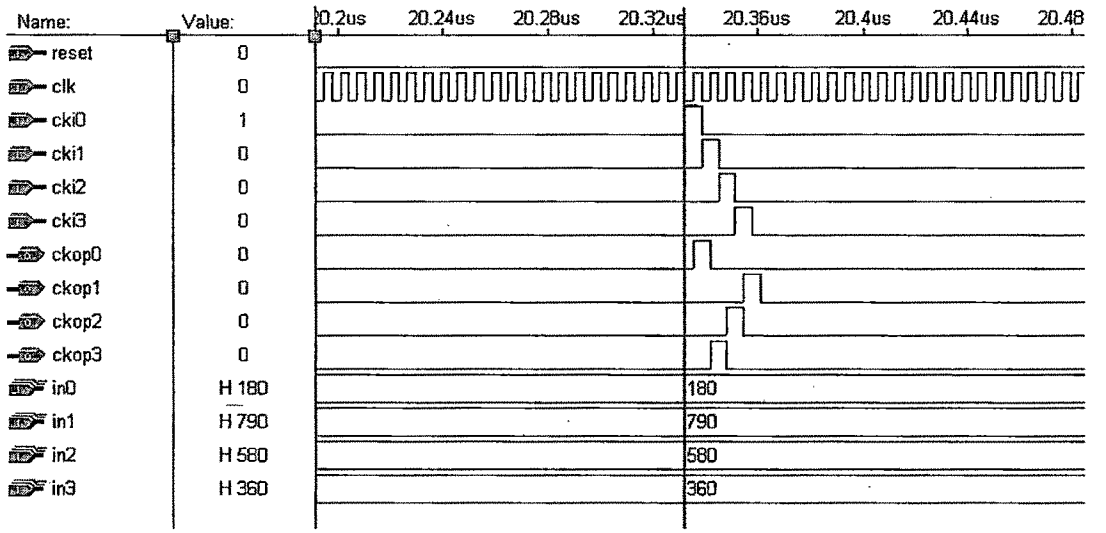

Figure 6.138 Simulation waveform for clk_swp

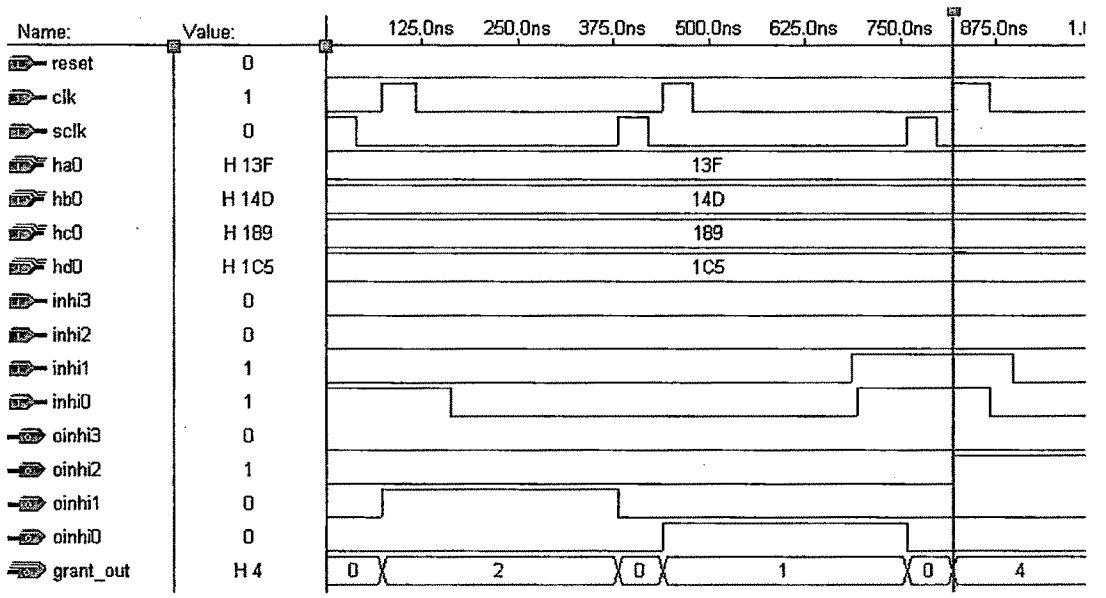

Figure 6.139 Simulation waveform for Globalschedipxn0

6.8 VLSI AREA REQUIREMENT OF m-DPA AND DSA

4x4 and 8x8 ATM crossbar switches with DPA, m-DPA and DSA algorithms are implemented using ALTERA'S QUARTUS II tool. Based on the implementations, Table 6.3 and 6.4 present the area analyses of DPA, m-DPA and DSA.

Table 6.3 DPA/ m-DPA/DSA area analysis for 4x4 ATM switch

Scheduling Algo./ Project	Device	Total Logic Elements	Total Pins	Total Memory Bits
DPA ATM_DPA_4x4	EP20k1500EB C652-1	5,656 / 51,840 (11 %)	87 / 488 (18 %)	33,920 / 442,368 (8 %)
mDPA atm_dap44m	EP20k1500EB C652-1	5,816 / 51,840 (11 %)	87 / 488 (18 %)	33,920 / 442,368 (8 %)
DSA ATM_DSA_4x4	EP20k1500EB C652-1	6,907 / 51,840 (13 %)	111 / 488 (23 %)	33,920 / 442,368 (8 %)

Table 6.4 DPA/ m-DPA/DSA area analysis for 8x8 ATM switch

Scheduling Algo./ Project	Device	Total Logic Elements	Total Pins	Total Memory Bits
DPA ATM_DPA_8x8	EP20k1500EB C652-1	22,778 / 51,840 (44 %)	178 / 488 (36 %)	133,376 / 442,368 (30 %)
mDPA ATM_DPA_88md	EP20k1500EB C652-1	23,178/51,840 (45%)	178 / 488 (36 %)	136,960/442,368 (31 %)
DSA ATM_DSA_88	EP20k1500EB C652-1	25,558 / 51,840 (49 %)	226 / 488 (46 %)	133,376 / 442,368 (30 %)

6.9 SUMMARY

In this chapter, we have presented two scheduling algorithms, m-DPA and DSA for solving the symmetric scheduling algorithm for input queued cross bar switches.

m-DPA is implemented using Altera's QuartusII 3.0 software in VHDL. This algorithm is feasible to implement in current VLSI technology. We have shown that for unevenly distributed and bursty traffic, cell loss rate is reduced, resulting in improvement in the throughput. Due to on-off traffic

pattern (C and D) the advantage of m-DPA vanishes after buffer size crosses threshold for buffer size variation.

A Dynamic Scheduling Algorithm, solving the symmetric scheduling problem with VOQ-based input queued crossbar switches is introduced and results from computer simulations and VHDL implementation are shown to prove the effectiveness of the algorithm in terms of rise in throughput (efficiency) by 2% to 5% at the cost of latency in traffic pattern B. It can be seen that the benefits of DSA are more pronounced with high QoS inputs outputs. It provides extremely high throughput (efficiency) and lowest average latency and delay variance to high QoS inputs outputs with only 2 % more area than DPA, in 4x4 switch and 5% more area than DPA in 8x8 switch. Also as seen from the results, benefits of DSA are more pronounced in cases of larger switches with small buffering capacity.