

The Simulation and Realization of Input-Buffer Scheduling Algorithm in Satellite Switching System

Yi Zhang, Quan Zhou, Jun Li, and Yanlang Hu

Abstract—Scheduling algorithm is a key technology in satellite switching system with input-buffer. In this paper, a new scheduling algorithm and its realization are proposed. Based on Crossbar switching fabric, the algorithm adopts serial scheduling strategy and adjusts the output port arbitrating strategy for the better equity of every port. Consequently, it increases the matching probability. The algorithm can greatly reduce the scheduling delay and cell loss rate. The analysis and simulation results by OPNET show that the proposed algorithm has the better performance than others in average delay and cell loss rate, and has the equivalent complexity. On the basis of these results, the hardware realization and simulation based on FPGA are completed, which validate the feasibility of the new scheduling algorithm.

Keywords—Scheduling algorithm, input-buffer, serial scheduling, hardware design.

I. INTRODUCTION

ONBOARD switching is the development trends of satellite communication. Satellite can complete the data switch among multibeam by the switching technology, which satisfies the communication of different ground stations and consists the space-earth integration wireless communication network. Consequently, the satellite switching will greatly accelerate the development of satellite communication. At the present time, there are onboard switching and routing unit in the broadband satellite [1], [2].

In order to transfer datagram quickly, the onboard switching unit adopts Fixed-Length mechanism to switch cells. Input-buffer, Output-buffer and Shared-buffer are three different buffer strategy of switching unit [3]. If the switching unit has N ports, the access speed is $(N+1)$ times the port speed for Output-buffer, $2 \times N$ times for Shared-buffer, and 2 times for Input-buffer. Consequently, it is necessary to study the Input-buffer strategy in the satellite switch systems, in order to extend switch capacity and support higher port speed.

There are Head of line blocking (HOL) in the Input-buffer switcher [4] and [5] indicate the VOQ (Virtual Output Queue) technology can improve this question, which increase the throughput to 100%. The key to achieve high performance of

VOQ is designing a high efficiency and fairness scheduling algorithm, by which the input port and idle output port are matched. So it is important to study a high performance input-buffer scheduling algorithm with low complexity in satellite switching system based on crossbar switch fabric.

The common input-buffer scheduling algorithms mostly contain PIM (Parallel Iteration Matching), iSLIP (iterative round-robin matching with SLIP), and so on. The above algorithms have three steps in every iteration, which are request, response and accept. In these steps, all input and output ports compete freely for all input and output ports. Therefore the algorithms have poor performance and complexity to implement when the payload is higher. The serial scheduling idea [6], [7] overcomes the drawbacks above and alleviates the blocking of input/output ports by serial mode. The serial scheduling algorithm has the better performance than iSLIP under high payload, and is also easy to implement. A new scheduling algorithm based on the serial scheduling idea is proposed in this paper. The algorithm adjusts the output port arbitrating strategy for the better equity of every port to increase the matching probability, and then the scheduling delay and cell loss rate are greatly reduced.

This paper is arranged as follows: the new scheduling algorithm we propose is described in Section II, in which the idea of the algorithm and its performance analysis are introduced in detail; the simulation results by OPNET is analyzed in Section III; then the hardware realization and simulation based on FPGA are showed in Section IV, including the hardware design of scheduling module, the relation between operating frequency and port speed, and the simulation results based on FPGA. At last, Section V concludes the paper.

II. THE IMPROVED INPUT-BUFFER SCHEDULING ALGORITHM

A. Algorithm Ideas

Satellite channel has the characters of long delay and the chip resources for satellite are limited, so that the scheduling algorithm of onboard switch needs to possess the better performance, such as short scheduling delay and low cell loss rate. In this paper, the original output serial scheduling algorithm is improved, and the new algorithm arbitrates the output port with minimum requesting input ports first. Then the output ports which have more requesting input ports are arbitrated late. This new algorithm can improve the

Yi Zhang is with the Science and Technology on Space Microwave laboratory China Academy of Space Technology (Xi'an), CAST, Xi'an China (phone: 86-29-85613415; e-mail: lbzytogether@163.com).

Quan Zhou, Jun Li, and Yanlang Hu are with the Science and Technology on Space Microwave laboratory China Academy of Space Technology (Xi'an), CAST, Xi'an China.

performance of onboard switch by increasing the matching probability among input and output ports. Every port adopts polling pointer to arbitrate, and the pointer is updated after completing the matches in order to ensure equity.

The improved algorithm proposed by this paper has two steps:

First, every input port sends requesting signal to the output ports where the cells in its queues may arrive to.

Second, all output ports which receive the requests from input ports are arbitrated.

In the second step, the output port with minimum requests is arbitrated first. And the output ports with more requests are arbitrated late. When the output port is arbitrated, the unmatched input port which is pointed by the polling pointer is chose. Then every input port is notified that if their requests are licensed, and the polling pointers of the output port are updated.

After completing the arbitration of one output port, the next output port is arbitrated in order, until all output ports are arbitrated.

For different scheduling algorithm, it is difficult to compare their performance by setting up a mathematical model. Accordingly, this paper analyzes the performance of proposed algorithm first. Then the analysis results are proved by simulations. At last, the feasibility is validated by the design based on FPGA.

B. Performance Analysis

The major indexes to evaluate the performance of scheduling algorithm for Crossbar are bandwidth utilization, scheduling delay, fairness, and so on. The bandwidth utilization is the ratio between the number of matched input/output ports and the number of total ports. If the total ports are invariable, the bandwidth utilization is higher with more matched input/output ports, and the average delay of cell is shorter with less waiting time in the buffer queues. When the buffer queues are limited, more ports number of every matching will reduce the cell loss ratio.

1. Influence of Arbitration Order on Bandwidth Utilization

Firstly, we analyze and compare the influence of arbitration order on matched input/output ports.

Theorem 1: Suppose the output port i and output port j have received some requests from input ports, the amounts of which are k_1 and k_2 , and $k_1 > k_2$. The matched probability of port i , when matching the port j first, is larger than the matched probability of port j when matching the port i first.

Proof:

- In case there are no iterative requesting input ports between output port i and output port j , the two matching order have the same matched numbers.
- In case there are k iterative requesting input ports between output port i and output port j ($k < k_1, k_2$), the results as follows.

If the input ports which the first matching output port has chosen do not belong to the iterative requesting input ports, the order doesn't influence on the matching result. Then we consider mostly that the input ports which the first matching output port belong to these k iterative requesting input ports.

When matching the output port i first, the probability of choosing the iterative requesting input ports is k/k_1 . Then the output port j will be matched. The number of optional input ports is k_2-1 . Such being the case, the matched probability of output port j is:

$$p1 = \frac{k}{k1} \times \frac{k2-1}{k2}$$

In the same way, if matching the output port j first, the matched probability of output port i is:

$$p2 = \frac{k}{k2} \times \frac{k1-1}{k1}$$

Because $k_1 > k_2$, the conclusion is $p_2 > p_1$.

Prove over.

In conclusion, the bandwidth utilization will be advanced through arbitrating the output port which has fewer requests from input ports first.

2. Compare of the Most Waiting Delay in the Case of Keepfull

Theorem 2: Suppose the cells have arrived in the case of keepfull, which means that all the VOQ queues are full at every time. With the original serial scheduling algorithm, the most waiting delay of the cells waiting for switching in the VOQ queues is N^2 , and for the improved algorithm the most waiting delay is N .

Proof: On the assumption that the cell becomes the HOL (Head Of Line) cell of queue $Q(i, j)$ at the time of t .

- After the time of t , there are N -time iterations beginning to arbitrate from output port j by the definition of original algorithm. Because the Round Robin pointer is modified only after the first arbitration, there must be one time that the pointer of output port j points to i in the N -time iterations. Consequently, the cell will be switched to the output port j in the N^2 -slots.
- In the case of keep full, the improved algorithm can match all input and output ports, because the Round Robin pointers will be updated after every successful matching and the pointers are different of every output port. Every output port will receive N requests from input ports. If the arbitration order is unchanged, there must be one time that the pointer of output port j points to i . Consequently, the cell will be switched to the output port j in the N -slots.

Proof over.

According to the analysis above, the improved algorithm proposed in this paper arbitrates the output port with minimum requests first. Although the output ports with more requests are arbitrated late, the matched probability is greater than the output ports which have fewer requests. So the improved algorithm will match more input/output ports than original algorithm in every cell slot. And the new algorithm improves the performance of onboard switch, such as scheduling delay, cell loss ratio, and so on. In the other way, the algorithm updates the pointers of output ports after every successful matching in order to guarantee the equity.

III. THE ANALYSIS AND SIMULATION RESULTS BY OPNET

This chapter compares the performance between the improved algorithm and other algorithms by OPNET simulation. The iSLIP, which can be convergence after less $\log_2 N$ -th iterations [8], is a representative of the scheduling algorithms. Therefore this chapter will simulate these three algorithms, which are 4-SLIP algorithm, original output serial scheduling algorithm and the improved algorithm. Then the average scheduling delay of these algorithms is compared and analyzed by the simulation results, which has the high load and the burst length of cells is 32. And the cell loss ratio with limited buffer queues is also compared between the original output serial scheduling algorithm and the improved algorithm.

In Fig. 1, when the burst cells are arrived, the improved algorithm has the minimum average delay than the others obviously, and the delay of original output serial scheduling algorithm is less than the 4-SLIP algorithm. The output serial scheduling algorithm avoids the synchronization in parallel iterative algorithm to increase the matched ports in every scheduling and reduce the average delay, so the delay of this algorithm is less than iSLIP algorithm. In addition, the improved algorithm in this paper arbitrates the output port with minimum requesting input ports first, which reduces the average delay by increasing the matching probability among input/output ports. And the improved algorithm updates the polling pointer after completing the matches in order to ensure equity.

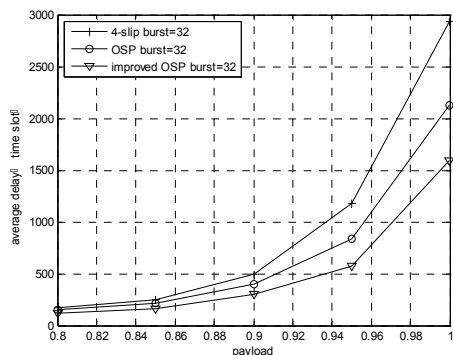


Fig. 1 Comparison of average delay in the 16×16 switching network, burst length=32

Fig. 2 compares cell loss ratio between original output serial scheduling algorithm and the improved algorithm with different VOQ lengths, for which the payload is 0.65 and 0.95, the burst length is 10. The conclusion is found from the figure that the improved algorithm has the smaller cell loss ratio than original output serial scheduling algorithm. From another point of view, the cell loss ratio with the lower payload (payload is 0.65) is smaller than the higher payload (payload is 0.95) in both two algorithms. And the cell loss ratio can be decreased by increasing some buffer for the lower payload (payload is 0.65). But there need more buffer for the higher payload (payload is 0.95).

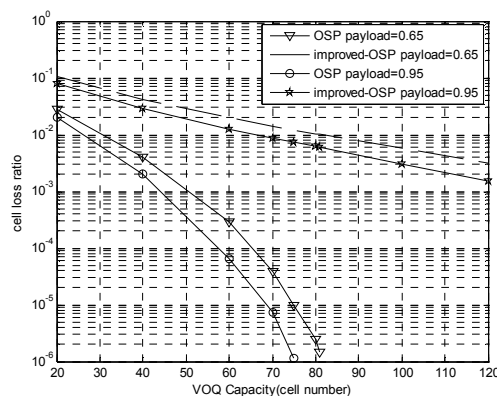


Fig. 2 The relation among VOQ, scheduling algorithms and cell loss rate in the 16×16 switching network, burst length=10

IV. HARDWARE REALIZATION AND SIMULATION BASED ON FPGA

A. Hardware Design of Scheduling Module

Fig. 3 shows the implementation framework of the scheduling module for input-buffer scheduling algorithm in satellite switching system. The scheduling module is composed of timing generation, pointer generation, requesting process, arbitration, output control, and so on.

The timing generation module generates the output port which needs to be polled currently by requesting of the improved algorithm, and the polling pointer of this output port is generated by the pointer generation module. The arbitration module completes the arbitration of the output port by the requesting and current polling pointer, then the arbitration results is delivered to the output control module, and the round robin pointer is updated at the same time. When the output control module has received arbitration results of all output ports, the results are delivered to corresponding input ports and the switch module in order to switch the cell for next time slot.

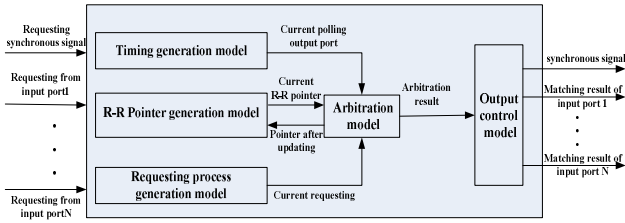


Fig. 3 Implementation framework of scheduling module

B. Relation between Operation Frequency and Port Speed

The improved scheduling algorithm proposed by this paper needs $N+1$ clock rhythms to complete one schedule for the $N \times N$ onboard switches. The scheduling module must calculate the ports matching results of next time slot during the switching of only one cell. Consequently, the total time of scheduling should be less than the time during which one cell can be switched.

Suppose the port speed of Switches is $Vbps$, the operating frequency of chip in the scheduling module is MHz , and the port number is N .

Because the onboard switches is based on fix cell of 64 bytes in the switch module, the arriving cell number per second is $V/(64 \times 8)$, and the cell arriving period is $T = (64 \times 8) / V$ seconds, that is to say a cell should be switched during the time T . On the other hand, there needs $N+1$ clock rhythms for every scheduling, and the time of one scheduling is $(N+1) / M$ seconds.

In order to satisfy $(64 \times 8) / V \geq (N+1) / M$, the relation between operating frequency and port speed is $M \geq (N+1) \times V / (64 \times 8)$.

On the basis of the analysis results above, if the operating frequency of scheduling module is 50MHz and the port number of onboard switches is 16, the time of one scheduling is $\frac{16+1}{50MHz} = 340ns$.

Then the port speed of onboard switches which the scheduling module can support is $V \leq (64 \times 8 \times M) / (N+1) \approx 1.5Gbps$.

C. Simulation Results Based On FPGA

In this chapter, the simulation results based on FPGA and the matching results computed by the improved algorithm are compared. If the two results accord, that is to say the algorithm is realizable.

The input of input-buffer scheduling algorithm based FPGA is requesting of every input port. The matching results of every input port are got by simulation. If this input port is matched with output port j , the position of j in the result delivered to the input port is set as 1.

Then the algorithm is validated by simulation during one cell time slot. The matrix A in Fig. 4 shows the input information supposed in the simulation. The expression $A(i, j) = 1$ means that the input port i has cells to be sent to output port j .

0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 4 Matrix A

In this example, the round robin pointers of all output ports are [3,6,7,8,9,1,2,10,11,12,13,14,15,16,5,4]. According to the improved algorithm, the results of this matching are (0,14), (2,2), (4,3), (5,13), (6,5), (7,6), (8,4), (9,8), (10,7), (11,11), (12,9), (13,10), (14,15), (15,12). The result of (i, j) means that the input port i is matched with output port j .

In this scheduling, the input port 3 is not matched. In this paper, the FPGA from Xilinx company, which belongs to Virtex-II and has 3000,000 gates, is chosen to implement the improved input-buffer serial scheduling algorithm. The operating frequency is 50MHz, and the port speed in this simulation can attain the frequency of 1.5Gbps. The simulation platform is ModelSim SE 6.1f. In order to close to the actual situation, the timing simulation based on FPGA is adopted to validate the hardware design of the improved algorithm. Fig. 5 shows the simulation result by ModelSim. The simulation result is compared with the computed result, and the two results accord, that is to say the simulation is correct. Consequently, the improved input-buffer scheduling algorithm is realizable.

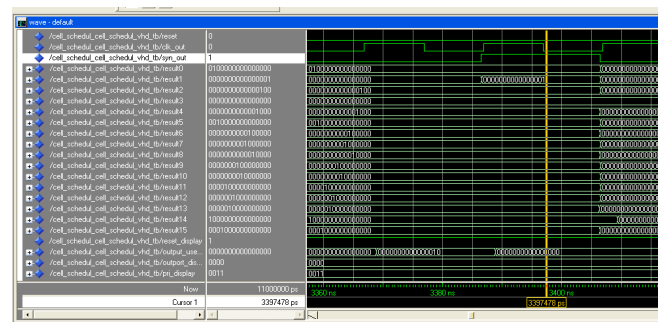


Fig. 5 The simulation result of improved algorithm by ModelSim

V. CONCLUSION

This paper improves the original serial scheduling algorithm in order to increase the performance of onboard switches. The simulation and comparison of several different scheduling algorithms are completed with the burst cells arriving. The results show that the proposed algorithm has the better performance than others in average delay and cell loss rate. When the buffer is limited, the improved algorithm has the smaller cell loss ratio than original output serial scheduling

algorithm. At last, the improved input-buffer scheduling algorithm has the equivalent complexity and is realizable by hardware simulation and analysis.

As an important algorithm in the Crossbar switch fabric with input buffer, the proposed algorithm can be used in the ATM switches and IP router, Furthermore, it can also be used in multistage switching fabric as a basic switch unit. Consequently, this algorithm can be considered to adopt in the onboard switches for which the chips are limited.

REFERENCES

- [1] Winds: wideband internetworking engineering test and demonstration satellite [J]. *Satellite & Network*, 2007(8):54-55.
- [2] Jun Li, Quan Zhou. Connection Admission Control in Satellite ATM Switching System: A New Improved Strategy [J]. *Journal of Astronautics*, 2006, 27 (3):513-517.
- [3] Xi sheng Chen. ATM Switching Technology [M]. Beijing: People Posts and Telecommunications Press, 2000:59-61.
- [4] Hakyong Kim, Kiseon Kim. Performance Analysis of the Multiple Input-Queued Packet Switch with the Restricted Rule [J]. *IEEE/ACM Transactions on networking*, 2003, 11 (3):478-487.
- [5] Wei Li-hua, Tang Yu-hua. Reseach of scheduling input-queue algorithms with Crossbar [J]. 2006, 23(3):22-24.
- [6] Sun Zhi-gang, Su Jin-shu, Lu Xi-cheng. ISP: A High Performance Crossbar Arbitrating Algorithm [J]. *Chinese J. Computers*. 2000, 23 (10):1078-1082.
- [7] Yi Zhang, Quan Zhou, Jun .Li. An Input-Buffer Scheduling Algorithm in Satellite Switching System [J], *Journal of Electronics and Information Technology*, 2009. 31(6): 1429-1432.
- [8] Hao Zeng-hui, Li Wen-jiang. Study on scheduling algorithms for VOQ input-buffer switches [J]. *Radio Communications Technology*. 2006, 32(6):59-61.