Journal of Theoretical and Applied Information Technology
31 August 2012. Vol. 42 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ONE FAIR SCHEDULING ALGORITHM OF INPUT BUFFER SWITCH

[1] DAIYUN WENG, [2] LI YANG
School of Information Engineering, Chongqing City Management College, Chongqing, China

ABSTRACT

As functional components like switch structure and cache do not need acceleration, the input buffer program becomes the mainstream structure of high-performance devices. Fair resource allocation is a necessary condition for achieving service quality control. The switch scheduling should take both scheduling efficiency and fair resource allocation into account. An iterative matching scheduling algorithm iPFQ for input buffer switch was brought out based on iSLIP. Simulation experiments were used to verify average delay, throughput as well as link bandwidth allocation fairness under different loads when it is uniform distribution and non-uniform Bernoulli arriving probability. The experimental results show that iPFQ can achieve high scheduling efficiency under fair resource allocation.

Keywords: *Input Buffer, Service Quality, Iterative Scheduling Algorithm, Fair Scheduling*

1. INTRODUCTION

The rapid development of fiber optic transmission technology eliminates bandwidth bottleneck problem which restricts network development in long-term and makes high-speed networks possible [1]. As bandwidth continues to improve, Internet whose core is IP is gradually becoming unified information exchange infrastructure to provide emerging network applications like video conferencing, visual computing, medical imaging, etc. The diversity of applications means diversification of quality of service requirements. Service quality control has become an important issue in the design of high-performance network devices [2-4].

The Quality of Service (QoS) means the ability of network to provide different levels of services for different users [5]. In essence, QoS does not create new resources. Therefore, appropriate resource supply is the prerequisite for network to provide QoS support. QoS control just manages and applies network resources effectively based on application needs and network status. The key elements of network QoS control are traffic differentiation and fair resource allocation. It is possible to cater for needs of different applications by differentiating traffic and allocate resources based on different QoS needs.

Ref. [6] pointed out that the bandwidth allocation is fair if the service rate of each stream is proportional to its share of resource reservation. The fair scheduling policy of output buffer switch has been in-depth researches and many results have been made. Ref. [7] made a very good overview. The paper is mainly based on the input buffer architecture of high performance network equipment scheduling algorithm efficiency and fairness issues. Although switches occupy the main work, routers reach packets from the input link to output link due to the random nature of packet arrival. If measures were not taken within routers and switches, fair scheduling would not be truly realized [8]. Therefore, the complexity of input buffer switch is higher than that of output buffer switch. Till now, fewer literatures are focused on fair scheduling of input buffer switches [8-10].

Throughput and fairness are two main performance indexes in the scheduling algorithm design of input buffer switch. The scheduling algorithm with higher throughput may have poor fairness, such as iSLIP [11]. The transmission scheduling which is only based on fair resource allocation principle cannot achieve higher throughput [8]. Therefore, the fairness of resource allocation and throughput should be compromised and balanced in switch scheduling. The paper proposes an iterative matching scheduling algorithm iPFQ for input buffer switch which was brought out based on iSLIP and is organized as follows: section 2 introduces related works; section 3 proposes fair scheduling algorithm of input buffer switch; section 4 verifies algorithm performance with simulation experiments; section 5 concludes our work.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. SCHEDULING ALGORITHM OF INPUT BUFFER SWITCHES

The scheduling of input buffer switch is the essential problem of bipartite graph matching. At present, practical scheduling algorithms are usually iterative matching algorithms such as iSLIP [11]. In case of each time of scheduling, the iterative matching algorithm should conduct times of matching processes. Each process has three phases including REQUEST, GRANT and ACCEPT. In the REQUEST, each input that has not been matched sends request to all possible outputs. In the GRANT, output ports that have not been matched select an input from many requests to send response. In the ACCEPT, un-matched input selects a response to recognize. Conditions that the input sent GRANT and output received GRANT are considered to be establishing connections. After scheduling, the input-output pair can transmit a packet.

The main difference between different iterative matching algorithms is selective method in grant/accept phase. The earliest iterative matching algorithm is PIM [13]. In the GRANT of PIM, each output port randomly selects an input port to send request signal. In the ACCEPT, each input port also randomly selects a response signal to send ACCEPT. The problem of PIM is that it is difficult to realize random selector with hardware and to assure fairness of random selection [8]. Basid Round-Robin matching algorithm (RRM) uses cyclic priority strategies for contention arbitration. In the GRANT, it outputs priority of port. In the ACCEPT, input port sends accept signal to output port with highest priority and update priority of each output. The advantage of RRM algorithm is that hardware implementation is relatively simple. In the GRANT, update of each input port priority in the input port is receiving ACCEPT before. No matter whether GRANT has been transmitted by output port or been accepted by corresponding input port, the input port priority would change. Therefore, there is synchronization in RRM. As to synchronization, the higher load of switch, the lower throughput of it. The iSLIP simply improved priority update strategy in RRM. In the GRANT, output port only updates after receiving the accept signal input port of the processing of input port-priority. If the ACCEPT was not received, the priority of each input port remained unchanged. Priority update strategy simply eliminates synchronization and greatly improves switch performance. The Ref. [15] has proved that throughput of iSLIP can reach up to 100%.

The focus of PIM, RRM and iSLIP is scheduling efficiency and the target of scheduling algorithm is to achieve most matching, namely most input-output pairs. In case of conflict, switch equally treats each input port. Output link bandwidth evenly distributes among each input port. However, average does not mean fair. In the Statistical Matching (SM) algorithm [13], it is the output port not the input port which conducts iterative processing. Output port randomly selects an input port to send GRANT, no matter whether input port received GRANT which is waiting for packet transmission or not. The probability of output port selecting some input port to send GRANT is proportional to the reservation of bandwidth in the output link. As SM sends GRANT without considering occupation status in input port queue, the input port receiving GRANT may not have packet queuing for transmission. Therefore, the blindness of SM sending GRANT results in low throughput of switch.

Weighted Probabilistic Iterative Matching (WPIM) algorithm [9] adds fair scheduling mechanism. It divides time into frames containing fixed number of slots and determines packet forward number in a frame with resource reserving mechanism. In the GRANT, WPIM randomly selects one stream which does not exceed reserved number to GRANT. WPIM achieves effective separation of stream by restricting packet number which can be transmitted in a frame. However, WPIM in delivery volume did not reach the reserved share of the flow under the same probability of selection. The output link bandwidth is still evenly distributed in the activity flow among the excess sent, the reserved share of the flow no relationship, the experiment also proved this point [8].

Iterative Fair Scheduling (IFS) [8] running generalized processor sharing (GPS) is each output port to maintain virtual time. The engine computes starting and ending time under GPS server for each arrival packet according to resource reserving number. Output port selects the least virtual starting time from conflicting packets to providing service. In the ACCEPT, input port uses First Come First Served (FCFS) for arbitration. IFS refers to fair queuing idea from flow-level to achieving fair allocation of link bandwidth. But the computation has high cost, which is not easy for implementation with hardware.

The idea of Iterative Deficit-round-robin (IDRR) is to assign a quota that is proportional to resource reserved for each stream. After input port

Journal of Theoretical and Applied Information Technology
31 August 2012. Vol. 42 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

establishing connection with output port, it continuously transmits packets until the quota finished. Each input port records active stream information with list out-FlowList. In the GRANT, the un-matched output port selects the first one from this list to send REQUEST. In the ACCEPT, input post selects the first output port from inFlowList receiving GRANT to send ACCEPT. After connection between input and output is established, it will transmit quota packets continuously. The essence of IDRR is still round-robin. It can achieve fairness of output link bandwidth allocation by setting quota that is proportional to resource reserved, which is also easy to be implemented with hardware. The problem of IDRR is that other active streams in the same input may not access to service, resulting in larger delay jitter.

3. ITERATIVE PORT-BASED FAIR SCHEDULING ALGORITHM

3.1 BASIC IDEA

In high-speed network environment, the scheduling time of router and switch should be as little as possible. The scheduling algorithm must be easily implemented with hardware. As to fair sharing of output link bandwidth, flow-based fair scheduling algorithm is the best. However, due to too many active flows in the core network, it is difficult to implement it with hardware. The port-based fair scheduling algorithm is more practical.

The iterative port-based fair scheduling algorithm (iPFQ) proposed in the paper is also based on iSLIP, which is also an iterative matching algorithm based on port. Similar to iSLIP, the output port of iPFQ utilizes iterative priority scheduling strategy for arbitration among several conflicting input ports. Input port determines among many received GRANT with FCFS principle. Different from iSLIP, the probability of an output linking with the arriving packets from the input port is authorized to use the probability of an output link and reaching the stream from the input port is proportional to the amount of resources reserved in the output link in order to ensure the fairness of the link bandwidth allocation. The iPFQ mainly improves iterative priority wheel configuration method of iSLIP so that the number of some input port in wheel is proportional to that of reserved bandwidth in this output link.

In the iPFQ, each output port maintains an iterative priority wheel as shown in Fig. 1. The signal in wheel is input port identifier. The wheel rotates in a certain direction using pointer to element with the highest priority currently. The priority of other elements decreases correspondingly. The maximum element number in priority is called length of wheel. It takes an integer larger than N, which is usually a multiple of N. Where, N is port number of router or switch.

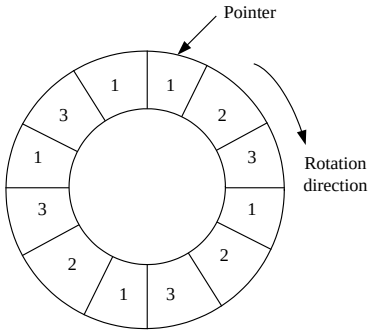

Figure 1: Iterative priority wheel

In terms of stream that expects to access bandwidth assurance in output link, it is necessary to perform resource reservation in advance. When the stream is conducting resource reservation in switch, it can acknowledge basic information of stream, resource reservation bandwidth. The problem to implement resource reservation is not the focus in the paper. We just assume that switch understands resource reservation information in output link and bandwidth.

Define the packet from input i and destination is output j belonging to flow $f(i, j)$. The reserved bandwidth of $f(i, j)$ in output link j whose capacity is C_j is $R_{i,j}$. The output link j is link connecting with port j. Based on formula, the number $P_{i,j}$ that input port i in iterative priority wheel of output port j can be computed. The LC_j in formula is element number in priority wheel of output port j. Although each output port can set element number of maintaining iterative priority wheel based on specific status of link bandwidth, the element number of each output port in the implementation can be equally set.

$$P_{i,j} = \left\lceil \frac{R_{i,j} \times LC_j}{C_j} \right\rceil \quad (1)$$

The granularity of the output link bandwidth allocation determines minimum output link bandwidth that can be reserved. As the input port sign should be at least once in some iterative list, the reserved minimum link bandwidth is C_j/LC_j. It shows that element number of wheel is related to link bandwidth allocation granularity. More number

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

in priority wheel means smaller bandwidth allocation granularity. However, the storage and management of priority of wheel may be more complex. Therefore, selection of priority wheel number should comprehensively take factors as implementation complexity and bandwidth allocation granularity into account. In addition, if an input port sign shows many times in the wheel, it should try to evenly allocate it in the wheel so as to avoid output port consecutive multiple scheduling grant signal granting the same stream and cause other long time not to get the service of the phenomenon. The paper will give another priority wheel configuration algorithm.

3.2 iPFQ Algorithm

In the GRANT, output port of iPFQ select one port with highest priority from many REQUESTs according to iterative priority strategy to GRANT. The method is as following: check whether received priority round of the highest priority element points to input port to send the request signal. If so, send GRANT to this input port and the selection process ends. Otherwise, the wheel rotates and highest priority points to next one to continue the process.

In the ACCEPT, input port select one from received GRANT based on FCFS and send ACCEPT to corresponding output port. As the conflicting packets in input port cone from same upstream node, if all nodes in the network use same scheduling strategy, the packet from first arrival node is also the first one leave upstream node. So it is natural to conduct arbitration with FCFS strategy.

The specific iPFQ algorithm is as following:

Step 1: Initialization. Based on resource reserved status, each output port computes number of each input port sign in the priority wheel $P_{i,j}(i, j = 0,1,2,\cdots,N-1)$ and try to allocate input port sign in wheel evenly as possible. Each scheduling needs $\log_2 N$ iterative matching process, where N is port number of router and switch.

Step 2: Each iterative matching process can be divided into three phases:

(a) Request phase: Fro any un-matched input port i, if $I(i, j) > 0(i, j = 1,2,\cdots,N-1)$, send REQUEST signal to output port j. where, $L(i, j)$ packet number waiting for transmission in output port j corresponding to input port i.

(b) Grant phase. As to any un-matched output port, select one from received REQUESTs with highest priority. Send GRANT to corresponding input port and notify port that has not been selected.

(c) Accept phase. Input port selects a output port from received GRANT with FCFS to send ACCEPT signal. The output port received ACCEPT rotate priority wheel so that the highest pointer to next element. The input port establish connection with output port received this ACCEPT.

Step 3: After each time of scheduling, configure switch structure based on matching result. Establish connection between corresponding input port and output port. Transmit a packet from input to output.

3.3 Time Complexity Analysis

The iterative priority wheel length of iSLIP is N. All input port sign can be emerge in wheel only once. The length of iPFQ is a number larger than N, which is usually multiple of N. Some input port sign may be emerged in wheel for times. The number of input port in wheel is proportional to link reserved bandwidth. The increase of iterative priority wheel length is basis of fair allocation in iPFQ, which also result in time of GRANT in iPFQ larger than that in iSLIP.

In case of light network load and the GRANT of iPFQ, the priority wheel needs to rotate *LC*-1 elements to find signal access port for transmitting REQUEST, where *LC* is length of wheel. However, the iSLIP only needs to rotate *N*-1 elements to access REQUEST. As *LC* is larger than *N*, the arbitration time of iPFQ is longer than that of iSLIP in case of light load. In case of heavy load, output port usually receives REQUEST from input port corresponding to element with highest priority. Therefore, the time in GRANT of iPFQ and iSLIP is same to (0, 1).

3.4 Configuration Method of Priority Wheel

If a input port appears many times in the wheel, how to determine location of this port in wheel has nothing to do with fair allocation of link bandwidth, while it may affect delay jitter of packet and waiting time before other flow access to GRANT. Therefore, if a port appears many times in wheel, the port sign should be evenly allocated in wheel. Here we give a configuration method. Without loss of generality, take priority wheel configuration at output port j as example.

Assume the reserved bandwidth in output link j of flow $f(i, j)$ is $R_{i,j}$. The number of each input port appear in iterative priority wheel j is $P_{i,j}(i, j = 0,1,2,\cdots,N-1)$.

Journal of Theoretical and Applied Information Technology
31 August 2012. Vol. 42 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
int i, j, k, wheel[LCj-1];
for (i=0; i<LCj; i++)
wheel[i]=-1;
i=0;
while(i<LCj) {
for(k=0;k<N;k++)
if(Pk,j→0) wheel[i++]=k;
}
```

4. SIMULATION EXPERIMENTS

Typically, isLIP is considered as scheduling algorithm with highest efficiency [8] and iFS is algorithm with best fair performance [10]. The simulation assesses efficiency and bandwidth allocation fairness of three scheduling algorithms as iPFQ, iSLIP and iFS. The experiment results show that iPFQ can achieve similar throughput with iSLIP, while access to same fairness with iFS.

4.1 SCHEDULING ALGORITHM EFFICIENCY

The main evaluation criterion of efficiency is average delay and throughput of packet. The paper examines algorithm efficiency under uniform Bernoulli arrival and on-uniform Bernoulli arrival.

(1) Uniform Bernoulli arrival

Under uniform Bernoulli arrival, the load of input link is packet arrival probability in a slot. Meanwhile, the packet destination is uniformly distributed among all output ports. In the experiment, size of VOQ is 16×16, namely switch has 16 inputs and 16 outputs. The packet length is 64Byte and each time of scheduling needs 4 iterations. Fig. 2 shows tread of packet average delay changes along input link load of three algorithms. We can see that average delay of iPFQ is same to that of iFS and close to that of iSLIP. When the link load greater than 0.9, the average packet delay of three algorithm grows to converge. The iSLIP is scheduling algorithm with minimum average delay [8]. The delay of iPFQ and iSLIP is almost same to each other. Therefore, under uniform Bernoulli arrival, iPFQ can also reach 100% throughput.

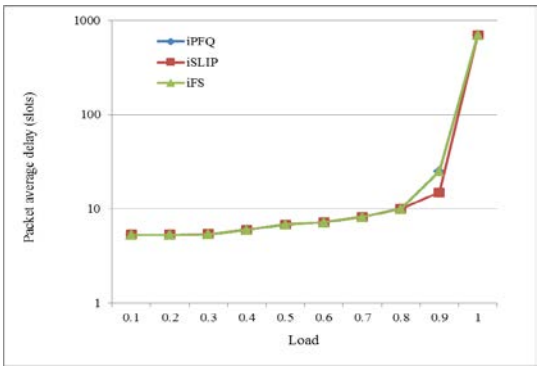

Figure 2: Average packet delay under uniform Bernoulli arrival

(2) Non-uniform Bernoulli arrival

The paper uses flow arrival model in [9] to assess scheduling algorithm efficiency under non-uniform Bernoulli arrival. Assume switch has 16 ports, in which 4 ports connect to server and 12 connect to clients. Each client generates 40% flow to server, which is evenly distributed on each server. The remaining 60% flow destinies to other clients and the flow also distributed evenly. Each server generates 96% flow to 12 clients evenly.

Figure 3 shows packet average delay changes along load under three scheduling algorithms. We can see from the figure that three curves representing iSLIP, IFS and iPFQ almost overlap. The throughput of switch can reach 79%. Assuming input link load is l, the output link load to server is $L_{server}=12\times0.4\times l/4+4\times0.04\times l/4$. When $L_{server}=l$, $l \approx 0.806$. It indicates that when input link load is 0.806, the output link to server is overload. At this moment, the packets accumulate in input cache and average packet delay increase sharply.

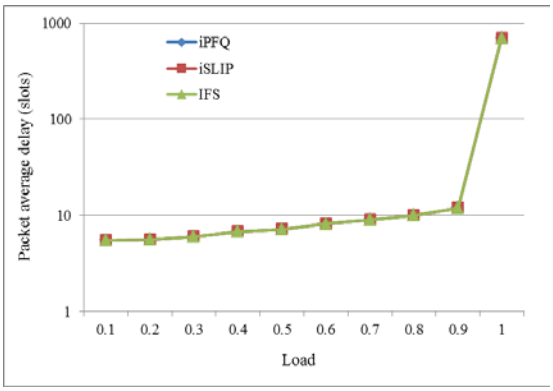

Figure 3: Packet average delay under non-uniform Bernoulli arrival

Journal of Theoretical and Applied Information Technology
31 August 2012. Vol. 42 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2 FAIRNESS OF BANDWIDTH ALLOCATION

To clearly show results, the VOQ size in simulation is 4×4. The element number of each output priority wheel is 40. Each input port has 4 flows to different output port. Without loss of generality, assume the flow *f*(0, 0), *f*(1, 0), *f*(2, 0) and *f*(3, 0) reserve 10%, 20%, 30%, 40& output link capacity connecting to output port 0. Actually, the arrival speed of 4 flows is same as 85% link load. The arrival speed of other flow is about 5% of link load. Set $\Gamma = [\lambda_{i,j}]$ as flow arrival intensity matrix; $\lambda_{i,j}$ is arrival speed of *f*(*i*, *j*); I is load of input link. Then:

$$\Gamma = [\lambda_{i,j}] = l \begin{bmatrix} 0.85 & 0.05 & 0.05 & 0.05 \\ 0.85 & 0.05 & 0.05 & 0.05 \\ 0.85 & 0.05 & 0.05 & 0.05 \\ 0.85 & 0.05 & 0.05 & 0.05 \end{bmatrix} \quad (2)$$

Fig. 4 and Fig. 5 show actual service speed of output port 0 with iSLIP and iPFQ. If input link load less than 29.4% of output link capacity, the load on output link 0 is less than link capacity.

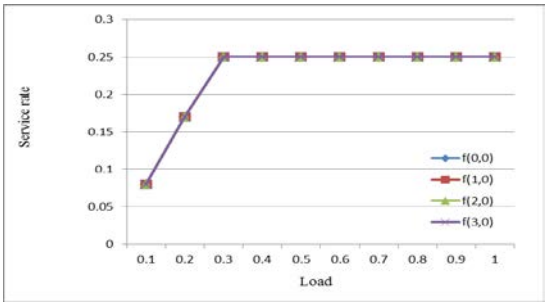

Figure 4: Flow service speed with iSLIP

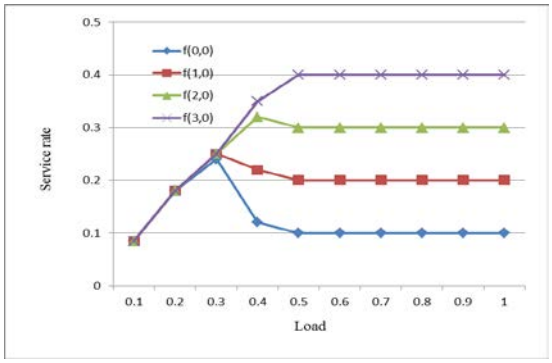

Figure 5: Flow service with iPFQ

If the input link 0 do not overload, the flow service speed is about actual arrival speed of flow. The scheduler as iSLIP and iPFQ tah run ccording to work conserving mechanism will not be in idle when packets waiting for transmission in input port. Therefore, in case of slight load, the output link can transmit. At this moment, it has no sense ask for resource reservation.

In case of output link 0 overload, the output port 0 with iSLIP may evenly allocate 4 flows within output link bandwidth. The actual service rate of each stream is 25% of the output link capacity, without considering different resource reservation. The reason is that element number in priority wheel of iSLIP is same to port number. The input port sign only appears once in wheel, which has nothing to do with resource reservation amount in case of different input link. When the load is heavy, there are almost packets in each input queue waiting for transmission by output port 0. Therefore, output port 0 may receive REQUEST from 4 input output ports generally. The GRANT arbitration in port 0 uses Round-Robin strategy and output link serves 4 flows in turn. Link bandwidth is actually divided equally among 4 flows as shown in Fig. 4. With iPFQ, when input link load greater than 24.9% of output capacity, iPFQ can isolate flow exceed resource reserved from practical transmission rate, so that it may not affect normal packet transmission, as shown in Fig. 5. If input link load is within 29.4%-47.1%, there is always one or several flow arrival rate is less than the reserved bandwidth. As the input port sign number in wheel is proportional to corresponding resource reserved in output link, the unused reserved link bandwidth evenly share reserved bandwidth. Therefore, the iPFQ can evenly allocate output link bandwidth resource as desired.

5. CONCLUSIONS

Rapid development of transmission technology requires higher performance of core devices. Input buffer structure solves extension of high-performance router and switch. Meanwhile, diversity of applications needs QoS from network. How to improve the throughput of the routers, switches and to provide flexible, easy to achieve quality of service control mechanism has become one of the key issues in the design of high performance routers and switches. Based on iSLIP, an iterative matching scheduling algorithm iPFQ for input buffer switch was proposed. It achieves packet conflicts in output port with iterative priority scheduling in GRANT, which is easy for implementation with hardware. Experiment results show that iPFQ has both efficiency and fairness. Its performance as packet average delay and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

throughput is same as iSLIP. At the same time, it can also ensure fairness of output link bandwidth allocation as iFS.

ACKNOWLEDGEMENTS

This work was supported by "Management System Design of EPON and Software Implementation of OAM Protocol"(KJ111702) and "Application of Information Sensing and Information Processing Technique in WSN" (KJ111701), the Projects of Chongqing Education Committee Science-Technology Program

REFRENCES:

[1] Kar K., Lakshman T. V., Stiliadis D., "Reduced complexity input buffered switches", *Proceedings of Hot Interconnects*, pp. 13-20, 2000.

[2] Nong Ge., Hamdi M., "On the provision of quality-of- service guarantees for input queued switches", *IEEE Communication Magazine*, Vol. 38, No. 12, pp. 62-69, 2000.

[3] Metz C., "IP routers: new tool for gigabit networking", *IEEE Internet Computing*, Vol. 2, No. 6, pp. 14-18, 1998.

[4] Keshav S., Sharma R., "Issues and trends in router design", *IEEE Communication Magazine*, Vol. 36, No. 5, pp. 144-151, 1998.

[5] Cisco Corporation, Internetworking technology handbook, http://www.cisco.com, 2007.12.

[6] Demers A., Keshav S., Shenker S., "Analysis and simulation of a fair queuing algorithm", *Internetworking Research and Experience*, Vol. 1, No. 1, pp. 3-26, 1990.

[7] Hui Zhang, Service disciplines for guaranteed performance service in packet-switching networks, *Proceeding of the IEEE*, Vol. 83, No. 10, pp. 1374-1396, 1995.

[8] Nan Ni, Bhuyan L., "Fair scheduling in Internet routers", *IEEE Transactions on Computers*, Vol. 51, No. 6, pp. 686-701, 2002.

[9] Stiliadis D., Varma A., "Providing bandwidth guarantees in an input-buffered crossbar switch", *Proceedings of INFOCOM*, pp. 960-968, 1995.

[10] Zhang Xiao, Laxmi B., "Deficit round-robin scheduling for input-queued switches", *IEEE Selected Areas in Communication*, Vol. 21, No. 4, pp. 584-594, 2003.

[11] Mckeown N., "The iSLIP scheduling algorithm for input-queued switches", *IEEE/ACM Transactions on Networking*, Vol. 7, No. 2, pp. 188-201, 1999.

[12] Karol M., Hluchyj G., Morgan S., "Input versus output queuing on a space-division packet switch", *IEEE Transactions on Communication*, Vol. 35, No. 12, pp. 1347-1356, 1987.

[13] Anderson T., Owicki S., Saxe J., "High speed switch scheduling for local area network", *ACM Transactions on Computer Systems*, Vol. 11, No. 4, pp. 319-352, 1993.

[14] Yuval T., Hsin-chou C., "Symmetric crossbar arbiter for VLSI communication switches", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 4, No. 1, pp. 13-27, 1993.

[15] Mckeown N., Mekkittikul A., Anantharam V., "Achieving 100% throughput in an input-queued switch", *IEEE Transactions on Communication*, Vol. 47, No. 8, pp. 1260-1267, 1999.

[16] Parkekh A., Gallager R., "A generalized processor sharing approach to flow control in integrated service networks: the single-node case", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 3, pp. 344-357, 1993.