A Scheduling Scheme to Improve QoS Provisioning for IP traffic

Oladayo Salami, *Member, IEEE* and H Anthony Chan, *SrMember, IEEE*
Department of Electrical Engineering, University of Cape Town, South Africa.
oladayo@crg.ee.uct.ac.za, h.a.chan@ieee.org .

Abstract

Output Queuing (OQ) and Input Queuing (IQ) are the two basic queuing strategies implemented in routers. IQ has been identified as the simplest and the most scalable. However, IQ achieves only 58.6% throughput due to the Head Of Line (HOL) blocking effect. The Virtual Output Queuing (VOQ) strategy is a proffered solution to the HOL blocking. It has been shown that VOQ can achieve a 100% throughput with an effective scheduling algorithm. This paper proposes a Multi stage Queuing and Scheduling strategy which implements VOQ at the input and OQ at the output of the router. The scheduling algorithm for the VOQ proposed in this paper is an Iterative Probabilistic Scheduling.

1. Introduction

The ever increasing demand in network capacity is being met by the technological advancement that has evolved since the 1990's. For example new generations of Ethernet are operating at 1Gbps, emerging from 10Mbps. During the early 2000, evolving next generation networks and heterogeneous networks like 4G targeted speeds of 70-200Mpbs. In addition, large ISPs began to use OC-192 circuits that transmit at approximately 10Gbps. Also, optical switching technologies with line rate of 10-40Gbps are being developed to meet the increasing demand in bandwidth [1].

Different applications sending packets through these networks also generate varying traffic patterns. Consequentially, mission critical traffic can coexist within a network thereby posing a challenge to the design of network elements. The challenge is that network elements need to provide QoS for different traffic types and they must do so at line speed.

The router, which is a major network element in networking, has undergone several technological changes over the years. It has emerged from the store and forward paradigm to the store- processing and forward paradigm [2]. This shift in paradigm in turn has brought about two major design issues.

The first issue is whether packet processing functions of the router should be performed at the input or at the output. Packet processing functions which are the fundamental processing operation of the router on a packet include policing, classification, queuing, shaping, and scheduling [1].

The second issue focuses on how to implement the above functions to scale with increasing network speed.

This paper proposes a solution to these issues with a focus on queuing and scheduling. We propose a multistage queuing and scheduling (MQAS) architecture. MQAS is a two-stage architecture which implements Virtual Output Queuing (VOQ) at the input ports and First-In First-Out (FIFO) queuing at the output ports of the router. The scheduling algorithm for the VOQ proposed in this paper is the Iterative Probabilistic Scheduling (IPS).

The organization of this paper is as follows; section 2 gives a summary of the Hierarchical QoS architecture, routers and packet processing functions. Queuing and Scheduling issues are discussed in Section 3. Analytical and simulation results of the FIFO and PIM algorithms are presented in section 4. Section 5 presents the proposed Queuing Architecture and Scheduling algorithm and section 6 gives a conclusion.

2. Hierarchical QoS Architecture.

This section provides a comprehensive summary of the hierarchical QoS architecture and a brief overview of routers and packet processing functions.

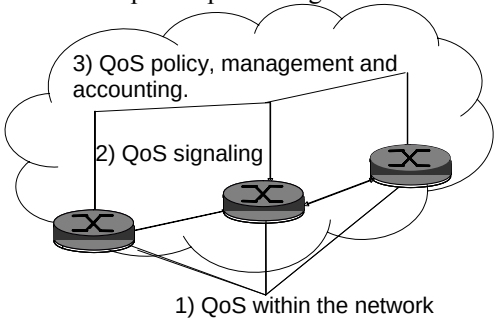

Fig. 1. Hierarchical QoS Architecture

Quality of service provisioning is implemented on a hierarchical architecture with three levels as shown in fig.1 [3]. The first level provides QoS within a single network element through classification, queuing, scheduling, and traffic-shaping techniques. At the second level, QoS signaling i.e. admission control and resource reservation are performed from end to end and between network elements. The third level provides QoS policy, management, and accounting, to control and administer end to end traffic across a network.

The focus of this paper is the first level of this architecture. This level provides QoS through packet processing functions performed within the router. Each of these packet processing functions monitors, controls, and ensures the provisioning of QoS for different traffic types. However QoS is provided to the traffic according to a Service Level Agreement (SLAs).

SLAs are implemented based on pre-specified and measurable QoS attributes. These attributes are delay, jitter, loss and throughput that traffic experiences in the network.

The router is a major network element which enforces these attributes through the packet processing functions. Routers are fundamental building blocks in any communication network. They are used to interconnect physical networks and also to forward traffic between them. They can be implemented at the access, edge, or core of a network. Routers knit together homogeneous and heterogeneous networks thereby creating an illusion of a unified network e.g. the internet [4]. A generic router has four major components which are the input ports, output ports, switching fabric and the processor

The input port is a point of attachment to a physical link and a point of entry for incoming packets. The operations carried out at the input port includes: buffering of packets, data link layer encapsulation and decapsulation, and packet destination look up.

The output port is the output interface to the transmission link. It also serves as a buffer for outgoing packet and support data link layer encapsulation and decapsulation.

The switching fabric is a hardware mechanism that interconnects input ports and output ports.

The processor is an important component within the router which performs complex packet processing functions. It is implemented in software, hardware or a hybrid of software and hardware known as network processor.

Packet processing is important in the provisioning of QoS. Without effective packet processing functions, it is almost impossible to deliver meaningful service guarantees during network congestion.

As packets arrive at the ingress policing ensures that the packets are within their SLA while being classified according to their QoS requirements. Queuing is buffering these packets within the router. Shaping ensures that these packets meet the requirement of the downstream network and do not cause congestion. Scheduling determines the optimum order to forward these packets.

The above processing functions can be implemented in software or hardware.

3. Queuing and Scheduling Issues.

In this section, we summarize the major issues of the queuing and scheduling functions in a router.

3.1. Queuing.

A queue is where the packet waits from its arrival time to its service time. A packet is required to wait only when the server is busy or there is congestion [5]. When this happens, queuing algorithms take effect.

These queuing algorithms provide strategies for buffering packets according to classification. Some examples of Queuing algorithms are Priority Queuing (PQ), First-In First-Out (FIFO), and Custom Queuing (CQ).

One important issue affecting the routers' scalability is where to strategically implement a queue for optimal performance in the router. Three categories of queuing strategies are 1) output queuing 2) input queuing 3) combined input and output queuing. These different queuing strategies and their challenges are discussed below.

The Output queuing (OQ) strategy is as shown in Fig. 2. Here packets are buffered at the output ports to maximize the throughput of the router. However, if packets destined for the same output port arrives simultaneously, then the buffer will have to queue these packets at a speed higher than the line speed. The required high speed places a scaling limitation on the router.

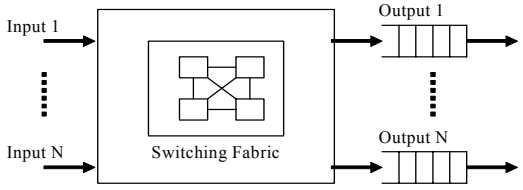

Fig. 2. Output Queuing

Input queuing (IQ) is as illustrated in Fig. 3. Packets are buffered at the input ports in this strategy and only one packet at a time, the first packet in any queue, is eligible for transmission at a time. Input queuing has no scaling limitations but it exhibits a performance bottleneck known as Head-Of-Line (HOL) blocking. HOL blocking happens when a packet at the head of a queue is blocked thereby preventing other packets behind it from being transmitted. It has been shown that HOL blocking reduces throughput to as low as 58.6% [6].

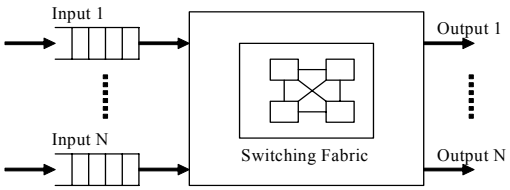

Fig. 3. Input Queuing.

Combined input-output queuing shown in Fig. 4 is a combination of input and output queuing. Packets are buffered at both the input and output ports. It is a good compromise between the performance of OQ and scalability of IQ.

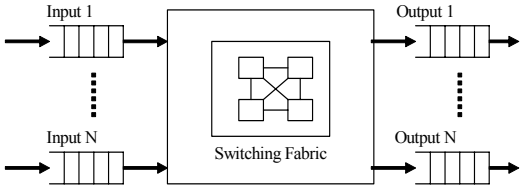

Fig. 4. Combined Input Output Queuing.

With all the advantages and drawbacks of the queuing strategies mentioned above, the simplest and most scalable approach is the input queuing [5]. However, input queuing has a draw back of the HOL blocking effect. To avoid the HOL blocking, Virtual Output Queuing (VOQ) in Fig. 5 was proposed [5].

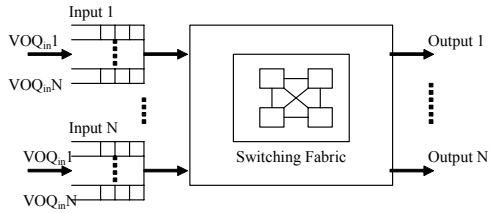

Fig. 5. Virtual Output Queuing.

VOQ is an input queuing strategy in which each input port maintains a separate queue for each output port. It has been shown that VOQ can achieve a 100% throughput performance with an effective scheduling algorithm. This scheduling algorithm should be able to provide a high speed mapping of packets from inputs to outputs on a cycle-to-cycle basis [6].

3.2. Scheduling.

Scheduling is a major component in QoS provisioning within the router. It is an algorithm which determines the order in which packets are served. The algorithm must be simple, fair and capable of preventing the starvation of any packet in a queue.

A scheduler can be either work conserving or non-work conserving [7]. A work conserving scheduler is one that is never idle if packets are waiting e.g. Fair Queue (FQ) and Weighted Fair Queue (WFQ) [8][9]. Conversely non-work conserving scheduling algorithms can be idle even though packets may be waiting for transmission e.g. hierarchical round robin, and jitter Earliest Due Date (Jitter EDD) [5].

The issue with packet scheduling in VOQ is similar to the bipartite graph-matching problem (BMP) which tries to find a conflict-free pairing of inputs to outputs [10][11]. The scheduler must retrieve the state information of all contending packets and perform a maximal matching of these packets to an output port. In addition, it must be able to arbitrate fairly among these packets under uniform and non uniform traffic.

Scheduling algorithms that satisfy these requirements are Longest Queue First (LQF) and Oldest Queue First (OCF) algorithm with $O(N^3 \log N)$ complexity. They also include Parallel Iterative Model (PIM) with $O(\log N)$ complexity. However these algorithms are too complex for hardware implementation [11][12]. Other categories of scheduling algorithms with QoS guarantees have been identified but each has a trade-off. The performance of these QoS guaranteeing algorithms is summarized in table 1.

TABLE I
SUMMARY OF QOS GUARANTEEING SCHEDULING ALGORITHMS FOR VOQ STRATEGY.

ALGORITHM	TIME SLOT ASSIGNMENT	MAXIMAL MATCHING	STABLE MATCHING
Time complexity	$0(N^{2.5})$	$0(N^2)$	$0(N^2)$
Maximum throughput	100%	50%	50%
Differentiated service	Not supported	Not supported	supported
Best supported traffic	CBR	CBR	CBR and VBR

4. Analytical and Simulation Results

FIFO and PIM have been chosen for analysis in order to demonstrate the limitations of FIFO and improvement provided by PIM. These algorithms were implemented with the VOQ strategy. Simulation was carried out under a variety of input load or utilization (U) to get the average queuing latency (L) for different router sizes. Incoming packets were assumed to be an independent, identically distributed (i.i.d.) Bernoulli process with destinations uniformly distributed over all output ports.

Fig. 6 shows the FIFO Latency-Utilization (L-U) curve for routers with different number of ports (N).

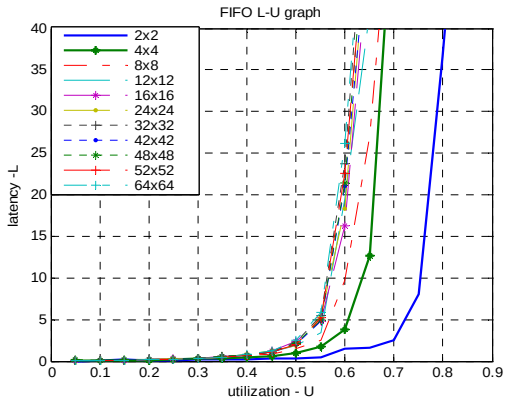

Fig. 6. FIFO Latency-Utilization Curve

TABLE 2
FIFO PERFORMANCE FOR DIFFERENT ROUTER SIZES

N	U
2	0.75
4	0.6533
8	0.5990
12	0.5858
16	0.5858
∞	0.5858

Table 2 summarizes the maximum input load (U) that can be handled by FIFO-enabled router before saturation. Fig.6 shows that for very large routers with N>8 the maximum throughput (U) that can be offered is approximately 58.58%. This limitation is as a result of the Head of Line (HOL) blocking effect. Therefore for routers with large N the performance of FIFO is limited to 58.58%. The simulation also shows that the 58.58% utilization is asymptotic with N, where small N gives asymptotes above 60%.

The L-U curve for the PIM algorithm with one iteration (PIM-1) in fig.7 also shows a poor utilization of the router as N increases. PIM-1 gives only a maximum utilization of 63% for routers with N>8. This is a slight increase to the 58.58% offered by FIFO. However, PIM with four iterations (PIM-4) gives a significant improvement by reducing latency and thereby increasing the utilization of the router.

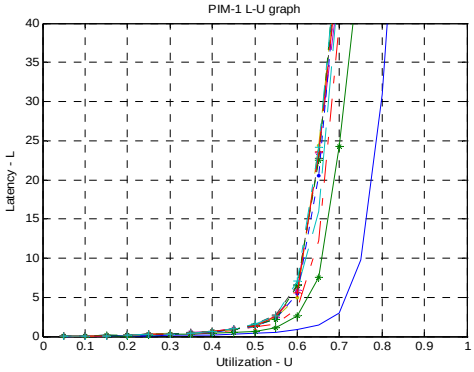

Fig. 7. PIM-1 Latency-Utilization Curve

PIM-4 remains stable with an offered input load in excess of 95% as shown in fig. 8.

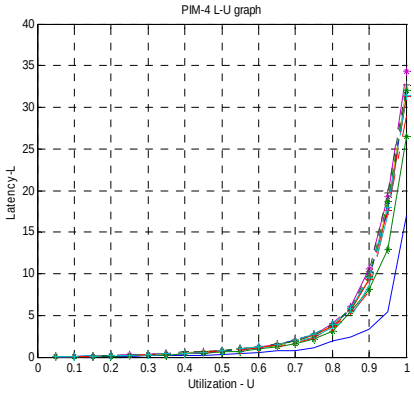

Fig. 8. PIM-4 Latency-Utilization Curve

5. Proposed Queuing Architecture and Scheduling Algorithm

In view of the limitations of the queuing strategies discussed in section 3, this research is implementing a multistage queuing and scheduling (MQAS) strategy in Fig. 9. MQAS is a two-stage queuing and scheduling architecture. It combines the performance of output queuing (OQ) with the scalability of the VOQ [13].
An Iterative Probabilistic Scheduling (IPS) is proposed for the VOQ stage while FIFO would be implemented at the OQ stage. IPS is proposed for the VOQ stage to improve on some of the shortcomings of the FIFO, PIM and the scheduling algorithms mentioned in table 1. The model of IPS is presented in section 5.2.

5.1. Multistage Queuing and Scheduling Architecture.

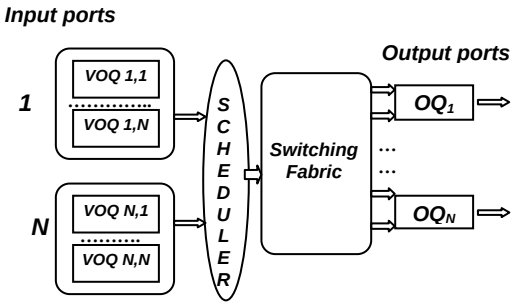

Fig. 9. Multistage Queuing and Scheduling Architecture (MQAS).

The VOQs in the first stage of this architecture maintains a separate queue at the input ports for each output port. These virtual output queues are implemented in the memory buffers at each input port. That means for

an N-output port router, each input port maintains N VOQs in its memory buffer, as indicated in Fig. 6. Therefore, for a router with N-input port and N-output port, the total number of virtual output queues to be maintained will be N^2. The total number of memory buffer implemented at each input is N.

As discussed earlier, achieving optimal throughput performance in VOQ depends on the scheduling algorithm implemented. In view of this, we propose an Iterative Probabilistic Scheduling (IPS) scheme for the VOQ.

The second stage of MQAS uses OQ strategy in which packets are queued on a FIFO basis. One queue is maintained per output port, and scheduler also uses FIFO to match packets to the output link. At the output, packets experience minimum delay because there is no HOL blocking so that the throughput is maximized. QoS is also guaranteed at these ports because contention is already resolved at the input ports.

5.2. The Iterative Probabilistic Scheduling (IPS).

IPS orders packets based on an estimation of the bandwidth required for transmission and of the waiting time. The weight of each packet is calculated using these two parameters which are retrieved during the scheduling process. IPS then determines the probability of transmitting each packet retrieved during a time slot. The packet with the highest probability is assigned the Highest Bandwidth Packet (HBWP) tag and transmitted to its corresponding output port. Only packets with the HBWP tag are served during a scheduling process [13] [14].

5.2.1. The Model

Consider a router with N input ports and N output ports implementing MQAS architecture. Assume an M/M/1 queue model: The first 'M' indicates that the arrival rate (λ) of packets is Poisson distributed, the second 'M' indicates an exponentially distributed service rate (μ), and '1' implies that the router is a single server system.

With MQAS architecture, it can be deduced that each input port has N virtual output queues (VOQ). Let 'i' represent an input port and 'j' an output port within the router. Therefore, $VOQ_{i,j}$ denotes a VOQ in input port 'i' queuing packets for output port 'j', and OQ_j represents an output queue in output port 'j'.

For all non-empty queue at the input the weight of a packet in $VOQ_{i,j}$ is given by:

$$WP_{VOQi,j} = eBW_{VOQi,j}*2 + eQ_{VOQi,j}*1, \qquad (1)$$

where $eBW_{VOQi,j}$ is the estimated transmission bandwidth given by the size of the packet. $eQ_{VOQi,j}$ is the estimated waiting time given by subtracting the current time from the last service time of the queue buffering the packet .

The probability of transmission of this packet is given by:

$$P_{voqi,j} = \frac{WP_{voqi,j}}{\sum_i WP_{voqi,j}} \qquad (2)$$

$0 \le P_{VOQi,j} \le 1$ for all time slots.

The summation $\sum WP_{VOQi,j}$ in equation 2 is the total $WP_{VOQi,j}$ of all packets already retrieved at that instant. For example, if two packets have been retrieved and a third is being retrieved, the $\sum WP_{VOQi,j}$ is for these three packets. The $WP_{VOQi,j}$ of other packets are not considered until their parameters are retrieved.

The Iterative Probabilistic Scheduling algorithm operates in three stages which are explained in the next subsections.

5.2.2. The Initialization Process of the IPS Algorithm.

1) At a time slot, input ports with packets in their $VOQ_{i,j}$ send requests for transmission ($REQ_{i,j}$) to corresponding OQ_j . If an input port has packets in all its VOQs it sends a request to each output port as shown in Fig. 10.

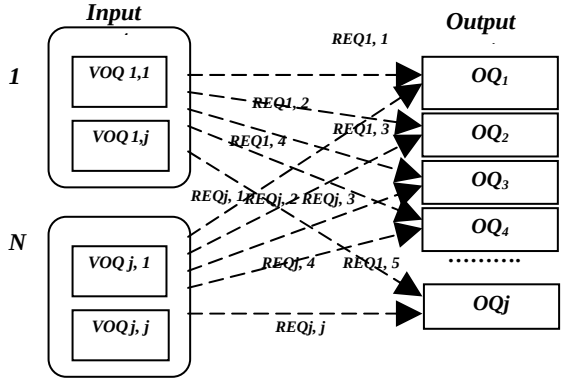

Fig. 10. Input Requests to Output.

2) IPS iterates only through the output queues with requests made to them and chooses an output queue (OQ_j) to serve a packet.

3) It forms a set $\{Z\} \forall$ i with $REQ_{i,j}$ to OQ_j such that for any j chosen at a time slot all elements in {Z} must be ≤N

4) IPS grants only the request of one input per time using the algorithm described in the next subsection.

5.2.3. Operation of the IPS Algorithm.

The flow chart in Fig.11 outlines the operation of the IPS algorithm. The calculation of PVOQi,j in the flow

chart is subject to the normalizing condition: $\sum_i P_{VOQi,j} = 1$

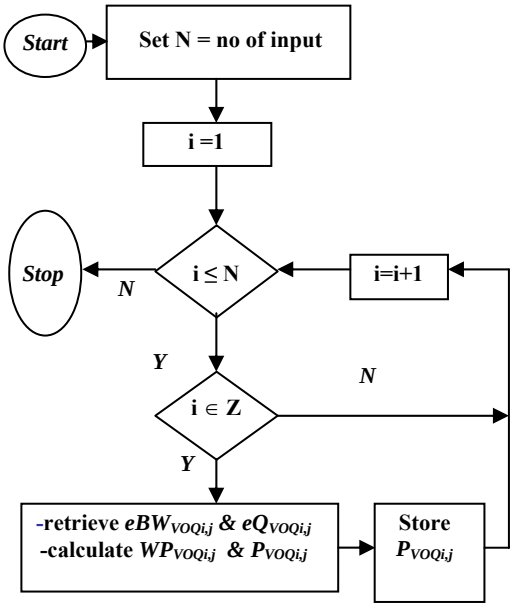

Fig. 11. Flow Chart of IPS algorithm.

5.2.4 Scheduling Process of the IPS Algorithm.

After retrieving the state of all packets contending for transmission, IPS performs the following process to schedule a packet to send to OQ_j

STEP 1: Pick the highest $P_{VOQi,j}$

STEP 2: Append the HBWP tag to the packet with this probability.

STEP 3: Transmit packet to OQ_j

At any scheduling time slot, only the packet appended with the HBWP tag is scheduled.

At the output port of the MQAS architecture, packets are mapped to the output link on the basis of FIFO. There is an increased throughput at the output ports because no packet major packet processing is carried out.

6. Conclusion.

From the analysis of the IPS algorithm model, IPS executes all operation within O(N) time complexity. Therefore it is fast to implement in a high-speed network. In addition, it is a simple algorithm which can provide priority and guaranteed service for VBR and CBR traffic. IPS also ensures fairness and prevents starvation of packets through the weight assigned to the parameters eBW and eQ. As a result, no packet is left indefinitely without being served. Also, IPS is a work conserving scheduler because it continues to serve packets as long as input ports keep sending requests to the output ports.

7. References

[1] D. E. Comer, *Network Systems Design using Network Processors IXP2XXX version*, 1st ed. (New Jersey: Pearson Prentice Hall, 2005).

[2] Dan Decasper, Guru Parulkar, Sumi Choi, John DeHart, Tilman Wolf, and Bernhard Plattner, A scalable, high performance active network node, *IEEE Network*, *31(1)*, January 1999, 8–19.

[3] S. Vegesna, *IP Quality of Service*, ISBN 1578701163, (Cisco Press, 2001).

[4] Dan Decasper, Zubin Dittia, Guru Parulkar, and Bernhard Plattner. Router Plugins – a modular and extensible software framework for modern high performance integrated services routers, *Proc. of ACM SIGCOMM 98*, Vancouver, BC, September 1998.

[5] J. Soldatos, E. K. Vayias, and G. Kormentzas. On the Building Blocks of Quality of Service in Heterogeneous IP Networks, *IEEE Communications Surveys & Tutorials*, 2005.

[6] G. Nong and M. Hamdi, "On the Provision of Quality-of-Service Guarantees for Input Queued Switches, *IEEE Commun. Mag.*, *38(12)*, Dec. 2000, 62–69.

[7] J.Crowcroft, "Scheduling and Queuing Management," www.cl.cam.ac.uk/Teaching/2003/DigiComm2/sched.pdf.

[8] A. Demers, S. Keshav, and S. Shenker, Analysis and Simulation of a Fair Queuing Algorithm, *Proc. ACM SIGCOMM '89*, 3–12.

[9] I. Stoica, S. Shenker, and H. Zhang. Core-Stateless Fair Queuing. *IEEE/ACM Trans. Net., 11(1)*, Feb. 2003, 33–46.

[10] I. Faynberg and H.-L. Lu. An Architectural Framework for Support of QoS in Packet Networks. *IEEE Commun. Mag., 41(6)*, June 2003, 98–105.

[11] N. McKeown and T. E. Anderson. A Quantitative Comparison of Iterative Scheduling Algorithms for Input-Queued Switches. *Comp. Networks and ISDN* Sys., *30* 1998, 2309–26.

[12] N. McKweon. Scheduling Algorithms for Input-Queued Cell Switches. *Ph.D. dissertation, Dept. Elect. and Comp. Eng.,* University of California at Berkeley, 1995.

[13] Oladayo Salami and H. Anthony Chan. Multi stage Queuing and Scheduling of IP traffic built on Network Processors for QoS Provisioning, *Proc.SATNAC2005*, Drakensberg, South Africa, Sept 2005, 64-65.

[14] O. Salami and H. A. Chan. Iterative Probabilistic Scheduling of IP traffic, *Proc. of 3rd IEEE CCNC'06*, Las Vegas, U.S.A., January 2006, 1336-1327.