Iterative Probabilistic Scheduling with Parallel Processing Process

[1]Avtar Singh, [2]Dr. Sandeep Kumar
[1]Singhania University Pacheri badi, Jhunjhunu, Rajasthan, India
[2]Shree Atam Vallabh Jain College, Ludhiana, Punjab, India

Abstract
Iterative Probabilistic Scheduling (IPS) is a scheduling algorithm that uses probabilistic approach to schedule traffic and to allocate recourses. It uses Virtual Output Queuing (VOQ) strategy and model is based on Poisson arrivals and M/M/1 queuing system. We proposed IPS with Parallel Processing (WPP) that adds another loop to normal IPS so that more than one packet can be scheduled simultaneously. The VOQs performs the calculation of probability and determines the HOQ packet with high probability which is the one to be schedule. Proposed IPS WPP is compared with normal IPS algorithm.

Keywords
IPSWPP, HoL, VOQ, IQ, MM

I. Introduction

IPS WPP follows the same procedure as IPS except for the fact that IPS WPP has the ability to schedule packets in parallel, which is what makes it more superior than IPS. IPS is a probabilistic scheduling algorithm which schedules traffic and allocates resources to traffic using a probabilistic approach. It is designed to work with the virtual output queuing strategy in which it schedules packets in a probabilistic manner and guarantees probabilistic quality of service to queued packets. IPS combines the properties of the Maximal Matching (MM) and Stable Matching (SM) algorithms. We are concerned with input-queued (IQ) switches. [1]

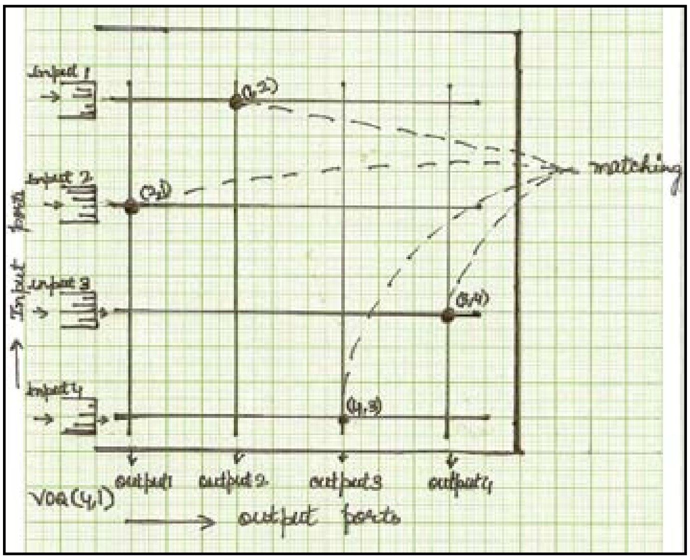

Fig. 1: An Input-Queued Switch and a Matching of Inputs & Outputs

Fig. 1 shows that a 4×4 IQ switch fabric, by which we mean the switch has 4 input ports and 4 output ports. (Not all ports need be used.) Packets arriving at input 'i' destined for output 'j' are stored in Virtual Output Queue VOQ (i,j). In each timeslot the switch can choose a matching from inputs to outputs [9]. Fig. 1.shows a matching how one packet is transmitted from input port 1 to output port 2 and so on. Since VOQ (4, 1) is empty no packet is transmitted. Input queuing has no scaling limitations but it exhibits a performance bottleneck known as Head-of-Line (HoL) blocking. To avoid the HoL blocking, VOQ in fig. 2.

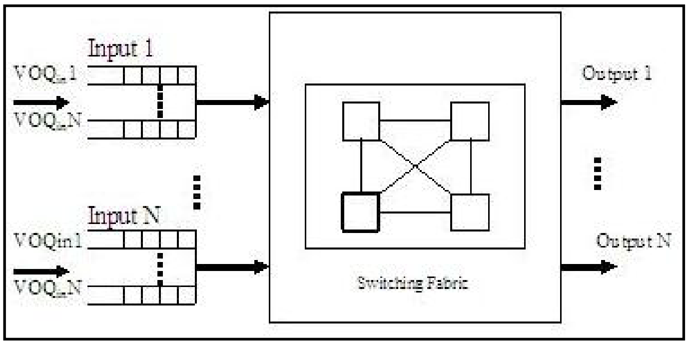

Fig. 2: Virtual Output Queuing

VOQ is an input queuing strategy in which each input port maintains a separate queue for each output port [7, 13].

II. IPS Model

A. Poisson Arrivals

Poisson arrivals occur such that:

- T = increment of time; no matter how large or small.
- t = interarrival time between events

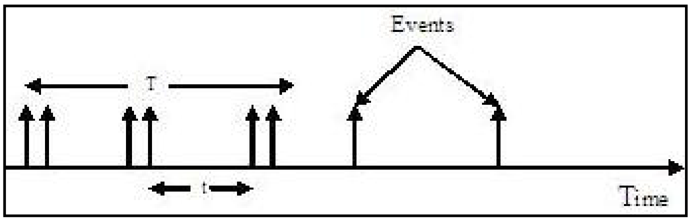

Fig. 3: IIlustration of an Arrival Process

As shown fig. 3, has a certain value called the interarrival time probability density. The following equation gives the resulting probability that the interarrival time't', is equal to some value 'x' when the average arrival rate is 'λ' events per seconds:

$$Pr(t=x)=\lambda e^{-\lambda x} \quad (1)$$

The famous Poisson distribution gives the probability that 'n' independent arrivals occur in (T) seconds: [5]

$$P_R(n,T) = ((\lambda T)^n/!n)^* e^{-\lambda T} \quad (2)$$

B. M/M/1 Queuing System

IPS schedules packets by using information from the queues and Head-of-Queue (HoQ) packets. The M/M/1 system has random length bursts with a negative exponential distribution (or memory less) burst arrivals. Fig. 4 shows that M/M/1 single server model:

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

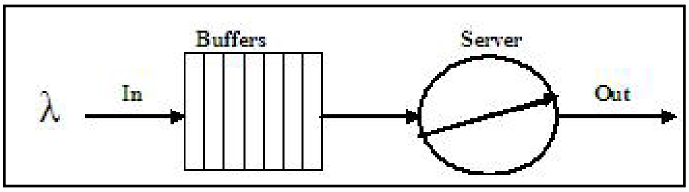

Fig. 4: M/M/1 Single Server Model

The M/M/1 case the units buffers are bursts of units represents by the fig. 5. The M/M/1 system do not model the switch buffer accurately since it is in units of bursts and not cell; however, the modeling of random burst lengths is more appropriate to many traffic sources such as FR.

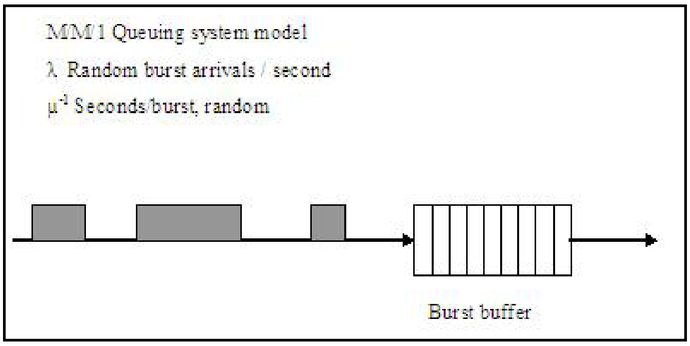

Fig. 5: Application of M/M/1 Queuing System With Cells

The probability that there are n bursts waiting in the M/M/1 queue is given by the following equation:

Prob[n burst in M/M/1 queue delay] = $p^n(1-p)$ (3)

When dealing with M/M/1 queues, as in `packet switching, there is an average of N users in the queue:

N = (p / 1 - p) (4)

Where p is the probability that the queue is not empty and (1 - p) is the probability that the queue is empty. The average queuing delays (i.e. waiting time), or w, in the M/M/1 system is given by the following equation: [5, 14]

w = Avg [M/M/1 queu delay] = ((p / u) /1-p) (5)

It calculates the weight of a packet $W_{VOQi,j}$ and queuing time $T_{VOQi,j}$ of every HoQ packet at each non-empty input queue, and uses this information to calculate the probability. $VOQ_{i,j}$ whereby 'i' represents an input port and 'j' represents an output port, this can be interpreted as, $VOQ_{i,j}$ is a VOQ in input port 'i' queuing packets for output port 'j'. OQ_j will then represent an output queue in port 'j'. The weight of a packet in (W) $VOQ_{i,j}$ is given by equation [1]

$W_{VOQi,j\,j} = 2* A_{VOQi,j} + T_{VOQi,j}$ (6)

Where $A_{VOQi,j}$ is the approximated transmission bandwidth given by the size of the packet and $T_{VOQi,j}$ is the approximated waiting time given by subtracting the time the queue buffering the packet was last served from the current time. Now that the weight of each packet has been calculated using bandwidth and waiting time, IPS then determines the probability $P_{VOQi,j}$ of transmitting each HoQ packet for that particular time slot. The probability is given by equation:

$P_{VOQi,j} = (W_{VOQi,j} / \sum W_{VOQi,j})$ (7)

$0 \leq P_{VOQi,j} \leq 1$ for all time slots. The sum $\sum W_{VOQi,j}$ in (7) above is the total of all HoQ packets retrieved at that time slot, i.e. if k packets have been retrieved and k+1 is being retrieved, the sum will be for those k+1 packets [1, 8].

III. IP Scheduling Process

1. Before IPS can calculate the probabilities of the HoQ packets in their $VOQ_{i,j}$ send request for transmission($REQ_{i,j}$) to corresponding OQi,j; this is shown in fig. 6.

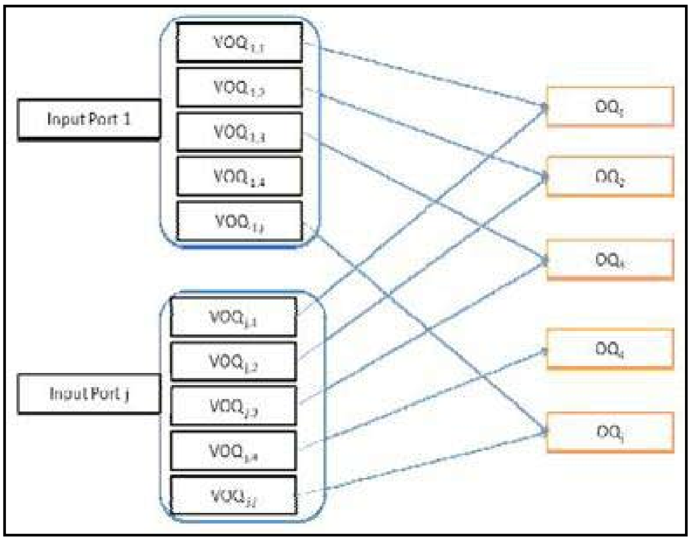

Fig. 6: Input Requests to Output Ports

Fig. 6, input port 1 has request in all of its $VOQ_{i,j}$ except for VOQ1,4 and input port 'j' has requests to all of its $VOQ_{i,j}$; this can be easily seen from arrows running from input ports to corresponding output ports [5-6].

When all requests have been made, IPS forms assert {Z} of inputs with $REQ_{i,j}$ t to $OQ_{i,j}$, this limits the size of set $Z \leq N$.

IPS then iterates through OQi,j with requests made to them and selects one of them(e.g. OQ_j) to serve [8, 11].

It goes through all $VOQ_{i,j}$ with request to OQ_j and perform the calculation of probability and determines the HoQ packet with high probability which is the one to be scheduled. This is illustrated clearly on the Algorithm 1 [8, 11].

Algorithm 1: Iterative Probabilistic scheduling

1. Read the value of Z;
2. Set N = Z, n = Size(Z);
3. Repeat for I = 1 to n ;
4. Pmax = 0;
5. If I <= n then

(i). Pick first element in the array Z;
(ii). Write $A_{VOQi,j}$;
(iii). Write $T_{VOQi,j}$;
(iv). Calculate $W_{VOQi,j} = 2*A_{VOQi,j} + T_{VOQi,j}$;
(v). Calculate $P_{VOQi,j} = (W_{VOQi,j} / \sum W_{VOQi,j})$
(vi). If $P_{VOQi,j}$ > Pmax then
(a). Set Pmax = $P_{VOQi,j}$;
(b). Set i = i + 1; and go to step 5;

6. Else go to step (b);
7. Else exit ();
8. Stop

IV. IPS With Parallel Processing (WPP)

The router consists of one dispatching processor (Pd), a few worker processors ($P_1 \ldots P_n$), and a transmitting processor (P_x). Fig. 7. Works, it will be simpler to explain the procedure by one packet throughout the whole process, which can be seen as undergoing 3 steps:

Step 1: P_d dispatches the packets to a worker processor;
Step 2: The worker Processors ($P_1 \ldots P_n$) processes the packet;
Step 3: The ($P_1 \ldots P_n$) sends the packet to the transmitting processor (P_n) [2-3].

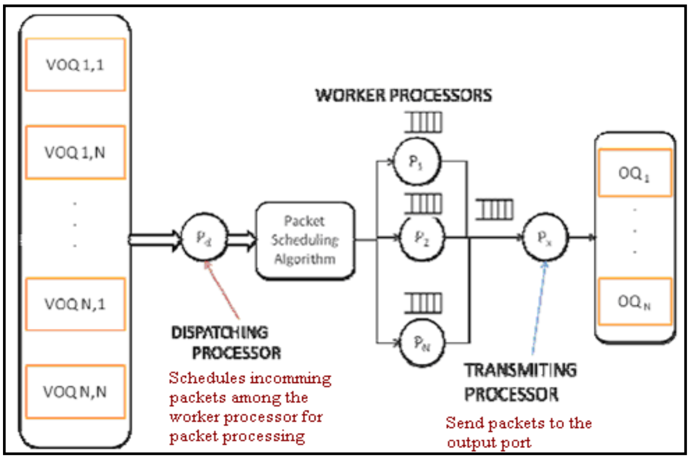

Fig. 7: Multiprocessing Packets Scheduling Architecture

Where there are 'n' worker processors, each worker processor Pi first receives some packets from the P_d (Step 1), then processes these packets (Step 2), and finally sends the packets to the P_x (Step 3) sequentially, again ensuring load balancing among the worker processors as shown in fig. 8:

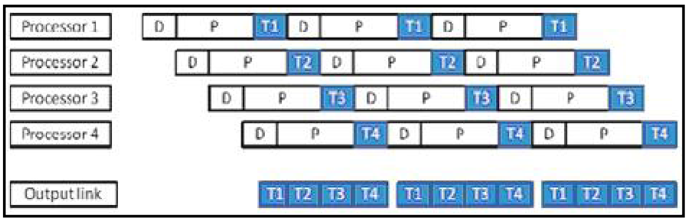

Fig. 8: Sequential Scheduling of Packets

- The number of packets to be dispatched to each Pi to produce the desired pattern should be calculated.
- There is a need to cater for the case where the packets are of variable length.
- A router processor configuration and any packet arrival rate (λ), there have to be a way to schedule packets in such sequential delivery pattern.
- Ensure fair scheduling among multiple flows having different reservations [2].

In general IPS WPP makes sure that during any iteration each output port receives a packet if there is a packet destined to it; hence packets equaling number of output ports are processed in parallel per iteration. Below is a relationship of number of processors (N_P) and number of packets (NPACKETS) that could be serviced in parallel:

- If N_P < NPACKETS, only NP requests can be granted thus allowing only NPACKETS to be processed in parallel.
- If N_P = NPACKETS, then all requests can be granted at a single time slot reaching maximal parallelization and full utilization of resources.
- If N_P > NPACKETS, again all requests can be granted but some of the processors will run idle hence under utilization of resources [10].

V. IPS WPP Scheduling Process

IPS WPP has the ability to schedule packets in parallel, which is what makes it more superior than IPS. The whole process of IPS WPP is summarized as shown Algorithm 2.

Algorithm 2: Iterative Probabilistic scheduling with parallel processing

1. Read the value of Z;
2. Set N = Z, n = Size (Z);
3. Repeat j = 1 to k;
4. If (j <= k) then
(i). i Set I = 1,Pmax = 0;
(ii). Write OQ;
(iii). If(i <= n) then
(a). Pick first element in the array Z;
(b). Write $A_{VOQi,j}$;
(c). Write $T_{VOQi,j}$;
(d). Calculate $W_{VOQi,j} = 2*A_{VOQi,j} + T_{VOQi,j}$;
(e). Calculate $P_{VOQi,j} = (W_{VOQi,j} / \sum W_{VOQi,j})$;
(f). If ($P_{VOQi,j}$ > Pmax) then
- Set Pmax = $P_{VOQi,j}$;
- Set i = i + 1;

else Go To Step (II);
else Service request with high Pmax;
else exit();
5. Stop

IPS WPP adds another loop on of normal IPS. The additional loop is not a problem since there are a few calculations being made per iteration hence the delay caused by calculations is insignificant, what is important is the number of packets that are being scheduled at the same time, which is equal to N_P. When the N_P = 1, then it is representation of IPS because now it means only one packet will be scheduled per iteration.

VI.TIME Complexity of IPS WPP

The efficiency of algorithm is measured by analyzing its execution time, i.e. the amount of time an algorithm takes to complete a task [8]. An algorithm can run slower on big tasks and faster on small tasks. Therefore the best way to compute the efficiency of an algorithm is to measure the number of times each instruction is going to be executed. This method is known as the "O notation" whereby the worst case execution time of an algorithm is analyzed as constant (O(k)) where k is a constant, linear (O(N)), Quadratic ($O(N^2)$) or even Polynomial ($O(N^n + N^n-1 + \ldots +1)$). To give time complexity of IPS WPP we first present its pseudo code: [1, 12]

```
1.  Insert  all inputs with REQ into set Z (array);
2.  Set n = number of objects in set Z;
3.  Set Pmax =  0;
4.  for ( j  = 1; j <=  worker_processors; j++) {
5.        for ( i = 1; i <= n; i ++) {
6.            get A and T;
7.            compute W and P;
8.            if ( P > Pmax)
9.             Pmax  = P;
            }
10.  Allocate packet with Pmax  to worker processor;
}[1]
```

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Table 1: Execution times of IPS WPP Algorithm

Instruction Number	Number of Times Executed
1	1
2	1
3	1
4	N_p
5	N
6	N
7	N
8	N
9	N
10	N_p

VII. Conclusion

High-speed routers must be robust and must have enough parallelism to support QoS, multicast etc. The algorithm has been described and compared with IPS for scheduling connections in an input-queued switch. IPS WPP differs from IPS, in that it schedules packets in parallel hence more packets are transferred to their respective output ports at a single iteration. A fully parallelized system IPS WPP run time grows quadratic ally as N increases.

References

[1] O. Salami,"Multi-stage Queuing with Iterative Probabilistic Scheduling of IP traffic forQoS Provisioning", Masters thesis, University of Cape Town, August 2006.

[2] J. Guo, J. Yao, L. Bhuyan,"An Efficient Packet Scheduling Algorithm in Network Processors", Department of Computer Science and Engineering, University of California.

[3] G. Welling, M. Ott, S. Mathur,"A cluster-based active router architecture", IEEE Micro, Vol. 21, No. 1, January/February 2001.

[4] H. Adiseshu, G. Parulkar, G. Varghese,"A reliable and scalable stripping protocol", SIGCOMM, pp. 131–141, 1996.

[5] Darren Spohn, Tina Brown, Scott Grau,"Data network Design", third edition published by Tata McGraw-Hill Companies, 2002.

[6] C.H. Papadimitriou, K.Steightz,"Combinational Optimization, Algorithm and Complexity", Prentice Hall, Englewood Cliffs,1982.

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

[8] O. Salami, H Anthony Chan,"A Scheduling Scheme to Improve QoS Provisioning for IP traffic", University of Cape Town.

[9] Mohsen Bayati, Balaji Prabhakar, Devavrat shah, Mayank Sharma,"Iterative Scheduling Algorithms", IEEE INFOCOM, 2007.

[10] N.McKweon.,"Scheduling Algorithms for Input-Queued Cell switches", Ph.D, dissertation, Dept. Elect. And Comp. Eng., University of California at Berkeley, 1995.

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

[12] I. Iliadis, W.E. Denzel,"Performance of packet switches with input and output queuing", IEEE ICC'90, Atlanta, GA, pp. 747-53, Apr. 1990.

[13] S.T. Chuang, A. Goel, N. McKeown, B. Prabhakar, "Matching output queueing with a combined input/output-queued switch", IEEE Journal on Selected Areas inCommunications, Vol. 17, No. 6, pp. 1030-39, Jun. 1999.

[14] L. Xiong, D. Platt,"Scheduling with Relaxed Constraint for Input-Queued Switch/Router with Crossbar Switching Fabric", DSCPC 2003/WITSP 2003, Gold Coast, Australia, Dec. 2003.

Avtar Singh received his Bachelor's Degree in Computer Science and Economics in 2003, Master's Degree in Computer Science in 2006, and Master of Technology Degree in Information Technology in 2008, all from Guru Nanak Dev University, Amritsar, India. He is pursuing a Ph.D. programme in Computer Science and Engineering as a research scholar from Singhania University, Distt. Jhunjhunu, Rajasthan, India. His research interests include "Iterative Probabilistic Scheduling Algorithms with Parallel Processing." At present, he is working as Assistant Professor in Computer Science at Guru Gobind Singh Khalsa College Sarhali, Distt. Tarn Taran, Punjab, India since 2009.