Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp377-383)

Modeling and Performance Evaluation of ATM Switches

KHALIL SHIHAB

Department of Computer Science, Sultan Qaboos University, P.O. Box 36, Al-Khod 123, Oman

Abstract: - In this work, we present a Colored Petri Net (CPN) model used for prototyping and modeling a complex switching scheme for ATM switches based on the combined I/O buffering technique. The new scheme is evaluated here within an overall setting that includes the traffic regulations enforced by the leaky bucket algorithm. It is based on splitting the traffic coming into input lines into two priority queues (High and Low) where traffic destined to busy ports is directed to the low priority queue. This scheme was compared against pure combined I/O buffering and simulation results showed that the new scheme pays back in decreasing transmission delay only when the traffic increases beyond a certain level. In addition the sensitivity analysis of the bucket size showed also that this latter affects the performance of the system only at high loads.

Keyword: Combined I/O Buffering, Priority Queues, ATM Switches, Leaky Bucket regulator

1. Introduction

ATM networks offer solutions for various traffic demands supporting a wide range of traffic types such as voice, real-time video, images and data. In most ATM networks, the traffic is regulated at the source using the leaky bucket algorithm. A leaky bucket regulator consists of a bucket (buffer) of a certain depth (size) leaking at a specified smooth constant rate. This is achieved by storing temporary bursts of incoming cells in the buffer. The buffer size defines the maximum burst that can be accommodated. If the buffer is full, the incoming cells are in violation and are therefore discarded.

There are two parameters associated with a leaky bucket regulator: the burst parameter and the leak rate parameter. The burst parameter, denoted by ß, is the size of the bucket. The leak rate parameter is denoted by ?. The number of cells that may be transmitted by a leaky bucket regulator in any interval of length I is bounded by $b + \lfloor r.I \rfloor$ [1].

ATM switches involve a number of input and output ports. Incoming cells on the input lines are switched to the appropriate output ports based on the addressing information embedded in the cells. A problem occurs when cells arriving at two or more input lines want to go to the same output port in the same cycle. Solving this problem is one of the key issues in the design of all ATM switches [2]. We can solve this problem using an input queue at each input port to store incoming cells, and in every cycle zero or one cell is taken from each input buffer and zero or one cell is sent to each output port. This solution is easy to implement and doesn't require any memory speedup over the line speed. But it suffers from the head of line blocking, which degrades the throughput to 60% or less [3]. The head of line blocking occurs when some packets left at the front of the input buffer prevent other packets further back in the buffer from getting a chance to go to their chosen output, even though there may be no contention for those output [4].

As a solution for the head of line blocking problem in input queuing switches, many researchers proposed that each input port maintains a separate queue for all cells destined to each output port (virtual output queuing), thus completely eliminating the head of line blocking [5], and [6]. In virtual output queuing zero or one cell is taken from one of the input queues at each input port and zero or one cell is delivered to each output port. This requires a memory that runs at the same speed as the line rate. A scheduling algorithm is used to determine which cell to select from the different queues at each input port. The scheduling algorithm chooses the best match between the input ports and the output ports in order to optimize a certain criterion. Some of the criteria used are: maximizing the switch throughput, minimizing the delay, or emulating an output buffered switch.

In [7] and [8] the authors addressed some of the issues in designing switches for very high-speed

networks. An algorithm that uses longest normalized queue first for scheduling input queuing switches to smooth the traffic shape in order to guarantee a faster delivery and a fair scheduling policy is presented in [9]. It has been proved in [10] that a speedup of 2 - 1/N is both necessary and sufficient for a combined input/output queuing buffer to emulate output queuing switch. In [11] the authors proposed a scheduling algorithm that can achieve the maximum efficiency at each switch and the scheduling algorithm is independent of the input traffic model.

Most of these algorithms either perform well under a uniform and independent input traffic but fails under a non-uniform or correlated traffic, or require complicated matching algorithms, such as bipartite graph matching that is very difficult to run in one cycle especially with very high-speed line rates and/or large N.

In [3], a new combined input/output buffered switch architecture is proposed that uses two priority queues at each input port and a simple scheduling algorithm that could be implemented in one cycle time in order to minimize the delay. In this scheme the queue of each input port is split into two queues, one is called high priority queue and the other is called a low priority queue. Each output port has only one queue (see Figure 1). In every cycle zero or one cell is taken from each input queue and zero or one cell is sent to each output queue. Both high priority and low priority input buffers and output buffers are assumed to be a simple First-In First-Out (FIFO) buffer.

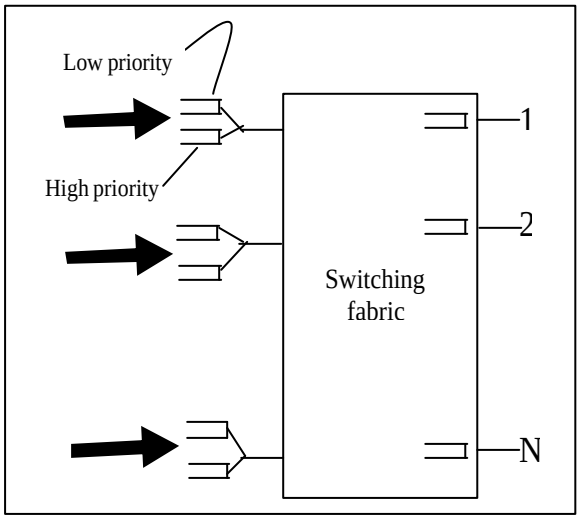

Fig. 1: Combined I/O Buffered Switch Architecture

It is also assumed that each output buffer is limited to a maximum number of cells (Max). If the output buffer is full, then all cells directed to this output port are blocked. In addition, a threshold value T is associated with each output queue. When a cell arrives in the input port, it is sent to the high priority queue. However, if the number of cells in the output queue reaches the threshold (T), then, newly arriving cells are sent to the low priority queue. Therefore, selecting cells to be sent to the output port will start from the high priority queues. Cells in low priority queue will not be sent out but only when all high priority queues are empty.

In this work we propose a Colored Petri Net (CPN) model to study the performance of the combined I/O switch architecture of Figure 1 under a leaky-bucket-regulated traffic. The CPN model is serving as a prototype of the proposed new combined I/O buffering scheme within a realistic setup. The setup comprises a set of nodes, each involving a set of processes engaged in communication through a central switch.

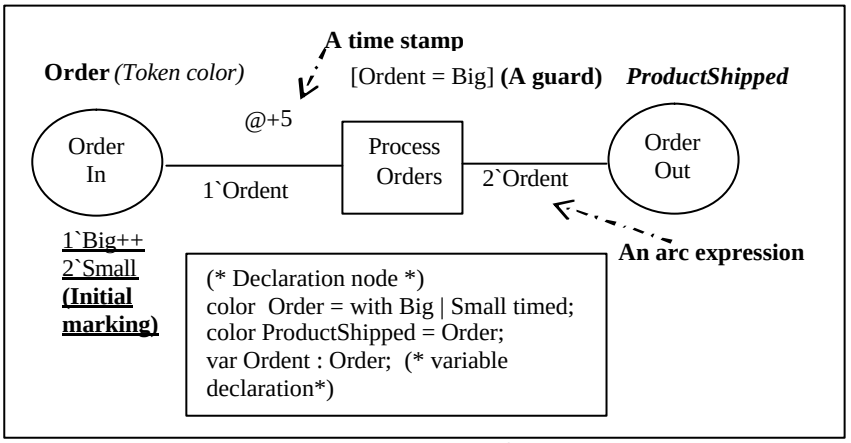

Fig. 2: Components of a CPN

The system is emulating an ATM LAN. The study focuses first on comparing priority based buffering and non-priority based one. Then, varying the bucket size

2. System Modeling

2.1 Colored Petri Nets

A Petri net is a network of interconnected locations and activities, with rules that determine when an activity can occur, and specify how its occurrence changes the states of the associated locations. Petri Nets can be used to model and simulate systems of any type. They are particularly useful in facilitating the design and analysis of complex distributed

systems that handle discrete flows of objects and information [12].

CPNs represent an extension of Petri Nets. They are graphical models that use the concept of colored tokens to represent data structures and state conditions. The presence of data or state conditions is marked by colored tokens in locations. The locations are represented graphically by ellipses called places. Each place is associated with a token color that specifies the type of data that may reside in the location. Activities are represented by rectangles called transitions, which govern the occurrence of events in the system. Places can be either input or output places for a transition. Places and transitions are linked through directed arcs modeling the flow of data. Each arc has an associated arc expression that controls the transition's occurrence. This expression specifies the number of tokens consumed by the transition, or the number of tokens that produce after its occurrence.

When the number of tokens in each input place of a transition satisfies the corresponding arc expression, then the transition is said to be enabled. An enabled transition can fire (i.e., occur) at any time. When it fires, it consumes as many tokens from its input places, as specified by their corresponding arc expressions, and produces as many tokens in its output places as specified by their corresponding arc expressions. Additional conditions for the enabling of the transition can be specified through the guard of the transition. All the Boolean conditions specified in the guard must evaluate to true for the transition to be enabled.

A declaration node is another component of a CPN that is used to record the token color, constants, variables, and function definitions. CPN modeling and simulation is supported by various simulation packages such as the Design/CPN tool [13] used in this study. In this tool, the different parts of a CPN model are constructed in different CPN pages. This helps making use of the CPN hierarchy constructs that enable the designer to break the complexity of the modeled system into different layers with different abstraction levels.

Figure 2 depicts a small CPN diagram used for processing shipping orders. The transition Process Orders has one input place, Order In, and one output place Order Out. The token color Order is associated with the place Order In, and the equivalent token color

ProductShipped is associated with the place Order Out. The token color Order is declared to hold Big and Small as data values. A variable Ordent is declared in the declaration node. The guard of the transition specifies that Ordent should be bound only to tokens having the value Big. In this state the transition is enabled. When it occurs it will consume the one token Big and produces two instances of Big into the output place.

Place fusion is one hierarchy construct that enables a place to be present physically on different CPN pages while representing a single conceptual place. The place Order Out might be declared as a fusion place that will be used to connect to another CPN page that completes the process of shipping with additional operation. Substitution transition (ST) is the second hierarchy construct that enables to hide lower level design details into the lower abstraction level. In the upper level, the ST behaves like a single transition, while in fact it represents a more complex activity hidden in the lower level. The transition Process Order can be designed to represent a ST that hides more elaborate details of product shipping that is hidden in the current level of the model.

Finally, the concept of time stamps that may be associated with timed tokens is used for the purpose of performance evaluation. Each timed token will bear an associated time stamp at each creation in accordance to a global clock maintained by the simulator. In figure 2, the transition Process Order has an associated timestamp (denoted by @+5) that specify that the occurrence of the transition takes 5 time units. The associated timestamps of the produced Big tokens will be augmented with 5 units. See [12, 13].

2.2 Overview of the Proposed CPN Model

Our CPN models an ATM LAN consisting of an ATM switch connecting a number of hosts running the combined input/output buffered switching algorithm described above. As depicted in figure 3, the CPN model comprises a set of nodes (*PC1* to *PC5*), each connected to an input line (*L1* to *L5*).

The traffic then goes through the ATM switch to any appropriate destination in the LAN. Each node generates traffic through five different processes, directed to specific destinations. Figure 4 depicts a view of the different processes generating traffic (modeled by the Substitution transitions Gen1 to Gen5). Figure 5 depicts one of such processes.

In Figure 5 an "on/off" model is used to capture the fact that sources alternate between an active (on) period (a generation state) during which packets are periodically emitted and a silence (off) period (a silence state), in which no packets are produced. In case there is only one source, this yield an Interrupted Poisson Process (IPP). When N similar

IPPs are multiplexed, one obtains a Markov Modulated Poisson Process (MMPP) with N states, where the state number indicates the number of active sources.

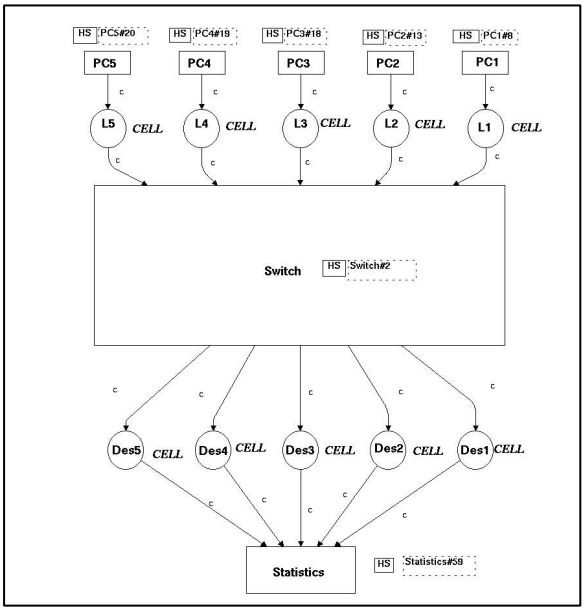

Fig. 3: CPN Top page modeling.

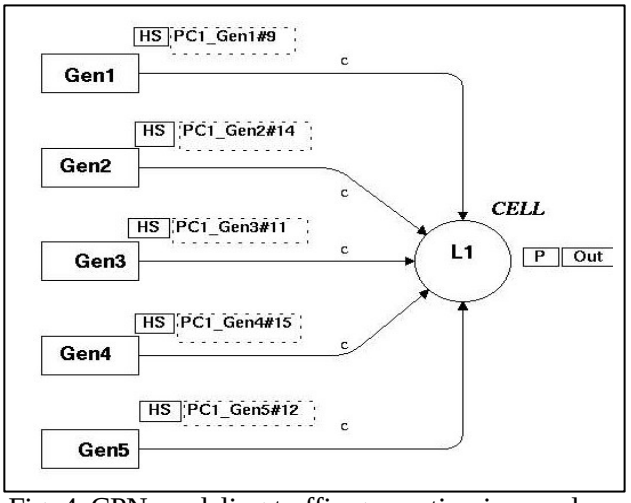

Fig. 4: CPN modeling traffic generation in a node

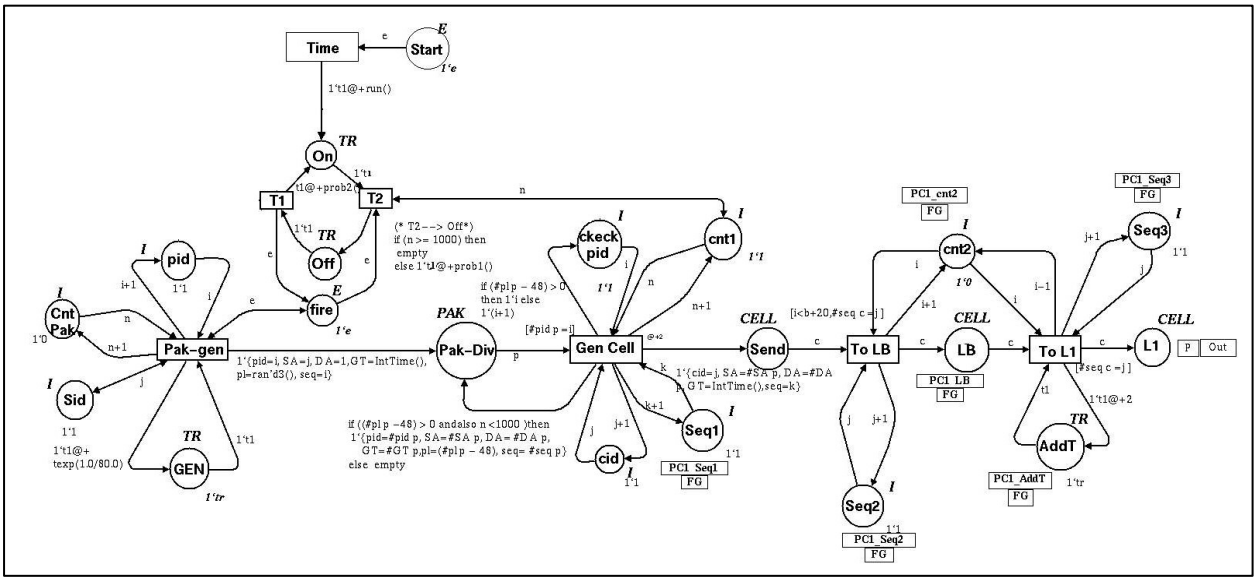

Fig. 5: A Traffic generation process with Leaky Bucket

Generation of traffic bursts of up to 64 Kbits is modeled by transition Pak-gen. Transitions T1 and T2 simulate the IPP traffic generation. When there is a token in the place fire, the transition Pak-gen fires and generates a packet. The firing of this transition is governed by the time stamp of the token on the place Gen. The time stamp associated with the token t is calculated using the function texp() that generates exponentially distributed variables with specified means. The generation of the next packet will not occur until the time of the global clock becomes greater or equal to the time stamp associated with the token t.

The IPP process is enforced in the same way, using the same exponential function. The transition Gen-Cell models the breaking of the bursts into streams of cells. The packets arrive at the Pak-Div place then go thru the process of splitting. In each cycle the size of the packet is reduced by 48 bytes, and a cell is generated in the place send. The transitions To-LB and To-L1 represent the leaky bucket regulator.

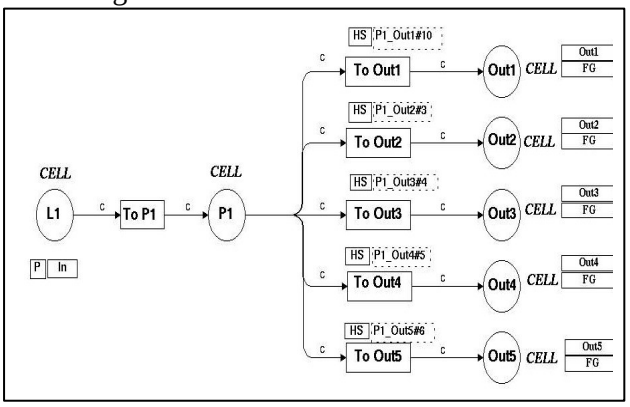

Fig. 6: CPN page describing Cell Switching to specific ports

The place LB models the Bucket. This place is a global fusion so all the traffic coming from different processes in the same node and will be multiplexed thru the same regulator. The transition To-LB will be blocked if the number of cells in the place LB is equal to the Bucket size b. The transition To-L1 generates cells at a periodic "leak rate" in the place L1 modeling the transmission line.

Figure 6 models the continuity of this process. Cells from place L1 are placed in the place P1. From P1 cells are sent to the appropriate port (Out1 to Out5) depending to their respective destinations by means of the substitution transitions To-Out1 thru To-Out5. The details of the switching enforced by any of these transitions are shown in figure 7.

Figure 7 models the combined I/O buffered switching algorithm. Cells from place P1 represent traffic coming from an input port of the switch. These cells are routed to either the high-priority queue modeled by the place HPQ, or to the low-priority queue modeled by LPQ. The transitions To-HPQ and To-LPQ enforce the checking of the threshold T value and of reaching the maximum occupancy of the buffer (specifically done by the transition To-LPQ). The place Out1-Q represents the queue at the port. The number of cells in this place is maintained in the place Check-t. Whenever any of the transitions send2 or send3 fires, a cell is put in the queue and the cell counter token in Check-t is incremented by one; and whenever the transition send4 fires, the cell-counter is decremented by one. Furthermore, the transition send2 has higher precedence than send3, since the latter cannot occur unless no cells are found in the HPQ place.

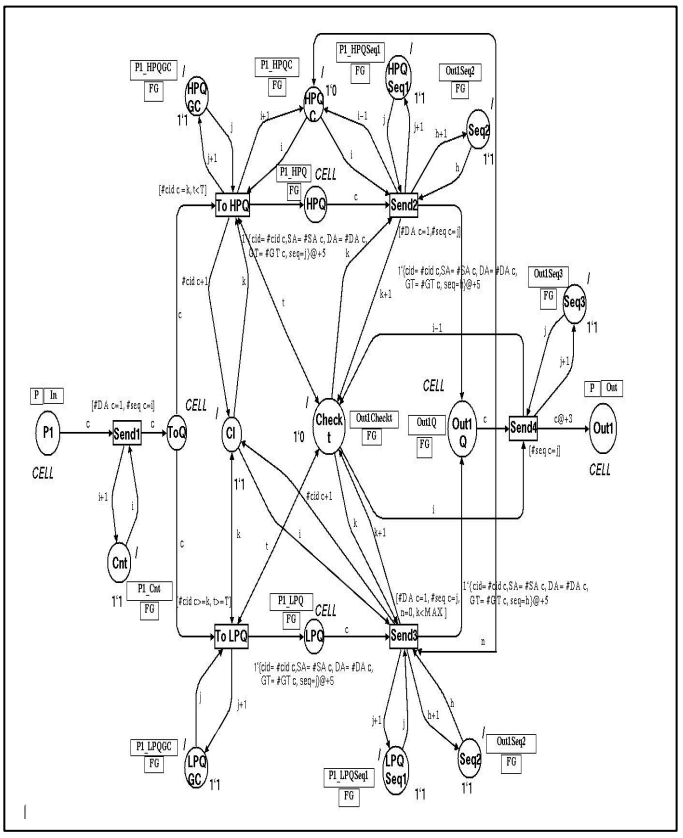

Fig.7: Cell Switching through the new Combined I/O Buffering scheme

This control is enforced using the content of the counter place HPQ-C that is used to maintain the number of High-priority cells present in the HPQ. The arc connecting the place HPQ-C to the transition send3 is used to enforce this control through the transition guard. The transition Send4

outputs cells from physical queue into the place Out1 modeling the specific destination.

Those cells are used later on to compute the delay incurred by the cell using the current value of the global clock minus the generation time that is recorded inside the token representing the cell. Other control places (e.g., CL, HPQ Seq1, LPQ Seq1, Seq2, etc.) are used to ensure the right sequencing and ordering of cells according to their arrival to match the FIFO requirement of ATM cell transmission.

Particularly the places Seq2 together with Seq3 are used to maintain the right sequence in sending a cell from HPQ or LPQ out to the output port. The place Out1-Q is declared as a global fusion place. It is used to accumulate cells from the various input lines generating traffic that is destined to go through the same output port.

3. Simulation and Results

The simulation is based on comparing the performance of the Cell Switching algorithm with priority queues with the case of pure Combined I/O buffering with no priority queuing enforced. The CPN model of the pure Combined I/O buffering is similar to the model described above with the omission of the splitting of the traffic into high and low priority queues depicted in figure 7.

Furthermore, we explore the sensitivity of the queuing scheme to the size of the leaky bucket.

Figure 8, plots the cell transmission delay as a function of the workload for both the Priority Based I/O queuing and Non-Priority based queuing. These results show that the effectiveness of the Priority Based I/O queuing starts when the load increases above a certain level. This is justified by the fact that the Priority Based I/O queuing imposes an overhead of traffic splitting (into high and low priority queues) that pays back only at high loads. This calls for exploiting this feature to implement an adaptive switching scheme that can use either of the two schemes (priority queue based or non-priority queue based schemes) depending on the load to which the switch is being subject to.

The plot of figure 9, describes the sensitivity of the priority queue based combined I/O switching scheme.

It can be noted from the plot that the effect of the bucket size on the average transmission delay incurred by cells in the switch for the priority queue-based combined I/O buffering is apparent only at high loads where increasing the bucket size starts paying off in terms of reduced transmission delays.

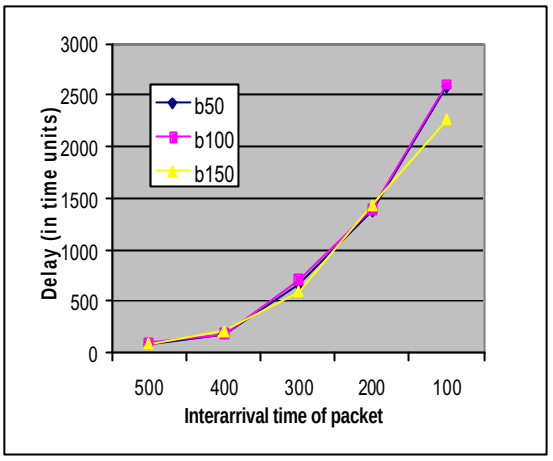

Fig. 8: Priority Vs. Non-Priority based switching

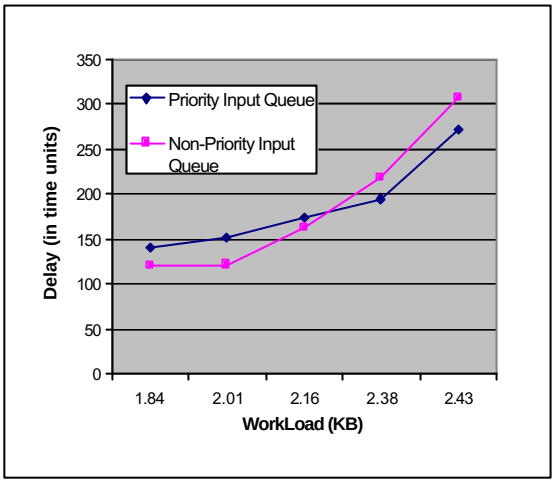

Fig. 9: Delay Vs. Workload for different bucket sizes

4. Conclusion

We have presented a Colored Petri Net (CPN) model for the Combined I/O Buffered Switching algorithm that is based on queue splitting into high and low priority queues to overcome the Head of Line blocking problem in ATM switches. The CPN was useful for capturing the essence of the Leaky Bucket cell generation at the hosts as well as the detail of the dynamic queue splitting process. Simulation results based on the CPN model have been derived. These results show that the priority based I/O queuing outperforms the non-priority based queuing beyond a minimum level of workload and that the size of leaky bucket affects performance only at very high loads.

References

[1] Raha Amitava, Kamat Sanjay, Jia Xiaohua and Zhao Wei, *"Using Traffic Regulation to*

Meet End-to-End Deadlines in ATM Networks", IEEE Transactions on Computers, vol. 48, No.9, pp. 917-935, Sept. 1999.

[2] Tanenbaum Andrew, *Computer Networks*, 3rd Ed, 1996, Prentice-Hall Inc.

[3] Gojko Babic, Raj Jain, Arjan Durresi, "ATM Performance Testing and QoS Management" in F. Golshani, Ed., "The IEC ATM Handbook" to be published by International Engineering Consortium, Chicago, IL, 1999

[4] Peterson Larry & Davie Bruce, *Computer Networks A Systems Approach*, 1996, Morgan Kaufmann Publisher Inc.

[5] M. Karol, and M. Hluchy "Improving the performance of input-queued ATM packet switches" INFOCOM 92 PP 110-115.

[6] T. Anderson, S. Owicki, J. Saxe, and C. Thaker "High speed switch scheduling for local area networks" ACM Transactions on Computer Systems, Nov. 1993 pp 319-352.

[7] G. Nong, and M. Hamdi "Burst-Level Scheduling Algorithms for Non-Blocking ATM Switches with Multiple Input Queues" IEEE Communication Letters Vol. 4, No. 6, June 2000.

[8] K. Choudhury, and E. L. Hahn, "A New Buffer Management Scheme for Hierarchical Shared Memory Switches" IEEE/ACM Transaction on Networking v 5, No. 6 pp 728-738 Oct. 98.

[9] S. Li and N. Ansari, "Scheduling Input-Queued ATM Switches with QoS Features," Seventh International Conference on Computer communications and Networks IC3N'98, pp 107-112 Oct. 12-14, Lafayette, LA. 1998.

[10] S.-T. Chuang, A. Goel, N. McKeown, and B. Prabhakar "Matching Output Queueing with a Combined Input Output Queued Switch" Computer Systems Technical Report CSL-TR-98-758 Stanford University, 1998.

[11] S. Chaudhry, and A. Choudhray "Time Dependent Priority Scheduling for Guaranteed QOS Systems" Proceedings of the 6th International Conference on Computer Communications and Networks, Las Vegas, NV Sept. 1997.

[12] Kurt Jensen, "*Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*", Vol.1 and Vol.2, Monographs in Theoretical Computer Science, Springer-Verlag, 1992, 1994.

[13] Kurt Jensen, S. Christensen, P. Huber, and M. Holla, "*Design/CPN: A reference anual*", C.S. Dept, University of Aarhust, Denmark, 1996.

[14] Durresi, A., V. Paruchuri, R. Kannan, S.S. Iyengar, "Optimized Broadcast Protocol for Sensor Networks," IEEE Transactions on Computers , Volume 54, Issue 8, August 2005, pp. 1013 - 1024