1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

BURST TRAFFIC SHAPING FOR INPUT-QUEUEING ATM SWITCH

*Jiunn-Jian Li,** *Jeen-Fong Lin,** *Tsai-Duan Lin***
jjli@cc.hwh.edu.tw jflin@vlsi.ee.hwh.edu.tw lintd@cc.hwh.edu.tw

* Department of Electronic Engineering
Hwa-Hsia College of Technology and Commerce, Taipei, Taiwan, R.O.C.

** Department of Management Information System
Hwa-Hsia College of Technology and Commerce, Taipei, Taiwan, R.O.C.

ABSTRACT

A new window scheme is proposed to improve the performance of nonblocking switches with input queueing under bursty traffic. This scheme shapes the incoming bursty traffic into random traffic virtually. It is shown that the proposed scheme alleviates the effect of burst traffic on input-queueing switches. Performance for the proposed scheme is also studied, together with the comparison with the performance of conventional window policy.

1. INTRODUCTION

Broadband Integrated Services Digital Network (B-ISDN) is intended to provide a wild variety of services. These services show very specific characteristics with respect to bit rates ranging from few Kb/s such as the telemeter to several Gb/s such as LAN interconnection, or concerning the required quality of service with information loss, delay, the variance of delay, and end-to-end synchronization, etc. Asynchronous Transfer Mode (ATM) is chosen as the final transfer mode for implementing the B-ISDN.

The heart of an ATM network is the switching fabric. It introduces many challenges that have become a major topic for researchers. Many different switching fabric designs have been proposed to satisfy the high-performance requirement for ATM. The user information is carried in cells and transferred asynchronously. Due to this statistical behavior and the fact that a theoretically unlimited number of virtual connections can shared the same link, it is mandatory for an ATM switch to synchronize to the incoming cell streams and examine the header of each cell to identify the virtual connection, translate into a new header for the next switching node, and derive routing and control information for the switch network. Furthermore, cell arrivals to the ATM switch are unscheduled. The consequence is that multiple cells will simultaneously compete for a common intermediate link or output port. This phenomenon is referred to as the cell contention. Queueing is required to resolve the cell contention that occurs when several cells are destined for the same resource, i.e., the intermediate link or the output port.

Input queueing places a separate buffer at each input port. A cell arriving at an input line enters the buffer. If it cannot proceed, it remains waiting in the buffer to be processed later. From the implementation point of view, classical input queueing dispenses with operating at some multiple of the trunk speed and consequently has the advantage of lower complexity and lower cost. Examples of input-queued switches are proposed in [1], [5], [6], [8], [9], [10].

Cell contentions are usually classified into two categories: internal contention and output contention. The former may occur at any intermediate link and the latter at any output port. Both kinds of contentions degrade the switch performance. In this paper, the ATM switches considered are internally nonblocking, namely, there are no internal contentions. To avoid output contention, a contention resolution mechanism is required to arbitrate arriving cells destined for the same output port. Two major solutions have been proposed: the three-phase algorithm [5] and the ring reservation algorithm [2, 7]. If the first-in-first-out (FIFO) queueing discipline is adopted, the queueing of the cell at the head of queue will prevent the subsequent cells from access the free output ports. This phenomenon is referred to as the head-of-the-line (HOL) blocking and reduces the switch throughput. Due to the HOL blocking, a large nonblocking switch with FIFO input queues has a throughput of about 0.586 compared to that of the ideal output-queued switch for random traffic.

Windowing is a famous technique to reduce the HOL blocking by allowing non-HOL cells to contend for the switch outputs [4]. In the beginning of a time slot, the input ports not selected to transmit their HOL cells have their second cells contending for the remaining output ports. The contending process repeats up to w times in each input port until a cell wins the contention, where w is referred to as the window size. The windowing scheme implies that each input queue must be a first-in-random-out (FIRO) buffer. A window size $w = 1$ corresponds to input queueing with FIFO buffers. As w increases, the throughput improves on a random traffic assumption [4]. However, the overhead, i.e., the duration spent on the arbitration phase is directly proportional to w. Alternatively, adopting windowing technique with window size w means operating w times in a time slot the three-phase or ring reservation algorithms. It is difficult to speedup w times of the switching internal rates.

A practical version of the windowing technique, called the time reservation algorithm, was proposed by Matsunaga *et al* in [8]. This algorithm features both advance time scheduling of the windowing technique and pipeline processing of the ring reservation algorithm.

The input-port expansion scheme can be regarded as a variation of the window scheme [4], in which each input port is expanded into r ports before cells enter an $rN \times N$ switch. Within one time slot up to r cells from each queue are 'windowed' and can be presented to the inputs for contention. Note that there is a slightly different between the original window policy and the input-port expansion scheme. The latter allows up to r cells to be swept from each input queue within one time slot and is expected to have a higher maximum throughput. As r increases, the throughput improves on a random traffic assumption.

The random traffic assumption provides tractable performance results but may not be realistic. An input line of the ATM switch, typically operating at a rate of about 100 Mb/s and carrying the cell stream formed by the multiplexing of cells from thousands of 10s kb/s sources such as voice services, is suitably represented by the random traffic model, in which the destintions of cells are uncorrelated and uniformly distributed among all output ports. This assumption may not be realistic and cannot represent many other services. For example, in high-speed data transfer, a file has to be splitted into cells and then transmitted one after another. Another example is the video services, which is allowed to use a large portion of the line capacity. The stream of cells presented to the input line of the switch has strong correlation.

A string of cells with the same destination is referred to as a *burst*. The approaches based on window policy improve the switch performance on a random traffic assumption. The improvement, however, diminishes quickly under bursty traffic. This is because the bursty traffic conditions make it likely that the first w (or r) cells in the input queue have the same destination.

2. PRELIMINARY

In general, previous contention resolution algorithms process only the HOL cell of each FIFO. The window policy [4] can be used to reduce the HOL blocking by allowing the non-HOL cells to contend for the outputs. Operation of such a policy is divided into arbitration phase and transmission phases. In the beginning of an arbitration phase, the input port not selected to transmit its HOL cell has its second cell contending for the remaining idle outputs. The contending process repeats up to w times until a cell wins the contention, where w is referred to as the window size. In the window policy, each input buffer must be a first-in random-out (FIRO) queue.

A window size $w = 1$ corresponds to input queueing with FIFO buffers. As w increases, the throughput performance improves on a random traffic assumption. However, the overhead, i.e., the duration spent on the arbitration phase is directly proportional to w. A practical version of the window policy was proposed by Matsunaga *et al* in [8]. The practical version is called the time reservation algorithm.

3. PROPOSED STRUCTURE

3.1 Basic Idea

If there is a method that could shape the bursty traffic into nearly uniform traffic, the previous schemes in Reference [4] could be applied. To help explanation, Figure 1 plots the input queue using the window policy, where the letter inside the cell represents the destined output port. In the switch that adopts the window policy or the input-port expansion scheme, only output ports 2 and 3 are probed in a time slot if the window size $w = 4$.

Imagine that a $Q_I \times w$ shaper whose function is to abstract the leading cells from each bursts in the input queue and then rearrange the cells of the first w leading cell of the bursts into an interleaving from, where and hereafter Q_I denotes the capacity of the

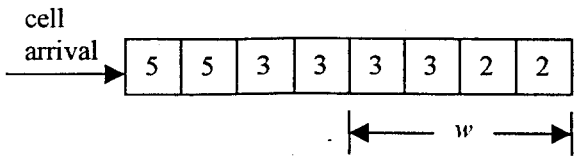

Figure 1. FIFO queueing (window size $d = 4$).

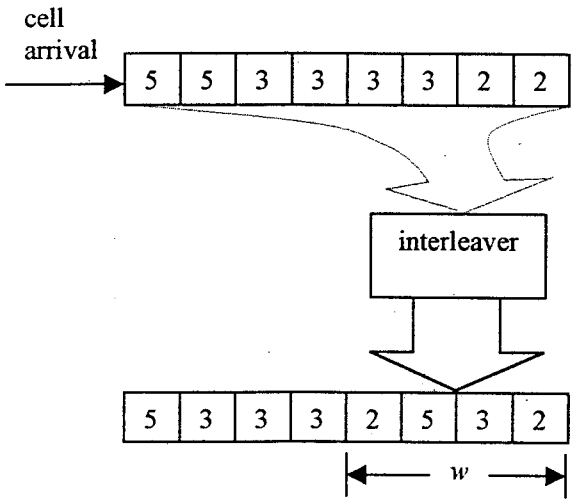

Figure 2. FIRO queueing with a interleaver (window size $w = 4$).

input queue. In the beginning of a time slot, the shaper examines the destination addresses of the queued cells, abstract the leading cells from the bursts, and put the abstracted cells to the front of the queue. This process is demostrated in Figure 2.
After leading cells of the bursts in the particular input queue are rearranged into an interleaving form, one would expect the switch to have a higher throughput because the output ports that can be probed are expanded. The shaping process, however, introduces extra hardware complexity because it must perform two tasks slot by slot: first to locate the delimiter between bursts and then rearrange the leading cells.

3.2 Structure of Input Queue

Each input port consists of one single FIRO queue and one queue control circuit. Suppose that the capacity of the FIRO is Q_I, measured in cells. The control circuit employs 3 types of registers served as pointers to regulate the FIRO. There are Q_I burst registers (*BR*s). The *i*-th *BR* corresponds to the *i*-th burst in the FIRO and is labeled by the notation BR_i. Each BR_i consists of 3 fields: one pointer ($BR_i \cdot ptr$) indicates the leading cell of the associated burst, one counter ($BR_i \cdot ctr$) indicates the burst length, and a field ($BR_i \cdot dest$) to record the destination address of the burst. The leading cell of the *i*-th burst is labeled by $[BR_i]$ in square

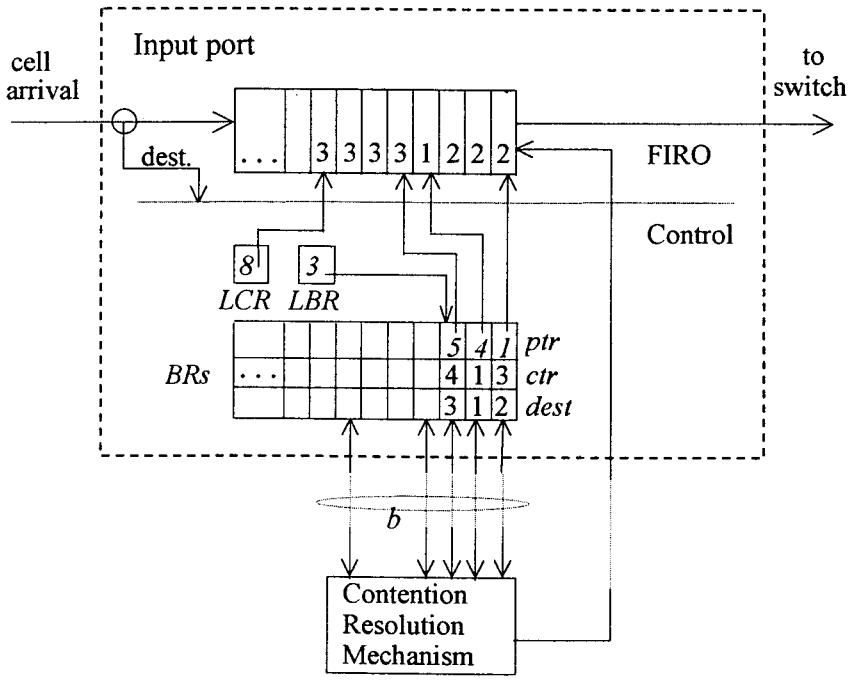

Figure 3. Basic structure of input port using the proposed shaping policy.

brackets. A last BR register (*LBR*) points to the last used BR. A last cell register (*LCR*) points to the last used room of the FIRO. Logically, *LCR* records the value of the total number of cells and ABR records the total number of bursts in the FIFO queue. The proposed structure is illustrated in Figure 3, where the FIRO holds three bursts (whose destinations are respectively 2, 1 and 3; and the individual burst lengths are respectively 3, 1, and 4).

The proposed structure is easily applied to the window-based contention-resolution algorithm. At the beginning of each time slot, assume the first w cells pointed by the *BR*s in one input queue successively contend for the access to the switch outputs. The cell pointed by the first *BR* contends first. Those input ports not selected to be transmitted then contend with the cells pointed by the second *BR* for access to any output ports that are not assigned to received cell in this time slot. This contention process repeats up to w times in each time slot to allow the leading cell of the first w bursts in a input queue contend for any remaining idle output ports, until the input selected to transmit a cell. Accordingly, the cells pointed at by the *BR*s form a 'window' in an input queue.

3.3 Control Operations

To keep the correct operation of the window-based algorithm, the contents of each *BR*, *LCR* and *LBR* has to be updated slot by slot as follows.

(i) At the beginning of the time slot: If a cell C with destination address D arrives at the input port:
  (a) If the FIRO is full ($LCR = Q_I$), the new arrival is dropped.
  (b) If the FIRO is not full:
    - If the new arrival is the leading cell of a new burst ($D \neq BR_{LBR} \cdot dest$), then the following operations are performed:

$$LBR{+}{+} \quad ^{*}$$
$$BR_{LBR} \cdot dest \leftarrow D$$
$$BR_{LBR} \cdot ctr \leftarrow 1$$
$$LCR{+}{+}$$
$$\text{FIRO}_{LCR} \leftarrow C$$

    - If the new arrival is the member of the *LBR-th* burst ($D = BR_{LBR} \cdot dest$), then the following operations are performed:

$$BR_{LBR} \cdot ctr{+}{+}$$
$$LCR{+}{+}$$
$$\text{FIRO}_{LCR} \leftarrow C$$

(ii) At the end of the time slot: If a cell, say $[BR_m]$, is selected by the window policy to be transmitted, then the cells which locate behind $[BR_m]$ are shifted forward, and the content of each BR_n, $n \geq m$, is updated as follows.
  (a) If the cell $[BR_m]$ is the last cell of the *m-th* burst ($BR_m \cdot ctr = 1$):

$$\forall m \leq n < LBR,\ BR_n \cdot ptr \leftarrow BR_{n+1} \cdot ptr$$
$$LBR{-}{-}$$
$$LCR{-}{-}$$

  (b) If cell $[BR_m]$ is not the last cell of the *m-th* burst ($BR_m.ctr > 1$):

$$BR_m \cdot ctr{-}{-}$$
$$\forall m < n \leq LBR,\ BR_n \cdot ptr{-}{-}$$
$$LCR{-}{-}$$

□ □ □

The proposed scheme is very similar in operation to the window policy. At the beginning of each time slot, if the cell $[BR_i]$, $1 \leq i < b$, is blocked, a chance is given to the cell $[BR_{i+1}]$, until either a cell is selected or the cell $[BR_b]$ is reached, whichever comes first.

4. PERFORMANCE EVALUATION

Performance of a nonblocking switch using the proposed scheme is analyzed by means of computer simulations. The simulations assume the uniform geometrically bursty traffic model described as follows. Each input port alternates between bursty and idle periods (see Figure 4). There is no correlation between different bursts and the distribution of each burst is uniformly distributed

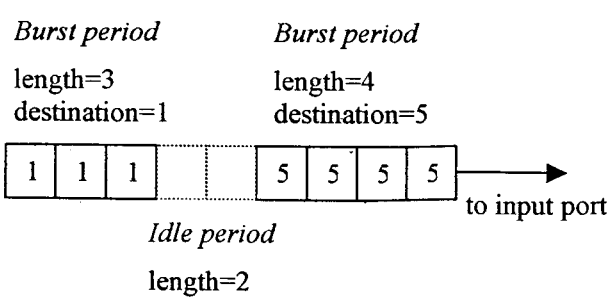

Figure 4. Bursty cell arrivals to an input port.

* Let p be a pointer. The notation "p++" increases p to point to the next element of whatever kind of object p points to. Similarly, the notation "p--" decreases p to point the previous element.

among the outputs. The probability that a burst lasts for i time slots is $P(i)=p(1-p)^{i-1}$, $i\geq 1$. The probability that an idle period lasts for j time slots is $Q(j)=q(1-q)^{j}$, $j\geq 0$. Therefore the mean burst length and idle period length are $L=\sum_{i=0}^{\infty} iP(i)=1/p$ and $Li=\sum_{j=0}^{\infty} jQ(j)=(1-q)/q$, respectively. Giving p and q, the offered load can be found by $\lambda=L/(L+Li)$.

Note that the uniform random traffic is a special case with $p=1$ and $q=\lambda$. That is, the burst length is deterministic and always lasts one cell long.

In addition, the case of full load $\lambda=1$, for which the gap between bursts is always zero, is considered here to obtained the maximum throughput. Other simulation conditions are: (1) Switch size N is set at 32 (The simulation results are not very sensitive to N for $N>16$), and (2) The capacity of each input queue is set at 256. The statistics of about 10^8 cells are collected over all input queues for each data point.

Throughput is the typical parameter used to describe the switch performance. It is defined as the average number of cells are transmitted in a output port in one time slot. Figure 5 plots the maximum throughput as a function of mean burst length L. The solid curves represent the cases where the proposed scheme is used and the dotted curves represent the cases where the conventional window policy [4] is used. As expected, the maximum throughput of proposed scheme is better than that of the conventional for any fixed mean bursty length L and window size b (or w). Even at $L=1$ (traffic is random), there is still a small gap in maximum throughput between these two schemes because the proposed one always searches the next cell with different destination. The advantage of the proposed scheme over the conventional increases as the traffic becomes more bursty.

The maximum throughput of the proposed scheme approaches an asymptotic value quickly as L increases. As shown, the different in maximum throughput between two extreme cases of $L=1$ (traffic is random) and $L=20$ (traffic is very bursty) is small. As the window size b increases, the effect of bursty traffic on input queueing is further alleviated as if the switch operated under random traffic.

The simulation results show that the proposed scheme possesses the following characteristics. It is independent of the mean bursty length virtually. It preserves the cell sequence. This will make easy output buffer construction. Two alternative cell scheduling algorithm are proposed in References [3] and [8]. As the window policy, these two algorithms

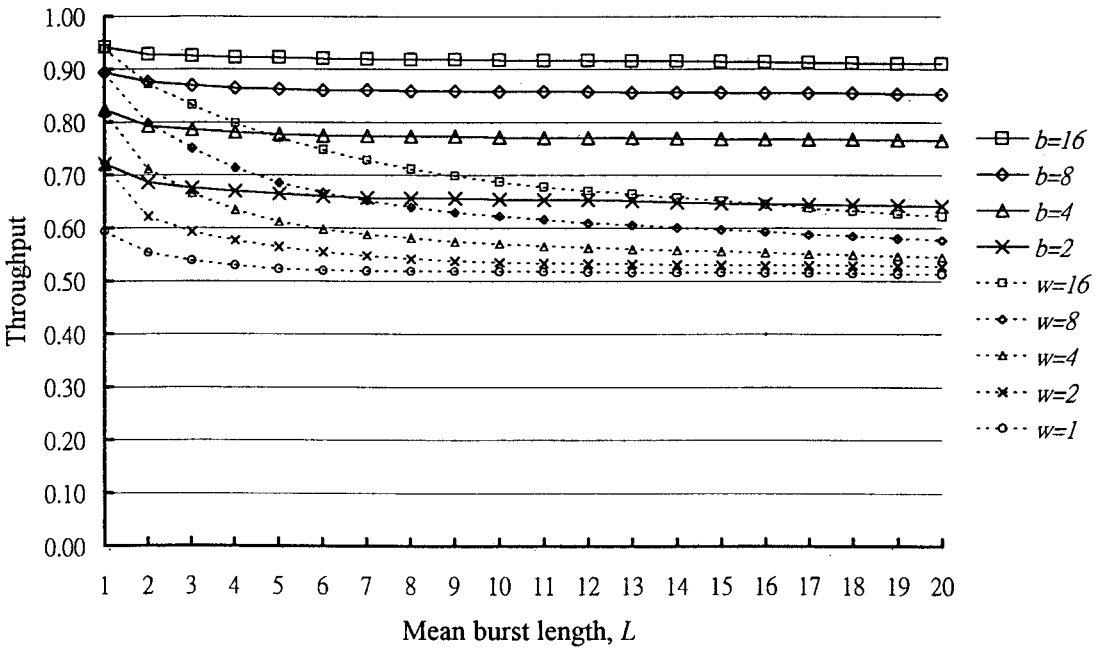

Figure 5. Maximum throughput against mean burst length (Solid curves represent the proposed scheme and dotted curves represent the conventional window policy [4]).

exclude the HOL effect and present excellent throughput under random traffic. The proposed scheme can also be applied to them to obtain better performance under bursty traffic.

5. CONCLUSION

This paper has proposed a new method to shape the burst traffic to virtual random traffic. We have studied performance for the proposed scheme. It is shown that an input-queueing switch using the proposed scheme can achieve better performance than the window policy. Moreover, the proposed scheme can be generally applied to ATM switches with input queues.

ACKNOWLEDGMENT

The authors wish to acknowledge Prof. Cheng-Ming Weng and Pi-E Lin for their help and encouragement.

REFERENCES

[1] N. Arakawa, A. Noiri, and H. Inoue, "ATM switch for Multimedia Switching System," ISS'90, pp. 9-14, 1990.

[2] B. Bingham and H. Bussey, "Reservation-Based Contention Resolution Mechanism for Batcher-Banyan Packet Switches," Electron. Lett., Vol. 24, No. 13, pp. 772-773, June 1988.

[3] W.-T. Chen, H.-J. Liu, and Y.-T. Tsay, "High-Throughput Cell Scheduling for Broadband Switching System," *IEEE J. Select. Areas Commun.*, Vol. 9, No. 9, pp. 1510-1523, Dec. 1991.

[4] M. Hluchyj and M. Karol, "Queueing in High-Performance Packet Switching," *IEEE J. Select. Areas Commun.*, 1988, Vol. 6, No. 9, pp. 1587-1596.

[5] J. Y. Hui and E. Arthurs, "A Broadband Packet Switch for Integrated Transport," IEEE J. Select. Areas Commun., Vol. 5, No. 8, pp. 1264-1273, Oct. 1987.

[6] C. T. Lea, "Design and Performance Evaluation of Unbuffered Self-Routing Networks for Wideband Packet Switching," *IEEE Trans. Commun.*, Vol.29, No. 7, pp.1075-1087, July 1991.

[7] T. T. Lee, "A Modular Architecture for Very Large Packet Switches," IEEE Trans. Commun., Vol. 38, No. 7, pp. 1097-1106, July 1990.

[8] H. Matsunaga and H. Uematsu,"A 1.5 Gb/s 8×8 Cross-Connec Switch Using a Time Reservation Algorithm," IEEE J. Select. Areas Commun., Vol. 9, No. 8, pp. 1308-1317, Oct. 1991.

[9] E. D. Re and R. Fantacci, "Efficient Fast Packet Switch Fabric with Shared Input Buffers," *IEE Proc. Pt.-I*, Vol. 140, No, 5, pp. 372-380, Oct. 1993.

[10] Q. Ta and J. S. Meditch, "A High Speed Integrated Services Switch Based on 4×4 Switching Elements," *INFOCOM'90*, pp. 1164-1171, 1990.