INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. **12**, 297–308 (1999)

Accurate modelling of the queueing behaviour of shared buffer ATM switches

Mahmoud Saleh[1] and Mohammed Atiquzzaman[2]*

[1] *School of Computer Science and Computer Engineering, LaTrobe University, Melbourne, 3083, Australia*
[2] *Department of Electrical and Computer Engineering, University of Dayton, Dayton, Ohio 45469-0226, U.S.A.*

SUMMARY

A model for the analysis of multistage switches based on shared buffer switching for Asynchronous Transfer Mode (ATM) networks is developed, and the results are compared with the simulation. Switches constructed from shared buffer switches do not suffer from the head of line blocking which is the common problem in simple input buffering. The analysis models the state of the entire switch and extends the model introduced by Turner to global flow control with backpressure mechanism. It is shown that buffer utilization is better and throughput improves significantly compared with the same switch using local flow control policy. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: ATM switches; performance modelling; simulation techniques, queuing techniques; analytical modelling

1. Introduction

In recent years, broadband ISDN (BISDN) has received increasing attention for its capability to provide a wide variety of services like video communication, graphic applications, and high speed data communications. One of the most promising approaches for BISDN is the Asynchronous Transfer Mode (ATM). An ATM network transfers all information in fixed length cells called *cells*, and is characterized by simplified protocols, high speed links, and high capacity switching nodes. The core of the switching fabric, referred to as *interconnection network* (IN), includes all the equipment required to route the cells through the switching fabric. Among the proposed architectures for ATM switches, multistage interconnection networks (MINs) have attracted a great deal of attention due to the features they offer, such as self-routing capability and suitability for VLSI implementation. The analytical modelling of unbuffered and buffered MINs have been widely studied in the literature.[1–5]

Jenq[2] developed a model for analysing Banyan networks consisting of 2×2 crossbar switching elements with a single buffer at each input of the switch. The traffic at the input to the switch is uniformly directed to all the outputs. Szymanski[9] extended Jenq's model to arbitrary switch sizes and buffer sizes. Turner[6] developed a similar model for switching networks with buffers shared between the inputs and outputs. His model assumes independence between buffer slots, i.e. the cells arriving in consecutive slots are statistically independent. The consecutive stages use flow

* Correspondence to: Mohammed Atiquzzaman, Department of Electrical and Computer Engineering, University of Dayton, Dayton, Ohio 45469-0226, U.S.A. E-mail: atiq@engr.udayton.edu.

CCC 1074–5351/99/040297–12$17.50
Copyright © 1999 John Wiley & Sons, Ltd.
Received 1 August 1998
Accepted 1 December 1998

control (backpressure) to avoid cell loss inside the network. Monterosso[10] developed a new method to evaluate the performance of a shared buffer ATM switch based on the exact model of the switching element, and without backpressure mechanism. This model, while accurate, is computationally intractable for networks constructed with large size switching elements. Bianchi[11] introduced an alternative approach to reduce the computation while maintaining the high accuracy. All of the models described in References 6, 10 and 11 use local flow control to forward a cell towards the output of the network.

In local flow control, a cell can be forwarded only if the next stage can accept a cell at the beginning of the cycle. In this paper, we extend Turner's model[6] for Delta-b networks[1] to enable it to analyse shared buffering with global flow control. In global flow control, the decision regarding whether a cell can be forwarded depends on whether the next stage can accept a cell after the stage has forwarded its cells in the current cycle. The simultaneous operations of forwarding and receiving cells at a buffer during a cycle are allowed. Global flow control provides *better buffer utilization*, and *improves the overall performance* of the network significantly.

The paper is organized as follows. In Section 2, we develop our model based on the assumptions we introduce in Section 2.1. Construction of the corresponding simulation network, and additional considerations are explained in Section 3. In Section 4, we examine our model with some numerical examples, and compare the results with the simulation and local flow control. Concluding remarks and further possible work are given in Section 5.

2. Analysis of the shared buffer ATM switch

A multistage ATM switch consists of a number of small crossbar switching elements (SE) interconnected by a permutation function. The switches can be broadly classified into two main categories, namely internally *blocking* and internally *non-blocking*. In an internally non-blocking switch two or more cells at different input ports can be simultaneously forwarded to two different output ports. A switch is called internally blocking if two or more cells with distinct output port destinations cannot always be transferred to the output ports due to routing conflict within the switch. For instance, resource contentions occur in switches when more than one cell access the same internal link. Buffers are used in the SEs to store the cells which lose the routing conflicts in an internally blocking switch. The cells are queued in the buffers for transmission during subsequent cycles. In a shared buffer switch, the buffers are shared between the inputs and outputs of the switching elements. An internal backpressure mechanism (called flow control) between the stages prevents cells from being dropped inside the switch.

In this section, we develop a model for shared buffer Delta network utilizing *global* flow control in contrast to *local* flow control developed in Reference 6. In global flow control, acceptance of a cell in the next stage's switch depends not only on the state of that switch, but also on whether some cells in that switch are forwarded to its successors during the same cycle. This allows more efficient buffer utilization, and considerably better performance as explained in Section 4. A recursive definition of Delta network with shared buffering is illustrated in Figure 1. It has N inputs and N outputs and consists of a number of stages consisting of $d \times d$ switching elements. Each switching element has B buffers shared between the inputs and outputs. A Delta permutation function is used to connect the adjacent stages. A switch of size N built from switching elements (SE) of size $d(D_{N,d})$ is formed by interconnecting N/d switching elements to d subnetworks of size $D_{N/d,d}$ each. The permutation function is such that the jth outlet of each SE is connected to the jth subnetwork.

Copyright © 1999 John Wiley & Sons, Ltd.

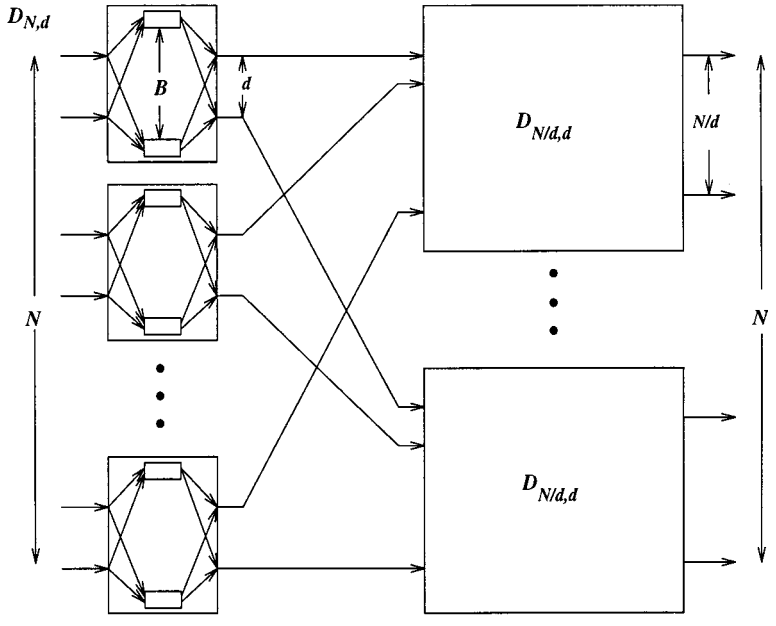

Figure 1. Recursive definition of a shared buffer Delta network of size $N \times N$ and using $d \times d$ switching elements, each having B buffers

2.1. Assumptions

We model each switch as a $B + 1$ state Markov chain, where B is the total amount of buffer space in a switch. The buffer space is shared between all the inputs and outputs of the switch. The following assumptions are made regarding the network, and its operation:

1. The network operates synchronously, i.e. the cells are submitted to the network at the beginning of the time slots (cycles). This reflects an ATM switch in which all cells have fixed lengths, and fit exactly into one time slot.
2. *Destination tag* is used to route a cell. The routing conflict inside the switch is resolved randomly, i.e. if two or more cells are destined to one output, one is chosen at random to be routed.
3. The state of the switching elements (SEs) at a particular stage in the switch are statistically indistinguishable, and the state of a stage is determined by the state of an SE in the stage.
4. The arrival of cells at each input of the switch is a Bernoulli process, i.e. the probability that a cell arrives during a time slot is constant, and the arrivals are independent of each other. Destination addresses are distributed uniformly.
5. A *backpressure* mechanism with global flow control ensures that no cell is lost inside the network. Thus, a cell leaves the switch if there is a space for it in the next stage's SE, or if the space becomes available during the same cycle. An acknowledgement policy is used to advise the receipt of a cell in the next stage's SE. Unacknowledged cells contend with other cells in the following cycles.
6. There is no blocking at an output link of the switch, i.e. the output can always accept a cell.

Copyright © 1999 John Wiley & Sons, Ltd.

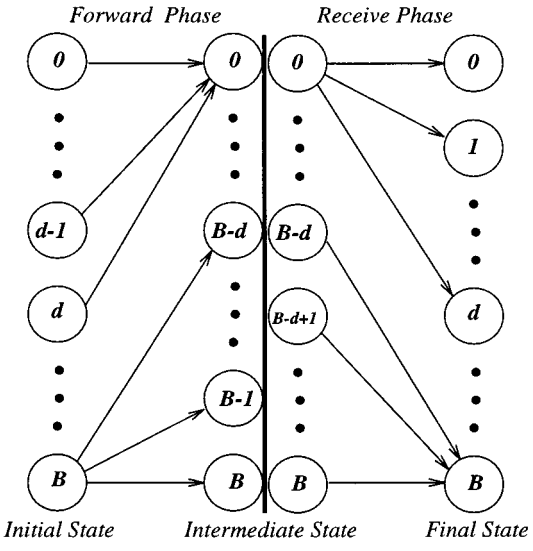

Figure 2. State diagram of a two phase switch operation

For analysis purposes, we assume that each cycle is divided in two phases as illustrated in Figure 2 which shows the state transitions of a buffer state between two cycles. During the *forward phase*, the cells destined to the output ports are forwarded to the SEs in the next stage, and the SE goes to an *intermediate* state. The forward phase is followed by the receive phase when it receives cells from SEs in the previous stage. During the *receive phase*, the available cells at the inputs of a switch are placed in the buffers, the corresponding acknowledgements are issued, and the SE goes to the *final state*. The above division of a cycle into forward and receive phases are required to model the enqueuing and dequeing of cells in a buffer during the same clock cycle. If the available space is less than the number of arriving cells, cells are selected randomly to determine the cell to be routed.

2.2. *Analysis of the ATM switch*

For the analysis, we introduce the following notation:

$\lambda_i(s1, s2)$: Probability that a switch in stage i contains $s2$ cells at the beginning of the next cycle given that it contained $s1$ cell at the beginning of the current cycle.

$\tau_i(s1, s3)$: Probability that a stage i switch contains $s3$ cells at the end of the forward phase given that it contained $s1$ cells at the beginning of the cycle, where $s1 \geqslant s3$.

$\sigma_i(s3, s2)$: Probability that a stage i switch contains $s2$ cells at the end of the current cycle given that it contained $s3$ cells at the beginning of the receive phase where $s3 \leqslant s2$.

$\theta_i(n1, n2)$: Probability that a stage i switch contains $n2$ cells at the beginning of the receive phase of the current cycle given that it contained $n1$ cell at the beginning of the receive phase of the previous cycle.

$\pi_i(s)$: Steady state probability that a stage i switch contains exactly s cells at the beginning of a cycle.

Copyright © 1999 John Wiley & Sons, Ltd.

$\tilde{\pi}_i(s)$: Steady state probability that a stage i switch contains exactly s cells at the beginning of the receive phase.

a_i: Probability that a cell is ready to enter a stage i buffer.

b_i: Probability that a successor of a stage i switch provides an acknowledgement to an output line given that a cell was submitted to the successor during the same cycle.

$Y_d(r, s)$: Probability that a switch that contains s cells, contains cells for exactly r distinct outputs of a switch.

SUCC_i: a successor switching element of stage i switching element. A switching element which receives cells from a switching element in the previous stage is called a successor switching element.

The objective is to calculate the steady state vector Π_i for every stage. Previous authors[6,11] have shown that the buffer states in the switching elements can be solved by Markov chain. The analysis presented below uses a similar approach and reasoning as used in Reference 6. The steady state vector Π_i is obtained by solving the matrix equation:

$$\Pi_i = \Pi_i \cdot \Lambda_i \tag{1}$$

where Λ_i is the transition matrix of a stage i.

$$\Pi_i = [\pi_i(s)], \quad s = 0, \ldots, B$$

$$\Lambda_i = [\lambda_i(s1, s2)], \quad s1 = 0, \ldots, B; \quad s2 = 0, \ldots, B$$

The state transition matrix elements are obtained by multiplying the state probabilities at the end of a cycle and after the forward phase after cells have been forwarded to the next stage.

$$\lambda_i(s1, s2) = \sum_{s3 = \max(0, s1 - d)}^{s1} \tau_i(s1, s3) \cdot \sigma_i(s3, s2) \tag{2}$$

where d is the number of inlets and outlets of a SE. The intermediate state probability $\tau_i(s1, s3)$ can be expressed as the destination probability of cells in a switch ($Y_d(r, s1)\binom{r}{s1 - s3}$) and the probability that the next stage can accept packets (b_i).

$$\tau_i(s1, s3) = \sum_{r = s1 - s3}^{\min(d, s1)} Y_d(r, s1) \binom{r}{s1 - s3} b_i^{(s1 - s3)} (1 - b_i)^{r - (s1 - s3)}. \tag{3}$$

b_i is a function of the number of cells in a buffer and whether the cells can be forwarded to the next stage, and can be expressed as follows[6] (see explanation below):

$$b_i = \sum_{h=d}^{B} \tilde{\pi}_{i+1}(B - h) + \sum_{h=1}^{d-1} \tilde{\pi}_{i+1}(B - h) \times \left[\sum_{r=0}^{h-1} \binom{d-1}{r} a_{i+1}^r (1 - a_{i+1})^{d-1-r} + \sum_{r=h}^{d-1} \binom{d-1}{r} a_{i+1}^r (1 - a_{i+1})^{d-1-r} \cdot \frac{h}{r+1} \right] \tag{4}$$

Equation (4) states that given that a cell was submitted to a SUCC_i through a particular output link, the link definitely receives an acknowledgment if SUCC_i has greater than or equal to d buffer spaces by the end of the forward phase or the total number of submitted cells to that SUCC_i is less

Copyright © 1999 John Wiley & Sons, Ltd.

than the number of available spaces by that time. Otherwise, the acknowledgment will be given depending on the number of submitted cells, and the intermediate state of the SUCC_i. The value of b_i for $i = k$ (where $k = \log_d N$ is the number of stages in the switch) is always equal to 1 due to the assumption that the output of the network can always accept a cell.

$$\sigma_i(s3, s2) = \begin{cases} \sum_{w=s2-s3}^{d} \binom{d}{w} a_i^w (1-a_i)^{d-w}, & s2 = B \\ \binom{d}{s2-s3} a_i^{(s2-s3)} (1-a_i)^{d-(s2-s3)}, & s2 < B \end{cases} \tag{5}$$

$Y_d(r, s)$ is recursively calculated using the following equation:[6]

$$Y_d(r, s) = \begin{cases} 1, & s = r = 0 \\ 0, & (s > 0 \wedge r = 0) \vee s < r \\ \dfrac{r}{d} Y_d(r, s-1) + \dfrac{d-(r-1)}{d} Y_d(r-1, s-1), & 0 < r \leqslant s \end{cases} \tag{6}$$

Equation (6) is independent of the stages, thus a table of required values can be created once and used for the rest of the calculations.

$$a_i = \begin{cases} \rho, & i = 1 \\ \sum_{j=0}^{B} \pi_{i-1}(j)[1-(1-1/d)^j], & \text{otherwise} \end{cases} \tag{7}$$

where ρ is the offered load to the network. The intermediate state vector is calculated similarly to the initial state vector. In particular,

$$\tilde{\Pi}_i = \tilde{\Pi}_i \cdot \Theta_i$$

where $\tilde{\Pi}_i = [\tilde{\pi}_i(s)]$, $s = 0, \ldots, B$.

$$\Theta_i = [\theta_i(n1, n2)], \quad n1 = 0, \ldots, B; \; n2 = 0, \ldots, B \tag{8}$$

$$\theta_i(n1, n2) = \sum_{k=0}^{B} \sigma_i(n1, k)\, \tau_i(k, n2) \tag{9}$$

To calculate Π_i and $\tilde{\Pi}_i$ we need equations (2)–(7). But equations (4) and (7) depend on the values of $\tilde{\Pi}_i$ and Π_i, respectively. Thus obtaining the steady state values requires an iterative computation. Though we are considering the steady state condition of the network, our experience shows that a fast convergence to the steady state values could be obtained by starting the calculation from the rest condition of the network. In particular, at the beginning, all of the stages can accept cells, and none have any cell to submit to their successor stages. Thus the value of b_i for all of the stages is initially 1, and the value of a_i for every stage, except the first is 0. Having these values we can calculate σ_i and τ_i for every stage, and calculate $\tilde{\Pi}_i$ and Π_i thereafter. Then the new values of $\tilde{\Pi}_i$ and Π_i are used to calculate the new values for a_i and b_i. The iteration continues until the

Copyright © 1999 John Wiley & Sons, Ltd.

steady state conditions are reached. After finding the steady state values, our merits of measurements could be obtained. The throughput of the SE is given by

$$\sum_{s1=0}^{B} \sum_{s3=0}^{s1} (s1 - s3)\,\tau_k(s1, s3)\,\pi_k(s1) \tag{10}$$

The average delay of stage i is given by Little's law. In particular the per stage delay is obtained from

$$\frac{\sum_{s=0}^{B} s\pi_i(s)}{\sum_{s3=0}^{B} \sum_{s2=s3}^{B} (s2 - s3)\,\sigma_i(s3, s2)\,\tilde{\pi}_i(s3)} \tag{11}$$

In equation (11), the numerator represents the equivalent average queue length in the stage i buffer, and the denominator is the average arrival rate of the stage i switch. Total delay is obtained by summing the per stage delays.

3. Validation of analytical model

We validate the model presented in Section 2 with the simulation study. The same assumptions as made for the analysis apply to the simulation network. Moreover, the following considerations are carried out too:

(a) At each cycle, a cell is generated with probability ρ (offered load to the network's input). The generated cell is independent of the cells generated at previous cycles and other input ports. Each cell consists of the following information:

   1. a source tag which represents the address at which it is generated,
   2. a destination tag, the address to which the cell is destined,
   3. the current network clock, used for measurement of the instantaneous delay.

(b) The results from the first 500 cycles of the network's operation are ignored to allow the network to reach a steady state condition. The network is then run for another 2000–5000 times until the average throughput stablizes with the accuracy of 10^{-6}.

(c) Conflict in the buffers for accessing a particular output as well as contention to seize a buffer space in the next stage is resolved using a random number generator with a different seed value from that of the cell generator.

The network operates as follows:

1. The cells at the last stage's buffers are sent to the output links of the network, and the instantaneous throughput and delay are measured for every link.
2. For each stage from stage $k - 1$ to 0:

   (i) The SE's buffer is examined for every output link of the SE, a copy of all cells interested in that output are placed in a list, and the list is sent to the corresponding input link of the next stage.

   (ii) If the number of available buffer spaces in the SUCC_i is less than the valid input lists, lists are selected at random.

Copyright © 1999 John Wiley & Sons, Ltd.

(iii) If the number of the elements in the selected list is more than one, one is selected randomly and is put in the new buffer, and the original cell is removed from its buffer.

3. A new set of cells is generated at stage 1 with probability ρ, and cells are placed in the first stage's buffer if there is any room. Unaccepted cells are discarded.

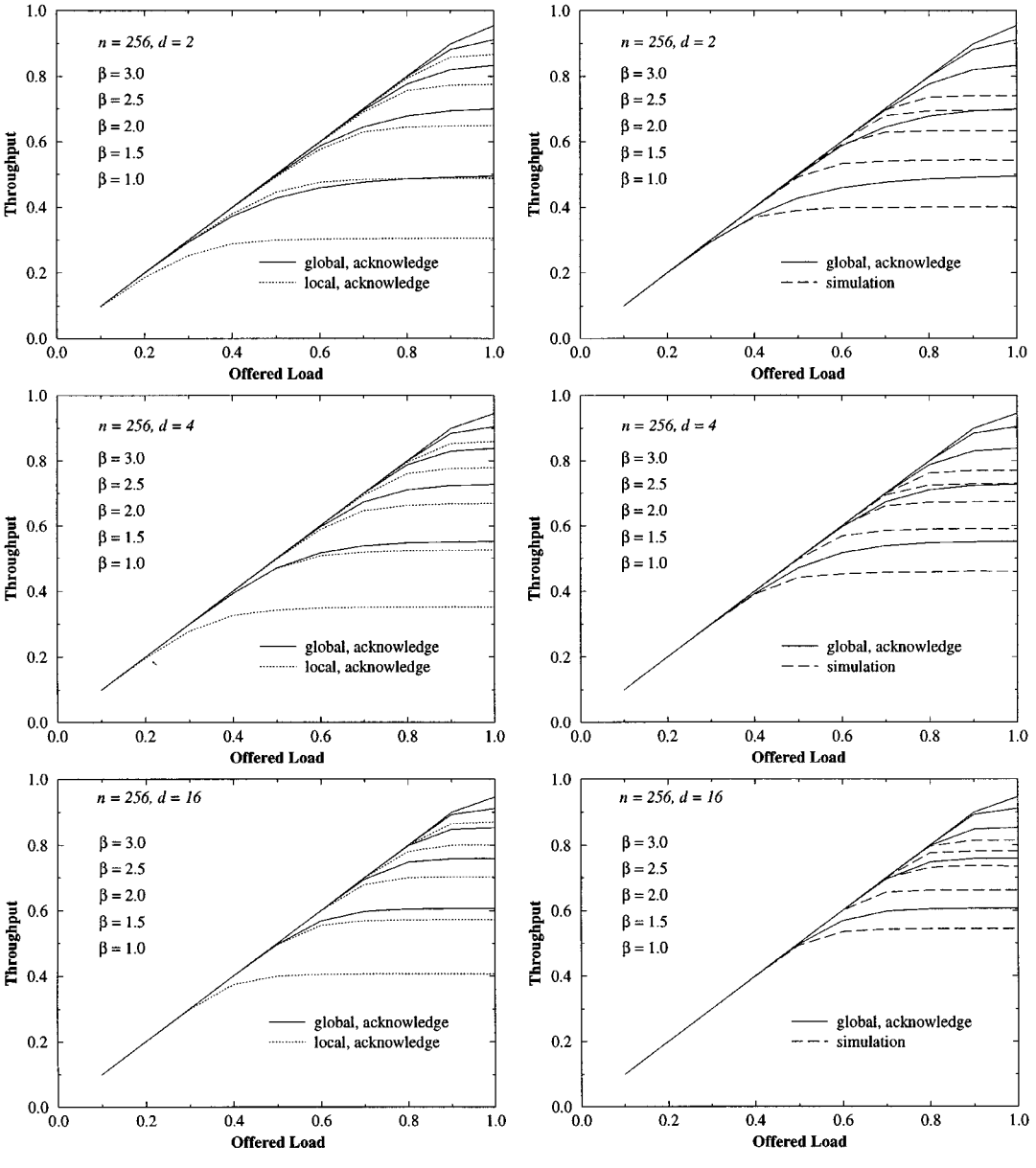

Figure 3. Comparison of throughput of the switch versus offered load ρ for global and local acknowledge, for $N = 256$

Copyright © 1999 John Wiley & Sons, Ltd.

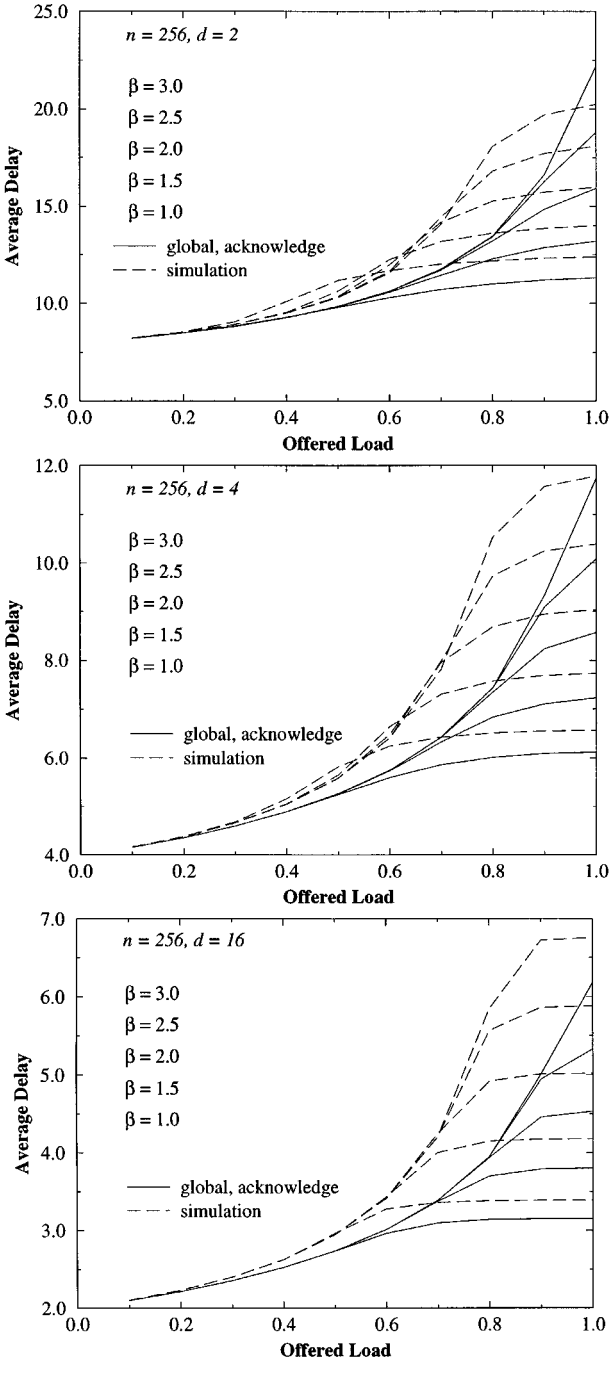

Figure 4. Delay versus offered load for global acknowledge method of flow control for $N = 256$

Copyright © 1999 John Wiley & Sons, Ltd.

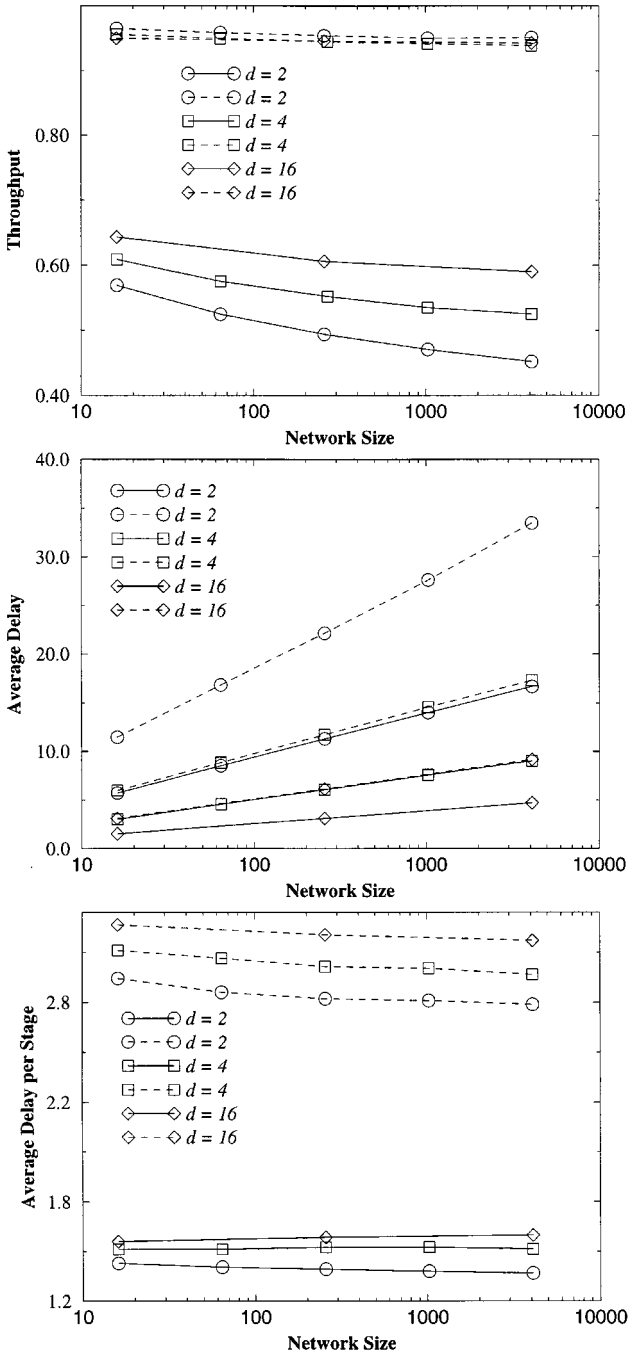

Figure 5. Throughout and delay versus network size for different values of β, and $\rho = 1$. Solid curves represent $\beta = 1$, and dashed curves represent $\beta = 3$

Copyright © 1999 John Wiley & Sons, Ltd.

4. Numerical results

Figures 3–5 compare the results obtained from our model with the simulation, and with the results obtained from the model described in Reference 6 for network size $N = 256$. In these figures, β is the buffer size per input/output link.

The analytical model is more accurate when the offered load to the network is low, and when a bigger buffer size is used. However, the model is optimistic under high input load due to the fact that the model assumes that there is no correlation between the cells in the buffers, whereas the correlation becomes significant under high offered loads. As the results in Figure 3 suggest, global flow control policy provides significantly higher throughput than local flow control, especially when β is small. This is because global flow control uses the buffer spaces more efficiently.

Figure 4 contrasts the delay curves of the model and the simulation. The results from the model are the best for small β value. In Figure 5, the throughput and delay versus network size for $\beta = 1$ and 3 for $\rho = 1$ are given. According to the results, under full input load, the per stage delay decreases as the buffer size increases, since the total number of buffer spaces increases inside the network. However, for a particular network size and β, the per stage delay grows when the switch size increases.

5. Conclusion

In this paper, we have developed a model for the analysis of multistage interconnection networks based on shared buffer switching elements for ATM networks. Our model employs global flow control and acknowledgement mechanism in contrast to local flow control described in Reference 6. In local flow control, a cell can be forwarded to the next stage depending on the state of the corresponding switch at the beginning of a cycle, whereas in global flow control, the simultaneous operation of forwarding and receiving the cells is allowed. This results in better buffer utilization, and higher performance. The new model is quite accurate under low-medium input load probability. However, its inaccuracy grows when the input load probability approaches 1, due to the output correlation of cells in a switch. An extension to this work could be to consider this correlation which leads to more accurate results. Other extensions could include the analysis of shared buffer switches under bursty traffic and other non-uniform traffic patterns.

References

1. J. H. Patel, 'Performance of processor-memory interconnections for multiprocessors', *IEEE Trans. Computers*, **C-30**(10), 771–780 (Oct. 1981).
2. Y.-C. Jenq, 'Performance analysis of a cell switch based on single-buffered Banyan network', *IEEE J. Selected Areas Commun.*, SAC-1(6), 1014–1021 (Dec. 1983).
3. H. Yoon, K. Y. Lee and M. T. Liu, 'Performance analysis of multibuffered cell-switching networks in multiprocessor systems', *IEEE Trans. Comput.*, **39**(3), 319–327 (Mar. 1990).
4. T. H. Theimer, E. P. Rathgeb and M. N. Huber, 'Performance analysis of buffered banyan networks', *IEEE Trans. Commun.*, **39**(2), 269–277 (Feb. 1991).
5. S. H. Hsiao and R. Y. Chen, 'Performance analysis of single-buffered multistage interconnection networks', *3rd IEEE Symp. on Parallel and Distributed Processing*, Dec. 1–5, 1991.
6. J. S. Turner, 'Queueing analysis of buffered switching network', *Proc. 13th Int. Teletraffic Congress*, Copenhagen, Denmark, June 1991, pp. 35–40.
7. M. Atiquzzaman and M. S. Akhtar, 'Performance modeling of switching fabric for ATM switching node', *Australian Broadband Switching and Services Symp.*, July 15–17, 1992, pp. 581–588.
8. M. Atiquzzaman and M. S. Akhtar, 'Performance of buffered multistage interconnection networks in non-uniform traffic environment', *7th Int. Parallel Processing Symp.*, Apr. 13–16, 1993.

Copyright © 1999 John Wiley & Sons, Ltd.

9. T. Szymanski and S. Shaikh, 'Markov chain analysis of cell-switched Banyans with arbitrary switch sizes, queue sizes, link multiplicities and speedups', *Proc. INFOCOM 89*, Apr. 1989, pp. 960–971.
10. A. Monterosso and A. Pattavina, 'Performance analysis of multistage interconnection networks with shared-buffered switching elements for ATM switches', *Proc. INFOCOM 92*, May 1992, pp. 124–131.
11. G. Bianchi and J. S. Turner, 'Improved queueing analysis of shared buffer switching networks', *Proc. INFOCOM 93*, 1993, pp. 1392–1399.

Authors' biographies:

Mahmoud Saleh received his PhD from the Department of Computer Science and Computer Engineering of La Trobe University, Australia. His research interests include B-ISDN and ATM networks.

Mohammed Atiquzzaman received the MSc and PhD degrees in Electrical Engineering and Electronics from the University of Manchester Institute of Science and Technology, England in 1984 and 1987, respectively. He had been academic staff member at La Trobe University and Monash University, Melbourne and King Fahd University of Petroleum and Minerals, Saudi Arabia. Currently he is faculty member in the department of Electrical and Computer Engineering at University of Dayton, Ohio. He is on the editorial board of *IEEE Communications Magazine*, *Computer Communications* journal and *Telecommunication Systems* journal. He has been the guest editor of special issues on *ATM Switching* and *ATM Networks* of the International Journal of Computer Systems Science and Engineering, special issue on *Projection-based Transforms* in the Image and Vision Computing journal, *Next Generation Internet* in the European Transactions on Telecommunications, three feature topics of IEEE Communications Magazine on *Traffic Management and Switching for Multimedia*, *Optical Networks*, *Communication Systems and Devices* and *IP Telephony*. He has also served in the technical program committee of many national and international conferences including IEEE INFOCOM and IEEE Annual Conference on Local Computer Networks. His current research interests are in Multimedia over Broadband ISDN and ATM networks, congestion control, ATM switching, multiprocessor systems, interconnection networks, parallel processing and image processing. He has over 90 refereed publications in the above areas.

Copyright © 1999 John Wiley & Sons, Ltd.