

STATE UNIVERSITY OF NEW YORK AT  
STONY BROOK

CEAS TECHNICAL REPORT 631

Performance Evaluation of a Starlite-lite  
Architecture for ATM Switching

E. Foo and T.G. Robertazzi

June 19, 1992

# Performance Evaluation of a Starlite-like Architecture for ATM Switching

Eugene Foo, Thomas Robertazzi  
Department of Electrical Engineering  
State University of New York at Stony Brook  
Stony Brook, N.Y. USA, 11794-2350

June 9, 1992

## **Abstract**

A Starlite-like switching architecture for ATM applications is examined in this paper. The switching architecture consists primarily of a Batcher sorting network and a Banyan switching network. A performance evaluation is done using both analysis and simulation. A discrete time queueing analysis is utilized due to the slotted nature of the system. Cells that are unable to make it through the system due to contention for resources are queued in a central buffer. There is one queue for each output line. The queues are implemented using a First In First Out (FIFO) queueing discipline. Only one packet from each queue is recirculated back to the system during each time slot. The average number of visits that a packet makes to the buffer and the mean queue length at various input utilizations are analyzed and compared to simulation values.

Keywords: ATM, Starlite, Performance Evaluation, Analysis, Switching

# 1 Introduction

Present telecommunications networks are characterized by their specialized natures and are geared towards a particular class of traffic. Future telecommunications networks would need to be more versatile and be able to support services that have diverse requirements. They would incorporate recent advances in technology like optical fibre transmission, optoelectronics and VLSI as well as an evolutionary restructuring of system concepts in software and firmware. One proposal for future advanced telecommunication networks is the Broadband Integrated Services Digital Network (BISDN). CCITT has recommended Asynchronous Transfer Mode (ATM) as part of BISDN. Transfer mode refers to the switching techniques and methodology such as multiplexing and switching. ATM can be simply described as a form of fast packet switching with minimal functionality. The main characteristics of ATM are the high speeds at which the packets are switched as well as small packet sizes ("cell" in ATM terminology). To achieve high speeds, there is limited functionality in the cell headers as well as no error correction capabilities on a link to link basis.

Currently there is a great deal of attention being focused on switching architectures that are designed with ATM features in mind. Switching is the transfer of cells from logical input ATM channels to logical output channels. It is advantageous to define a switching fabric that consists of identical basic switching building blocks that can be used in a modular design. This paper explores a switching architecture that exhibits some of the desirable properties of switching fabrics :

- low or zero cell blocking probability
- output logical channel contention resolution
- an efficient routing scheme for cells

The switching fabric consists of switching elements that are connected by a multi-stage interconnection network (MIN). Contention occurs when two or more cells attempt to use a single resource. The resource may be an output link or an internal link of the MIN. One solution to this problem is the use of buffers at switching elements and at the outputs/inputs. A cell can only advance to the next switching element if it is able to



accept a cell. The use of buffers at switching elements requires a backpressure mechanism to act as feedback. We explore an alternative scheme to this that avoids any contention for an internal link. This is done using a switching architecture that is internally non-blocking. The only contention then is at the output links. There are two main methods of implementing this. One method is to provide redundant paths [6]. Another method is to preprocess the cells so that any potentially conflicting cells will not be permitted into the switching fabric.

An example of the latter method is the Batcher-Banyan network called Starlite [4]. This switching architecture comprises of two networks. The Batcher network is a bitonic sorting network that sorts cells by their destination addresses [1]. The Banyan network is an interconnection network that provides a unique path from any one input to any output using a multi-stage network of switching elements. Section two describes the switching architecture of Batcher-Banyan based switching fabrics. The analytical model and performance analysis are described in section three followed by the numerical analysis in section four.

## 2 Batcher-Banyan switching architecture

Our primary aim in this section is to describe a Batcher-Banyan based switching fabric that utilizes a central buffering scheme to resolve output contention. In particular, we will focus on obtaining an analytical model of the switching fabric. Output contention occurs when two or more cells are attempting to go to the same output during the same time slot. We would like to allow one cell to go through and recirculate the other cells so that they can try to get through the fabric on a future slot. The class of Banyan networks we are actually using belong to a subclass of Banyan networks called Delta networks. Delta networks have a very useful property in that they are self-routing. The destination address can be used to determine the routing required by the switching elements to route cells to the correct output. Banyan networks are comprised of switching elements that can either perform the straight-through or exchange connection. A switching element can exist in two states as illustrated in figure one.

Let  $d_i$  be the  $i$ th digit of the destination address. Then at the  $i$ th switching element,

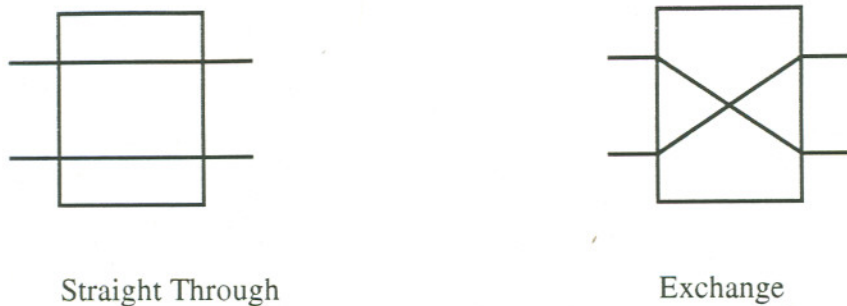


Figure 1: Allowable switch settings

the following rule applies:

$$d_i = \begin{cases} 0 & \text{take the upper path} \\ 1 & \text{take the lower path} \end{cases}$$

Any reference to a Banyan network should be taken to mean the particular sub-class of Delta networks. Banyan networks are intrinsically blocking networks. However when the input cells to it are

- compact ie. no inactive inputs in between the active inputs
- monotonic ie. the destination addresses of cells at consecutive inputs are in descending order

then the Banyan network will not be blocking as demonstrated in [7]. The two conditions are obtained by passing the cells through a Batcher network [1]. All input cells are then arranged in ascending order. Next they are concentrated in a compact arrangement. Cells that are duplicated are marked except for one cell. This cell is allowed to move to the Banyan network while the rest are removed and placed in the buffer. This switching methodology was first proposed in the Starlite architecture [4] which also incorporated multicast and broadcast capabilities. We will examine a similar architecture that excludes multicasting and broadcasting. The system has a Batcher Network, a purge network, a concentrator and a Banyan network. All cells that are unsuccessful in transerving the system are returned into a buffer. The buffer has  $N$  virtual queues where  $N$  is the number of external inputs as well as outputs.

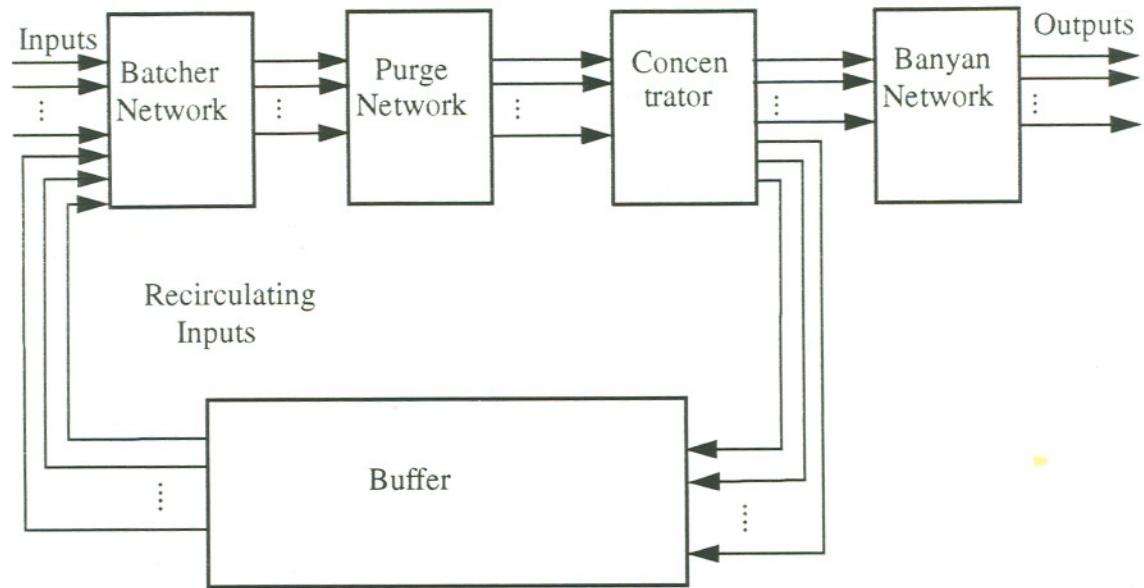


Figure 2: Block representation of the switching fabric

In practice, the memory management of the buffer would have an effect on its performance. However, we shall ignore such effects and assume the ideal case of no delay due to the memory management schemes.

## 2.1 Switching architecture description

Figure two shows a block diagram of the switching fabric. The sorter is the Batcher network and it receives its input cells from the external inputs as well as the recirculated cells from the buffer. It sorts the cells according to their destination addresses in ascending order and moves them to the purge network.

The purge network examines all the cells that it receives during a slot and picks out all those with the same output address. It picks one of these either at random or according to a predetermined criteria. An example of a criteria could be the oldest packet or a higher class priority. The selected cell is then marked so that it can be recognized by the concentrator. The concentrator will then arrange all the cells that have been marked in the ascending order of their output addresses. The remaining unsuccessful cells will then be moved into the recirculating buffer. Those with the same output address will go to the queue in the buffer that stores all the cells that are going to that output.

Each queue in the buffer is implemented with a First In First Out (FIFO) queuing



discipline. Only the cell at the head of the queue moves back to the Batcher network at the beginning of the slot, no matter how many cells there are in the queue. The reason for doing this is that there is no advantage in releasing more than one cell with the same output address back to the Batcher network since only one cell can get through per slot.

The cells received by the Banyan network will have been arranged in a compact form by the concentrator. All duplicated cells will have been sent to the buffer. Thus the two required conditions for non-blocking are satisfied so no cells will be blocked. Thus the Banyan network is able to successfully route all the cells it receives, though there may be some delay in doing so.

### 3 Performance Evaluation

The analytical model for the system is based on the fact that it can be modelled as a discrete time queueing system. Each discrete time interval is called a slot and cells are assumed to arrive at the beginning of each slot. It is assumed here for this study that cells can transverse the system in a single time slot. This assumption allows us to model the system using a 'black box approach'. Cells arriving at the external inputs can be assumed to follow independent and identical Bernoulli distributions. Hence in a single slot, the probability of a new cell arriving at any input is say  $p$ . Each cell has an equal chance of going to any of the  $N$  outputs. Therefore the probability of a cell going to a given output is  $1/N$  and successive cells are independent of each other also.

Due to the independence assumption, we can focus our attention on a particular output we shall call the tagged output. The results for the tagged output apply to any output. We shall consider a system that has  $N$  external outputs and  $N$  external inputs. The cells arriving at a particular output are from two possible sources.

- the external inputs
- the recirculating buffer

The probability of  $i$  cells arriving at the tagged output from the external inputs is

$$P(I = i) = \binom{N}{i} \left(\frac{p}{N}\right)^i \left(1 - \frac{p}{N}\right)^{N-i} \quad i = 0, 1, \dots, N; \quad (1)$$

where  $I$  is a random variable denoting the number of cells arriving at the tagged output from the external input.

For cells from the queue,

$$B = \begin{cases} 0 & q_{i-1} = 0 \\ 1 & q_{i-1} \geq 1 \end{cases} \quad (2)$$

where  $B$  is a random variable denoting the number of cells from the queue and  $q_{i-1}$  is the number of cells in the tagged queue at the beginning of the slot. Tagged queue here denotes the queue containing the cells that have the same output address as the tagged output. If the number of cells in the tagged queue at the end of the  $i$ th time slot is denoted by  $q_i$ , we then have

$$q_i = \max(0, q_{i-1} + a_i - 1) \quad (3)$$

where  $a_i$  is the number of cells arriving in the tagged queue during the  $i$ th slot.

The number of arrivals  $a_i$  has to be considered on the basis of whether the tagged queue was occupied or not during the previous slot. If the queue was occupied during the  $i-1$ th slot, then

$$a_i = a_{1i} = \binom{N}{i} \left(\frac{p}{N}\right)^i \left(1 - \frac{p}{N}\right)^{N-i} \quad i = 0, 1, \dots, N; \quad (4)$$

This is again a binomial distribution. In this instance, there are actually  $N+1$  inputs contributing to the number of cells going to the tagged output. Of these,  $N$  of them are the external inputs where the probability of cell generation is  $p$ . However the input from the tagged queue has a probability of cell generation that is 1 since the queue was previously occupied. Hence the fact that the queue has one cell guarantees that there is at least one cell going to the concentrator with the remainder going to the tagged queue during the next slot. Since we are concerned with only the number of cells going to the queue, there are only  $N$  inputs, ie. the external inputs.

If the queue is not occupied previously, then one has



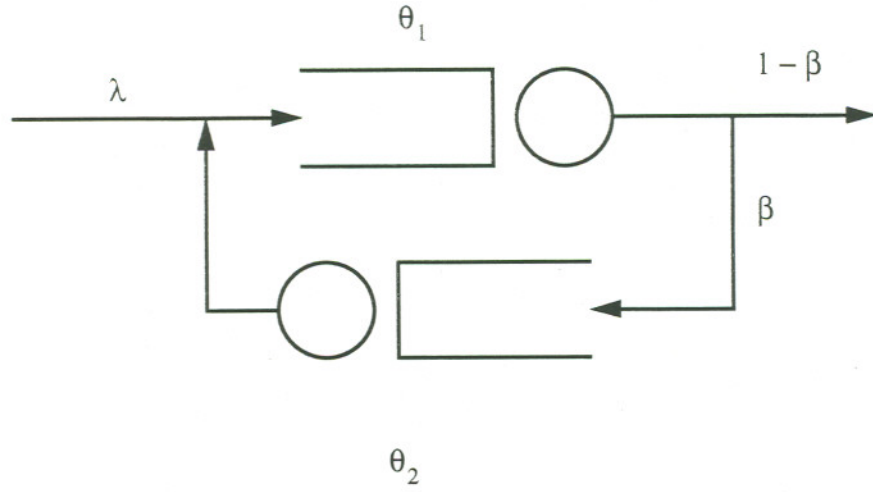


Figure 3: System queueing model

$$a_i = a_{0i} = \begin{cases} \binom{N}{1} \left(\frac{p}{N}\right) \left(1 - \frac{p}{N}\right)^{N-1} + \left(1 - \frac{p}{N}\right)^N & \text{if } i = 0 \\ \binom{N}{i+1} \left(\frac{p}{N}\right)^{i+1} \left(1 - \frac{p}{N}\right)^{N-(i+1)} & \text{if } i \geq 1 \end{cases} \quad (5)$$

There are two possible cases when  $i = 0$ . One case is when there is only one cell from all the external inputs. Since there is only one cell, it experiences no contention and therefore it exits to the concentrator with no cell going to the tagged queue. The other case is that no cells are generated from the external inputs at all. If  $i \geq 1$ , the probability is modified by adding one to the index  $i$ , since one cell will exit through the concentrator. The state transition diagram for the queue is shown in figure four.

Now that we have established the basic system, let us consider what we wish to obtain from the model. We can model the system as a queueing system as shown in figure three. Here  $\beta$  represents the proportion of cells entering the buffer. The remainder go on to the concentrator. The top queue represents the sorter and the purge network. The bottom queue is the buffer containing all the virtual queues. Let  $\lambda$  be the aggregate arrival from all the  $N$  inputs,  $\theta_1$  be the throughput of the upper queue and  $\theta_2$  be the throughput of the bottom queue. Then,

$$\begin{aligned} \theta_1 &= \lambda + \theta_2 \\ &= \lambda + \beta\theta_1 \end{aligned}$$

$$= \frac{\lambda}{1 - \beta} \quad (6)$$

The expected number of visits to the buffer is

$$E[\text{number visits to the buffer}] = \frac{\beta}{1 - \beta} \quad (7)$$

Let us consider an arbitrarily chosen cell that we tag and consider its progress in the system to obtain an expression for  $1 - \beta$ . The term  $1 - \beta$  can be written as

$$\begin{aligned} 1 - \beta &= \text{Prob}[\text{tagged cell gets through}] \\ &= \sum_m \text{Prob}[m \text{ cells forwarded through the concentrator}] \times \\ &\quad \text{Prob}[\text{tagged cell chosen from } m \text{ cells}] \end{aligned} \quad (8)$$

The term  $\text{Prob}[\text{tagged cell chosen from } m \text{ cells}]$  depends on the type of strategy used in selecting the cell. If we use a random strategy, then one simply has

$$\text{Prob}[\text{tagged cell is chosen from } m \text{ cells}] = \frac{1}{m} \quad (9)$$

The contribution to the  $m$  cells attempting to go through the concentrator comes from the external inputs and the recirculating inputs. Thus the probability becomes

$$\begin{aligned} \text{Prob}[m \text{ cells forwarded}] &= \sum_{m=1}^N \text{Prob}[m_i = m - 1] \text{Prob}[Q \geq 1] + \\ &\quad \sum_{m=1}^N \text{Prob}[m_i = m] \text{Prob}[Q = 0] \end{aligned} \quad (10)$$

where  $m_i$  is the number of cells from the external inputs and  $Q$  is the number of cells in the queue. Local balance equations can be obtained from the state transition diagram illustrated in figure four. These can be used to solve recursively for the state probabilities. It is possible in theory for  $n$  to range to infinity. Practically speaking,  $N$  will be finite. The computation of  $q_n$  can stop once it reaches an infinitesimally small value.

$$q_1 \triangleq \frac{1 - a_{00}}{a_{10}} q_0$$

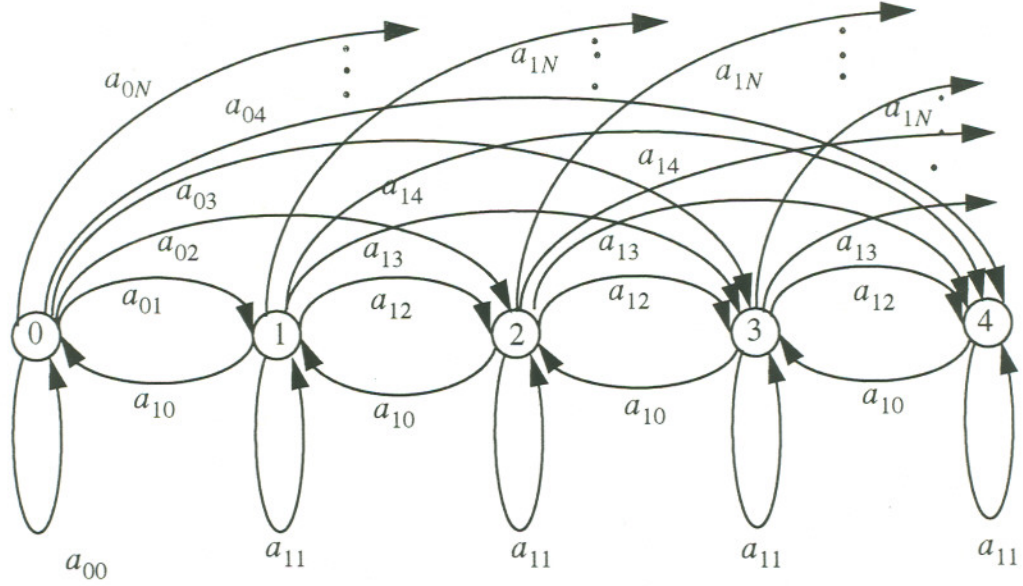


Figure 4: State Transition Diagram

$$\begin{aligned}
 q_2 &\triangleq \frac{1 - a_{11}}{a_{10}} q_1 - \frac{a_{01}}{a_{10}} q_0 \\
 q_3 &\triangleq \frac{1 - a_{11}}{a_{10}} q_2 - \frac{a_{12}}{a_{10}} q_1 - \frac{q_{02}}{a_{10}} q_0 \\
 q_4 &\triangleq \frac{1 - a_{11}}{a_{10}} q_3 - \frac{a_{12}}{a_{10}} q_2 - \frac{a_{13}}{a_{10}} q_1 - \frac{a_{03}}{a_{10}} q_0 \\
 &\vdots \\
 q_n &\triangleq \frac{1 - a_{11}}{a_{10}} q_{n-1} - \sum_{i=2}^{n-1} \frac{a_{1i}}{a_{10}} q_{n-i} - \frac{a_{0n-1}}{a_{10}} q_0
 \end{aligned} \tag{11}$$

Let  $q_0 \triangleq c$ , an arbitrary constant. We can now solve for the values of  $q_i$  recursively and then renormalize the values obtained to account for the value of  $q_0$  having been to be assumed arbitrarily. These equations are very similar to those obtained for output buffering obtained by Karol et al in [5]. In our model, the equations take into account that the cells in the queues are recirculated into the system thus leading to (4) and (5). In output queueing, the cells are not recirculated but are queued at the outputs.



## 4 Numerical results

Using the values of  $q_i$  obtained, the number of inputs  $N$  and the probability of cell generation  $p$ , at each input,  $1-\beta$  can then be calculated and is shown in table one. Comparable results from a simulation model are shown in table two. They show a very close agreement with each other. The expected number of the visits to the buffer can be easily computed from (7). The analytical results and simulation results are shown in tables three and four. The results at high loads show that a cell on the average would expect to make less than one trip to the buffer. While this is a useful figure of merit, this is not equivalent to the amount of delay that a cell has to endure passing through the system. This is because a cell may only make a single trip to the buffer, but due to the other cells in front of it, it may have to stay in the buffer for some time.

Tables five and six show the mean queue lengths for the analytical and the simulation results respectively. The analytical results are obtained from the recursive solution of the steady state probabilities in (11). The probability of a packet getting through the system is plotted in figure five for the analytical case and in figure six for the simulation. Plots of the analytical and simulation results for  $N=16,64$  and  $256$  for expected number of visits to the buffer are in figure seven. Figure eight shows the plot of the mean queue lengths again for the same sized switch as before. The plots reveal that even with our assumption of infinite buffer lengths, at every high traffic loads, the size of the queues in the buffer are still within a reasonable size. The advantage of such a scheme it will be possible to tradeoff increased queue lengths for greatly reduced probability of loss of blocked cells due to finite buffer space.

## 5 Conclusion

A performance evaluation of a Starlite-like switching architecture has been carried out. Analytical as well as simulation results have been obtained. Both results show very good agreement with each other. We have obtained a useful figure of merit in the expected number of trips that a cell expects to make to the buffer. In addition, we have also obtained the average queue size in the buffer under different loading conditions. Since this was with

no cells lost due to blocking, this allows us to have an idea of what queue sizes would be required in implementing a non-blocking switching network for ATM applications.

## 6 Acknowledgement

The research in this paper was supported by the SDIO/IST under the Office of Naval Research grant N00014-91-J4063

## References

- [1] K. Batcher, "Sorting Networks and their Applications", *AFIPS Conference Proceedings 1968 SJCC*, pp. 307-314
- [2] A. E. Eckberg and T.-C. Hou, "Effects of Output Buffer sharing on Buffer Requirements in an ATDM packet switch", *Proceedings of IEEE Infocom 88*, New Orleans, pp. 459-465.
- [3] M. Hluchyj and M. Karol, "Queueing in Space-Division Packet Switching", *Proceedings of IEEE Infocom 88*, New Orleans, pp. 334-343.
- [4] A. Huang and S. Knauer, "Starlite: A wideband digital switch", *Proceedings of IEEE Globecom 84*, pp. 121-125.
- [5] M. Karol, M. Hluchyj and S. Morgan, "Input versus Output Queueing on a Space-Division Packet Switch", *IEEE Transactions on Communications*, Vol. Com-35, No. 12, pp. 1347-1356.
- [6] R. Melen and J. Turner, "Nonblocking Networks for Fast Packet Switching", *Proceedings of Infocom 89*, Ontario, Canada pp. 548-557.
- [7] J. Hui, "Switching and Traffic Theory for Integrated Broadband Networks", Kluwer Academic Publishers, Boston, 1990.

Table 1: Analytical values of  $1-\beta$

1- $\beta$ : Analytical results						
	Number of inputs N					
p	8	16	32	64	128	256
0.1	0.977	0.975	0.975	0.974	0.974	0.974
0.2	0.951	0.948	0.947	0.946	0.946	0.946
0.3	0.922	0.918	0.917	0.916	0.915	0.915
0.4	0.890	0.886	0.884	0.883	0.882	0.882
0.5	0.855	0.851	0.849	0.848	0.847	0.847
0.6	0.817	0.813	0.811	0.810	0.809	0.809
0.7	0.774	0.771	0.770	0.769	0.769	0.769
0.8	0.728	0.727	0.726	0.726	0.726	0.726
0.9	0.677	0.679	0.679	0.680	0.680	0.680
0.92	0.667	0.669	0.670	0.670	0.671	0.671
0.94	0.656	0.658	0.660	0.661	0.661	0.661
0.95	0.650	0.653	0.655	0.656	0.656	0.656
0.96	0.645	0.648	0.650	0.651	0.651	0.652
0.97	0.639	0.643	0.645	0.646	0.646	0.647
0.98	0.634	0.638	0.640	0.641	0.641	0.642
0.99	0.628	0.633	0.635	0.636	0.637	0.637
1.0	0.624	0.629	0.631	0.632	0.632	0.632



Table 2: Simulation values of  $1-\beta$

1- $\beta$ : Simulation results						
	Number of inputs N					
p	8	16	32	64	128	256
0.1	0.976	0.975	0.974	0.974	0.974	0.974
0.2	0.951	0.948	0.947	0.946	0.946	0.946
0.3	0.923	0.919	0.917	0.915	0.915	0.915
0.4	0.892	0.886	0.884	0.883	0.882	0.882
0.5	0.857	0.850	0.849	0.848	0.846	0.847
0.6	0.818	0.813	0.811	0.810	0.809	0.809
0.7	0.774	0.771	0.770	0.769	0.769	0.769
0.8	0.729	0.726	0.726	0.725	0.726	0.726
0.9	0.678	0.679	0.679	0.680	0.680	0.680
0.92	0.667	0.669	0.669	0.670	0.670	0.671
0.94	0.656	0.658	0.659	0.661	0.661	0.661
0.95	0.650	0.654	0.654	0.655	0.656	0.656
0.96	0.645	0.648	0.650	0.651	0.651	0.651
0.97	0.638	0.643	0.645	0.646	0.646	0.647
0.98	0.632	0.637	0.640	0.641	0.642	0.642
0.99	0.628	0.632	0.635	0.636	0.637	0.637
1.0	0.623	0.628	0.631	0.632	0.633	0.633

Table 3: Analytical results: Expected number of visits to the buffer

Analytical results: Number of visits to buffer						
	Number of inputs N					
p	8	16	32	64	128	256
0.100	0.024	0.026	0.026	0.027	0.027	0.027
0.200	0.052	0.055	0.056	0.057	0.057	0.057
0.300	0.085	0.089	0.091	0.092	0.093	0.093
0.400	0.124	0.129	0.131	0.133	0.134	0.134
0.500	0.170	0.175	0.178	0.181	0.181	0.181
0.600	0.224	0.230	0.233	0.236	0.236	0.236
0.700	0.292	0.297	0.299	0.300	0.300	0.300
0.800	0.374	0.376	0.377	0.377	0.377	0.377
0.900	0.477	0.473	0.473	0.471	0.471	0.471
0.920	0.499	0.495	0.493	0.490	0.490	0.490
0.940	0.524	0.520	0.515	0.513	0.513	0.513
0.950	0.538	0.531	0.527	0.524	0.524	0.524
0.960	0.550	0.543	0.538	0.536	0.536	0.534
0.970	0.565	0.555	0.550	0.548	0.548	0.546
0.980	0.577	0.567	0.563	0.560	0.560	0.558
0.990	0.592	0.580	0.575	0.570	0.570	0.570
1.000	0.603	0.590	0.585	0.582	0.582	0.582

Table 4: Simulation results for the expected number of visits to the buffer

Simulation results: Number of visits to buffer						
	Number of inputs N					
p	8	16	32	64	128	256
0.100	0.025	0.026	0.027	0.027	0.027	0.027
0.200	0.052	0.055	0.056	0.057	0.057	0.057
0.300	0.083	0.088	0.091	0.093	0.093	0.093
0.400	0.121	0.129	0.131	0.133	0.134	0.134
0.500	0.167	0.176	0.178	0.179	0.182	0.181
0.600	0.222	0.230	0.233	0.235	0.236	0.236
0.700	0.292	0.297	0.299	0.300	0.300	0.300
0.800	0.372	0.377	0.377	0.379	0.377	0.377
0.900	0.475	0.473	0.473	0.471	0.471	0.471
0.920	0.499	0.495	0.495	0.493	0.493	0.490
0.940	0.524	0.520	0.517	0.513	0.513	0.513
0.950	0.538	0.529	0.529	0.527	0.524	0.524
0.960	0.550	0.543	0.538	0.536	0.536	0.536
0.970	0.567	0.555	0.550	0.548	0.548	0.546
0.980	0.582	0.570	0.563	0.560	0.558	0.558
0.990	0.592	0.582	0.575	0.572	0.570	0.570
1.000	0.605	0.592	0.585	0.582	0.580	0.580



Table 5: Analytical results for the mean queue size

Analytical results: Mean queue size						
	Number of inputs N					
p	8	16	32	64	128	256
0.1	0.005	0.005	0.005	0.005	0.006	0.006
0.2	0.022	0.023	0.024	0.025	0.025	0.025
0.3	0.056	0.060	0.062	0.063	0.064	0.064
0.4	0.117	0.125	0.129	0.131	0.132	0.133
0.5	0.219	0.234	0.242	0.246	0.248	0.249
0.6	0.394	0.422	0.436	0.443	0.446	0.448
0.7	0.715	0.766	0.791	0.804	0.810	0.813
0.8	1.400	1.500	1.550	1.575	1.587	1.594
0.9	3.544	3.797	3.923	3.987	4.018	4.034
0.92	4.628	4.959	5.125	5.207	5.249	5.269
0.94	6.428	6.893	7.131	7.248	7.306	7.335
0.95	7.835	8.415	8.730	8.883	8.954	8.990
0.96	9.832	10.60	11.08	11.33	11.43	11.48
0.97	12.72	12.82	14.74	15.34	15.56	15.62
0.98	16.85	18.55	20.56	23.60	23.70	23.92
0.99	22.73	25.05	29.43	36.09	43.72	48.18

Table 6: Simulation results for the mean queue size

Simulation results: Mean queue size						
	Number of inputs N					
p	8	16	32	64	128	256
0.1	0.005	0.005	0.005	0.006	0.006	0.006
0.2	0.022	0.023	0.024	0.025	0.025	0.026
0.3	0.056	0.059	0.062	0.067	0.069	0.065
0.4	0.117	0.126	0.129	0.132	0.133	0.133
0.5	0.216	0.234	0.242	0.245	0.248	0.249
0.6	0.389	0.417	0.434	0.441	0.446	0.451
0.7	0.715	0.762	0.782	0.805	0.810	0.817
0.8	1.384	1.500	1.547	1.582	1.597	1.599
0.9	3.269	3.758	3.941	3.933	4.102	4.076
0.92	4.291	4.867	5.199	5.326	5.353	5.377
0.94	6.195	7.089	7.341	7.324	7.511	7.400
0.95	8.198	7.969	8.237	8.721	9.147	8.994
0.96	9.349	10.26	10.37	11.46	11.63	11.71
0.97	13.52	14.19	16.49	15.41	15.90	15.84
0.98	18.80	20.94	22.25	23.28	24.67	23.52
0.99	38.12	33.33	43.98	42.76	41.32	40.42

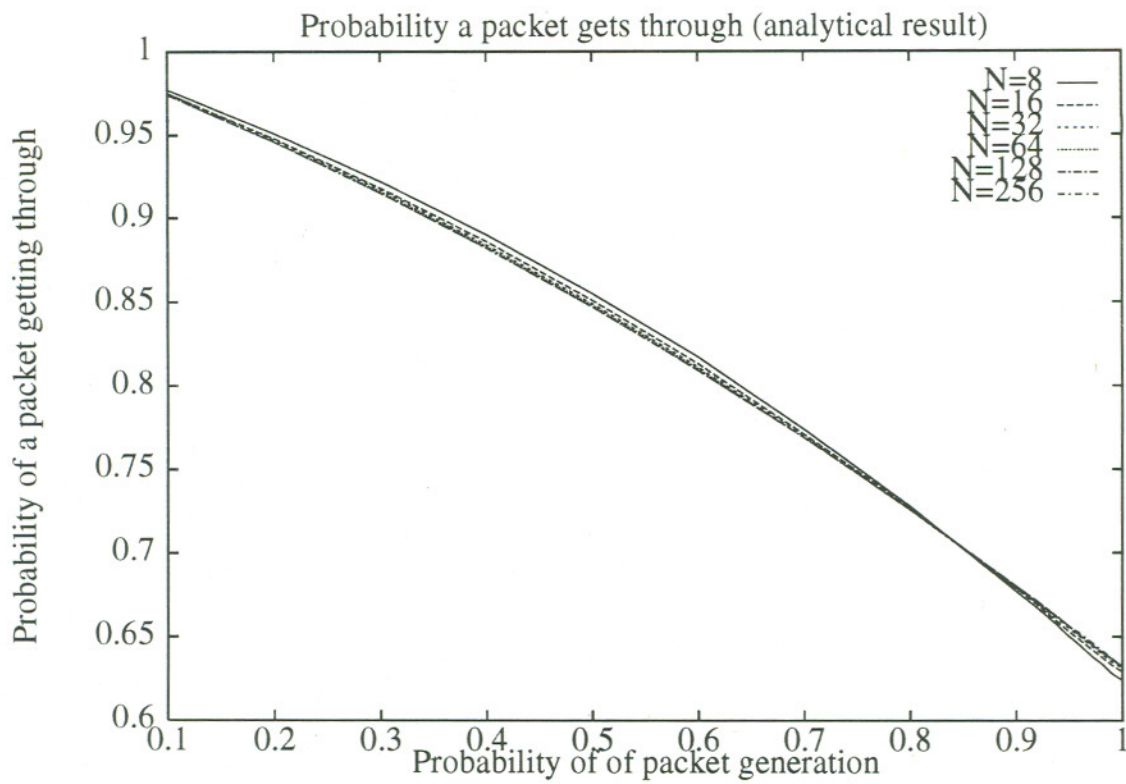


Figure 5: Graph of  $1-\beta$  for analytical results



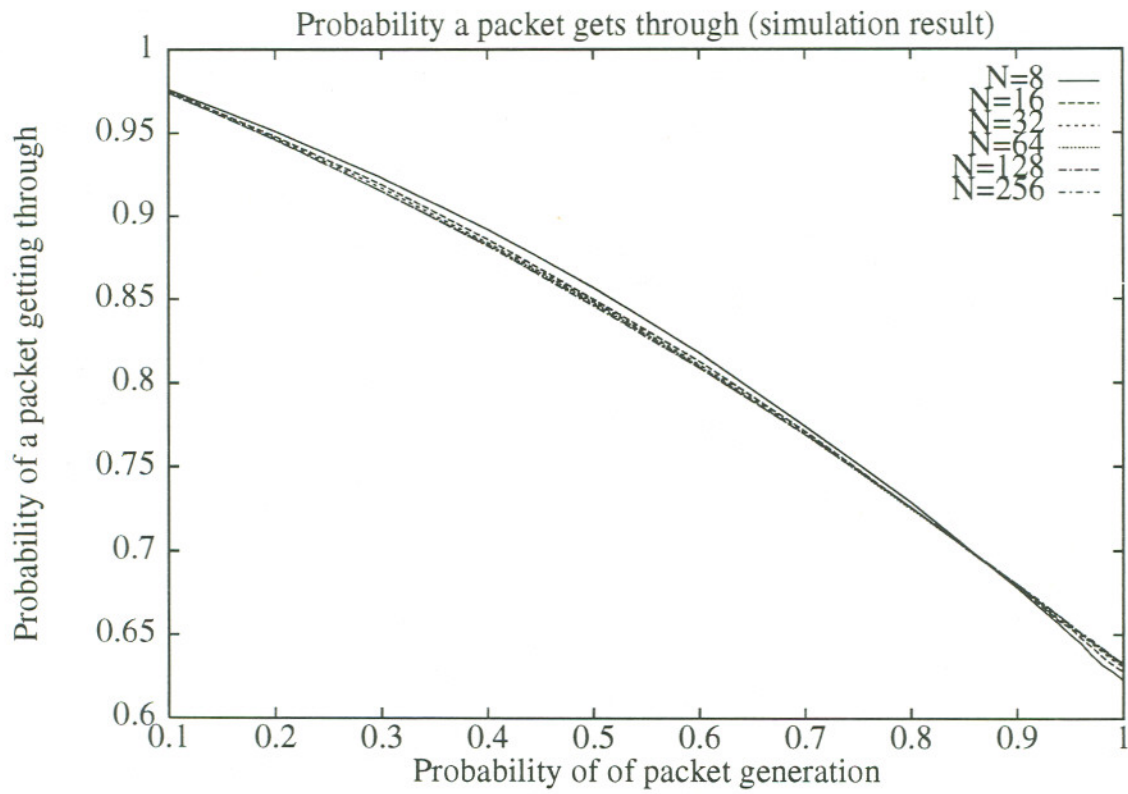


Figure 6: Graph of  $1-\beta$  for simulation

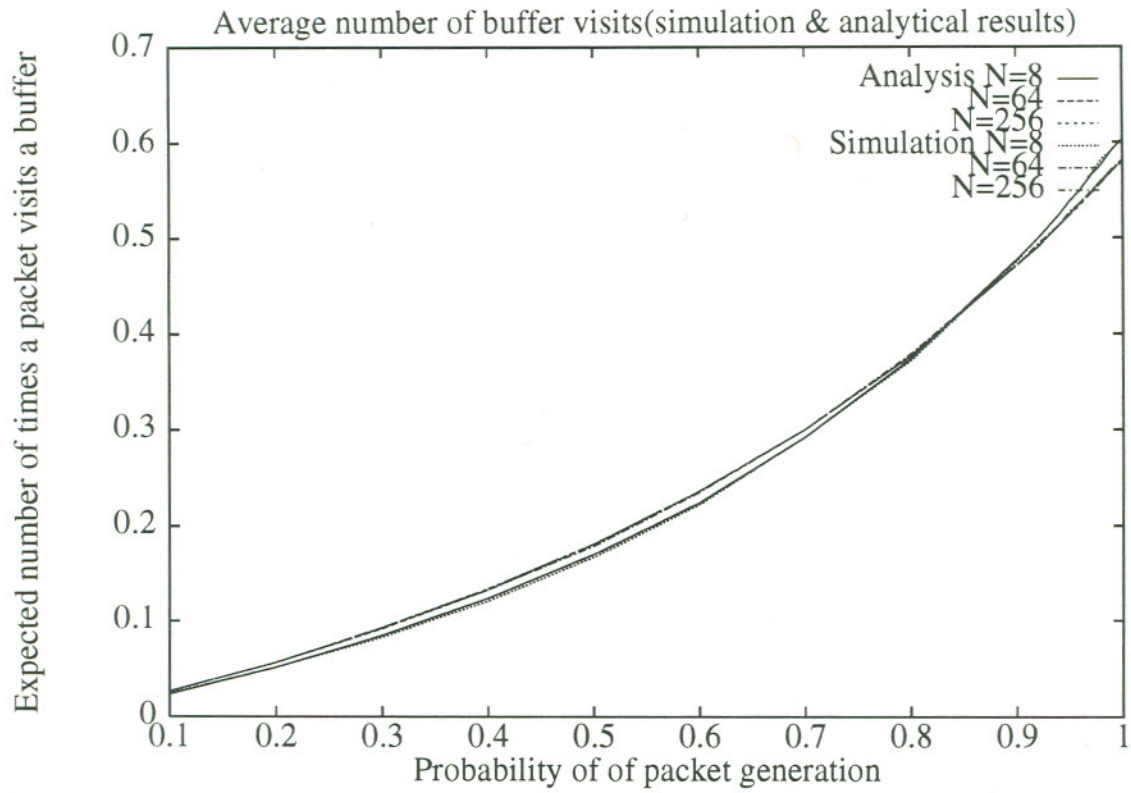


Figure 7: Graph of expected number of buffer visits for analysis and simulation

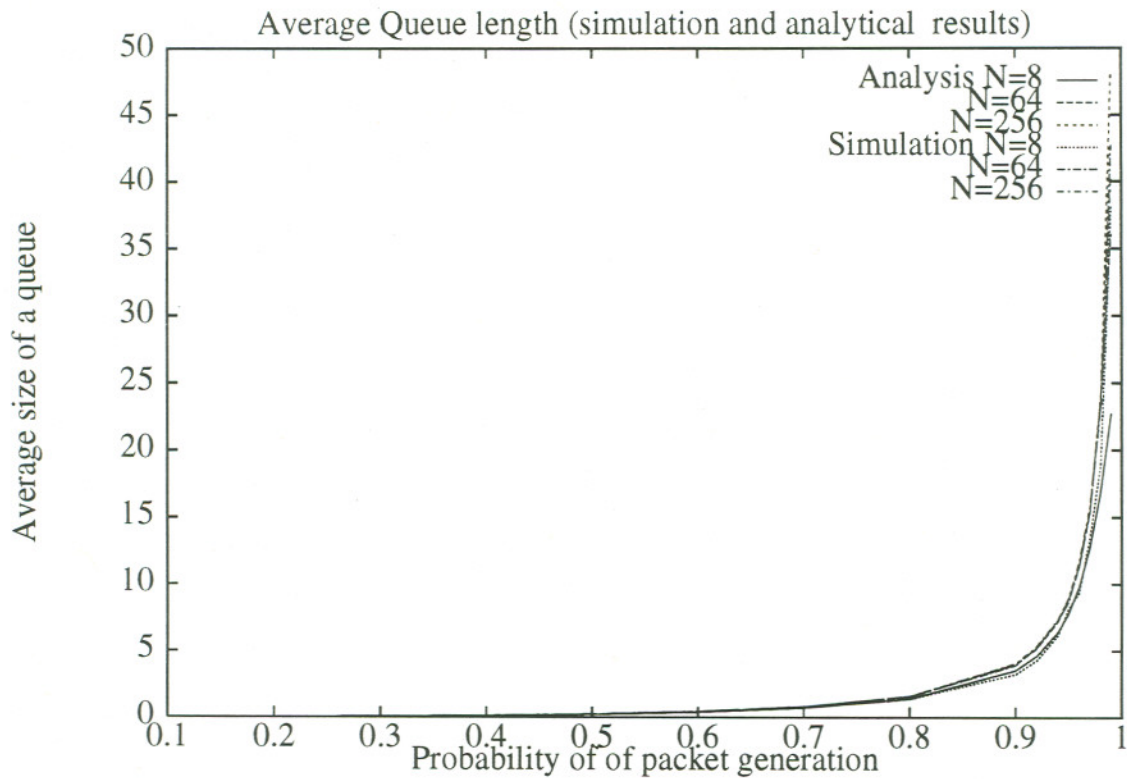


Figure 8: Graph of average queue lengths for analysis and simulation