

# Performance Analysis of an ATM Local Computer Network

Nader Mirfakhraei

Advanced Telecommunications Institute  
Stevens Institute of Technology, Hoboken, New Jersey 07030, U.S.A.

## Abstract

*This paper evaluates the performance of a local switching network using a Manhattan street network (MSN) for high-speed ATM applications. The MSN has a cyclic structure and belongs to the class of deflection-routing networks. We propose the use of shared buffering in the structure of each node. With shared buffers the network operates efficiently since the occurrence of deflections is minimized. We develop a new analytical model for the MSN and use it to study the traffic performance of this network. The key new result of this study is the need to scale the bandwidth of the MSN links in proportion to the network size. Through this evaluation, we determine how much speed advantage an MSN requires to become a practical switch. Numerical results of the analysis are compared with numerical results of an extensive simulation developed for this network.*

## 1 Introduction

The *Manhattan-street network* (MSN) is a regular mesh network that can be realized as a collection of horizontal and vertical rings as shown in Figure 1. The MSN has been presented for implementation in geographically distributed computer

<sup>0</sup>This work was supported by the National Science Foundation, Ascom Timplex, Bell Communications Research, Bell Northern Research, Digital Equipment Corporation, Italtel SIT, NEC America, NTT, SynOptics Communications and Tektronix.

<sup>1</sup>The work was done while the author was with Computer and Communications Research Center (CCRC) at Washington University, St. Louis, MO.

networks [8] [9]. The links resemble the streets and avenues in Manhattan which alternate in direction. With the *torus*-shaped topology embedded in the network, at each node there are two links arriving and two links leaving regardless of the network size. Although it is not indicated in the figure, at each node an incoming link and an outgoing link connect the network to a local processor which generates and receives data sent through the network. Routing in the MSN relies on *deflection*. When two cells request the same outgoing link, only one of them is forwarded on the preferred link while the other one is deflected on the second link. By maintaining this rule in the system, once a cell is admitted to a network, it is not discarded if congestion occurs. Instead, the cell is misrouted temporarily but will ultimately reach its destination. However, deflection of cells onto longer paths causes additional delay.

This paper is organized as follows. In Section 2, the structure of a node with shared buffering and its potential effects on performance are discussed. In Section 3, the performance of the network under deflection routing using a stochastic model for the queueing-deflection process is analyzed, and some numerical results are presented. In Section 4 results of the performance evaluation on the system throughput, delay and average number of deflections are presented and compared with the results of a simulation.

## 2 Nodes with Shared Buffering

Consider Figure 1 and assume the network has no internal buffers to store incoming traffic. In a

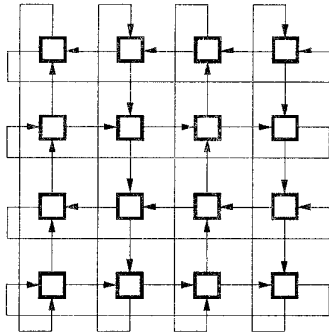


Figure 1: A  $4 \times 4$  Manhattan-street network.

deflection-routing network, if two cells from the neighboring nodes arrive at the same time and both cells require use of a given output, the conflict is resolved by temporarily deflecting one of the cells on an undesired link and allowing the other one to be forwarded to the desired link. However, deflection routing by itself as a contention resolution strategy can waste bandwidth within the network by requiring cells to follow suboptimal routes. If a node has no internal storage, deflection is necessary to avoid blocking or discarding the cell. It is apparent that augmenting the Manhattan street networks with buffers at every node potentially offers better performance. In this context, [3] and [4] have examined the effect of *output buffering* in the Manhattan-street network. With this strategy, a cell must be routed to a buffer upon its arrival. If the desired buffer is full, the cell has to be deflected on the other one and it can not be changed at a later time if the desired buffer allows admission of new arrivals.

We extend this work and combine the two output buffers of each node, letting the inputs and outputs share an internal queue to reduce the frequency of the full buffer condition and hence reducing the number of deflections further. With this buffering structure cells are stored in the shared buffer slots regardless of their destinations, and deflection occurs only when all buffer slots are occupied. The structure of a node in a two-connected network is shown in Figure 2. Note that the routing within the network is implemented by  $2 \times 2$  nodes, but as mentioned be-

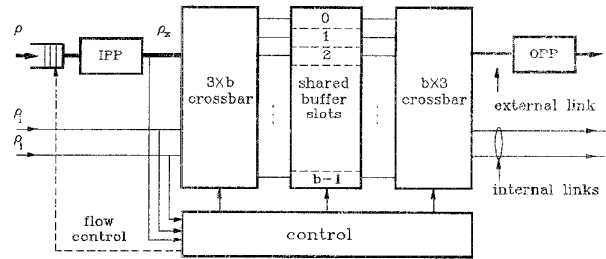


Figure 2: The proposed structure of a node.

fore, there is one incoming external link and one outgoing external link joining each node to a local processor. In the ATM context, this processor would implement the interface to an external transmission link. Thus, each node is in fact a  $3 \times 3$  switch. With the shared buffer structure, an incoming cell is immediately stored in one of the  $b$  buffer slots. A  $3 \times b$  crossbar implements the switching functions. At each cell cycle, a cell competes to access the desired outgoing link based on a priority field, and the winning cell is allowed to exit through a  $b \times 3$  crossbar.

The method of shared buffering in the MSN would potentially reduce the number of deflections compared to the similar but nonbuffered networks or MSNs with different buffering strategies. The second advantage of using the shared buffer scheme is the nature of such a queuing structure for providing better performance. In [1] a comprehensive study on the performance of the shared buffering strategy for switching networks has been made. The results of this study show that switching systems with shared buffers at each switch element provide better overall performance.

## 2.1 Description of Node Operation

When cells arrive at a node, the node determines how they should be routed. There are four possibilities; type-1 cells are transmitted to the local processor (through the external link), type-2 cells are addressed to the row output (internal link), type-3 cells are addressed to the column output (internal link), and finally type-4 cells can be sent to either the row or col-

umn. The last possibility arises because the grid structure of the MSN provides a multiplicity of equally good paths between most source-destination pairs. The routing decision for a cell is made when the cell arrives and is stored in the control circuitry of the node.

The MSN provides no flow control between adjacent nodes in the network, but does provide flow control back to the local processor. This means that the node must be prepared to receive two cells each cycle. To ensure that two cells can be received and stored in one cycle, the output control always transmits enough cells to ensure that two buffer slots are available for arriving cells. If at the start of a node operational cycle, the shared buffer is full and all stored cells are of the same type and none is of type 4, one of the stored cells is selected for deflection and sent to a row or column output that would not otherwise be used.

## 2.2 Port Processors

At the incoming external link of a node, an *input port processor* (IPP) is used to convert the ATM cell to the local network format. The IPPs have lookup tables to determine the routing information for cells. Due to the limited capacity of the network, an input buffer, as shown in Figure 2 accepts new arrivals if there are empty slots; otherwise the cells are discarded. Once a cell is admitted to the network, due to the queueing-routing strategy explained earlier no cells are lost. At the output side, *output port processors* (OPPs) accept cells from the MSN and resequence them. A *resequencing operation* is needed for misordered cells. This misordering is caused by the multiplicity of paths and the resulting variation in delays in the queues throughout the network.

## 3 Performance Analysis

In this section, an exact analytic model for the Manhattan-street network based on the proposed node structure is presented. We start with the state of the input buffer which is explicitly modeled by a discrete time birth-death process. Ac-

cording to the operational description of a node given earlier, the input buffer is granted permission to deliver cells to the shared buffer so long as the number of cells in the shared buffer is less than  $b - 1$ . This mechanism is carried out by a *grant* flow control as shown in Figure 3, and the probability that a grant is given is denoted by  $\varphi$ . Thus,  $\varphi$  depends on the state of the shared buffer. Let  $\pi(\mathbf{j})$  be the probability that the shared buffer at a given node contains  $j_1, j_2, j_3$ , and  $j_4$  cells of type 1, 2, 3, and 4, respectively where  $\mathbf{j}$  is a state vector  $[j_1, j_2, j_3, j_4]$ , then  $\varphi$  can be expressed as:

$$\varphi = 1 - \left[ \sum_{\mathbf{j}, j_1+j_2+j_3+j_4=b-1} \pi(\mathbf{j}) + \sum_{\mathbf{j}, j_1+j_2+j_3+j_4=b} \pi(\mathbf{j}) \right]. \quad (1)$$

Using  $\varphi$  and the external offered load  $\rho$ , all states of the input queue can be determined from the transition probabilities of the Markov chain describing the input buffer. In general, The probability that an input port buffer contains exactly  $s$  cells,  $\pi_x(s)$ , is computed recursively starting from  $\pi_x(0)$  and is given by:

$$\pi_x(s) = \pi_x(s-1)\rho\bar{\varphi} + \pi_x(s)(\bar{\rho}\bar{\varphi} + \rho\varphi) + \pi_x(s+1)\bar{\rho}\varphi, \quad (2)$$

where  $\alpha$  is the number of input buffer slots, and  $\bar{\varphi}$  and  $\bar{\rho}$  are  $(1 - \varphi)$  and  $(1 - \rho)$ , respectively. Next, let  $\rho_x$  be the probability that a cell is available to enter the shared buffer from the input buffer. This term is calculated from the state of the input queue when the queue is nonempty:

$$\rho_x = 1 - \pi_x(0). \quad (3)$$

As defined above, let  $\rho_i$  be the probability that the row (or column) predecessor of a given node has a cell for it. By the symmetry of the network and the uniform traffic assumption,  $\rho_i$  is also the probability that a node sends a cell on its row (or column) output. Thus:

$$\rho_i = \sum_{\mathbf{j} \in X} \pi(\mathbf{j}) + \frac{1}{2} \sum_{\mathbf{j} \in Y} \pi(\mathbf{j}). \quad (4)$$

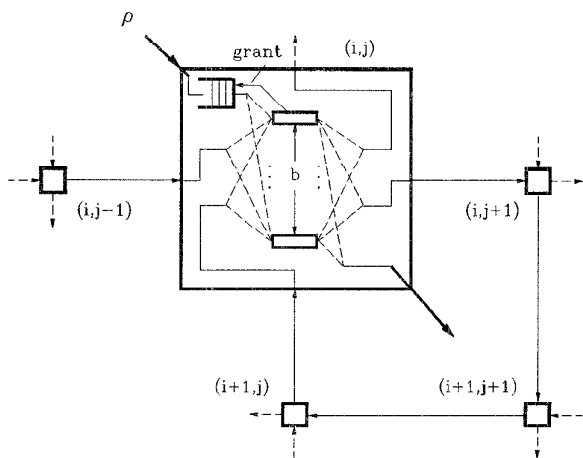


Figure 3: Modeling a node for queueing-deflection analysis.

where  $X$  consists of all state vectors  $[j_1, j_2, j_3, j_4]$  that satisfy the following:

$$\begin{aligned} & [j_2 > 0] \vee [j_2 = 0 \wedge j_3 = 0 \wedge j_4 > 1] \\ \vee & [j_2 = 0 \wedge j_3 > 0 \wedge j_4 > 0] \\ \vee & [j_1 = 0 \wedge j_2 = 0 \wedge j_3 = b \wedge j_4 = 0] \end{aligned}$$

and  $Y$  consists of all state vectors  $[j_1, j_2, j_3, j_4]$  that satisfy the following:

$$[j_2 = j_3 = 0] \wedge [j_1 + j_4 = b] \wedge [j_4 \leq 1]$$

Also, with term  $\pi(\mathbf{j})$  the condition under which a cell must be deflected can explicitly be determined. Let  $\delta$  be the probability that a node deflects a cell in a given cell cycle. A cell is deflected when all of the shared buffer slots are occupied by cells of type 1 or 2 or 3. This statement is expressed by:

$$\delta = \frac{1}{2} \rho_x \sum_{\mathbf{j} \in Z} \pi(\mathbf{j}), \quad (5)$$

where  $Z$  consists of all state vectors  $[j_1, j_2, j_3, j_4]$  that satisfy the following:

$$[j_1 = b] \vee [j_2 = b] \vee [j_3 = b].$$

Now, in order to compute the probability that a given number of cells enter a node, we first need

the traffic load for cells of each specific type. Let  $\rho_{x2}$ ,  $\rho_{x3}$ , and  $\rho_{x4}$  be the offered loads from the input buffer for cells of type 2, 3, and 4 respectively. Notice here that, no cells of type 1 arrive from the external link as indicated in Figure 4. The use of a uniform traffic model leads to balanced cell arrivals of type 2 and 3 from each internal link and thus  $\rho_{x2} = \rho_{x3}$ . One can notice that in the MSN there are a number of nodes in which a cell can take either outgoing link to reach to its destination using a shortest possible path. This is a consequence of the topological structure of the MSN which provides a multiplicity of shortest paths between most pairs of nodes. Here, these types of nodes are referred to as *don't-care nodes*. Let  $\nu$  be the overall percentage of don't-care nodes among the other nodes. It can easily be shown that in a network with  $N^2 - 1$  nodes (destination node is not included):

$$\nu = \frac{1}{N^2 - 1} \cdot \left( \frac{1}{2} N^2 - N + 1 \right) \cdot 100\%. \quad (6)$$

For details of computation of  $\nu$ , see [12]. We take the percentage of cells that are of type 4 to be equal to  $\nu$ . This gives

$$\rho_{x4} = \nu \rho_x, \quad (7)$$

and also:

$$\rho_{x2} = \rho_{x3} = \frac{1}{2} (1 - \nu) \rho_x. \quad (8)$$

Similarly for  $i \in \{1, 2\}$ , let  $\rho_{i1}$ ,  $\rho_{i2}$ ,  $\rho_{i3}$ , and  $\rho_{i4}$  be the internal traffic arriving from an internal link that contain only cells of type 1, 2, 3, and 4 respectively. Relying on the fact that no cells are lost within the network, the exiting load on the external link (the predecessor link of the OPP) should be equal to  $\rho_x$ . This load is clearly the result of the cell transmission from the two internal links. Therefore each internal link has an equal contribution of  $\rho_x/2$  to the exiting traffic from the external link as shown in Figure 4. The internal traffic loads can be written as:

$$\rho_{i1} = \frac{1}{2} \rho_x, \quad (9)$$

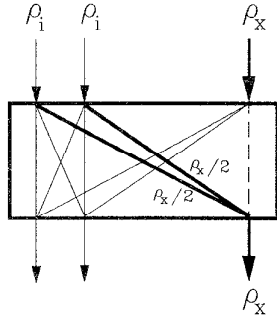


Figure 4: Distribution of traffic load in a node.

$$\rho_{i2} = \rho_{i3} = \frac{1}{2}(1 - \nu)(\rho_i - \frac{1}{2}\rho_x), \quad (10)$$

$$\rho_{i4} = \nu(\rho_i - \frac{1}{2}\rho_x). \quad (11)$$

Note that  $\rho_i$  can be computed by  $\rho_i = \ell\rho_x/2$  where  $\ell$  is the average length of a path taken by a cell. Now, we can compute the probability that a given number of cells of various types enter a node. Assuming that an arriving cell from an external link never exits from the same node, we define the generating function  $g(x, y, z)$ :

$$g(x, y, z) = ((1 - \rho_x) + (\rho_{x2} + \rho_{x3})y + \rho_{x4}z) \\ ((1 - \rho_1) + \rho_{11}x + (\rho_{12} + \rho_{13})y \\ + \rho_{14}z)^2. \quad (12)$$

This expression can be expanded into a polynomial on  $x$ ,  $y$ , and  $z$ . In this form, the coefficient of  $x^{j_1}y^{j_2+j_3}z^{j_4}$  represents the probability that exactly  $j_1$  cells of type 1,  $j_2 + j_3$  cells of types 2 and 3, and  $j_4$  cells of type 4 enter in a single cycle. We call this probability  $p(j_1, j_2, j_3, j_4)$ , or  $p(\mathbf{j})$  where  $\mathbf{j}$  is a vector  $[j_1, j_2, j_3, j_4]$ . Notice that this analysis relies on the assumption that the types of arriving cells are independent.

We model the shared buffer by a Markov process. We define  $\lambda(\mathbf{j}_1, \mathbf{j}_2)$  to be the probability that if a shared buffer is in state  $\mathbf{j}_1$  in the current cycle, it will be in state  $\mathbf{j}_2$  during the next cycle. Let  $Q(\mathbf{j})$  be the vector  $\mathbf{k} = [k_1, k_2, k_3, k_4]$  that represents the number of departing cells of each type when the state is  $\mathbf{j}$ . The transition probability is

then computed by:

$$\lambda(\mathbf{j}_1, \mathbf{j}_2) = \sum_{\mathbf{h}} p(\mathbf{h}) \cdot \quad (13)$$

$\mathbf{j}_1 + \mathbf{h} - Q(\mathbf{j}_1) = \mathbf{j}_2$

The steady state probability for the shared buffer can then be determined from the balance equations:

$$\pi(\mathbf{j}_2) = \sum_{\mathbf{j}_1} \lambda(\mathbf{j}_1, \mathbf{j}_2)\pi(\mathbf{j}_1). \quad (14)$$

We now take into account the impact of routing where a cell takes an average of  $\ell$  hops from its source to its destination. Let  $\ell_s$  be the average length of a shortest path from a source to destination assuming no deflections. For simplicity, realize the re-orientation of all addresses with respect to the destination address at  $(0,0)$  [9]. The average distance can be computed by summing over all possible source addresses  $(i, j)$  as follows:

$$\ell_s = \frac{1}{N^2 - 1} \sum_{i=-(N/2-1)}^{N/2} \sum_{j=-(N/2-1)}^{N/2} (|i| \\ + |j| + \varepsilon), \quad (15)$$

where  $\varepsilon$  is the number of additional hops a cell takes as a result of the fact that streets are always one way. It can be shown [12] that:

$$\ell_s = \frac{1}{N^2 - 1} (N^3/2 + N^2/4 - N + 4[N/4]^2 \\ + 4[N/4]). \quad (16)$$

To complete the analysis we need to determine the impact of deflection on the length of paths taken by cells. Thus, let  $\ell_d$  be the average number of hops taken by cells as a result of deflections. Then,  $\ell_d = g \cdot \Delta$ , where  $\Delta$  is the average number of deflections in a given routing, and  $g$  is the number of additional hops a cell takes per deflection. Examining the Manhattan street network, each deflection can cost a cell at most four additional steps, hence  $g \leq 4$ . For computing  $g$ , we need to calculate the sum of all additional hops cells take including all nodes and average it over the total number of nodes. In this case, one can

see that don't-care nodes in which cells never get deflected must be excluded from the calculation. The number of don't-care nodes can be obtained from Equation (6) (see [12] for details). The exact value for  $g$  can be given by:

$$g = \begin{cases} \frac{4(N^2-4)}{N^2+2N-4} & \text{if } \frac{N}{2} \text{ is even} \\ \frac{2(N^2+6N-16)}{N^2+2N-4} & \text{if } \frac{N}{2} \text{ is odd.} \end{cases}$$

Next,  $\Delta$  can be expressed in terms of  $\delta$ , the probability that a cell gets deflected which was given in Equation (5), therefore:

$$\Delta = \delta(\ell_s + \ell_d). \quad (17)$$

By substituting  $\Delta = \ell_d/g$  in the above equation, we have  $\ell_d = g\delta(\ell_s + \ell_d)$ , hence:

$$\ell_d = \left(\frac{g\delta}{1-g\delta}\right)\ell_s. \quad (18)$$

Let  $\ell$  be the average shortest path length including deflections. The value of  $\ell$  is given by:

$$\ell = \ell_s + \ell_d = \left(\frac{1}{1-g\delta}\right)\ell_s. \quad (19)$$

Given, the above equation, one can compute the steady-state probabilities for a representation node of the MSN using an iterative numerical computation. Given the steady-state probabilities, we can easily calculate the required performance metrics of interest. The *throughput* ( $T$ ) can be extracted directly from the state of the shared buffers. According to the previous discussion, as long as there is one cell of type 1 in the buffer, one of the cells is guaranteed to exit from the node. This is basically due to the fact that there is no flow control mechanism at the node output. The throughput is then equal to:

$$T = \sum_{\mathbf{j}, j_1 \neq 0} \pi(\mathbf{j}). \quad (20)$$

The average *delay* ( $D$ ) is defined to be the number of cell cycles that a cell spends in an  $n$ -port network to reach its destination. Between the time a cell enters the network until it leaves, it passes

through an average of  $\ell + 1$  nodes. In the first stage of this queueing sequence, a cell stays in the input queue before entering the network. In the next  $\ell$  nodes, at each stage a cell is queued in the shared buffer. At each of these nodes, the cell is of type 2, 3, or 4. Finally, a cell stays in the shared buffer of the destination node where it is a type-1 cell. Let  $D_x$  be the delay incurred by the input queue, and  $D_1, D_2, D_3$  and  $D_4$  be the delays that a cell incurs in shared queues where it is respectively of type 1, 2, 3, and 4. The total average delay can then be computed as follows:

$$D = D_x + \ell\left(\frac{1}{2}(1-\nu)D_2 + \frac{1}{2}(1-\nu)D_3 + \nu D_4\right) + D_1. \quad (21)$$

Recall that  $\nu$  represents the fraction of don't-care nodes. In general, delay in a queue is computed by the ratio of the queue length and the arrival rate. Let the length of an input queue be denoted by  $Q_x$ . Let  $Q_k$  be the number of cells of type  $k$  in a shared queue, where  $k \in \{1, 2, 3, 4\}$ . We can define these two quantities by:

$$Q_x = \sum_{0 \leq j \leq \alpha} j\pi_x(j), \quad (22)$$

$$Q_k = \sum_{\mathbf{j}} j_k \pi_x(\mathbf{j}). \quad (23)$$

The arrival rate for the input buffer is in fact the external offered load  $\rho$ . The arrival rate for the shared buffer depends on the type of cell and in a generic form is expressed by:

$$A_1 = \rho_x = 2\rho_{11}, \quad (24)$$

$$A_k = \rho_{xk} + 2\rho_{1k}, \quad k \in \{2, 3, 4\}. \quad (25)$$

By Little's law  $D_x = Q_x/A_x$  and  $D_i = Q_i/A_i$  for  $i \in \{1, 2, 3, 4\}$ . Then, by substituting  $D_x$  and  $D_i$  into Equation (21) the total average delay can be obtained.

## 4 Performance Evaluation Results

This section compares numerical results of the analysis with results of an extensive simulation

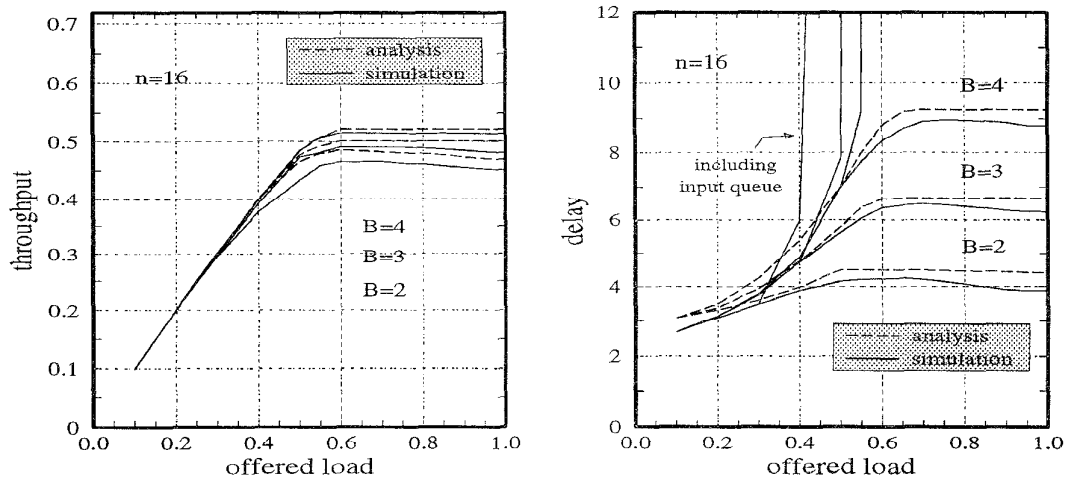


Figure 5: Throughput ( $T$ ) and delay ( $D$ ) versus offered load ( $\rho$ ) when the number of shared buffers per port of node ( $B$ ) varies.

experiments. We do not describe the structure of the simulator in this paper. For details of the simulation structure and description of each subprogram see [12]. We give the results of the average number of cells leaving the network per cell cycle per external link which is referred to as throughput ( $T$ ). The average delay ( $D$ ) is also measured from the time a cell enters the network to the time the cell leaves the network. By comparing the delay with the length of the least-cost path from source to destination, we can assess how efficiently the network is operating. The average number of deflections ( $\Delta$ ) per exiting cell is another measurement by which various results of the simulation are verified. We evaluate the effects of increasing offered traffic loads, varying speed advantage, and increasing network size on the network performance. Speed advantage refers to the ratio of the internal link speed to the external link speed.

#### 4.1 Effect of Increasing Offered Traffic Load

The first set of results includes Figure 5 and Table 3. These results compare the numerical outputs of the analysis given in the previous section with the simulation. Note that these results do not include the effect of speed advantage at this

point. Figure 5 (left) shows the throughput with three different numbers of buffer slots per port of each node ( $B$ ). With  $B = 2$  and  $n = 16$ , at offered loads exceeding  $\rho = 0.4$ , the curve starts to flatten out since the network begins to reach its full capacity.

By further increasing the load, the throughput does not improve further and even falls slightly since the growing incoming traffic increases the number of deflections as shown in Table 3. The flattening curves in Figure 5 (right) give the delay through the network, while the rising curves add the delay in the input queue. Notice the delay becomes large when the network capacity is reached. It is also worth noting that most of the total delay is incurred within the input queue when the load is heavy. Finally, we compare the average number of deflections ( $\Delta$ ) obtained from the simulation and analysis. We notice a *loop correlation effect* at large offered loads in Figure 5 (left) and Table 3 where the decrease of throughput slows down the increase of deflections, thus avoiding further decline of the throughput.

Higher throughputs can be obtained using larger buffers. Both the simulation and analysis show an improvement in the throughput when the size of the shared buffer increases. However, the overall throughput cannot exceed an upper

Table 1: Comparison of simulation and analysis on the average number of deflections ( $\Delta$ ) versus offered load ( $\rho$ ) when the number of shared buffers per port of node ( $B$ ) vary.

offered load ( $\rho$ )	analysis			simulation		
	$B = 2$	$B = 3$	$B = 4$	$B = 2$	$B = 3$	$B = 4$
0.1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.2	0.00287	0.00000	0.00000	0.00324	0.00000	0.00000
0.3	0.01492	0.00094	0.00000	0.01689	0.00119	0.00000
0.4	0.06827	0.01342	0.00289	0.07583	0.01463	0.00315
0.5	0.16270	0.05229	0.03031	0.18739	0.05840	0.03384
0.6	0.27219	0.15430	0.10842	0.31072	0.17829	0.12156
0.7	0.40320	0.28902	0.22082	0.45424	0.31027	0.24673
0.8	0.56009	0.42964	0.36183	0.62049	0.46047	0.39090
0.9	0.63034	0.49648	0.41461	0.70302	0.56844	0.46931
1.0	0.65884	0.51038	0.42996	0.72171	0.57942	0.47338

bound as seen in Figure 5. This mainly expresses the maximum capability of the network in handling the traffic at this rate of offered load. As was expected, by enlarging the capacity of the shared queue, the number of deflections is reduced as well. We will show later that a speed advantage is needed to improve performance further. Comparisons on the results of simulation and the analysis exhibit admissible accuracy in the performance evaluation as seen in the plots.

#### 4.2 Effect of Network Size

The next set of plots given in Figure 6 focuses on the effect of the network size on the throughput. In Figure 6 (left) we have examined the impact of different network dimensions including  $n \in \{16, 64, 144, 256\}$  ( $N \in \{4, 8, 12, 16\}$ ). In these experiments, the offered load is fixed at  $\rho = 0.9$ . As expected, enlarging the network size  $n$  reduces the throughput. In this case, the system performance is sensitive to adding more buffer slots to the shared queue, but improving the system throughput with this approach is impractical and it works out only to a certain point as explained earlier.

Figure 6 (right) clearly demonstrates the effect

of network size on the throughput when the speed advantage is  $3N/4$ . We used this value since the average number of hops in the networks we simulated is close to this figure. Note that the speed advantage varies with the network size, thus enlarging the network dimension has little impact on the throughput as shown in the figure.

#### 4.3 Effect of the Speed Advantage

In the previous discussion, we observed that the network seems to be incapable of handling high traffic loads. We also noticed that increasing the number of shared buffers can improve the throughput, but this approach is useful only to a certain point beyond which the throughput even decreases.

The next set of results shown in Figure 7 evaluates the effectiveness of a speed advantage ( $s$ ) on the network performance. The simulation repeats all previous experiments but letting the internal data paths operate at a rate  $s$  times faster than the rate of the external links where  $s \in \{2, 3, \dots, 10\}$ . The results are given together with the previous observation of rate  $s = 1$ . With the speed advantage of 2 and 3, throughput improves approaching the ideal case. The plots

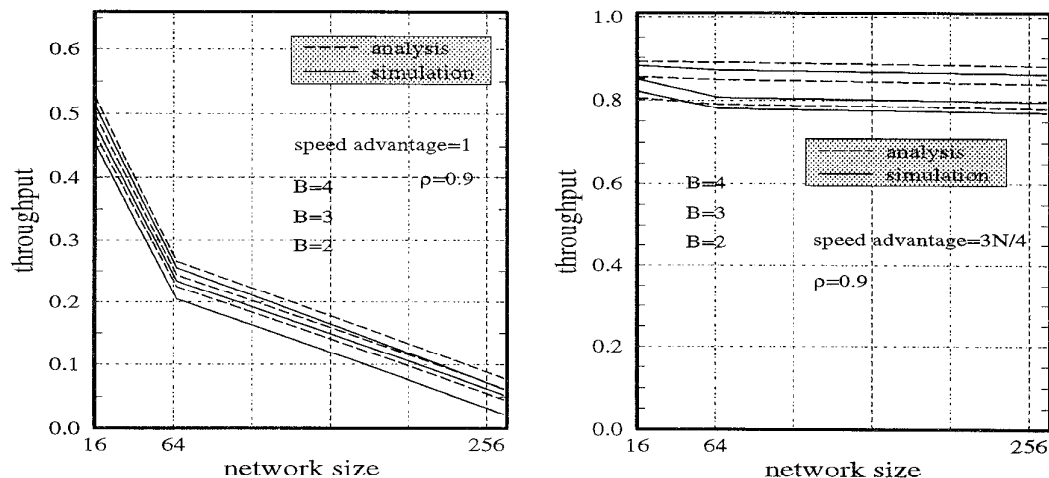


Figure 6: The effect of enlarging network size ( $n$ ) on throughput ( $T$ ) for three different numbers of buffers per port of node ( $B$ ). Left: the speed advantage is one, right: the speed advantage is  $3N/4$  ( $n = N^2$ ).

show that in the neighborhood of  $s = 5$ , that is  $s = 1.25N$ , the network performs ideally meaning that it delivers the entire offered load to external links regardless of the traffic load.

## 5 Summary

In this paper we presented the design and analysis of a local ATM switching system based on the Manhattan-street network (MSN). We presented an ATM-compatible implementation for each node of the MSN. We studied the traffic performance of this network through a completely new analytic approach and computed the average throughput and routing delay. The results of extensive simulation experiments were compared with the results of the analytical approach. The use of shared buffering in the nodes resulted in lower delays and better performance compared to the previous work in the literature since the average number of deflections was reduced. Both throughput and delay decline slightly at large offered loads. This behavior verified the increase of deflections at higher offered loads. One other contribution of the performance evaluation in capturing the MSN performance characteristics was the measurement of the system throughput with different ratios of the internal link speed to external

link speed. The results of this evaluation showed the necessity of the speed advantage as we scale up the system.

## Acknowledgments

The author would like to thank Dr. Jonathan S. Turner for his valuable comments and discussions, and for continued support and encouragement in this work.

## References

- [1] G. Bianchi and J. S. Turner, "Improved queueing analysis of shared buffer switching networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 482-490, Aug. 1993.
- [2] R. G. Bubenik and J. S. Turner, "Performance of a broadcast packet switch," *IEEE Transactions on Communications*, vol. 37, no. 1, pp. 60-69, Jan. 1989.
- [3] A. K. Choudhury and N. F. Maxemchuk, "Effect of a finite reassembly buffer on the performance of deflection routing," *Proceedings of IEEE ICC '91*, pp. 1637-1646, May 1991.

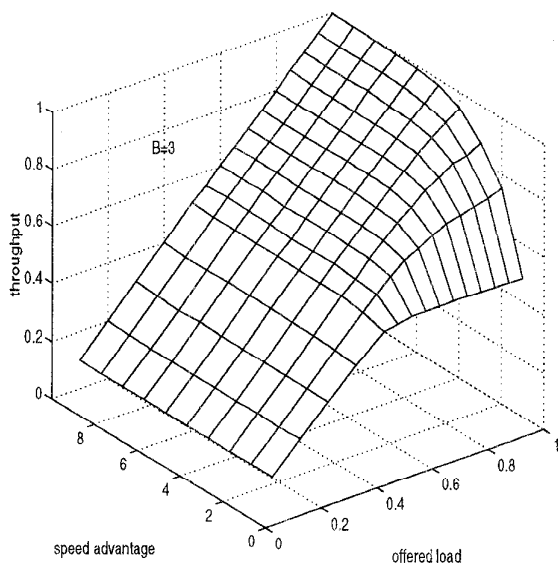


Figure 7: The effect of speed advantage ( $s$ ) on the throughput ( $T$ ) of the network when offered load ( $\rho$ ) varies ( $n = 16$ ).

- [4] A. K. Choudhury and V. O. K. Li, "Performance analysis of deflection routing in the Manhattan street network," *Proceedings of IEEE ICC'91*, pp. 1659-1665, May 1991.
- [5] P. Coppo, M. D'Ambrosio, and R. Melen. "Optimal cost/performance design of ATM switches," *Proceedings of IEEE INFOCOM 92*, pp. 446-458, May 1992.
- [6] A. G. Greenberg and J. Goodman, "Sharp approximate models of deflection routing on mesh networks," *IEEE Transactions on Communications*, vol. 41, no. 1, pp. 210-223, Jan. 1993.
- [7] Y. C. Jenq, "Performance analysis of a packet switch based on a single-buffered banyan network," *IEEE Journal on Selected Areas in Communications*, vol. SAC-1, no. 6, pp. 1014-1021, Dec. 1983.
- [8] N. F. Maxemchuk, "The Manhattan-street network," *Proceedings of IEEE GLOBECOM '85*, pp. 255-261, Nov. 1985.
- [9] N. F. Maxemchuk, "Routing in the Manhattan-street network," *IEEE Transactions on Communications*, vol. COM-35, no. 5, pp. 503-512, May 1987.
- [10] N. Mirfakhraei, "Performance analysis of a large-scale switching system for high-speed ATM networks," *Proceedings of IEEE Global Telecommunications Conference, GLOBECOM '94*, San Francisco, Calif., vol. 1, pp. 145-149, Nov. 27 - Dec. 1, 1994.
- [11] N. Mirfakhraei, "Design of a buffered switch for a gigabit ATM switching network," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 1, pp. 11-18, Jan. 1995.
- [12] N. Mirfakhraei, "Design and analysis of an ATM switch based on a Manhattan street network," *Washington University, Computer and Communication Research Center technical report*, WUCCRC-94-16, 1994.
- [13] T. Szymanski and S. Shaikh, "Markov chain analysis of packet-switched banyans with arbitrary switch sizes, queue sizes, link multiplicities and speedups," *Proceedings of IEEE INFOCOM '89*, Apr. 1989.
- [14] J. S. Turner, "Design of a broadcast packet switching network," *IEEE Transactions on Communications*, vol. 36, no. 6, pp. 734-743, June 1988.
- [15] J. S. Turner, "An optimal nonblocking multicast virtual circuit switch," *Proceedings of IEEE INFOCOM '94*, pp. 298-305, Jun. 12-16, 1994.