

ABSTRACT

Title of Thesis: ON ROUTING AND PERFORMANCE EVALUATION
OF BUFFERED SPARSE CROSSBAR CONCENTRA-
TORS

Degree candidate: Rahul Ratan

Degree and year: Master of Science, 2002

Thesis directed by: Professor A. Yavuz Oruç
Department of Electrical and Computer Engineering

We investigate the routing and performance of sparse crossbar packet concentrators under a buffered network model. The concentration property in packet switching concentrators is defined with respect to packets instead of input/output ports. This allows such concentrators to function as generalized connectors (with some constraints). This altered functionality for a packet concentrator over its circuit switched counterpart translates into differences in performance measures like complexity and delay.

A model for constructing sparse crossbar packet switching concentrators with optimal cross point complexity has been introduced in literature. We use this construction to model the performance of a sparse crossbar packet concentrator and

relate performance measures to its complexity, connectivity and buffer requirements.

In this thesis, we address issues of routing and performance evaluation over such optimal sparse crossbar fabrics, in particular their relation to complexity and buffer requirements. We present an analysis of the packet loss suffered in such concentrators when excess packets are dropped. We go on to analyze the best performance possible when packets are stored and serviced in FIFO order. These results lead us to formulate a routing algorithm which tries to emulate the best case performance on the sparse crossbar. We present theoretical and simulation results for the best case performance and the algorithm. We find that the algorithm is efficient and allows concentration to be done with negligible loss of performance on the sparse crossbar.

ON ROUTING AND PERFORMANCE EVALUATION
OF BUFFERED SPARSE CROSSBAR
CONCENTRATORS

by

Rahul Ratan

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2002

Advisory Committee:

Professor A. Yavuz Oruç, Chair
Professor Richard La
Professor Charles. B. Silio

© Copyright by

Rahul Ratan

2002

DEDICATION

To my parents and my sister for their constant love and support.

ACKNOWLEDGMENTS

I am grateful to my advisor, Professor A. Y. Oruç for his advice, support and encouragement in both academic and personal matters. I thank Dr. Richard La and Dr. Charles. B. Silio for agreeing to serve on my committee and review the thesis. My special thanks go to Manish Shukla for his valuable suggestions, comments and friendship.

TABLE OF CONTENTS

List of Figures	vii
1 Introduction	1
1.1 Background	1
1.1.1 Packet-Switched Concentrators	1
1.1.2 Concentration in Network Switching	4
1.1.2.1 Traffic Aggregation at the Outputs	4
1.1.2.2 The Knockout Concept	5
1.2 Motivation	6
1.3 Contributions	8
1.4 Organization of the Thesis	9
2 Buffered Concentrators	10
2.1 Buffered Network Model	10
2.1.1 Concentrator Model	11
2.2 Buffered Concentrator Complexity Bounds and Construction	13
2.3 Structure of the Concentrator Construction	15

3	Packet Loss Analysis	19
3.1	Input-Output Model	19
3.1.1	Routing Schemes	21
3.1.1.1	Case I (Instantaneous Routing)	22
3.1.1.2	Case II (Routing with Finite Delay)	22
3.1.1.3	Case III (Randomized Routing)	23
3.2	Packet Loss Analysis	24
3.2.1	Case I: Deterministic Routing with no Delay (Instantaneous Routing)	24
3.2.2	Case II: Deterministic Routing with Finite Delay	30
3.2.3	Case III: Randomized Routing	34
4	Input Queued Concentrator	40
4.1	Queuing Model	41
4.1.1	System and Queue Contents	44
4.1.2	Probability Distributions and Tail Probabilities	49
4.2	Theoretical and Simulation Results	52
4.3	Outline of Analysis for Delayed Service	58
5	Algorithm for Concentration	63
5.1	Packet Concentrator as a Shared Buffer	63
5.2	Routing on Sparse Crossbar Structures	65
5.3	The Routing Algorithm	69

5.3.1	A Block-Packing Model	70
5.4	An Approximate Analysis	75
5.5	Simulation Results	78
6	Conclusions and Future Work	80
	Appendix A	83
A-1	Proof	83
	Appendix B	85
B-1	Properties of Probability Generating Functions	85
B-2	Probability Generating Functions for S and Q	86
B-3	Steady-State Probability Distribution	89
	Bibliography	90

LIST OF FIGURES

1.1	A circuit switched concentrator.	2
1.2	A packet switched concentrator with concentration capacity $c = 4$	3
1.3	An abstract reference model for space-division switches [1].	5
2.1	The buffered network model [2].	12
2.2	Full capacity buffered concentrators [2].	16
2.3	Sectioning of inputs in a $Q(n, m, v, w)$	18
3.1	Concentrator model with $v = 3$ and $w = 4$	20
3.2	Case I: Deterministic routing with no delay (Instantaneous routing).	24
3.3	Probability of loss, p_L , instantaneous routing, $nv = 50$	27
3.4	Upper bound for p_L , instantaneous routing, $nv = 50$	28
3.5	Lower bound for p_L , instantaneous routing, $nv = 50$	29
3.6	Throughput, ρ , instantaneous routing, $nv = 50$	30
3.7	Upper bound for ρ , instantaneous routing, $nv = 50$	31
3.8	Lower bound for ρ , instantaneous routing, $nv = 50$	32
3.9	Case II: Deterministic routing with finite delay.	33
3.10	Case III: Randomized (Distributed) routing.	34

3.11	Loss ratio l_r , $\lambda = 1$	36
3.12	Loss ratio l_r , $\lambda = 2$	37
3.13	Loss ratio l_r , $\lambda = 3$	37
3.14	Loss ratio l_r for different λ , $w = 10$	38
4.1	Late arrival model.	43
4.2	Mean queue length, \bar{Q} , at various output capacities (mw)	54
4.3	Calculated queue length variance, $\text{Var}(Q)$	55
4.4	Mean queue length variation with nv , $mw = 4$	55
4.5	Buffer size (Q_0) vs. input load (ρ) for a fixed probability of loss	56
4.6	Probability of cell loss, $\Pr(Q \geq X)$ for $mw = 1, 3$	57
4.7	Mean queue delay, \bar{W}_q , at various output capacities (mw)	59
4.8	Tail probability of cell queuing delay	60
5.1	Blocking due to sparse connectivity	66
5.2	Cell blocking in a $Q(n, m, v, w)$	68
5.3	Block-packing analogy.	73
5.4	Mean queue length, \bar{Q} for a $Q(8, 6, 1, 1)$	79
5.5	Mean queue delay, \bar{W}_q for a $Q(8, 6, 1, 1)$	79

Chapter 1

Introduction

1.1 Background

1.1.1 Packet-Switched Concentrators

Concentration is typically used to refer to the process of combining many low-speed input lines into a few high-speed output lines. A circuit switched concentrator is defined as a device with n inputs and m outputs ($m \leq n$) that can connect any k inputs ($k \leq m$) to some k outputs. No two inputs share an output and vice-versa.

A particular connection pattern of a 6 input, 5 output circuit switched concentrator is shown in Figure 1.1. Note that there is a one-to-one correspondence between the inputs and the outputs to which they are concentrated. Concentration occurs as long as an input is connected to an output, irrespective of the address of the output. This observation enables us to view an n input, m output concentrator as a $n \times m$ switch in which an active input connects to any available output without consideration for its address. This in a sense resembles deflection routing.

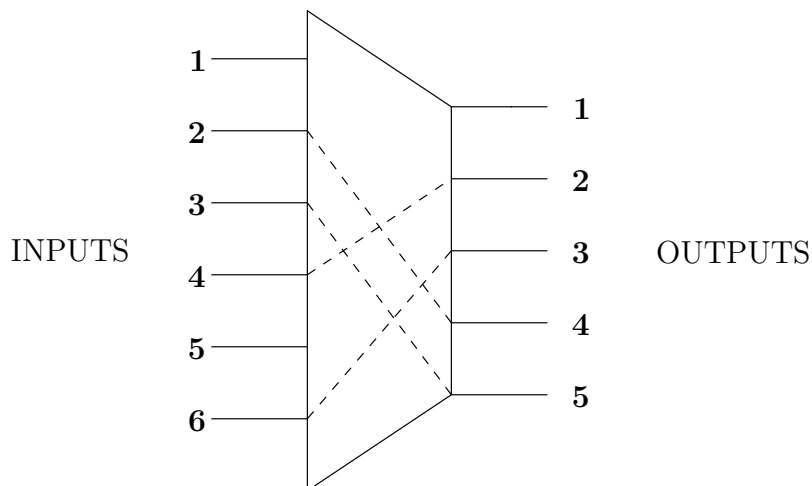


Figure 1.1: A circuit switched concentrator.

A packet switched concentrator, on the other hand, is a device with n inputs and m outputs which can route up to c packets to its outputs, c is called the concentration capacity. Thus, the maximum number of packets which can be concentrated at a time is c . In the following discussion, the time taken for concentrating all the possible “concentratable” packets from the inputs is called a *cycle*. A packet may not be “concentratable” when more than c packets arrive in a cycle or there is internal blocking in the concentrator fabric. Usually it is assumed that the input and output ports have a packet rate of 1 packet per cycle and $c = m$.

Since concentration is now defined with respect to packets transferred to the outputs, there is no restriction regarding the one-to-one correspondence between the inputs and the outputs to which they are concentrated. In fact, in a packet concentrator such a restriction would cause blocking, severely limiting the concentration capability and reducing the bandwidth flexibility gained by the statistically

multiplexed packets. A packet concentrator is shown in Figure 1.2.

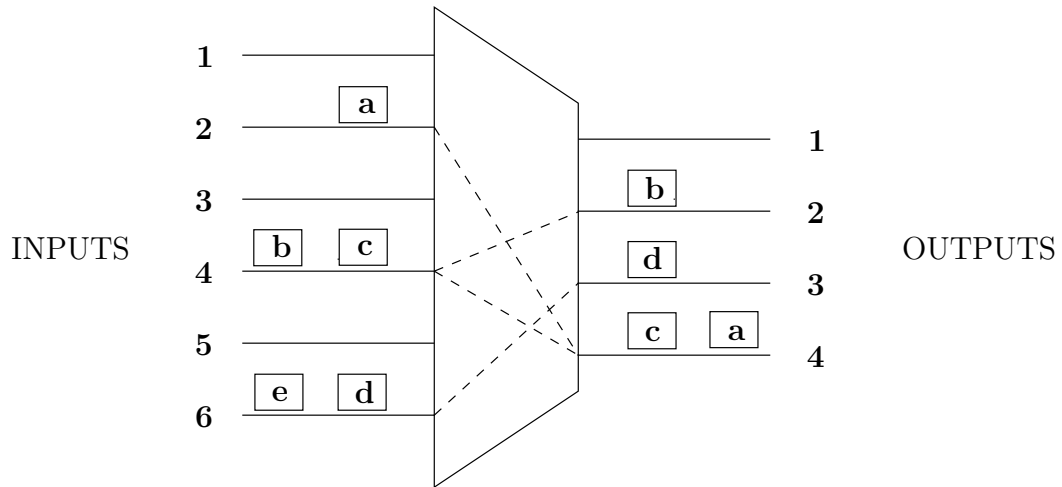


Figure 1.2: A packet switched concentrator with concentration capacity $c = 4$.

In a cycle, as shown in Figure 1.2, a single input can connect to multiple outputs (input 4 sends packets **b** and **c** to outputs 2 and 4 respectively), a single output can get packets from multiple inputs (output 4 gets packets **a** and **c** from inputs 2 and 4 respectively) and one input can connect to a single output as well (input 6 sends packet **d** to output 3). The maximum number of packets an input can potentially send (or an output can receive) in a single cycle is only limited by the input (output) packet rate. In Figure 1.2 this rate is equal to 2 packets per unit time at both the inputs and the outputs. The actual number of packets transferred by an input, however, depends on the concentration capacity, the distribution of packets at the inputs, the crosspoint pattern in the concentrator fabric (which in the optimal case is sparse and determined by the input/output rates and the capacity) and the algorithm used to assign packets to the outputs. In Figure 1.2,

the internal connectivity of the concentrator fabric is assumed to be such that all the outputs together cannot accommodate more than 4 packets in a unit time even though the output rate at each output is 2 packets per unit time. Thus, packet **e** is not “*concentratable*” i.e. input 6 cannot send packet **e** in this cycle as the capacity of 4 packets has already been met.

1.1.2 Concentration in Network Switching

1.1.2.1 Traffic Aggregation at the Outputs

The need for concentration arises from a lack of sufficient bandwidth to carry all of the traffic at once over a network. In circuit switched communication networks concentrators were initially used to combine and redirect calls, later this functionality was extended to include data when data began to be transferred via switched public networks. The advent of switched data networks led to the development of routers and switches meant specifically for data. It was realized that switching basically consisted of distributing and then concentrating traffic. Thus, conceptually a switch (especially a space division switch) can be expressed as a combination of distributors and concentrators. Distributors serve to separate the traffic streams at an input to the different outputs and concentrators collect the different streams coming to an output. Figure 1.3 illustrates this conceptually. In fact, even the set of cross points which access an output in a crossbar can be considered to constitute a concentrator, especially when they function as an arbiter in a centralized con-

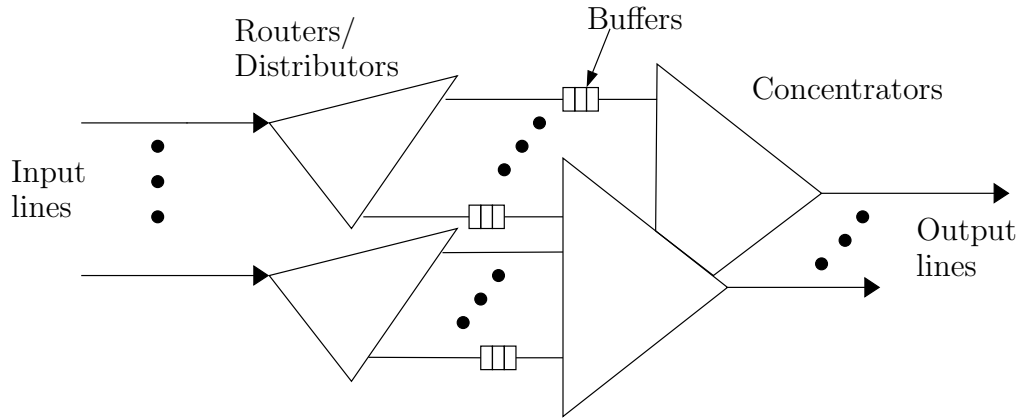


Figure 1.3: An abstract reference model for space-division switches [1].

attention resolution scheme [1]. Many switches explicitly incorporate concentrators and distributors as physical devices to achieve this function. The Multinet packet switch in [3], [4] and the circuit switch in [5] are examples of such switch designs.

1.1.2.2 The Knockout Concept

Concentrators are used extensively in high-speed packet switching architectures where they serve another important function apart from simply aggregating traffic. It is known that output buffered and shared memory switches achieve the best throughput/delay performance but the speedup required for routing presents a bottleneck [6]. This problem is solved in many proposed architectures [7], [8], [9], [10], [11], [12] by using packet concentration which reduces the requirement of both speedup and simultaneous buffer requirements.

The logic behind using concentrators within switches is that at any time it is highly unlikely that more than a certain number of inputs (say k) will have packets

addressed to a single output. Thus, up to k packets can be concentrated at each output and if in the unlikely event more than k packets arrive, packets can be either be buffered for later transmission or discarded (“knocked out”) while maintaining a low probability of loss. This is in general referred to as the knockout principle and is the basis for building output queued switches without requiring significant speedup [12].

1.2 Motivation

There is an extensive body of work on the analysis of circuit switched concentrators. Many bounds have been derived for quantities like crosspoint complexity and routing delay [13], [14] and the structures required achieving for such bounds are well-understood in terms of their complexity.

Despite the significance of concentration in packet switching architectures, the literature on it is limited. Work related to packet concentrators has mainly been done on two tracks:

1. Performance Analysis Related to Switching

Recent efforts have refined the knockout concept [15] and introduced more efficient switching fabrics [16] with similar packet loss properties, but all the analysis has been done in the context of using concentrators as part of a switch based on some form of the knockout principle. Research has focused mainly on determining the queuing delay, throughput and other

performance measures for overall switch structures of which concentration forms only one part [17], [18]. In a concentrator, the incoming packets do not need to be routed to a specific output, concentration is carried out as long as they are routed to some output, and in this respect, concentrators fundamentally differ from switches and this affects basic measures like the probability of loss and average packet delay. Moreover, concentration is an important operation in routing traffic and needs to be analyzed independent of its role in a network switch. A particular solution to multiple server queues with application to ATM concentrators has been found in [19], but no study has been done of the performance of a packet concentrator by itself, which takes into account different input, switching and output speeds and the related connection pattern.

2. Construction of Concentrator Structures

Explicit constructions for packet concentrators are few, and even these relate to specific topologies like the knockout concentrator [12], or shared buffer concentrators [3]. There has been little attempt to build a theoretical basis for arriving at packet concentrator structures in a methodical way. A buffered network model suitable for constructing a specific class of such concentrators viz sparse crossbars has been introduced only recently by Gündüzhan and Oruç [2]. This model considers bulk arrivals and difference in input and output packet rates and switching speeds, leading to crossbar concentrator structures that minimize crosspoint complexity. Since the

construction is in terms of crosspoints it can be used as a basis for modeling even non-optimal physical implementations like the knockout concentrator constructions of [15] and [20]. But the authors of [2] have left the issue of quantifying the packet loss open when the concentration capacity c is exceeded. Since the model is for minimum complexity crossbars, the connection pattern is sparse and non uniform. Thus, contention amongst inputs for an output can develop due to non availability of crosspoints. In such a situation the specific scheme used for concentration of packets and buffering of excess packets greatly affect the performance, these issues are also left unaddressed.

1.3 Contributions

In this thesis we present an analysis of the performance of sparse crossbar packet concentrators. This analysis is based on a theoretical model [2] which formulates explicit optimal constructions for packet concentrators with design parameters as number of inputs, number of outputs, input packet rates, output packet rates and concentration capacity c . Therefore, we are able to relate all these quantities and the structure to the performance of the concentrator. Firstly, we relate the degree of sparseness to packet loss by quantifying the packet loss (when excess packets are simply dropped) in two situations:

- (a) An arbiter/controller routes the packets to fully utilize the available out-

put capacity and crosspoints on the sparse crossbar.

- (b) The packets are sent randomly to any output without any control on the routing but the connection fabric is a full crossbar.

An analysis of the best possible performance of the concentrator when excess packets are buffered at the inputs is developed for independent arrivals and FIFO service discipline at the inputs. We show that this system is the same as a shared buffer implementation of the concentrator which can be mathematically represented as a multi-server discrete time queue. An algorithm is developed which emulates the shared buffer to the maximum extent possible under the sparse connection constraint using a round robin token passing scheme. The performance of both the shared buffer and the algorithm is evaluated via simulations in C. Comparison of their performance suggests that the algorithm is able to emulate the shared buffer very closely.

1.4 Organization of the Thesis

The rest of the thesis is as follows. In Chapter 2 the buffered packet concentrator model is described. Chapter 3 presents the loss analysis for the case when excess packets are dropped. In Chapter 4, simulation results are discussed and analysis is given for the case when excess packets are buffered. In Chapter 5 the algorithm for routing the packets over the concentrator is developed and results from its simulation are presented. Chapter 6 concludes the thesis.

Chapter 2

Buffered Concentrators

2.1 Buffered Network Model

The buffered network model in [2] is used to investigate the crosspoint complexity of networks with certain connection specifications between their inputs and outputs. The basic problem can be stated as follows: A collection of sets of callers, $C_i, 1 \leq i \leq p$ generate packets into buffers $X_i, 1 \leq i \leq p$ of size v which are then routed over a network to a collection of sets of receivers $R_i, 1 \leq i \leq q$ which consume these packets from buffers $Y_i, 1 \leq i \leq q$ of size w as shown in Figure 2.1. Assuming that the callers in any C_i can generate at most v packets/unit time and the receivers in any R_i can consume packets at least w packets/unit time, determine the crosspoint and/or buffer complexity of a network that can achieve a specified set of connections between the callers and receivers. The network portion of the model is composed of a set of switches (or equivalently crosspoints) and a set of buffers connected by some fixed topology. Its crosspoint complexity is given by the total number of crosspoints in all of its switches, and its buffer complexity

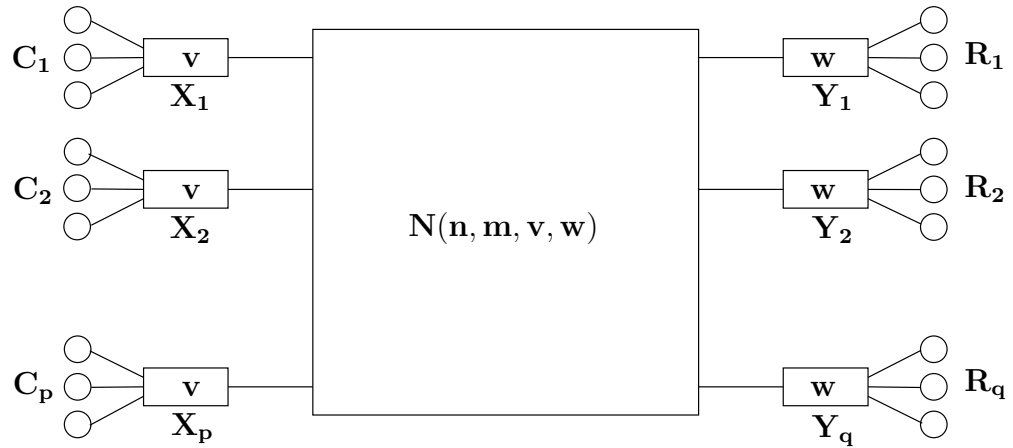
is given by the sum of the size (in terms of number of packets) of all its buffers.

As long as buffers are not overloaded, packets are delivered to their destinations. Buffers not only store packets but also redirect them to their destinations under the bounds of packet generation and consumption rates. For example, in Figure 2.1(b), it is easily verified that any seven of the eight packets at the inputs can be permuted to the seven buffer locations at the outputs assuming that the buffer locations are randomly accessible. For $v = w = 1$, we get the extensively studied space division networks like concentrators, generalizers, and rearrangeable and strictly nonblocking networks.

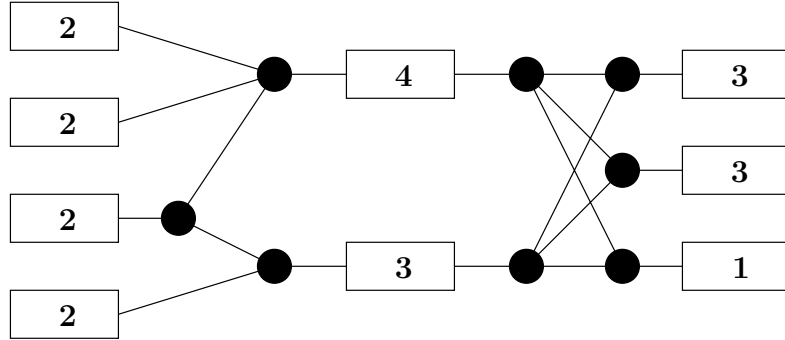
2.1.1 Concentrator Model

A buffered (n, m, v, w, c) -concentrator, denoted by $Q(n, m, v, w, c)$, is an n -input, m -output network with a buffer of size v at each input and a buffer of size w at each output. The parameters should satisfy $c \leq mw \leq nv$. In a $Q(n, m, v, w, c)$ any $p \leq c$ packets at any p input buffer locations can be routed to some p output buffer locations. When $c = mw$, the network is called a full capacity buffered concentrator, and is denoted by $Q(n, m, v, w)$. A special case of a buffered concentrator, called a buffered sparse crossbar concentrator arises when we restrict the number of switches between the input buffers and output buffers to 1.

It should be noted that the concept of input and output buffers can be viewed in two ways:



2.1(a): The buffered network model.



2.1(b): A buffered network. The dark circles correspond to the switches and the numbers in the rectangles represent the buffer sizes.

Figure 2.1: The buffered network model [2].

- (a) When the number of packets are fixed or the time is fixed with no new packets coming in (i.e. we are looking at a snapshot of the arrival process), we interpret the input and output buffers as memories which store v packets and w packets respectively.
- (b) In the dynamic case when new packets are constantly arriving, we interpret the input and output buffer parameters as the packet rates seen at the inputs and outputs. Thus the input packet rate, v packets per unit time, is the maximum number of packets which can arrive at an input in one unit of time (the unit of time chosen is arbitrary, we can scale the number of packet arrivals according to it). Similarly, the output packet rate, w packets per unit time, is the maximum number of packets which can arrive at an output in one unit of time.

2.2 Buffered Concentrator Complexity Bounds and Construction

The following lower bound on the crosspoint complexity of a sparse crossbar $Q(n, m, v, w, c)$ is derived in [2]:

Theorem 1 A buffered sparse crossbar $Q(n, m, v, w, c)$ has a crosspoint complexity:

$$|Q(n, m, v, w, c)| \geq \frac{m}{m - \lfloor \frac{c-1}{w} \rfloor} \left[n - \frac{w}{v} \left\lfloor \frac{c-1}{w} \right\rfloor - \frac{1}{v} + 1 \right], \quad \text{if } c/v \in \mathbb{Z}$$

For the full capacity case ($c = mw$), the bound in Theorem (1) becomes:

$$|Q(n, m, v, w, c)| \geq m \left(n - \frac{mw}{v} + \left\lceil \frac{w}{v} \right\rceil \right) \quad (2.1)$$

The bound is tight for the full capacity case. This is shown by constructing a concentrator which has crosspoint complexity equal to the right hand side of (2.1).

The concentrator is constructed as follows [2]:

- i. Connect each output disjointly to $\lfloor \frac{w}{v} \rfloor$ inputs. This requires $m \lfloor \frac{w}{v} \rfloor$ crosspoints and $n - m \lfloor \frac{w}{v} \rfloor$ inputs remain. Denote the set of inputs connected in this section by I_1 .
- ii. If $w/v \in \mathbb{Z}$, then go to step (iii). Otherwise, connect each output to a pair of the next $\frac{mw}{v} - m \lfloor \frac{w}{v} \rfloor + 1$ inputs in the following way ¹. Let $z = w - v \lfloor \frac{w}{v} \rfloor$, which is the minimum number of empty spaces at each output buffer after all packets coming from I_1 have been routed. Now suppose we divide a line of length mz into (open) intervals of length v , corresponding to the inputs to be connected to except the last one. Call these intervals A_i , $i = 1, \dots, mz/v$. We also divide the same line into (open) intervals of length z , corresponding to the outputs, and call them B_j , $j = 1, \dots, m$. Connect the input corresponding to A_i to the output corresponding to each B_j if $A_i \cap B_j \neq \emptyset$. Finally, connect the last input to all outputs which have only one input connected to them in this step. This step requires $2m$ crosspoints, and $n - \frac{mw}{v} - 1$ inputs remain. Denote the set of inputs

¹Note that $\frac{mw}{v}$ is an integer since $c = mw$ and $v|c$.

connected in this step by I_2 .

- iii. Connect each of the remaining inputs to all outputs, and denote these inputs by I_3 .

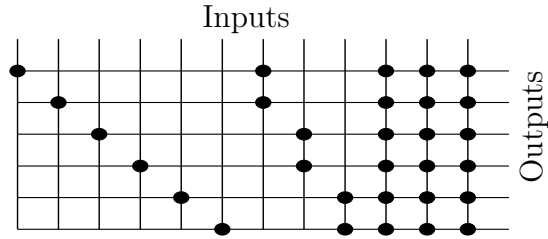
The procedure described requires $m \left(n - m \lfloor \frac{w}{v} \rfloor + \lfloor \frac{w}{v} \rfloor \right)$ crosspoints if $w/v \in \mathbb{Z}$, and $m \left(n - \frac{mw}{v} + 1 + \lfloor \frac{w}{v} \rfloor \right)$ crosspoints otherwise. But in both cases these expressions match the lower bound in (2.1). This concentrator construction is illustrated in Figure 2.2 for some values of n, m, v and w .

2.3 Structure of the Concentrator Construction

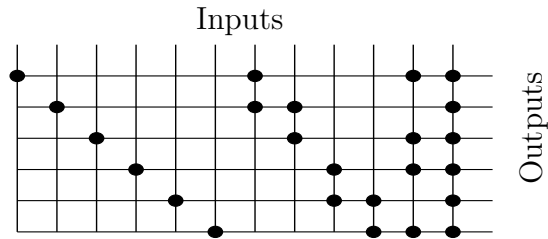
We make some relevant observations about the structure of the concentrator fabric obtained from the procedure described in Section 2.2. It will be important to keep these properties in mind when we formulate a scheduling algorithm for the concentrator in Chapter 5.

1. Each output is connected to an equal number of inputs. This implies that the number of crosspoints seen in each row of the constructions seen in Figure 2.2 is equal. We call this the output port connectivity and denote it by α . From the formula for the total number of crosspoints given in Section 2.2 we get:

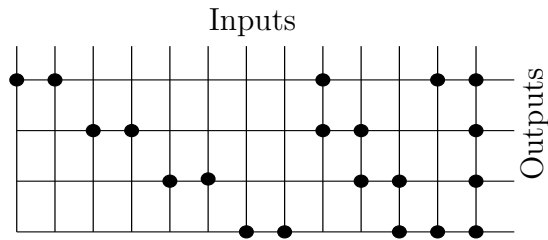
$$\alpha = \begin{cases} n - m \lfloor \frac{w}{v} \rfloor + \lfloor \frac{w}{v} \rfloor & \text{if } w/v \in \mathbb{Z}, \\ n - \frac{mw}{v} + 1 + \lfloor \frac{w}{v} \rfloor & \text{if } w/v \notin \mathbb{Z} \end{cases} \quad (2.2)$$



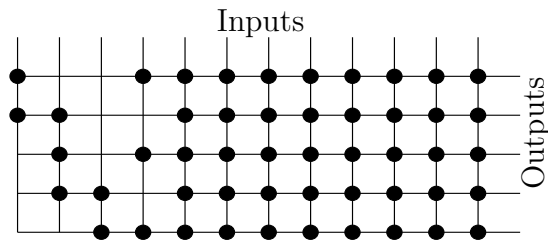
2.2(a): $n=12, m=6, v=2, w=3, z=1$



2.2(b): $n=12, m=6, v=3, w=5, z=2$



2.2(c): $n=13, m=4, v=4, w=11, z=3$



2.2(d): $n=12, m=5, v=5, w=3, z=3$

Figure 2.2: Full capacity buffered concentrators [2].

2. Based on the crosspoint connection pattern the inputs can be divided into three distinct sections, each corresponding to one step in the construction procedure.

- The slim (disjoint) section: This is the group of inputs corresponding to region I_1 in Section 2.2. This section occurs in the fabric only when $\lfloor \frac{w}{v} \rfloor \geq 1$, i.e., when the output port rate w is greater than or equal to the input rate v . Here each input is connected to at most one output, and no two outputs share an input. Thus, the connection pattern in this region allows an input to access only one output.
- The ladder section: This is the group of inputs corresponding to region I_2 in Section 2.2. This section occurs in the fabric only when $\frac{w}{v} \notin \mathbb{Z}$ i.e. w is not a multiple of v . In this section outputs can share inputs, so the structure is not fully disjoint, but it is also not fully connected. We can view this region as a transition between the sparse, fully disjoint slim section and the fully connected fat section (see next item).
- The fat section: This is the group of inputs corresponding to region I_3 in Section 2.2. This section is a fully connected crossbar, so every input in this section can connect to every output. Hence, there is no blocking constraint for these inputs.

These three sections are outlined for a $Q(12, 6, 3, 5)$ in Figure 2.3. In the rest of the discussion we refer to inputs in the slim section as slim inputs,

inputs in the ladder section as ladder inputs and inputs in the fat section as fat inputs.

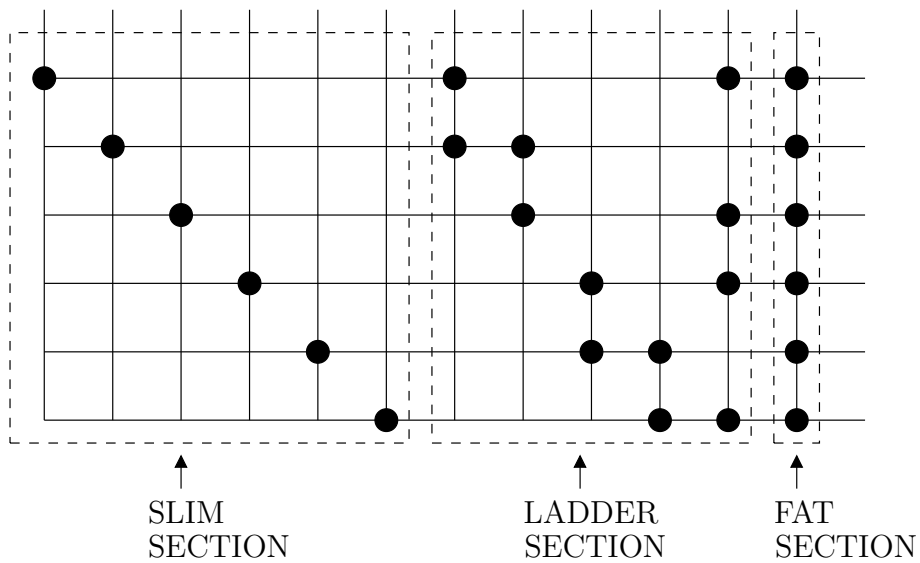


Figure 2.3: Sectioning of inputs in a $Q(n, m, v, w)$

Chapter 3

Packet Loss Analysis

In this chapter we present an analysis of the packet loss in the concentrator.

3.1 Input-Output Model

There are n input ports and m output ports in the concentrator. We consider a slotted discrete-time structure for the packet arrivals, with time divided into equal length time slots. All packets are also of equal length (called cells in ATM terminology). We assume bulk arrivals throughout the rest of the analysis. In consonance with the model given in Section 2.1 and the dynamic case interpretation of the input buffers in item (b) on Page 13, there are up to v cell arrivals in a time slot. The (up to v) new arrivals in a slot can in concept be interpreted as constituting a new batch arriving in that slot. More specifically, the probability that any position out of the maximum possible v positions in a batch is occupied by a cell is p . The probability that at input j ($1 \leq j \leq n$), in slot t ($t = 0, 1, 2, \dots$), position i in a batch ($1 \leq i \leq v$) is occupied by a cell, is equal to $p \forall j, i, \text{ and } t$.

In other words, the probability of a particular position being occupied in a batch is statistically independent of its position, subsequent and previous batches and the input port at which it enters the concentrator.

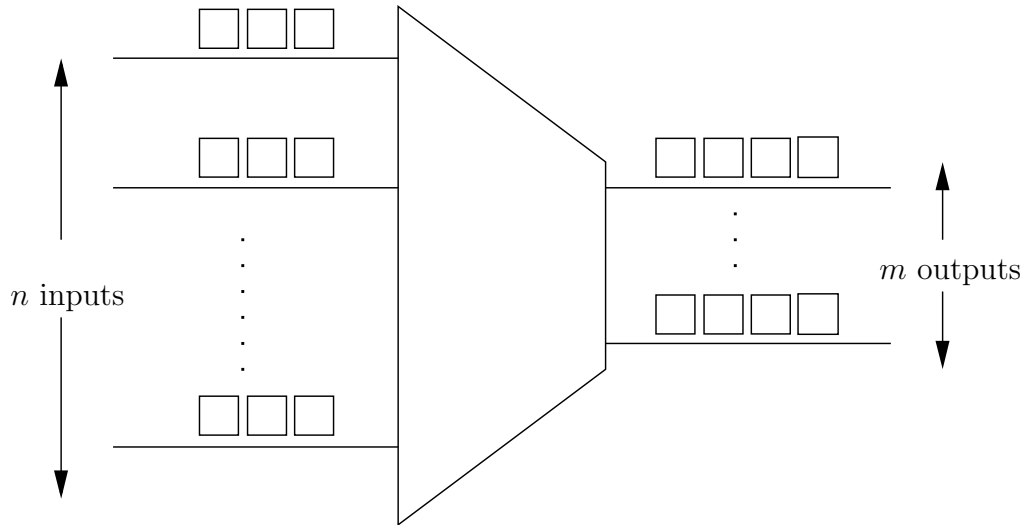


Figure 3.1: Concentrator model with $v = 3$ and $w = 4$.

Any internal crosspoint in the fabric has to route up to $\min\{v, w\}$ cells in one time slot and each output can accept up to w cells in the same time (see Figure 3.1). Thus, the crosspoints have to be able to switch w times per slot as up to m accesses to an output may be required in a single slot. We assume a full capacity ($c = mw$) sparse crossbar $Q(n, m, v, w)$ with a structure described in Section 2.2, unless stated otherwise. We also define the concentration ratio (λ) as the ratio between the maximum number of input cells to the maximum number of cells concentrated in a single time slot. Hence, $\lambda = nv/mw$ ¹.

This model is analogous to input smoothing described in [6], where frames of b

¹ $\lambda \geq 1$ as $mw \leq nv$.

cells are formed and then demultiplexed to b inputs by the internal fabric switches at $1/b$ times the input rate. However, unlike the model in [6], in this model, the crosspoints can switch multiple cells ($\min\{v, w\}$) in a time slot and so, multiple (up to v) cells in a particular slot enter the concentrator on a single input rather than multiple inputs. Clearly, this reduces the crosspoint complexity.

3.1.1 Routing Schemes

As stated earlier, the incoming cells do not need to be routed to a specific output, concentration is carried out as long as they are pushed to some output. Even without output specific routing, some routing control needs to be applied to prevent too many inputs from connecting to an output, leading to overflow and loss of cells while other outputs may have empty space. In the sparse crossbar fabric, certain inputs (in the slim region of the crossbar) have access to lesser number of outputs as compared to other inputs (in the fat region of the crossbar). The cells have to be assigned to outputs in a fashion which ensures that all inputs get a fair share of the available bandwidth i.e. the lower connectivity of some inputs should not affect the availability of output space for them. Thus, it is to be expected that routing strategy also affects the cell loss probability. With these assumptions in place, we consider three concentration schemes based on the routing strategy employed and the delay incurred in calculating the routes. The first two cases consider schemes which route cells taking into account the structure of the crossbar fabric and hence,

are referred to as deterministic schemes. In the third case we send cells at random to the outputs irrespective of whether they can accept more cells or not over a fully connected crossbar, hence we call it randomized routing. The different cases are described below:

3.1.1.1 Case I (Instantaneous Routing)

The routing is assumed to be very fast so that the time taken to route all the incoming cells at all the inputs in one slot is negligible compared to the frame period. Hence, we call this instantaneous routing. The concentrator fabric can be viewed as having pipeline buffers to store the cells while calculating their routes and then routing them within one time slot. In Figure 3.2, the pipeline buffers (as dashed boxes) are shown outside the concentrator for clarity. The parallel transfer of cells to the buffers indicates the fast routing speed. Alternatively, it can be assumed that the concentrator is self-routing, similar to those described in [12], [18]. Due to fast routing, the incoming cells in the subsequent time slot do not experience any blocking at the concentrator inputs and thus there is no additional loss due to time taken in routing cells.

3.1.1.2 Case II (Routing with Finite Delay)

The route calculation is slower in this case. The input buffer storing the cells while routes are calculated has a capacity equal to v and it is assumed that it takes an additional t slots to process and route all the cells arriving in one slot,

see Figure 3.9. Since there is no buffer space to store any new cells arriving during the time cells in the input buffer are being processed, all the cells at the inputs in the next t time slots are lost. This sort of scenario can occur in routing over large concentrators (large values of m and n) with centralized routing control because if the routing algorithm is not efficiently scalable then routing the cells may require more than one time slot.

3.1.1.3 Case III (Randomized Routing)

The inputs route the cells randomly to the outputs independent of whether an output port is full to its capacity or not. In other words, there is no deterministic control over routing the cells, and hence we call it randomized routing (see Figure 3.10). Under this assumption, an input port sends cells with equal probability to the output ports to which it is connected. Since full randomization of routing cannot occur in the optimized crossbar construction, we assume a full crossbar fabric in this case, unlike in the previous two cases with deterministic routing. This case is analyzed to get an estimate of how adversely the cell loss is affected by simply pushing cells to the outputs. Since we are using a full crossbar construction, we hope to get an estimate of the tradeoff between the crosspoint complexity and ease of routing for achieving a certain probability of loss by comparing this loss with that in the previous two cases. Also, it will be good to check how packet loss will be impacted by randomly setting crosspoints over an optimized concentrator fabric.

3.2 Packet Loss Analysis

We now present the packet loss analysis for the three cases discussed in Section 3.1.1.

3.2.1 Case I: Deterministic Routing with no Delay (Instantaneous Routing)

Key assumption: The cell routing inside the concentrator is instantaneous or it occupies a very small fraction of the slot.

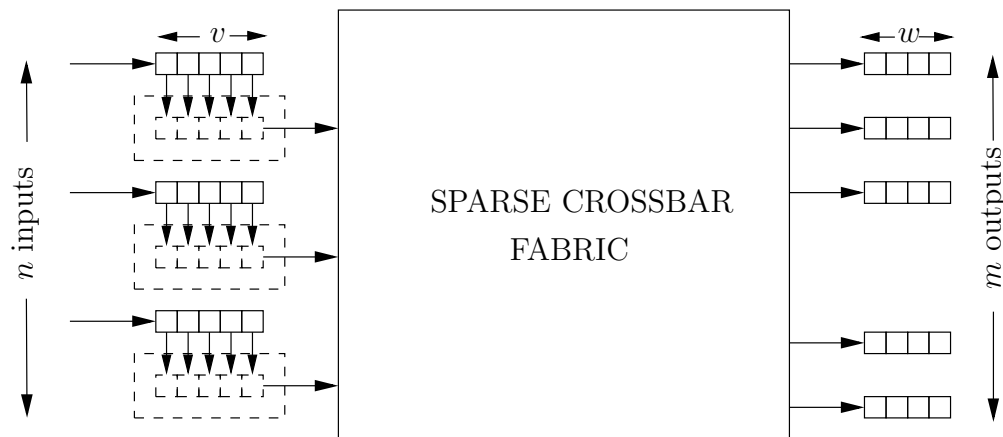


Figure 3.2: Case I: Deterministic routing with no delay (Instantaneous routing).

Let the random variable N_i represent the total number of cell arrivals at the i^{th} input in one slot. Then by the preceding assumptions:

$$\Pr[N_i = k] = \binom{v}{k} p^k (1-p)^{v-k} \quad (3.1)$$

For the combined process at all the n inputs, define $N_T = \sum_{i=1}^n N_i$. Then,

$$\Pr[N_T = j] = P(j) = \binom{nv}{j} p^j (1-p)^{nv-j} \quad (3.2)$$

Since the routing is instantaneous and the capacity of concentration is mw , we get probability of cell loss (p_L):

$$\begin{aligned} p_L = \Pr[N_T \geq mw + 1] &= \sum_{j=mw+1}^{nv} P(j) \\ &= \sum_{j=mw+1}^{nv} \binom{nv}{j} p^j (1-p)^{nv-j} \end{aligned} \quad (3.3)$$

A simple application of the Markov inequality

$$\Pr[N_T \geq x] \leq E[N_T]/x \quad (3.4a)$$

gives a bound on p_L :

$$p_L \leq nvp/(mw + 1) \quad (3.4b)$$

It is relevant to note that this inequality gives meaningful results only for low concentration ratios (nv/mw) and/or low input rate (p) because for other cases, the right side is greater than one and the inequality is trivial. A more refined version of the inequality is obtained from the Chernoff Bound [21]:

$$\begin{aligned} \Pr[N_T \geq mw + 1] &\leq e^{-s(mw+1)} E[e^{sN_T}] \\ &= e^{-s(mw+1)} \sum_{j=1}^{nv} \binom{nv}{j} e^{sj} p^j (1-p)^{nv-j} \\ &= e^{-s(mw+1)} (1-p + pe^s)^{nv}, \quad s > 0 \end{aligned} \quad (3.5)$$

Minimizing the expression on the right with respect to s :

$$\begin{aligned} p_L &\leq \exp \left\{ nv \left[\mathcal{H} \left(\frac{a}{nv} \right) + \left(\frac{a}{nv} \right) \ln p + \left(\frac{nv-a}{nv} \right) \ln(1-p) \right] \right\} \\ &= \left[\frac{nv p}{a} \right]^a \left[\frac{1-p}{(1-a/nv)} \right]^{nv-a} \end{aligned} \quad (3.6)$$

where $a = mw + 1$ and $\mathcal{H}(p) = -p \ln p - (1-p) \ln(1-p)$ is the entropy function.

However, the above inequality is valid only for $p < (mw + 1)/nv$

We can get tighter bounds by using some specific properties of the binomial function [21]. This leads to the following bounds on the probability of loss:

$$p_L \leq \min \left\{ \frac{a(1-p)}{a(1-p) - (nv-a)p} \binom{nv}{a} p^a (1-p)^{nv-a}, \quad 1 \right\} \quad (3.7a)$$

$$p_L \geq \binom{nv}{a} p^a (1-p)^{nv-a}, \quad a = mw + 1 \quad (3.7b)$$

Note that the above probability of loss is equal to the fraction of slots in which cell loss occurs. The plots of the exact probability of loss and the upper and lower bounds are given in Figure 3.3, Figure 3.4 and Figure 3.5 respectively. This probability of loss is not the ratio of number of cells lost to the total number of incoming cells. That is better represented by the loss ratio or its complementary quantity, throughput. This is the probability of loss as defined in other papers in the literature [6].

Let number of cells lost per slot = N_{lost}

$$\mathbb{E}[N_{lost}] = \sum_{j=mw+1}^{nv} (j - mw) \binom{nv}{j} p^j (1-p)^{nv-j} \quad (3.8)$$

Since $1 \leq j - mw \leq (nv - mw)$, from the above equation we get

$$p_L \leq \mathbb{E}[N_{lost}] \leq (nv - mw)p_L \quad (3.9)$$

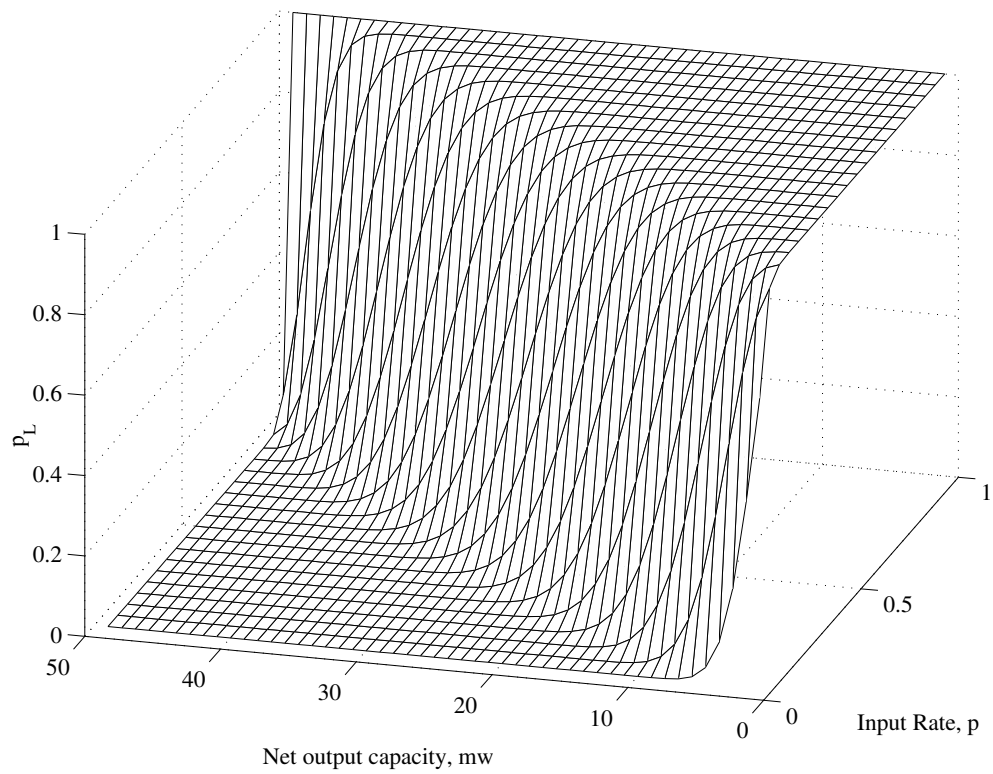


Figure 3.3: Probability of loss, p_L , instantaneous routing, $nv = 50$.

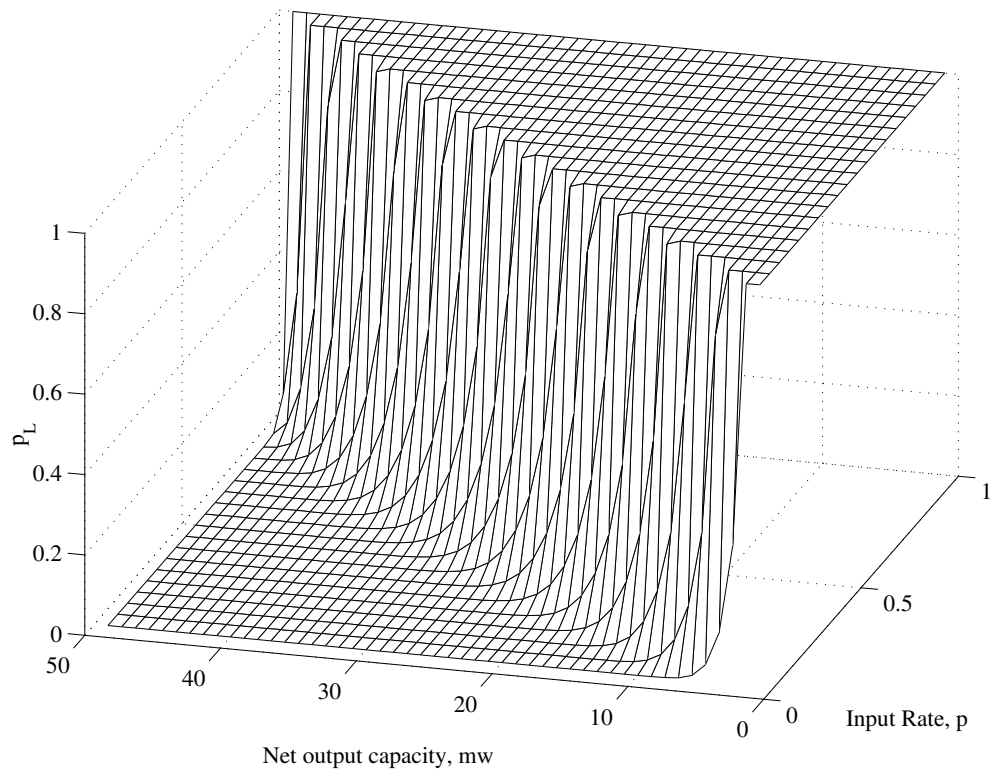


Figure 3.4: Upper bound for p_L , instantaneous routing, $nv = 50$.

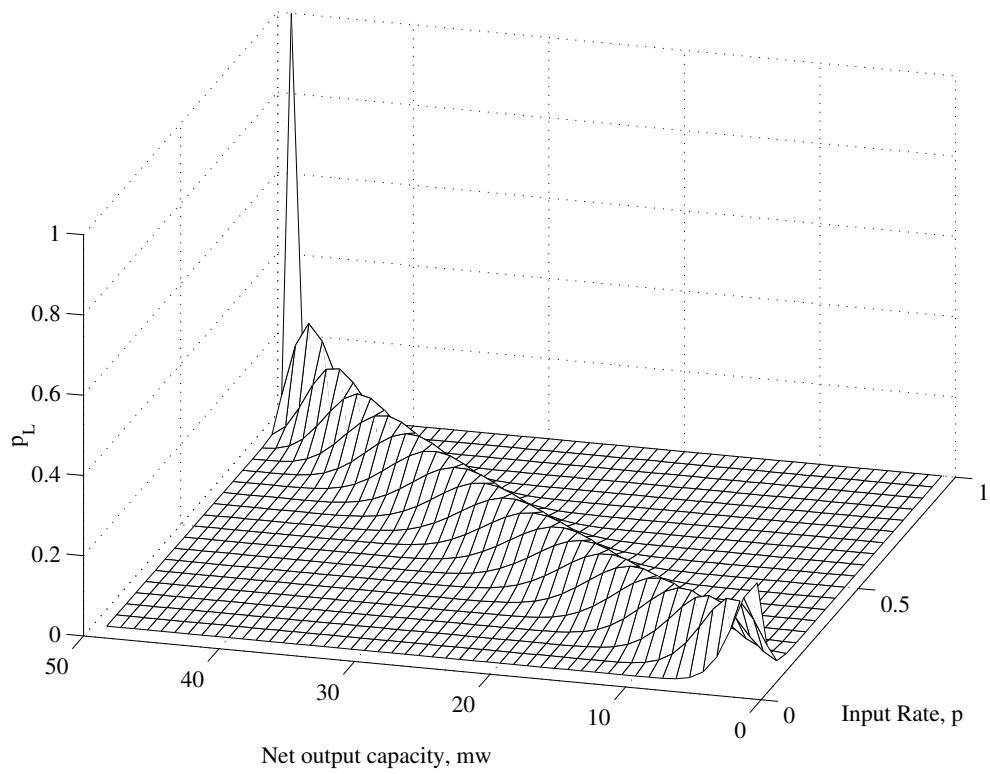


Figure 3.5: Lower bound for p_L , instantaneous routing, $nv = 50$.

where p_L is the probability of loss defined earlier.

Throughput, $\rho = 1 - E[N_{lost}]/E[N_T]$. Therefore, from (3.9)

$$\max \left\{ 0, 1 - \frac{p_L(nv - mw)}{nvp} \right\} \leq \rho \leq 1 - \frac{p_L}{nvp} \quad (3.10)$$

The plots of throughput and the upper and lower bounds are shown in Figure 3.6, Figure 3.7 and Figure 3.8 respectively for the case $nv = 50$.

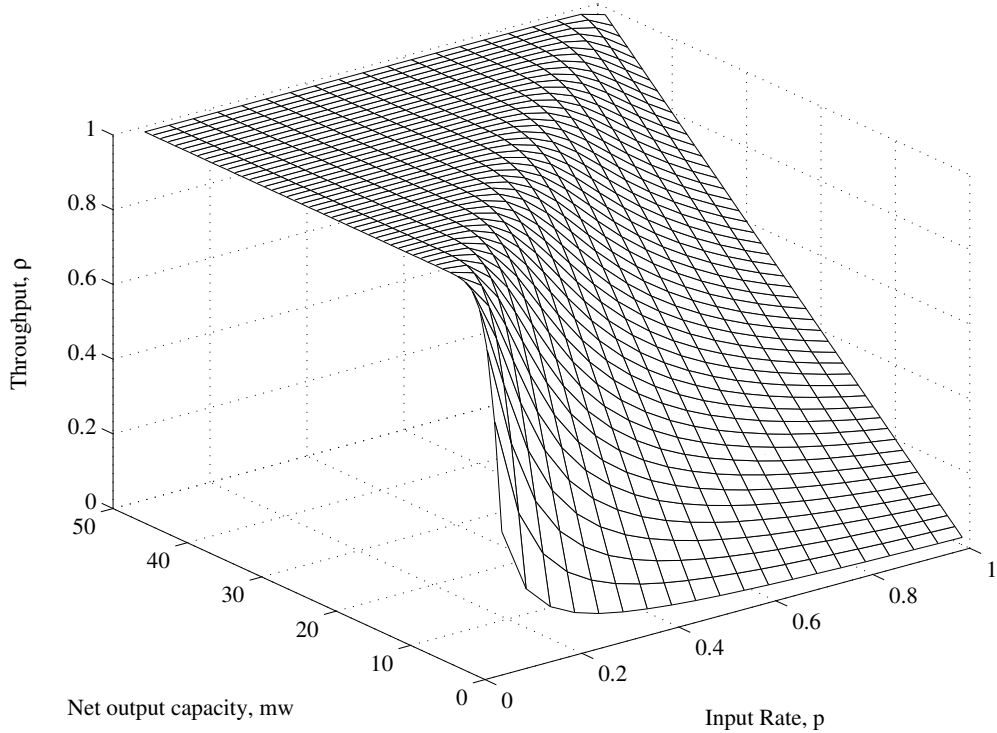


Figure 3.6: Throughput, ρ , instantaneous routing, $nv = 50$.

3.2.2 Case II: Deterministic Routing with Finite Delay

Key assumption: There are no buffers to store cells lost due to cell processing delay, and we loose all the cells at the inputs in the next t slots.

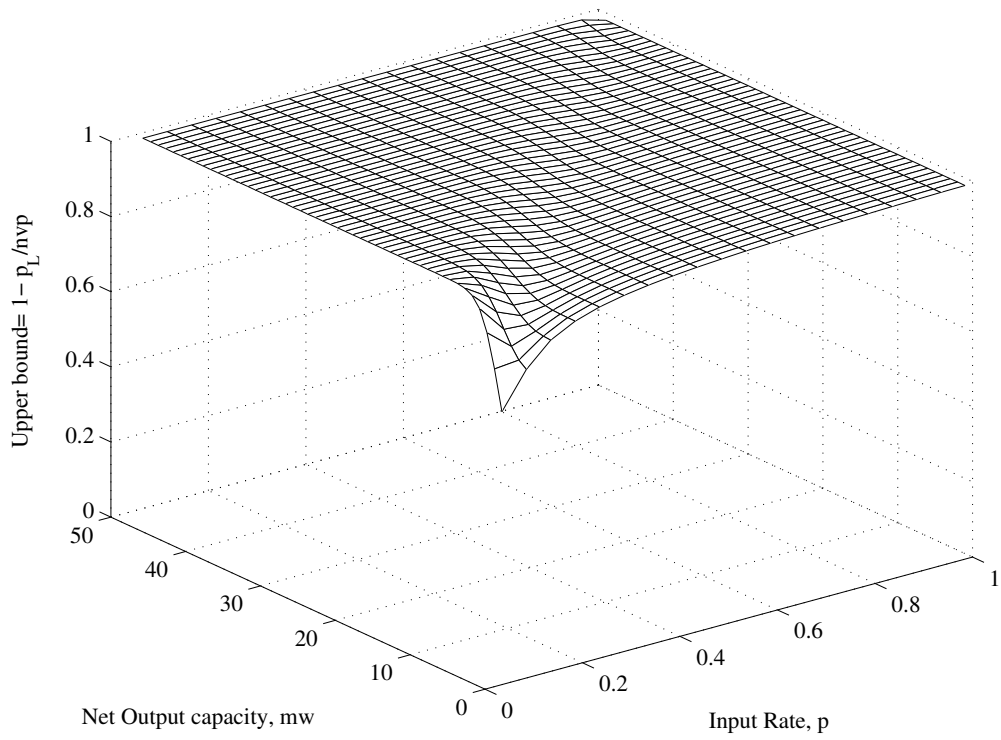


Figure 3.7: Upper bound for ρ , instantaneous routing, $nv = 50$.

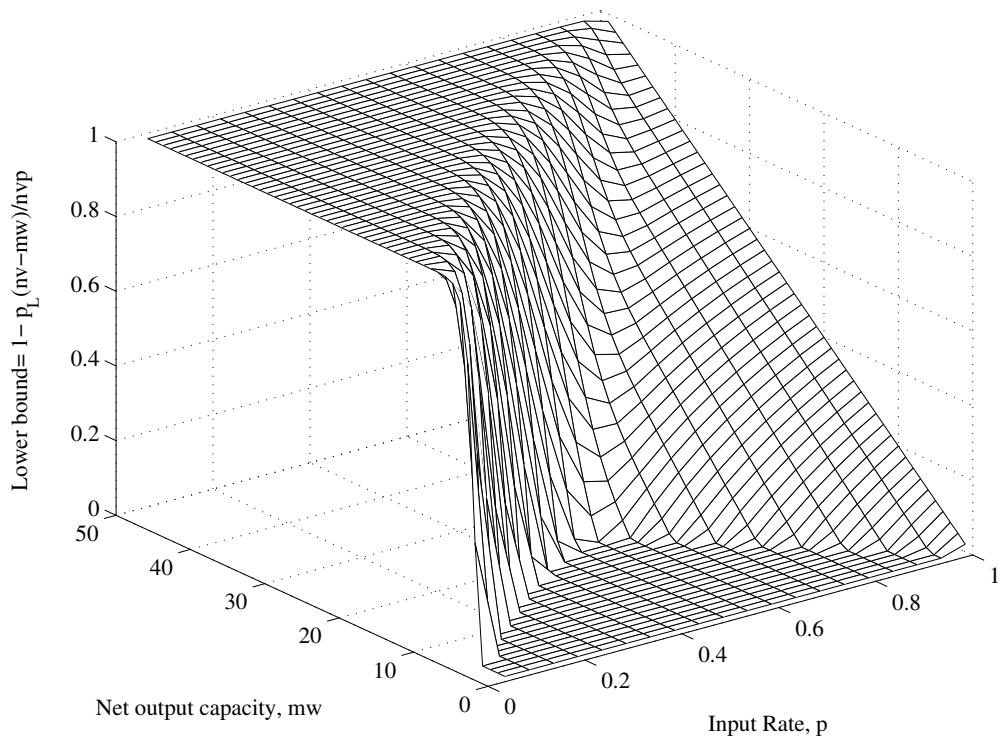


Figure 3.8: Lower bound for ρ , instantaneous routing, $nv = 50$.

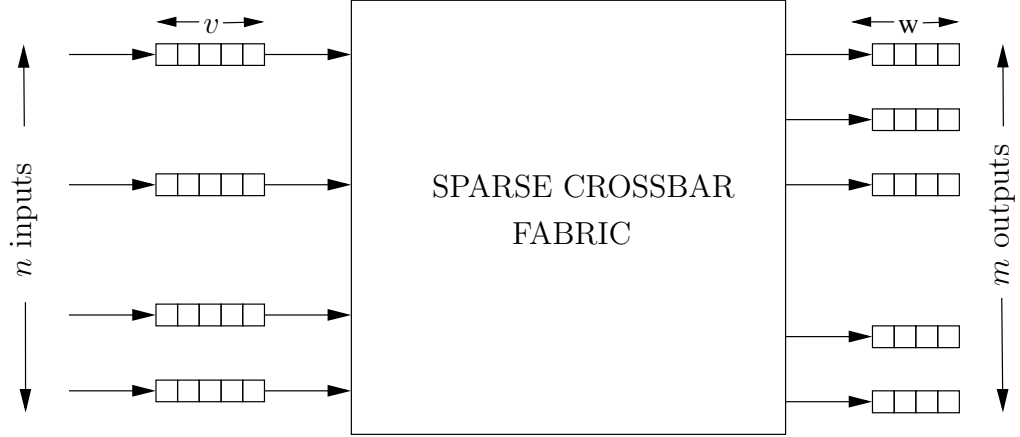


Figure 3.9: Case II: Deterministic routing with finite delay.

The probability of cell loss is:

$$\Pr[\text{cell loss}] = 1 - \Pr[N_T \geq mw \text{ and } 0 \text{ cells arrive in the next } t \text{ frames}]$$

Arrivals in different slots are independent, therefore

$$\Pr[\text{cell loss}] = q_L = 1 - (1 - p_L)(1 - p)^{tnv} \quad (3.11)$$

where p and p_L are as defined previously. This is the probability of loss for $t + 1$ slots combined

Thus, the expected number of cells lost per frame slot is

$$\begin{aligned} \mathbb{E}[\text{number of cells lost/slot}] &= \frac{t\mathbb{E}[N_T] + \mathbb{E}[N_{lost}]}{t + 1} \\ &= \frac{(t + 1)nv p - mwp_L - \sum_{j=1}^{mw} jP(j)}{t + 1} \end{aligned} \quad (3.12)$$

From the above expression, the intuitive result that the number of cells lost per slot approach $nv p$ (which is equal to the expected number of arrivals in a slot) as the processing delay is increased linearly is readily seen.

3.2.3 Case III: Randomized Routing

In this case it is assumed that the inputs route the cells randomly to the outputs irrespective of whether an output port is full to its capacity or not. Every input is connected to all the m output ports and that each cell at the input is equally likely to go to any of these m output ports. So here as already stated in Section 3.1.1.3 we are assuming a fully connected bipartite concentrator, unlike the optimized construction in the previous two cases. So a cell loss could occur for even less than mw cells/slot at the input because now a single output port can receive w cells/slot even when only w cells/slot come at all the n inputs combined. This is not the way a real concentrator would work but we are looking at this scenario as a worst case situation in terms of routing and thus derive an upper bound for the cell loss probability as the number of input ports increases linearly. The basic idea here is to look at a tagged output port and see the losses at that port when the inputs route cells with equal probability to the output ports.

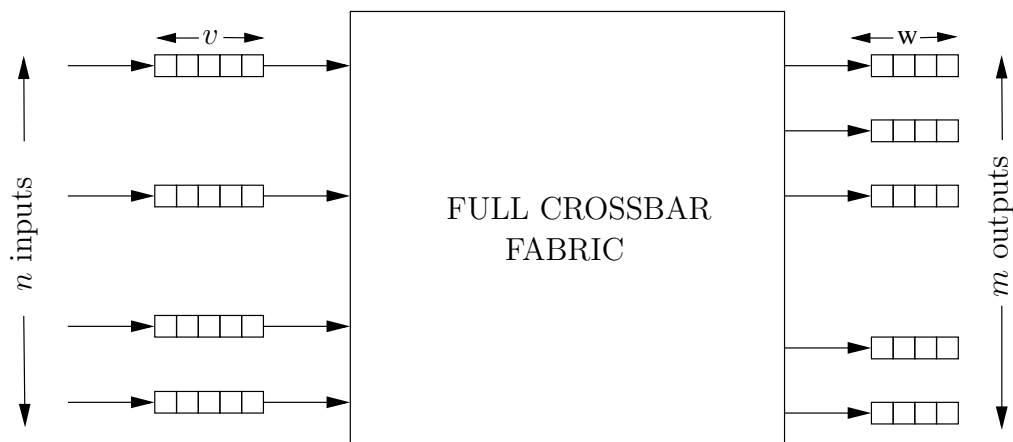


Figure 3.10: Case III: Randomized (Distributed) routing.

It is easy to see that the probability that a cell comes to the tagged output in a sub-slot is p/m . Let M be the number of cells arriving at an output port in one slot.

The distribution of M , assuming binomial distribution at the inputs as before is:

$$\Pr[M = j] = \binom{nv}{j} \left(\frac{p}{m}\right)^j \left(1 - \frac{p}{m}\right)^{nv-j} \quad (3.13)$$

The probability of cell loss is,

$$\hat{p}_L = \sum_{j=w+1}^{nv} \binom{nv}{j} \left(\frac{p}{m}\right)^j \left(1 - \frac{p}{m}\right)^{nv-j} \quad (3.14)$$

Taking the limit $n \rightarrow \infty$, while keeping the concentration ratio, λ ($= nv/mw$) constant we obtain after some manipulation (See Appendix A)

$$\lim_{n \rightarrow \infty} \hat{p}_L = 1 - \sum_{j=0}^w \hat{P}(j) \quad (3.15)$$

where $\hat{P}(j) = e^{-\lambda wp} \frac{(\lambda wp)^j}{j!}$

As expected, the resulting distribution in (3.15) is Poisson with parameter λwp .

Let number of cells lost/slot = M_{lost}

$$\mathbb{E}[M_{lost}] = \sum_{j=w+1}^{nv} (j - mw) \binom{nv}{j} \left(\frac{p}{m}\right)^j \left(1 - \frac{p}{m}\right)^{nv-j} \quad (3.16)$$

Again, taking the limit $n \rightarrow \infty$, while keeping λ constant we get

$$\lim_{n \rightarrow \infty} \mathbb{E}[M_{lost}] = (\lambda wp - w) \left(1 - \sum_{j=0}^w \hat{P}(j)\right) + (\lambda wp) \hat{P}(w) \quad (3.17)$$

Define loss ratio l_r as the ratio of the expected number of cells lost to the expected number of cell arrivals in a slot, i.e.

$$l_r = \frac{E[M_{lost}]}{E[M]} = \frac{1}{\lambda w p} \sum_{j=w+1}^{nv} (j-w) \binom{nv}{j} \left(\frac{p}{m}\right)^j \left(1 - \frac{p}{m}\right)^{nv-j} \quad (3.18)$$

Note throughput, $\rho = 1 - l_r$.

As $n \rightarrow \infty$

$$\lim_{n \rightarrow \infty} l_r = \left(1 - \frac{1}{\lambda p}\right) \left(1 - \sum_{j=0}^w \hat{P}(j)\right) + \hat{P}(w) \quad (3.19)$$

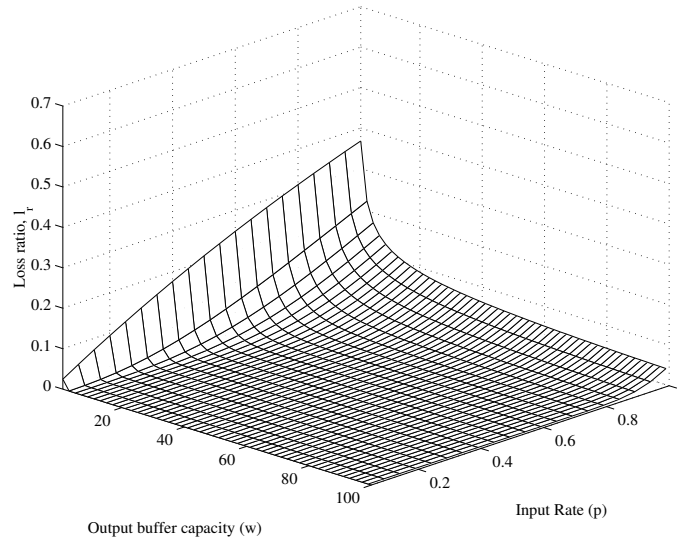


Figure 3.11: Loss ratio l_r , $\lambda = 1$.

The variation of the loss ratio l_r with output buffer capacity (w) and rate of traffic at the input (p) is shown in Figure 3.11, Figure 3.12 and Figure 3.13 for $\lambda = 1$, $\lambda = 2$ and $\lambda = 3$ respectively. Figure 3.14 shows the change in the variation of l_r versus p for different λ . These figures reflect the expected behavior of loss. As we increase the concentration ratio (λ), the loss (l_r) increases for a given value of output capacity (w) and input rate (p). At low values of p , before the l_r curve

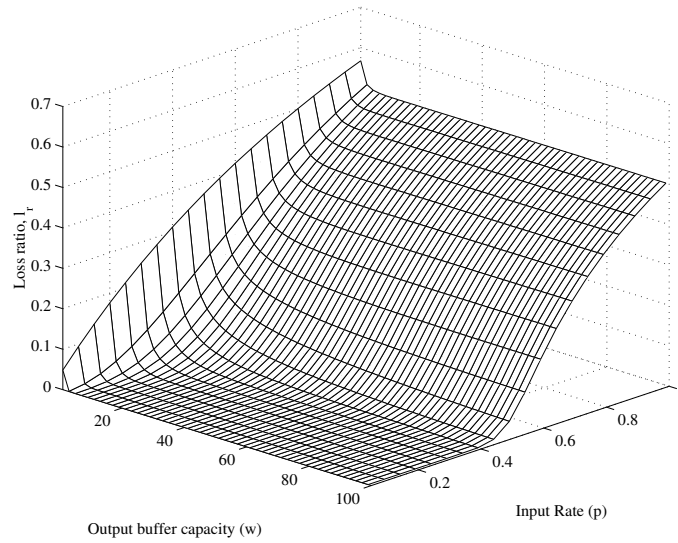


Figure 3.12: Loss ratio l_r , $\lambda = 2$.

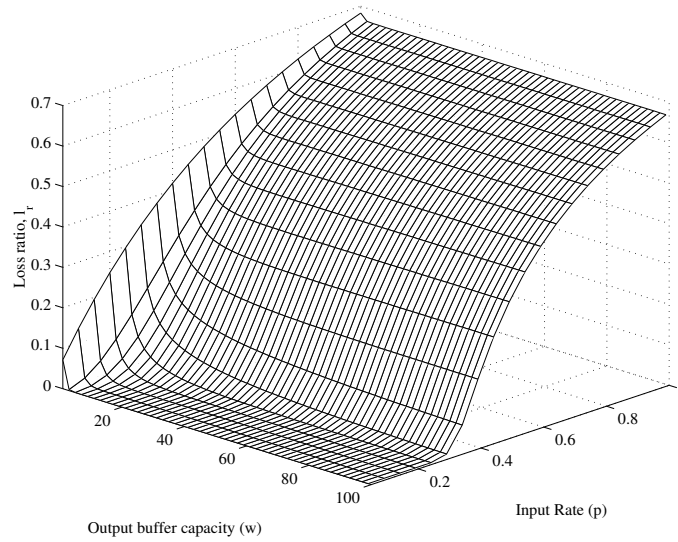


Figure 3.13: Loss ratio l_r , $\lambda = 3$.

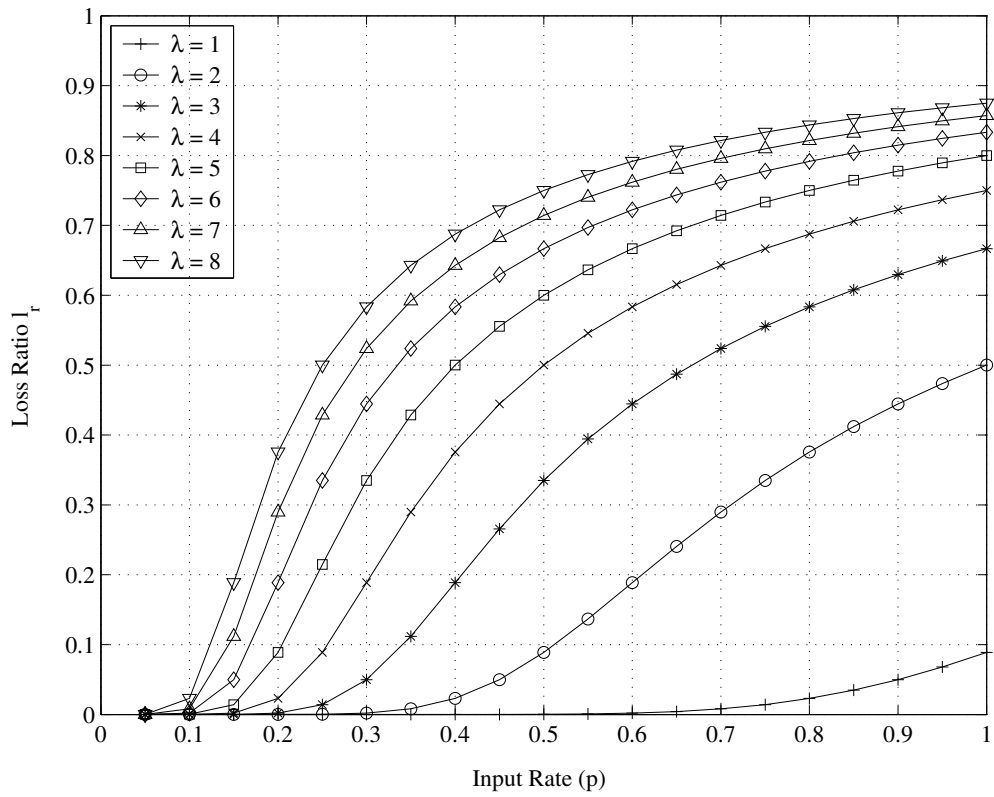


Figure 3.14: Loss ratio l_r for different λ , $w = 10$.

starts tapering off, not only the loss (l_r) but also the rate of change of loss ($\frac{\partial l_r}{\partial p}$) at a particular input rate p is higher for larger concentration ratios. This reflects the fact that an increase in net traffic at the inputs as compared to the capacity at the output increases the loss at low input rates (p).

Chapter 4

Input Queued Concentrator

In Chapter 3, the cell loss in a sparse crossbar $Q(n, m, v, w)$ was analyzed. There we assumed that the cells which cannot be concentrated are dropped. This is the worst case scenario because we can obviously get lower losses by queuing (storing) the excess or “*unconcentrated*” cells and concentrating them later. Therefore, it is of interest to characterize the performance of the concentrator when cells are queued to prevent excessive loss.

The first question to consider when queuing the cells is where to place the buffers for storing the excess cells. It is known that, for switches, output queuing can achieve high throughput, but it also requires a speedup equal to the number of inputs i.e for an $N \times M$ switch a speedup of N is required [6]. For concentration, the minimum average output speedup ¹ required for an $nv \times mw$ packet concentrator is equal to $\frac{nv}{mw}$. For a $Q(n, m, v, w)$ described in Section 2.1.1 the maximum

¹Average output speedup is equal to the ratio of the maximum number of cells which can arrive in one slot to the maximum number of cells routed by the non-output queued concentrator in one slot.

speedup required would be $\frac{\alpha v}{w}$, where α is the connectivity, i.e., the number of inputs connected to an output, as defined in Section 2.3. Note that since the output addresses are assigned by the routing control, the average speedup can be reduced to $\frac{\frac{nv-mw}{m}+w}{w} = nv/mw$ by assigning the excess cells uniformly across all the outputs. The first term in the numerator on the left hand side in the previous expression accounts for the excess cells which are delivered to an output. the sparse Since this implies a further speedup of the already fast crosspoints, output queuing is not considered.

Input queuing, on the other hand allows the buffers used for storing excess cells to operate without any speedup requirement. The buffers at the input ports accept cells at the rate of v cells per slot, and hence operate only as fast as the input port speed. The limitation is that we do not get throughput as high as that for the output queued case. In the next section we develop a Markov chain model for the input queued concentrator and analyze it using results from queuing theory.

4.1 Queuing Model

We consider a slotted discrete time system similar to that introduced in Section 3.1. To recap, the arrivals at the input occur in batches, one batch per slot, with a maximum batch size of v cells and the outputs can sink up to w cells per slot. We will consider the combined queue state for all the inputs together. We define some terms which will be used throughout the following discussion:

- S_t^i Number of cells in the system at input i at the end of time slot t .
- S_t Number of cells in the system at all the inputs at the end of time slot t ,
i.e., $S_t = \sum_{i=1}^n S_t^i$.
- Q_t^i Number of cells in input queue i (*excluding those in service*) at the end
of time slot t .
- Q_t Number of cells in all the input queues (*excluding those in service*) at the
end of time slot t , i.e., $Q_t = \sum_{i=1}^n Q_t^i$.
- A_t^i Number of cells in the batch arriving at input queue i in time slot t .
- A_t Number of cells arriving at all input queues time slot t , i.e., $A_t = \sum_{i=1}^n A_t^i$.
- D_t^i Number of cells serviced at input queue i in time slot t , i.e., number of
cells leaving the i^{th} input queue in time slot t .
- D_t Number of cells serviced at all input queues in time slot t , i.e., $D_t =$
 $\sum_{i=1}^n D_t^i$.

The following assumptions are used in the queuing model for the concentrator.

1. The input queues have unlimited storage capacity.
2. The input queues are emptied through the transmission of the cells they contain. The total number of output spaces via which the cells are removed (i.e., the number of servers in the queuing model for all the inputs combined together) is equal to $mw > 0$.
3. Time is divided into fixed length intervals, referred to as slots, such that one slot suffices for the transmission of w cells via each output link. The

transmission of a cell via an output channel of a queue starts at the beginning of a slot and ends at the end of this slot. Cells cannot leave the queue at the end of the slot during which they arrived in the queue. This is referred to as the late arrival assumption. See Figure 4.1.

Note that, in this case, when a new cell arrives (say at the end of the l^{th} slot) to find an empty queue and its service begins in the next slot (i.e., the $(l + 1)^{th}$ slot), then the time spent in the queue is measured from the $(l + 1)^{th}$ slot. So, in Figure 4.1, if the cells in batch A_{t-1} arrive to find an empty queue, i.e., $Q_{t-1} = 0$, then the cells from this batch which are serviced between time instants t and $t + 1$ will have delay equal to 0.

4. New cells enter the input queues according to independent batch arrival processes. The number of cells arriving in the input queue i during consecutive slots are modeled as *i.i.d.* random variables with a probability distribution, characterized by the generating function $A^i(z)$.

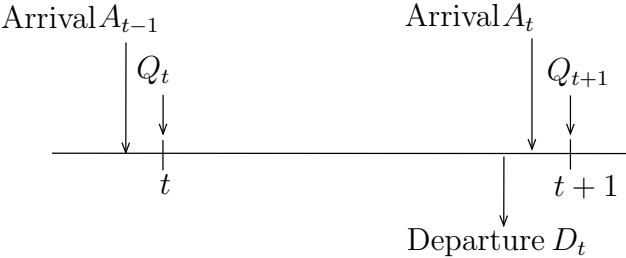


Figure 4.1: Late arrival model.

4.1.1 System and Queue Contents

Using the preceding assumptions, the following equation can be written for input buffer i .

$$S_{t+1}^i = S_t^i - D_t^i + A_t^i \quad (4.1)$$

where $0 \leq A_t^i \leq v$, $S_t^i \geq 0$, $S_t^i \geq D_t^i \geq 0$, $\forall i, t$.

If the routing control can always concentrate k , ($k \leq mw$) cells to the outputs when there are a total of k cells in all the input buffers, then the concentration property is always satisfied and we get a work-conserving $nv \times mw$ concentrator. Since this concentrator always concentrates the maximum number of cells possible, it should give the lowest delay and queue length when compared with all other concentrators which can concentrate upto mw cells in a slot. Specifically, for the concentration property to be satisfied by in a work-conserving fashion

$$D_t = \sum_{i=1}^n D_t^i = \min(S_t, mw) \quad \forall t > 0 \quad (4.2)$$

Using (4.1) and (4.2) we get

$$\begin{aligned} S_{t+1} &= \max(0, S_t - mw) + A_t \\ &= Q_t + A_t \end{aligned} \quad (4.3)$$

Here $Q_t = \max(0, S_t - mw)$ are the cells waiting in the queue (*excluding those in service*) in slot t , as defined in Section 4.1. This expression is apparent from the late-arrival assumption stated in Section 4.1. Let $\lim_{t \rightarrow \infty} \Pr(S_t = k) = \pi_k$ denote

the steady-state probabilities for the distribution of S_t ². We define the probability generating function *p.g.f.*

$$\begin{aligned}
S_t(z) &= \sum_{k=0}^{\infty} \pi_k z^k = \mathbb{E} [z^{S_t}] = \mathbb{E} [z^{S_{t+1}}] \\
&= \mathbb{E} [z^{\max(0, S_t + A_t - mw)}] \\
&= \mathbb{E} [z^{\max(0, S_t - mw)}] \mathbb{E} [z^{A_t}]
\end{aligned} \tag{4.4}$$

since S_t and A_t are independent. Since we are analyzing the steady state probabilities of S_t , A_t and Q_t with respect to t , we will refer to $\lim_{t \rightarrow \infty} S_t$, $\lim_{t \rightarrow \infty} A_t$ and $\lim_{t \rightarrow \infty} Q_t$ as S , A and Q and the steady state, i.e., $t \rightarrow \infty$ limits of the *p.g.fs* $S_k(z)$, $A_k(z)$ and $Q_k(z)$ as $S(z)$, $A(z)$ and $Q(z)$ respectively. Taking the limit ($t \rightarrow \infty$) on both sides in (4.4)

$$\begin{aligned}
\lim_{t \rightarrow \infty} S_t(z) &= \lim_{t \rightarrow \infty} \mathbb{E} [z^{\max(0, S_t - mw)}] \mathbb{E} [z^{A_t}] \\
&\Rightarrow S(z) = \mathbb{E} [z^{\max(0, S - mw)}] \mathbb{E} [z^A]
\end{aligned} \tag{4.5}$$

Now using (4.5) we get

$$S(z) = \left(\sum_{j=0}^{mw-1} \pi_j z^0 + \sum_{j=mw}^{\infty} \pi_j z^{j-mw} \right) \sum_{j=0}^{nv} \Pr(A = j) z^j \tag{4.6}$$

Since in our case $\Pr(A > nv) = 0$, therefore, the summation in the third term only goes upto nv . Set $A(z) = \sum_{j=0}^{nv} \Pr(A = j) z^j$ as the *p.g.f.* for the arrival process A . Continuing from the previous expression,

$$\begin{aligned}
z^{mw} S(z) &= \left(\sum_{j=0}^{mw-1} \pi_j z^{mw} + \sum_{j=mw}^{\infty} \pi_j z^j \right) A(z) \\
&= \left(\sum_{j=0}^{mw-1} \pi_j z^{mw} + S(z) - \sum_{j=0}^{mw-1} \pi_j z^j \right) A(z)
\end{aligned} \tag{4.7}$$

²The steady state exists if $\mathbb{E}[A] < mw$. This is explained in more detail on Page 47.

Grouping the terms for $S(z)$ and $A(z)$, we obtain the *p.g.f.* for the steady-state queue size:

$$S(z) = A(z) \frac{\sum_{j=0}^{mw-1} \pi_j (z^{mw} - z^j)}{z^{mw} - A(z)} \quad (4.8)$$

This is a standard approach in queuing analysis (see, for example [22]). Now (4.3) implies $S(z) = A(z) \cdot Q(z)$. Thus, $Q(z)$ can be written as

$$Q(z) = \frac{\sum_{j=0}^{mw-1} \pi_j (z^{mw} - z^j)}{z^{mw} - A(z)} \quad (4.9)$$

Equation (4.9) contains mw unknown constants π_j for $0 \leq j \leq mw - 1$. These can be determined by invoking the analyticity of $S(z)$ and $Q(z)$ inside the unit disk of the complex z plane and by using Rouché's theorem (see e.g., Appendix B Section B-2 on Page 86 and section 3.2.3.6, Appendix 3.A in [22]).

Using Rouché's theorem, one can show that the characteristic equation (CE), $z^{mw} - A(z)$, has exactly mw roots on and inside the unit circle $|z| = 1$. Note that these mw roots must coincide with those of the numerator because of the analyticity of $Q(z)$ in $|z| \leq 1$. Now, $A(z)$ is a polynomial of degree nv (because not more than nv cells can arrive in a single slot). For the concentrator $nv \geq mw$. Therefore, after canceling the mw factors in $Q(z)$ and using $Q(1) = 1 = (\sum_{i=0}^{\infty} \Pr(Q = i))$, (See Appendix B Section B-2) we have

$$Q(z) = \prod_{j=1}^{nv-mw} \left(\frac{1 - z_j}{z - z_j} \right) \quad (4.10)$$

where z_j , $1 \leq j \leq nv - mw$, are the roots of the CE, $z^{mw} - A(z) = 0$, *outside* the unit circle [23]. Bruneel and Kim [section 4.1.2 in [22]] also give an expression for

$Q(z)$ in terms of the mw roots *inside and on* $|z| = 1$, say z_j^* , $1 \leq j \leq mw - 1$ (and $z = 1$) as follows

$$Q(z) = (mw - A'(1)) \frac{(z - 1)}{z^{mw} - A(z)} \prod_{j=1}^{mw-1} \frac{z - z_j^*}{1 - z_j^*} \quad (4.11)$$

where $A'(1) = \left. \frac{\partial A(z)}{\partial z} \right|_{z=1} = E[A]$. The model developed so far corresponds to a GI/D/ mw queuing system. If the input process is binomial (similar to the one considered in Chapter 3) then the probability distribution and the generating function for A are

$$a_k \triangleq \Pr(A = k) = \binom{nv}{k} p^k (1 - p)^{nv-k} \quad (4.12)$$

and

$$\begin{aligned} A(z) = E[z^A] &= \sum_{k=0}^{nv} a_k z^k \\ &= (1 - p + pz)^{nv} \end{aligned} \quad (4.13)$$

Each batch of v arrivals at the input can be viewed to comprise v “spaces” or positions which the cells of that batch occupy. The parameter p corresponds to the probability that an arbitrary position is occupied in any batch at any input, this is the same interpretation for p as that described in Section 3.1. With A and $A(z)$ as described in (4.12) and (4.13), the GI/D/ mw system becomes a Binom/D/ mw system. A steady state for the GI/D/ mw system exists only if

$$E[A] = A'(1) < mw \quad (4.14)$$

for the Binom/D/ mw system this becomes

$$nvp < mw \Rightarrow \rho = \frac{nvp}{mw} < 1 \quad (4.15)$$

where ρ is the normalized offered load.

Note that the *p.g.f.* $A(z)$ in (4.13) can be written in terms of ρ as

$$A(z) = \left(1 - \frac{\rho mw}{nv} + \frac{\rho mw}{nv} z\right)^{nv} \quad (4.16)$$

Therefore, taking the limit as $nv \rightarrow \infty$ on both sides while keeping ρ constant we get the limit

$$\lim_{nv \rightarrow \infty} A(z) = e^{\rho mw(z-1)} \quad (4.17)$$

which is the *p.g.f.* for a Poisson process with parameter ρmw , i.e., A is a Poisson(ρmw) random variable.

The moments of the steady-state queue contents can be easily obtained from (4.10) by calculating derivatives on the unit circle. Accordingly, The mean steady-state queue size is given by

$$\bar{Q} = \sum_{j=1}^{nv-mw} \frac{1}{z_j - 1} \quad (4.18)$$

The mean for the queuing delay W_q follows from Little's law:

$$E[W_q] = \frac{E[Q]}{E[A]} \quad (4.19)$$

Note that W_q is the delay encountered by the cell while it is in the queue waiting to be serviced. It does not include the delay while the cell is being served. The total delay W , suffered by a cell is the sum of W_q and the time it takes to be served. Since the service time is deterministic and equal to one, therefore

$$E[W] = \frac{E[S]}{E[A]} = \frac{E[Q] + E[A]}{E[A]} = 1 + E[W_q] \quad (4.20)$$

Recall that $S = Q + A$, (from (4.3)). Similarly, the variance of Q and S is given by

$$\text{Var}(Q) = \sum_{j=1}^{nv-mw} \frac{1}{(z_j - 1)^2} + \sum_{j=1}^{nv-mw} \frac{1}{(z_j - 1)} \quad (4.21a)$$

$$\text{Var}(S) = \text{Var}(A) + \text{Var}(Q) \quad (4.21b)$$

4.1.2 Probability Distributions and Tail Probabilities

The roots z_j and z_j^* in (4.10) and (4.11) can be easily calculated by numerical techniques. The probability mass function (PMF) for Q can be obtained by taking the inverse z -transform of (4.10) or (4.11). Similarly, we can get the PMF of S , $\{\pi_i\}$, from the inverse z -transform of $A(z) \cdot Q(z)$. It is easier and faster to apply numerical inversion to (4.10) to get the PMF for Q . The PMF for the system delay W can then be calculated from the following relation derived in [23]

$$\Pr(W \leq j) = \sum_{i=0}^{mw} \pi_i \Pr(A^* \leq mwj) + \sum_{i=mw+1}^{mw(j+1)-1} \pi_i \Pr(A^* \leq mw(j+1)-i) \quad (4.22)$$

where $\Pr(A^* = i) = \Pr(A \geq i)/E[A] \quad \forall i \geq 1$.

It should be noted that this expression is true for service time equal to one slot and geometric inter-arrival times, and these conditions are satisfied in our case for the binomial arrival process.

Although we can get the entire probability distributions for Q and S from the inverse z -transform of $Q(z)$ and $S(z)$ respectively, usually it is of interest to get an estimate of the tail of the PMFs for finding loss probabilities. This helps in

estimating the buffer size for a given loss probability and input rate. Specifically, we can get closed form expressions for the tail probabilities and of the queue contents and cell delay without calculating the entire PMF. These formulas turn out to be extremely accurate and easy to evaluate. The method for this is outlined below.

The probabilities $\Pr(Q = n)$ can be determined, in principle, by applying the inversion formula for z -transforms and Cauchy's residue theorem on $Q(z)$ in (4.10) or (4.11). Other methods to do this include direct series expansion or partial fraction decomposition (e.g., see Appendix B Section B-3 for a derivation using partial fractions). We concentrate on the residue method here. Using this method $\Pr(Q = n)$ is obtained as the negative sum of the residues of $Q(z) \cdot z^{-n-1}$ in the poles of $Q(z)$. It can be seen, however, this sum is dominated, for large values of n by the term associated with the pole of $Q(z)$ with the smallest absolute value. The poles of $Q(z)$ are the roots of $A(z) = z^{mw}$ outside the unit disk ($\{z_j\}$ in (4.10)) in the complex z -plane. This equation can be shown to have the following properties if $\rho < 1$, i.e., if the equilibrium condition for the steady-state is met:

1. $A(z) = z^{mw}$ has exactly one real positive root, say z_0 , larger than 1 and the multiplicity of z_0 is one.
2. $A(z) = z^{mw}$ has no roots outside the unit disk with absolute value less than z_0 .
3. No other root exists with absolute value equal to z_0 if $A(z)$ is not a function

of z^M for some $M > 1$, $M \in \mathbb{Z}$, such that $\gcd(M, mw) > 1$.

$\Pr(Q = n)$ is equal to the coefficient of z^n in $Q(z)$. We know from complex analysis that the coefficient of z^n can be obtained as the negative sum of residues of $Q(z) \cdot z^{-n-1}$ in the poles of $Q(z)$, z_j , $j = 1, \dots, nv - mw$. From the preceding discussion we can see that the dominant term in the expression for $\Pr(Q = n)$ (coefficient of z^n in $Q(z)$) is the residue of $Q(z) \cdot z^{-n-1}$ in the pole z_0 . The characteristic equation, $A(z) = z^{mw}$, can be solved numerically to get z_0 easily. Since z_0 has multiplicity one, the residue at z_0 is simply equal to $\lim_{z \rightarrow z_0} (z - z_0) Q(z) z^{-n-1}$. As a result, the following approximation for the tail probability of the queue contents is obtained:

$$\Pr(Q = n) \simeq -b_q z_0^{-n-1}, \quad (4.23)$$

where

$$b_q = (1 - z_0) \prod_{\substack{j=1 \\ z_j \neq z_0}}^{nv-mw} \frac{1 - z_j}{z_0 - z_j} \quad (4.24)$$

Here, $b_q = \lim_{z \rightarrow z_0} (z - z_0) Q(z)$, is the residue of $Q(z)$ at z_0 . Therefore, the probability that the queue contents exceed a given threshold Q_0 obtained by summing (4.23) over appropriate values of n , gives for large n

$$\Pr(Q > Q_0) \simeq \frac{b_q z_0^{-Q_0-1}}{z_0 - 1} \quad (4.25)$$

4.2 Theoretical and Simulation Results

To evaluate the performance of a Binom/D/ mw system simulation experiments were also done to verify the theoretical results. The simulations were executed until the estimate of the average cell queuing delay (\bar{W}_q) and the average queue length (\bar{Q}) reached with probability 0.95 a relative width of the confidence interval equal to 5%. The estimation of the confidence interval width was obtained by the replication method [24]. The simulator was written in C.

All the results discussed below are for a concentrator with $nv = 8$. First we focus our attention on the statistics for Q . The plot for the theoretical value of mean queue delay \bar{Q} , is given in Figure 4.2(a) for different values of mw . Figure 4.2(b) shows the plot for simulated values of \bar{Q} for comparison. Note that the x -axis has been normalized to $\rho = nvp/mw$ for ease of comparison between different mw . The theoretical values for the variance of the queue size $\text{Var}(Q)$ for different mw are shown in Figure 4.2. We show only the values for large traffic loads ($\rho \geq 0.8$) as at lower loads \bar{Q} , \bar{W}_q and $\text{Var}(Q)$ have very low (< 1) values. We note that the analytical and simulated results are in good agreement. The influence of nv on \bar{Q} is shown in Figure 4.4 for a fixed $mw = 4$. The plot for $nv \rightarrow \infty$ was obtained using the Poisson *p.g.f.* in (4.17). The figure reveals that, on the average, more buffer space is occupied as nv gets larger, but the influence of nv becomes negligible as soon as nv gets sufficiently large. We can explain this phenomenon by considering

the variance of the arrival process in terms of ρ

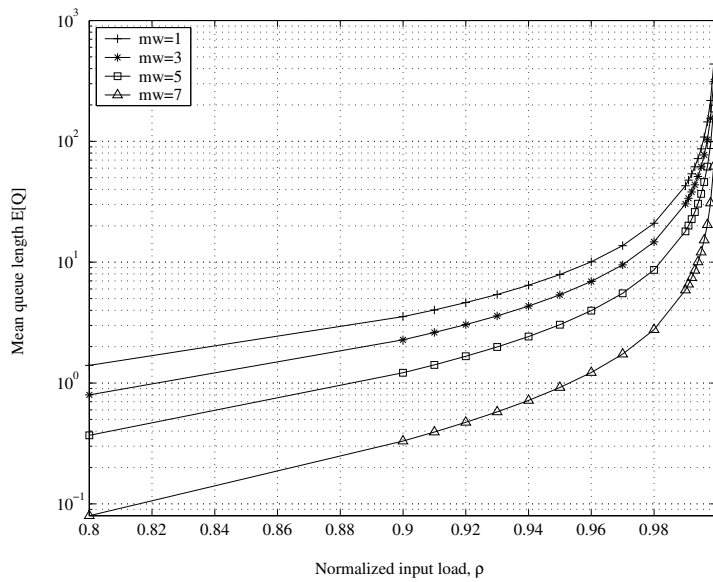
$$\begin{aligned}\text{Var}[A] &= A''(1) + A'(1) - [A'(1)]^2 = nvp(1-p) \\ &= \rho mw \left(1 - \frac{\rho mw}{nv}\right)\end{aligned}\tag{4.26}$$

This shows that the arrival variance and hence the congestion in the input buffer increases with nv but the rate of increase reduces steadily as nv becomes larger.

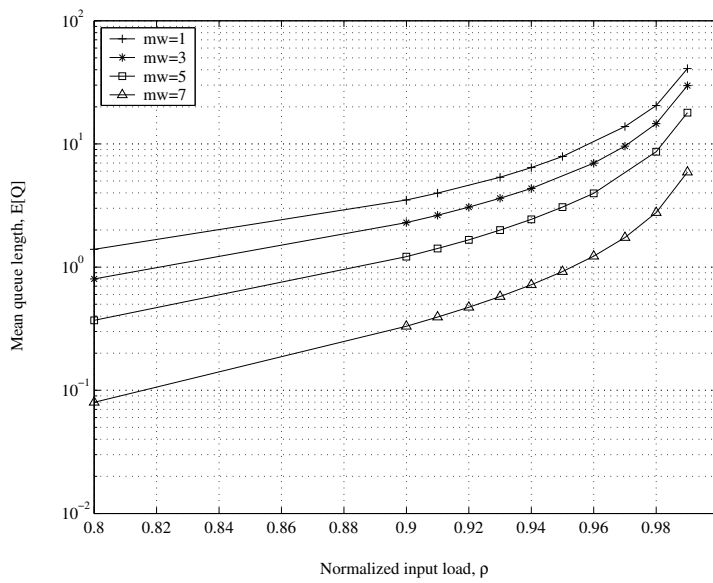
Also, the increase is most significant for large values of ρ and mw .

The queue size Q_0 at which the tail probability ($\Pr(Q \geq Q_0)$) becomes 10^{-6} and 10^{-9} for $mw = 1, 3, 5$ and 7 is shown in Figure 4.5(a) and Figure 4.5(b) respectively. All these plots are for $nv = 8$. From Figure 4.5(a) we can obtain the buffer size ($Q_0 + mw$) required to attain tail probability of 10^{-6} . For instance if $\rho = 0.9$ the required buffer size is given by 59, 45, 31 and 16 for $mw = 1, 3, 5$ and 7 respectively. These values are calculated for the infinite buffer capacity case. It can be shown that these values accurately approximate the loss ratio for finite size queues [22]. Specifically, these values are slightly larger than the finite capacity case cell loss ratio for high ($\rho \gtrsim 0.5$) traffic loads, whereas the inverse holds true at light loads.

Figure 4.6 shows the probability of having queue contents greater than X , i.e., $\Pr(Q \geq X)$ versus the total required buffer space $X + mw$ for different input loads ρ , here $\Pr(Q \geq X)$ is the exact probability distribution function, not an approximation of the tail (See Appendix B Section B-3). We can see that these results are almost identical to those in Figure 4.5(a) and Figure 4.5(b). This shows that the tail approximation is extremely tight.



4.2(a): Calculated mean queue length, \bar{Q}



4.2(b): Simulated mean queue length, \bar{Q}

Figure 4.2: Mean queue length, \bar{Q} , at various output capacities (mw)

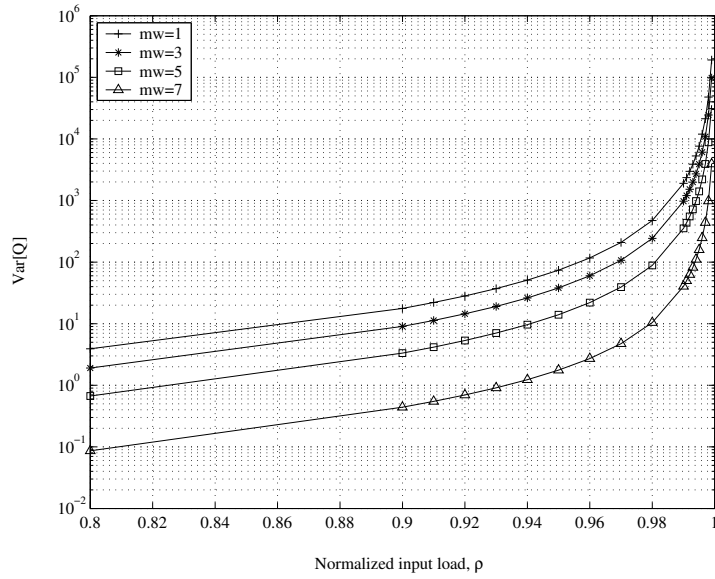


Figure 4.3: Calculated queue length variance, $\text{Var}(Q)$

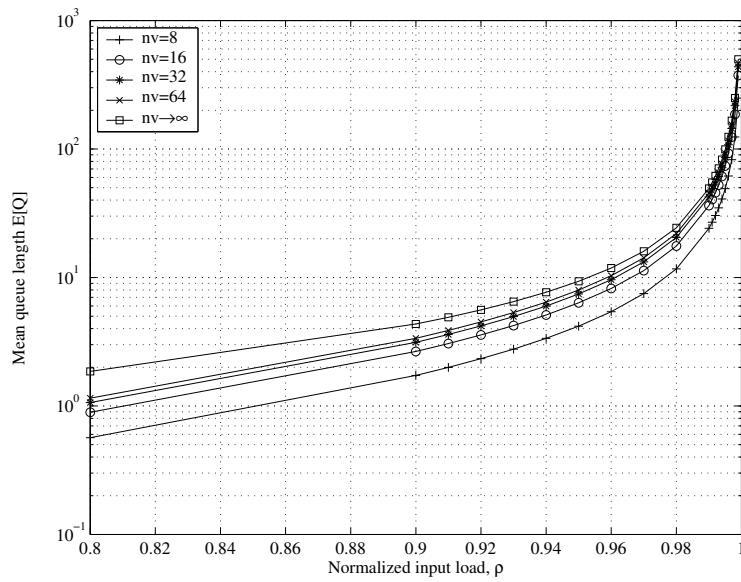
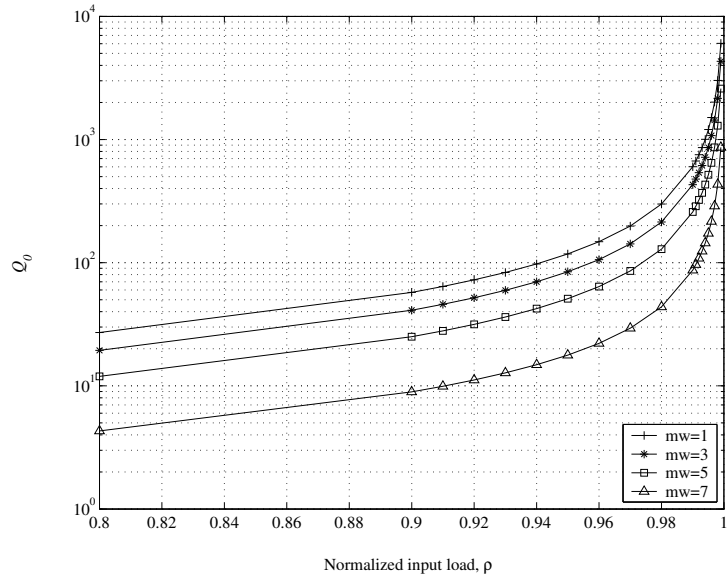
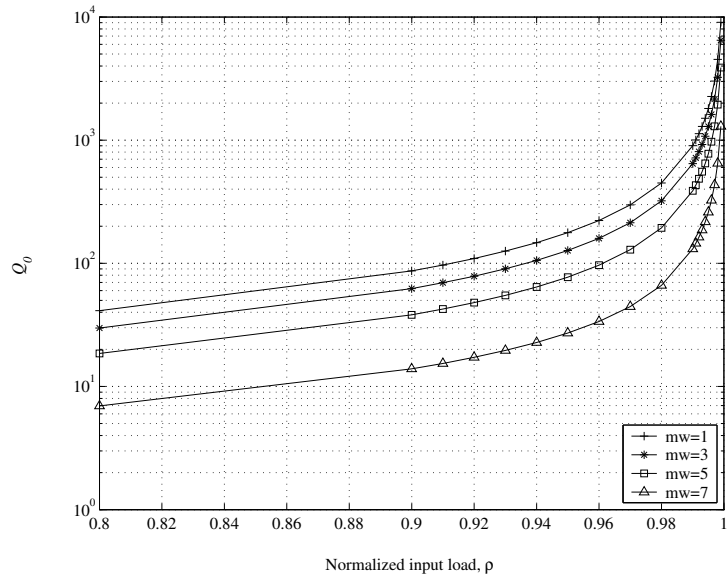


Figure 4.4: Mean queue length variation with nv , $mw = 4$

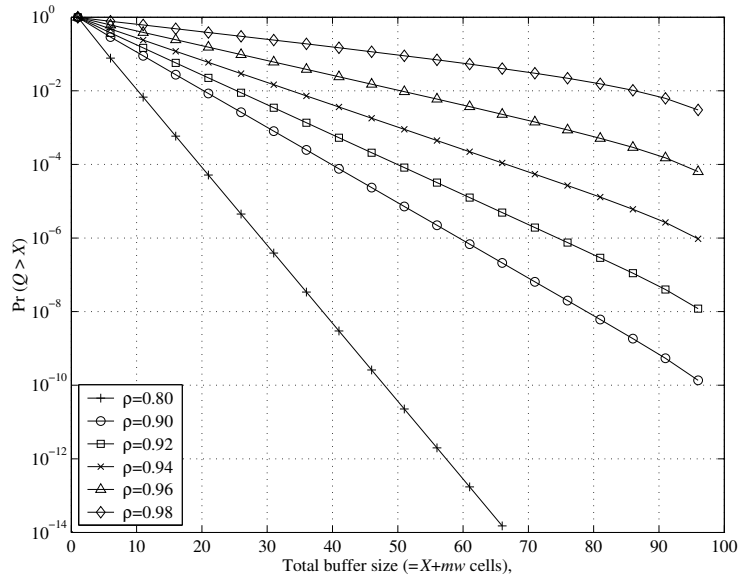


4.5(a): Queue size (Q_0) for $\Pr(Q \geq Q_0) = 10^{-6}$

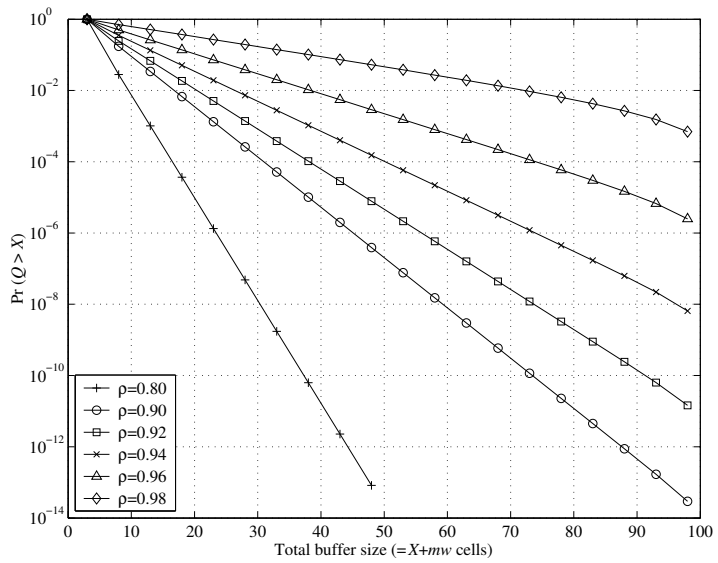


4.5(b): Queue size (Q_0) for $\Pr(Q \geq Q_0) = 10^{-9}$

Figure 4.5: Buffer size (Q_0) vs. input load (ρ) for a fixed probability of loss



4.6(a): $\Pr(Q \geq X)$ vs. total buffer size, $mw = 1$



4.6(b): $\Pr(Q \geq X)$ vs. total buffer size, $mw = 3$

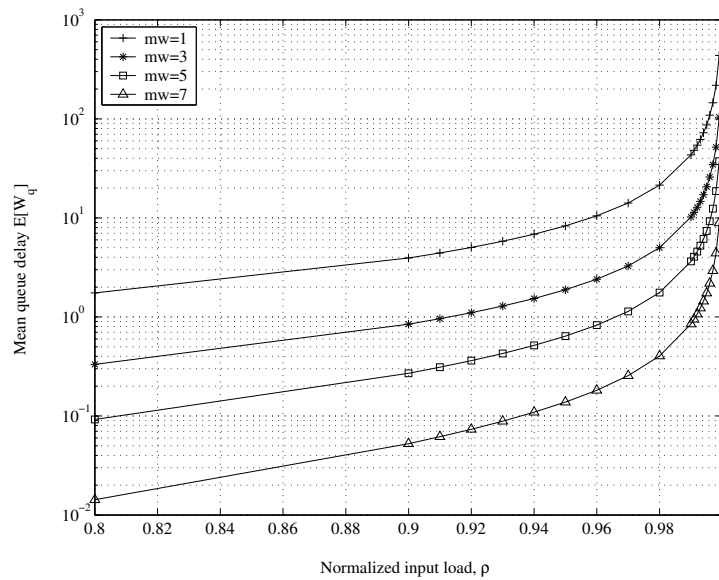
Figure 4.6: Probability of cell loss, $\Pr(Q \geq X)$ for $mw = 1, 3$

Now we focus on the delay characteristics of the concentrator. The analytical and simulated mean cell queuing delay \bar{W}_q is shown in Figure 4.7(a) and Figure 4.7(b) respectively for different values of mw . These figures show the same behavior for \bar{W}_q as that for \bar{Q} . The tail probabilities of the queuing delay W_q derived from (4.22) are shown for $mw = 1$ and $mw = 3$ are given in Figure 4.8(a) and Figure 4.8(b). These curves can be used to characterize the *delay jitter* or the degree of variability in the cell inter-departure times of the concentrator in terms of the 10^{-k} quantile of the queuing delay, i.e., the value of X^* such that $\Pr(W_q > X^*) = 10^{-k}$. Such measures are important for quantifying performance of real-time streaming data like voice and video.

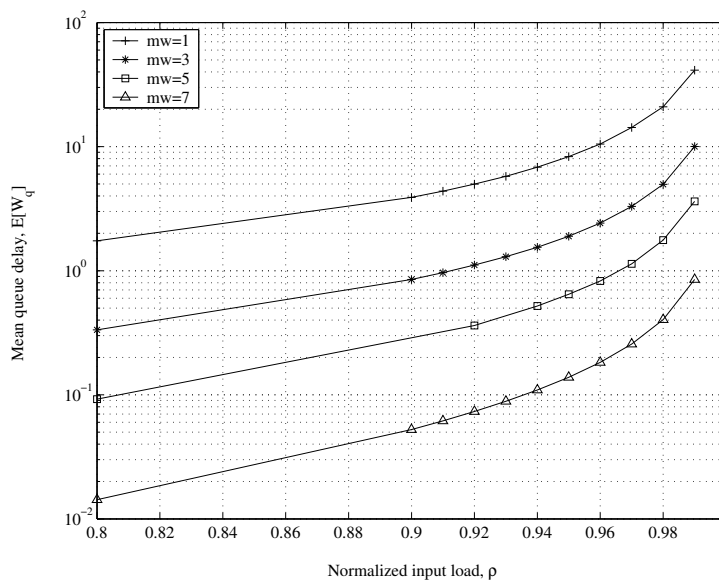
As all these figures illustrate, the queue size and the cell delay increase exponentially fast near the saturation region, $\rho = 1$. As the buffers fill up fast near $\rho = 1$, the buffer size required to maintain a given loss probability also increases. We can also make the observation that the average delay and queue size for a fixed ρ , decrease as mw increases, which is obvious considering that the total output capacity increases.

4.3 Outline of Analysis for Delayed Service

Throughout the previous discussion in this chapter we have assumed that the cells at the input get served in one time slot. Now we briefly outline the modification in analysis for the case when the service for a cell is deterministic but equal to d

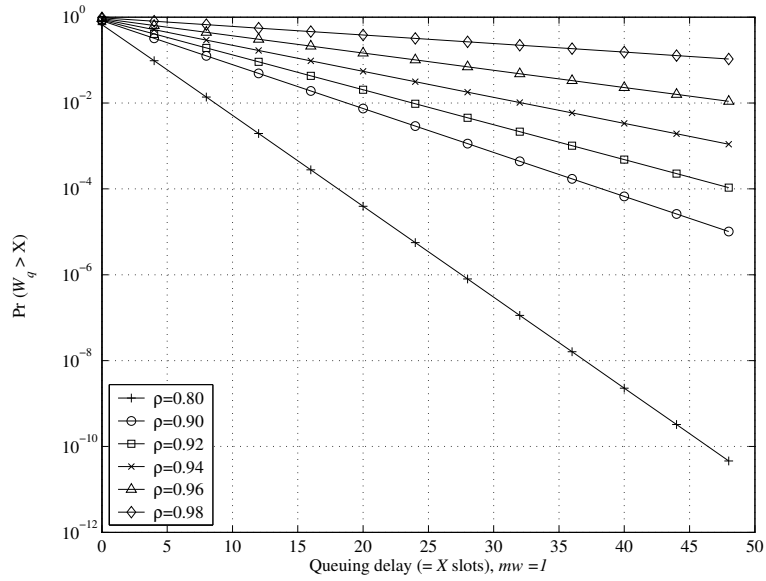


4.7(a): Calculated mean queue delay, \bar{W}_q

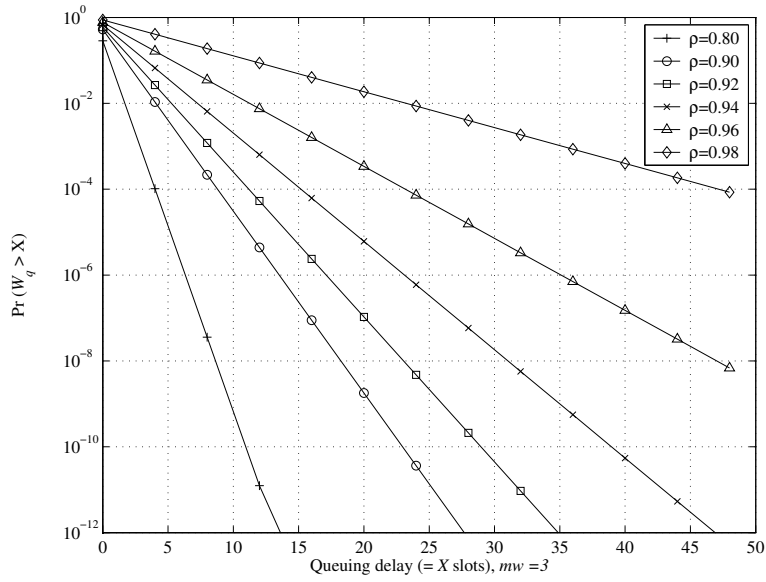


4.7(b): Simulated mean queue delay, \bar{W}_q

Figure 4.7: Mean queue delay, \bar{W}_q , at various output capacities (mw)



4.8(a): $\Pr(W_q > X)$, $mw = 1$



4.8(b): $\Pr(W_q > X)$, $mw = 3$

Figure 4.8: Tail probability of cell queuing delay

slots, where $d > 1$, $d \in \mathbb{Z}$. In such a case, the equation for the system contents is modified as follows:

$$\begin{aligned} S_{t+1} &= \max(0, S_{t+1-d} - mw) + A_t + A_{t-1} + \cdots + A_{t-d+1} \\ &= \max(0, S_{t+1-d} - mw) + \sum_{i=0}^{d-1} A_{t-i} \end{aligned} \quad (4.27)$$

This equation is apparent. Consider the system state in slot $t + 1$. Since it takes d slots for a cell to get serviced, the arrivals for the $d - 1$ most recent slots, i.e., $\{t - d + 1, \dots, t\}$, wait in the queue while the cells from the slot d slots ago, $t + 1 - d$, get serviced and leave the system. Following the same derivation as that given in Section 4.1.1 we can derive the *p.g.f.* for S and Q :

$$S(z) = [A(z)]^d \cdot Q(z) \quad (4.28a)$$

where

$$Q(z) = \frac{\sum_{j=0}^{mw-1} \pi_j (z^{mw} - z^j)}{z^{mw} - [A(z)]^d} \quad (4.28b)$$

Note the change in the denominator of $Q(z)$. The stability criterion for the steady-state to exist is

$$\rho = \frac{\mathbb{E}[A] \cdot d}{mw} < 1 \quad (4.29)$$

Similarly, using the analyticity of $Q(z)$ on $|z| = 1$ we get

$$Q(z) = \prod_{j=1}^{nvd-mw} \left(\frac{1 - z_j}{z - z_j} \right) \quad (4.30)$$

where z_j , $1 \leq j \leq nvd - mw$, are roots of the CE, $z^{mw} - [A(z)]^d = 0$, lying outside the unit circle in the complex- z plane. We can obtain the PMF for Q by taking the inverse z -transform of $Q(z)$ as given in (4.30). The PMF for S can then be determined by using (4.28a) and (4.30). The behavior of the various performance measures show the same trend as already discussed in Section 4.2.

Chapter 5

Algorithm for Concentration

5.1 Packet Concentrator as a Shared Buffer

In the previous chapter we evaluated the performance of the input queued crossbar concentrator. It was shown that under assumptions of *i.i.d.* arrivals and a discrete time scale an input queued $nv \times mw$ concentrator can be modeled by a GI/D/ mw queue, in particular, by a Binom/D/ mw queue for a Bernoulli process. These results were obtained assuming a work conserving queuing system i.e. $\min(k, mw)$ input cells are always concentrated in a slot if there are k , $k \geq 1$, cells at the input queues. A concentrator with capacity mw cannot do better than this as the maximum number of cells which can be accommodated at the outputs in one slot is mw . Thus, this analysis illustrates the best case performance possible for such a concentrator under identical input traffic streams.

Thus, the queuing model indicates that a shared buffer with an output rate of mw cells per slot would give the best performance for a $nv \times mw$ concentrator. An obvious way to achieve this performance is to put a shared buffer with the concen-

trator. However, a physical implementation of such a shared buffer is not scalable. It is easily seen that such a buffer would need to operate $mw/w = m$ times faster than the outputs and $nv/v = n$ time faster than the inputs. To overcome these speed limitations shared buffers are usually implemented as concentrators by parallel independent buffers filled in a round-robin (cyclic sequential) manner. The sequential assignment of cells across parallel input queues is enabled by additional structures like reverse banyan networks [3] or running adders [25]. All such methods increase the complexity of the concentrator and thus negate the advantage gained by reducing the complexity of the connection fabric in the sparse crossbar $Q(n, m, v, w)$.

This naturally leads to the question of how to achieve or to approach as close as possible to the performance of a shared buffer system using independent buffers and the sparse crossbar fabric. Obviously, this would involve a scheme or algorithm to assign output addresses to the cells at the inputs so that they can then be routed over the crossbar, if we do not want to increase the complexity by adding more hardware. Moreover, in addition to the constraint of trying to emulate a shared buffer, any such scheme also has to take into account the limitations posed by the asymmetric nature of the sparse fabric. Before formulating an algorithm which can route the cells we evaluate the limitations on routing posed by the sparse crosspoint structure.

5.2 Routing on Sparse Crossbar Structures

Recall from Section 2.3 that the minimum complexity connection fabric for any $Q(n, m, v, w)$ can be divided into three sections viz. the slim, ladder and fat sections. A sparse crossbar $Q(n, m, v, w)$ guarantees the concentration of any mw cells out of nv cells at the head of the input queues where each of the n inputs contributes not more than v cells. The structure in Chapter 2 was developed without considering packet losses or buffering, so each input cannot have more than v cells (equal to the maximum input rate) in a slot for concentration. But if we buffer the cells then it is possible that an input has more than v cells buffered while the total number of cells at all the inputs together, say x , is less than mw . If the input with more than v cells does not have the connectivity to transfer all its cells to the outputs then we cannot get the concentration of these x cells even though $x < mw$, thus, a shared buffer cannot be emulated in such a case. This is more likely at a slim input because of its lower connectivity. Such a situation is depicted in Figure 5.1.

In Figure 5.1 input 1 (I_1) and input 3 (I_3) have 5 and 2 cells respectively in their queues. All the other input queues are empty. The outputs can sink 4 packets/slot and the inputs can have a maximum batch arrival size of 2 i.e. $w = 4$ and $v = 2$. Assume capacity $c = mw > 7$. Thus, we see that even though the total number of cells is less than mw , the cells at the inputs cannot be concentrated to the

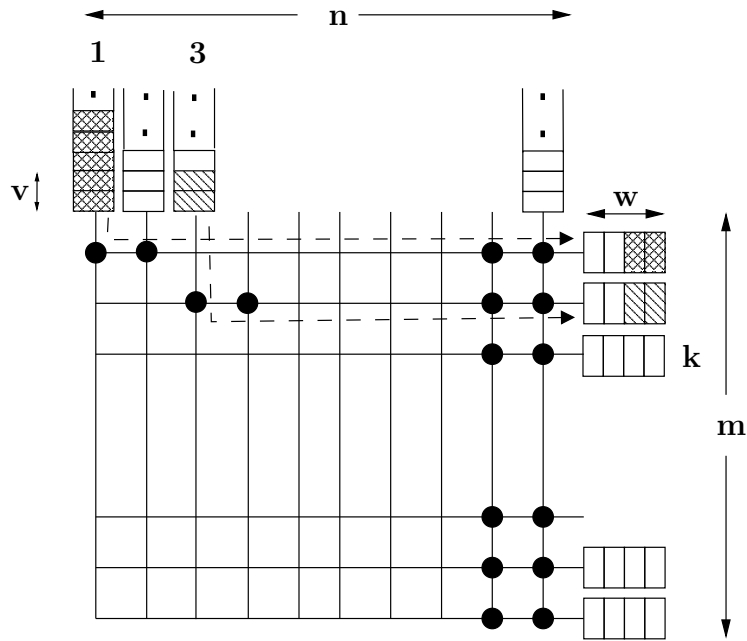


Figure 5.1: Blocking due to sparse connectivity

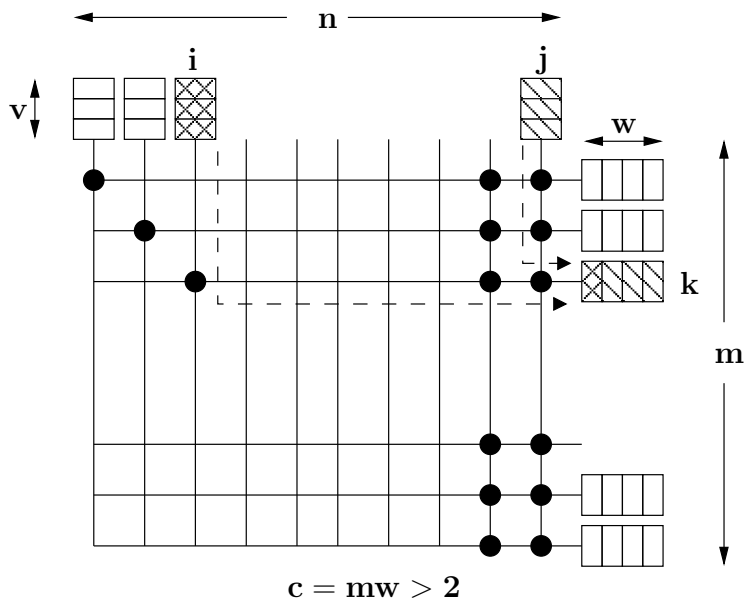
outputs ¹. We observe that this situation will arise more often when the more sparsely connected inputs have large queue sizes while other input buffers have a low occupancy. Thus, this observation indicates that the routing should be done in a fair manner for all the inputs. When no input gets a lower/higher share of service than rest of the inputs, then for identical traffic arrival patterns all the input queue lengths remain balanced and hence the inputs with lesser connectivity are not adversely affected. Since we will consider identical traffic at all inputs, a simple cyclic scheme should suffice to ensure fairness.

¹Note that according to our model each input routes not more than v cells per slot and hence input 1 routes only 2 cells to output 1. Even if input 1 could route faster (with faster crosspoints), it can never route more than 4 cells to output 1 in any case, and so capacity, c , is not achieved.

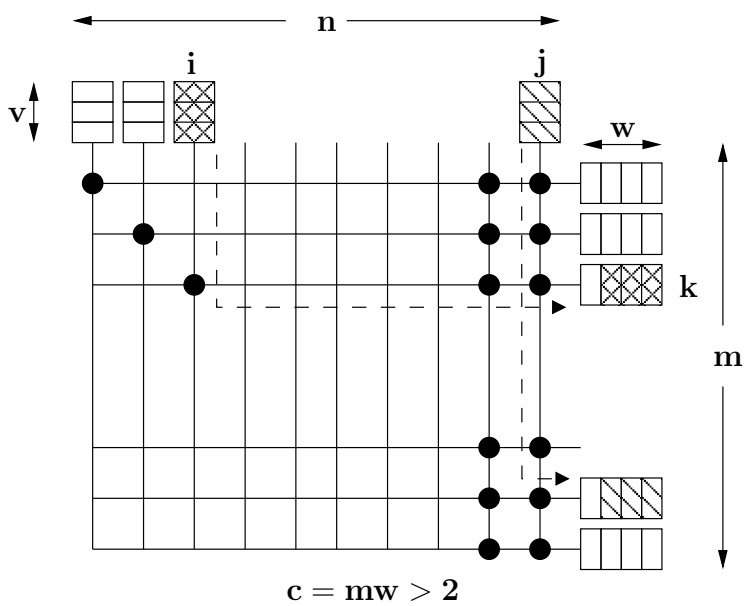
Now, if we assume that a scheduling algorithm proceeds in a cyclic round-robin fashion to ensure fairness amongst the inputs, then another kind of blocking can occur when, in a slot, we start the scheduling of cells from the fat or ladder section and proceed in a cyclic fashion. The inputs in the fat section can access any output and so could easily fill up the output which can be accessed by a slim (ladder) input if they assign outputs to cells randomly. This can be illustrated by the situation shown in Figure 5.2.

In Figure 5.2(a) we consider a $Q(n, m, v, w)$ with capacity $c (= mw) > 2v$. Assume that the slim input i has v cells to be scheduled in this slot and the fat input j also with v cells to be scheduled in this slot and all other inputs have no cells to be scheduled. The only output accessible to input i is output k , whereas input j is fully connected and can access all outputs. We start the scheduling cycle for this slot from input j as the cycle ended at input $j - 1$ in the previous slot. If we assign all the v cells at input j to output k then it is easy to see that input i can not concentrate all its cells to the output side even though the concentration capacity is not exceeded, i.e. the total number of cells at the inputs is less than c . Thus, even though the structure of the concentrator allows up to c cells at the inputs to be concentrated, we are not able to do that due to injudicious scheduling of input cells. The situation can easily be rectified by assigning the cells at input j to an output other than k as seen in Figure 5.2(b).

But we have to note that input j has no way of knowing the queue state (number of cells) of any of the slim inputs, if j is accessed before i in a slot. Hence, it does



5.2(a): Cell blocking.



5.2(b): Cell reassignment to prevent blocking.

Figure 5.2: Cell blocking in a $Q(n, m, v, w)$.

not know at which output blocking can occur. This situation can be eliminated by “marking” the requisite cells at input j for routing after the cells which have to be concentrated at the other inputs are routed. Note that the “marked” cells are not routed in the next slot, they are routed in the same slot but after the more sparsely connected inputs have been allowed to route cells. Thus, the routing algorithm needs to incorporate some sort of a cell marking scheme or in other words a priority scheme among the inputs whose cells are concentrated in the same slot with the most sparsely connected inputs getting the highest priority.

5.3 The Routing Algorithm

To summarize the previous section we list the properties the algorithm should possess to prevent blocking due to the fabric interconnection pattern and hence achieve good performance:

1. Fairness: All the inputs should see uniform service, which in the case of identical traffic means that the routing algorithm should cycle through the inputs in a round-robin fashion while assigning output addresses.
2. No blocking: The inputs should be given routing priority for concentration within a slot in proportion to the sparseness of their connections to the outputs to prevent the slim and ladder inputs from encountering blocking at the outputs. This means that the algorithm should first concentrate cells at the slim and ladder inputs in any given slot.

5.3.1 A Block-Packing Model

We show that an algorithm with the above properties can be mapped onto a block-packing problem. Within this framework, we can describe the concentration algorithm in an intuitive and geometric fashion. First, we define some terms used in the discussion below:

I_i Input i , $1 \leq i \leq n$.

O_j Output j , $1 \leq j \leq m$.

X Incidence matrix for the sparse crossbar connection fabric. This is a $m \times n$ matrix with $X_{ij} = 1$ if input j is connected to output i and $X_{ij} = 0$ otherwise.

R_i The set of all outputs which can be accessed by input i . Formally, $R_i = \{k : X_{ki} = 1\} \forall i = 1, 2, \dots, n$. I_i can access O_j only if $j \in R_i$.

Consider a box of width m and height w , so that it can hold a maximum of mw blocks. Every input cell is mapped onto a block. Upon being scheduled for concentration, up to v cells at the head of an input buffer are dropped into the compartmentalized box. See Figure 5.3 which shows a box for a $Q(12, 6, 2, 3)$. Each of the m columns of the box holds cells destined for a specific output i.e. column j holds cells routed to O_j . The label on a cell denotes the input port from which it has arrived. Note that cells with label i can only be dropped in columns belonging to R_i . We will drop cells into this box to show the output address assignment scheme. The cells in the bottom row of the box in Figure 5.3 at columns 1, 2 and

3 are cells from inputs 1, 2 and 8 assigned for concentration to outputs 1, 2 and 3 respectively.

Let the output port assignment of cells in slot t begin from I_1 , a slim input. At each input we consider the first v cells at the head of the queue for concentration in any given slot. A cell counter keeps track of the number of cells pushed into the box in one slot.

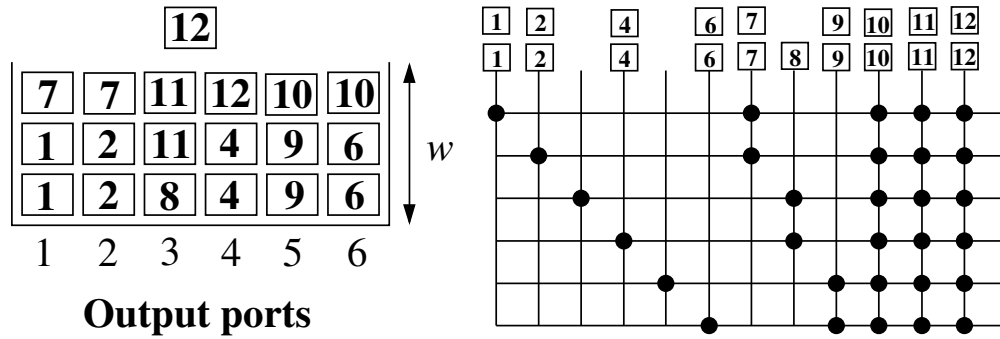
The first v cells at the buffer of I_1 are now dropped into the box in columns corresponding to its range R_1 .² The cell counter is updated by the total number of cells dropped in the box. Now we move to I_2 and drop the first v cells in its buffer into the columns corresponding to R_2 , update the cell counter by the number of cells dropped and proceed to the next input. If an input is in the fat or ladder section then the cells are not immediately dropped but are “marked” i.e. reserved for being dropped into the box at the end of the slot and the cell counter increased by the number of cells marked. We go on until the cell counter equals mw or we return back to I_1 . If there are any marked cells, then starting from the marked cells at the ladder inputs and proceeding towards the fat inputs, the marked cells are dropped into the remaining empty spaces. The column in which a cell is dropped corresponds to the output to which it is assigned. All the columns are cleared at the end of the slot by routing the cells to their outputs. Let the last input in

²If the number of cells in the buffer are less than v then all the cells in the buffer are dropped in the box. Since I_1 is slim, there is only one column in its range and we can route to it unambiguously.

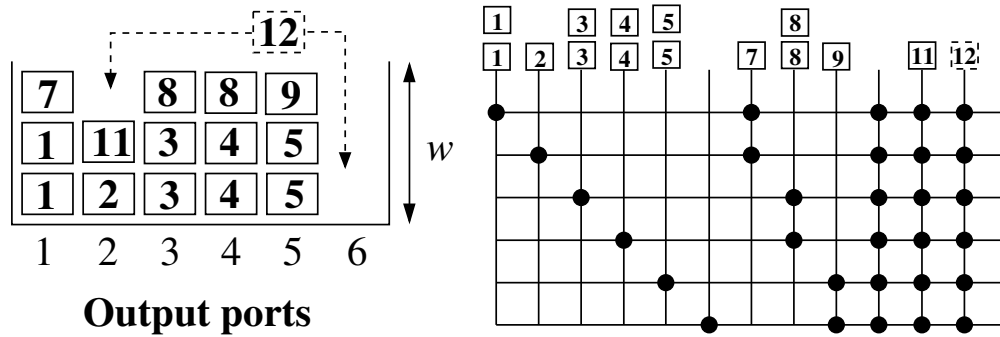
this cycle be denoted by I_{last} . In slot $t + 1$, the round is started from the input immediately after I_{last} in cyclic order i.e. if $I_{last} = n$, then in $t + 1$ concentration starts from input 1. There is an exception to this rule when I_{last} has $k, k \leq v$ cells at the head of its buffer but can drop only some of them, say $k', k' < k$ into the box because the box fills up to its capacity. In such a situation in slot $t + 1$ we start from I_{last} itself and drop the first $k - k'$ cells into the box or mark them, depending on whether I_{last} is a slim or ladder/fat input. These $k - k'$ cells are referred to as the residue.

By applying this algorithm to the example of Figure 5.3(a) it is easy to see that when we start from input 1 in slot t , $I_{last} =$ input 12 and one cell from input 12 cannot be put into the box. Thus, in slot $t + 1$, we will start from input 12 itself and this excess cell will be marked and dropped into the box only after the slim and ladder inputs have been served. See Figure 5.3(b). We can see that if the cell from input 12 in Figure 5.3(b) is not marked and instead routed to either of the outputs among $\{1, 3, 4, 5\}$ then one input cell from inputs in the set $\{1, 3, 4, 5\}$ is blocked.

We now give a heuristic to implement the algorithm outlined above. In the discussion below, scheduling of a cell at an input refers to routing of that cell to its output if the input is slim, and marking of a cell for subsequent routing at the end of the slot if the input is in the fat/ladder section. The sequence of steps is for a given slot, say t . For $t = 0$ we start from input 1. The steps in sequence are:



5.3(a): Time = t .



5.3(b): Time = $t + 1$.

Figure 5.3: Block-packing analogy.

1. Go to the starting input.
2. If the current input is the starting input and it has a residue from slot $t - 1$ then schedule the first $\min(mw, \text{size of residue})$ cells in the buffer.
 else schedule the first $\min(k_t, v, mw - N_t)$ cells at the head of the current input's buffer where k_t is the total number of cells in the current buffer in slot t and N_t is the number of cells scheduled at all inputs till now in slot t .
3. If ($N_t < mw$ and the starting input for slot t is not accessed a second time) go to the next input in cyclic order and repeat step 2 else proceed to step 4.
4. If mw cells have been scheduled i.e. $N_t = mw$ and if a residue exists at the last input accessed in slot t (I_{last}) then set starting input for slot $t + 1$ as I_{last} .
 else if $N_t = mw$ and no residue exists at I_{last} then set starting input for $t + 1$ as the next input in cyclic order after I_{last} .
 else (starting input is accessed a second time) if current input (which is the starting input) buffer being accessed is non empty and has less than v cells scheduled for slot t then schedule additional cells at this input until $N_t = mw$ or the buffer has no more cells, whichever comes first. Set the starting input for slot $t + 1$ as the next input in cyclic order.
5. Route all the marked cells for slot t to empty spaces. Advance time to $t + 1$ and repeat from step 1.

This algorithm ensures that the concentration capability of the crossbar structure is fully utilized in the sense that whenever there are $k \leq mw$ cells in the first nv locations at the input buffers, they are always concentrated in one slot. Note that this algorithm is not always work conserving as we do not consider more than v cells for concentration at one input in any slot. This is due to the limitation on the speed of the internal cross-points, if we can speedup the fabric then work conserving schemes can be implemented in the same fashion by considering more than v cells at an input for concentration in a slot.

Another observation which can be made is that we can use the concept of marking cells in a more general fashion to reserve output bandwidth for selected inputs. This can be achieved by using a suitable criterion, like a weight assignment to inputs based on a cost function to mark more cells at some inputs as compared to others.

5.4 An Approximate Analysis

We now present an approximate analysis of the $Q(n, m, v, w)$ with routing of cells according to the presented algorithm for saturated inputs. By saturation we mean that a new cell is always available to fill up the buffer as soon as a cell is routed to the outputs. It is easy to see that in the saturated case all the nv locations at the head of the input buffers will be full. Hence, we will always be able to find mw cells for concentration in a slot which in turn means that the system is

work conserving. Thus, we expect the statistics for the queue size and delay to approach the results derived for the shared buffer in Chapter 4 when the queues are saturated.

A polling model is used to analyze our routing algorithm in the saturated case. A polling model is a multiple queue (n queues in our case) cyclic service system where a server polls the queues for service in cyclic order [26]. Some terms used in describing polling systems are:

- Cycle time (C)—The time taken (in terms of number of slots in our case) between two successive pollings of the same input queue.
- Walk time or switch-over time (s_i)—The amount of time taken to switch from input i to input $i + 1$ in cyclic order.
- Gated system—A polling system is called a gated system when an input port transmits all the cells in its buffer when it is polled, but none of the messages that arrive after it is polled.
- Limited service—A limit is placed on the maximum number of cells which can be transmitted before the next input in sequence is polled.

Therefore, our system is best described by a v -limited gated service polling system. Note that in our system there is no time taken to switch from one input to the next except at the slot boundaries. For a gated limited system the following conditions are required for stability of the queues [27]:

$$\sum_{i=1}^n \rho_i = \rho < 1 \tag{5.1a}$$

where ρ_i is the offered load at input i and ρ is the total offered load, $\rho_i = \lambda_i \bar{h}_i$

where λ_i is the input cell rate and \bar{h}_i is average service time for input i .

$$\frac{\lambda_i \bar{s}}{1 - \rho} < k_i \quad (5.1b)$$

where k_i is the maximum number served at input i and $\bar{s} = \sum_{i=1}^n \bar{s}_i$ is the mean ring latency.

For the symmetric system of the concentrator $k_i = v \forall 1 \leq i \leq n$ and for the Bernoulli arrival process $\lambda_i = vp$. $E[C]$ and \bar{s} are related by the following balance equation:

$$E[C] = \frac{\bar{s}}{1 - \rho} \quad (5.2)$$

Substituting (5.2) in (5.1b) we get

$$\begin{aligned} vp \cdot E[C] &< v \\ \Rightarrow E[C] &< \frac{1}{p} \end{aligned} \quad (5.3)$$

Since we always find mw cells at the head of the queues,

$$E[C] = \frac{nv}{mw} \quad (5.4)$$

From (5.3) and (5.4) we get the stability criterion

$$\frac{nvp}{mw} < 1 \quad (5.5)$$

Recall that this is the same criterion for stability as derived in (4.15) for the shared buffer system. Also, even from (5.1a) we again get the same result as that in (5.5). Thus, we see that our system approaches the work conserving system of Chapter 4 near saturation.

5.5 Simulation Results

We simulated the performance of the proposed routing algorithm for a $Q(8, 6, 1, 1)$. The results were again executed until the estimate of the average queue length reached with probability 0.95 a relative width of the confidence interval equal to 5%. The estimation of the confidence interval width was obtained by the replication method.

The plots for the average queue length at an input \bar{Q} and the average queuing cell delay at an input \bar{W}_q analogous to those defined in Chapter 4 are given in Figure 5.4 and Figure 5.5 respectively. The equivalent shared buffer quantities are also plotted for comparison. Obviously, the average queue length at an individual input is $1/n$ times the value for \bar{Q} derived in Chapter 4 for the total queue, while the average delay, \bar{W}_q is the same. Therefore, for the case of routing using our algorithm we have added the average queue lengths for all the n inputs and then plotted the result to enable a comparison. We see that the two plots are almost identical. This shows that the algorithm is able to successfully emulate the shared buffer to a good degree.

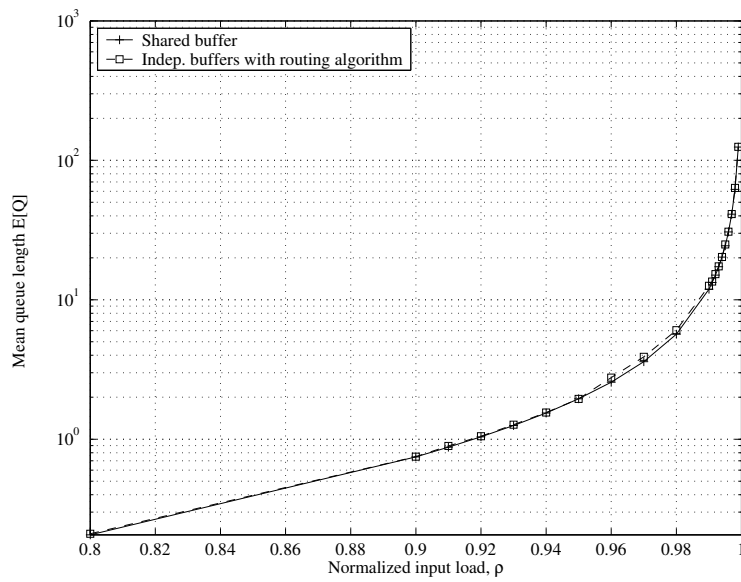


Figure 5.4: Mean queue length, \bar{Q} for a $Q(8, 6, 1, 1)$

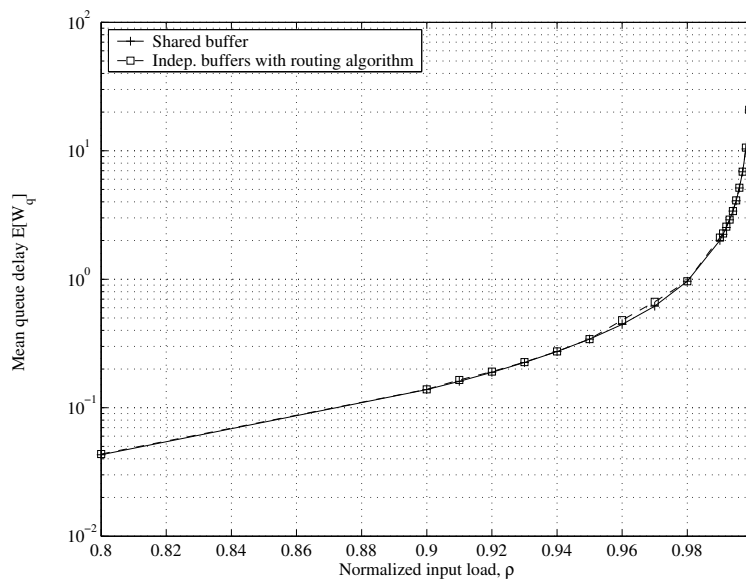


Figure 5.5: Mean queue delay, \bar{W}_q for a $Q(8, 6, 1, 1)$

Chapter 6

Conclusions and Future Work

Packet-switching concentrators having multiple packets being sourced at inputs and routed to outputs have a more generalized connection pattern than the traditional circuit-switched concentrators. Thus, it is of interest to investigate the construction and performance of such structures. In this thesis we started with a presentation of the construction of optimal complexity sparse-crossbar concentrator fabrics. In the subsequent development a few bounds were derived for packet loss when packets in excess of the concentrator capacity are simply dropped at the inputs. Here we considered cases when the routing over the concentrator is both random and deterministic. It was shown that for purely random routing the bounds converge to a limit when the concentration ratio is kept constant and the number of inputs is unbounded. After this we developed a statistical queuing model for the concentrator when excess packets are buffered at the inputs to reduce packet loss. It was shown that under the assumption of statistically identical traffic patterns at the inputs and a FIFO work conserving service discipline, the input queued concentrator behaves as a shared buffer. This analysis quantified

the best performance possible for the concentrator and gave us a yardstick with which to measure the goodness of any specific routing scheme for the concentrator. Finally, we tackled the problem of designing a routing algorithm which enables a sparse-crossbar concentrator with independent buffers at the inputs to emulate the performance of the shared buffer. A heuristic routing algorithm for the concentrator was developed after deriving the necessary features of the algorithm from certain key observations about the sparse crosspoint structure. It was shown that for identical input traffic patterns this algorithm is basically a round-robin polling scheme. Results were derived from simulation to show that such a routing scheme comes close to achieving the best performance possible. A few directions for further research and investigation include:

- Distributed routing: The round-robin polling structure of the proposed algorithm inherently makes it a centrally controlled scheme. It would be interesting to see how it can be modified to enable a distributed implementation.
- Implementation complexity: The crosspoint fabric gives us gains in terms of reducing the cross-point complexity, but we have not made an attempt in this thesis to quantify the complexity of the proposed routing scheme so that the overall tradeoff between the routing and the fabric structure can be fixed.
- Asymmetric traffic: Certain asymmetric traffic patterns may be efficiently concentrated by the sparse crosspoint fabric as it is itself inherently asym-

metric. Therefore, another direction for further investigation is determining the kind of traffic patterns which have a good match with the concentrator fabric. Equivalently, a weighting cost function could be assigned to inputs to model some quality of service criterion to vary the service provided to different inputs so that the service profile matches one of the asymmetric patterns mentioned above.

Appendix A

Consider a random variable $X \sim \text{Binom}(nv, p/m)$. We now show that the probability distribution of X is $\text{Poisson}(\lambda wp)$ in the limit $n \rightarrow \infty$ where $\lambda = nv/mw$.

A-1 Proof

Since X is Binomially distributed with parameters $(nv, p/m)$

$$x_k \triangleq \Pr(X = k) = \binom{nv}{k} \left(\frac{p}{m}\right)^k \left(1 - \frac{p}{m}\right)^{nv-k} \quad (\text{A-1})$$

Writing (A-1) in terms of $\lambda = nv/mw$, and taking the limit $n \rightarrow \infty$ on both sides, we get

$$\lim_{n \rightarrow \infty} x_k = \lim_{n \rightarrow \infty} \frac{nv!}{(nv-k)! k!} \left(\frac{\lambda wp}{nv}\right)^k \left(1 - \frac{\lambda wp}{nv}\right)^{nv-k} \quad (\text{A-2})$$

Using Stirling's formula, $n! \approx \sqrt{2\pi n} e^{-n} n^{n+1/2}$

$$\lim_{n \rightarrow \infty} x_k = \frac{nv^{nv}}{(nv-k)^{nv-k}} \frac{e^{-k}}{k!} \left(1 - \frac{\lambda wp}{nv}\right)^{nv} \frac{\left(\frac{\lambda wp}{nv}\right)^k}{\left(1 - \frac{\lambda wp}{nv}\right)^k} \quad (\text{A-3})$$

Using the limit, $\lim_{y \rightarrow 0} (1 + y)^{\frac{1}{y}} = e^y$, we get

$$\begin{aligned} \lim_{n \rightarrow \infty} x_k &= \frac{e^k e^{-k}}{k!} e^{-\lambda wp} (\lambda wp)^k \left(\frac{nv - k}{nv - \lambda wp} \right)^k \\ &= \frac{e^{-\lambda wp}}{k!} (\lambda wp)^k (1 + o(k/n)) \end{aligned} \tag{A-4}$$

Thus we see that as $n \rightarrow \infty$

$$x_k \triangleq \Pr(X = k) = (\lambda wp)^k \frac{e^{-\lambda wp}}{k!} \tag{A-5}$$

In other words, $X \xrightarrow{\mathcal{D}} \text{Poisson}(\lambda wp)$, i.e., X converges to a $\text{Poisson}(\lambda wp)$ random variable in distribution.

Appendix B

We derive an expression for the *p.g.f.* of the queue contents ($Q(z)$) of a GI/D/ mw system with batch arrivals in terms of the roots of the characteristic equation (CE), $z^{mw} - A(z) = 0$, where $A(z)$ is the *p.g.f.* of the *i.i.d.* batch arrival process. A closed form expression for the steady-state probabilities of the system contents is also derived in terms of the roots of the CE lying outside the unit disk in complex- z plane. Before proceeding to the proof, we list some general properties of *p.g.f.s.*

B-1 Properties of Probability Generating Functions

A discrete random variable, X , which takes non-negative integer values, has a *p.g.f.* $X(z)$ defined as

$$X(z) = \sum_{i=0}^{\infty} x_i z^i \quad (\text{B-1})$$

Where $x_i \triangleq \Pr(X = k)$. The value of $X(z)$ at $z = 1$ is

$$X(1) = \sum_{i=0}^{\infty} \Pr(X = i) = 1 \quad (\text{B-2})$$

Similarly,

$$\left. \frac{\partial X(z)}{\partial z} \right|_{z=1} = X'(1) = \sum_{i=0}^{\infty} i \Pr(X = i) = E[X] \quad (\text{B-3})$$

and

$$\left. \frac{\partial^2 X(z)}{\partial z^2} \right|_{z=1} = X''(1) = \sum_{i=0}^{\infty} i(i-1) \Pr(X=i) = E[X^2] - E[X] \quad (\text{B-4})$$

Therefore, the variance of X can be written in terms of the derivatives of $X(z)$ as

$$\text{Var}(X) = E[X^2] - (E[X])^2 = X''(1) + X'(1) - (X'(1))^2 \quad (\text{B-5})$$

Now we show that $X(z)$ is analytic inside and on the unit disk in the complex- z plane. Taking the value of $X(z)$ along any closed contour \mathcal{C} , where $\mathcal{C} = \{z : |z| \leq 1\}$, we get:

$$\begin{aligned} |X(z)|_{\mathcal{C}} &= \left| \sum_{i=0}^{\infty} x_i z^i \right|_{z \in \mathcal{C}} \\ &\leq \sum_{i=0}^{\infty} |x_i| |z|^i \\ &\leq \sum_{i=0}^{\infty} |x_i| \leq 1 \end{aligned} \quad (\text{B-6})$$

The inequality becomes an equality when $z = 1$. Since the power series is absolutely summable, $\forall \mathcal{C}$ described above, $X(z)$ is analytic in the region $|z| \leq 1$.

B-2 Probability Generating Functions for S and Q

We now derive closed form expressions for the system occupancy probabilities in steady-state for a GI/D/ mw system with batch arrivals.

Let S , A , and Q be the random variables for the number of cells in the system, arrivals in a single slot, and number of cells awaiting service in the queue in the

steady-state. See Section 4.1 for a more complete description of the definitions and underlying assumptions.

We can write the *p.g.f.* of the arrival process, $A=A(z)$ as

$$A(z) = \sum_{i=0}^{\infty} a_i z^i \quad (\text{B-7})$$

where $a_i \triangleq \Pr(A = i), i = 1, 2, \dots$. If the maximum batch size is nv then $a_i = 0 \forall i \geq nv$ and we get

$$A(z) = \sum_{i=0}^{nv} a_i z^i \quad (\text{B-8})$$

Therefore, $A(z)$ is now a polynomial of degree nv . It was shown in Section 4.1.1 that $S(z)$ is given by the following expression (See (4.8)):

$$S(z) = A(z) \frac{\sum_{j=0}^{mw-1} \pi_j (z^{mw} - z^j)}{z^{mw} - A(z)} \quad (\text{B-9})$$

where $\pi_j \triangleq \Pr(S = j), j = 0, 1, 2, \dots$ are the steady-state system state probabilities. We now state a key result used to derive the final result.

Theorem B-2.1 (Rouché's Theorem) *If $f(z)$ and $g(z)$ are analytic functions of z inside and on a closed contour \mathcal{C} , and $|g(z)| < |f(z)|$ on \mathcal{C} , then $f(z)$ and $f(z)+g(z)$ have the same number of zeros inside \mathcal{C} .*

From (B-2) we know that $z^{mw} - A(z) = 0$ has $z = 1$ as a root. Apply Theorem (B-2.1) to the CE, $z^{mw} - A(z) = 0$. Let $g(z) = -A(z)$, $f(z) = z^{mw}$ and $\mathcal{C} = \{|z| = 1\} - \{z = 1\}$. We know from (B-6) that $|g(z)|_{z \in \mathcal{C}} < 1$ and obviously $|f(z)|_{z \in \mathcal{C}} = 1$. Thus we see that both $f(z)$ and $g(z)$ are analytic on \mathcal{C} and $|f(z)| > |g(z)|$ on \mathcal{C} . Therefore, from Rouché's theorem both $f(z)$ and $f(z) + g(z)$

have the same number of zeros inside \mathcal{C} . It is obvious that $f(z) = z^{mw}$, has $mw - 1$ zeros inside and on \mathcal{C} with another zero at $z = 1$. Therefore, we can make the following statement:

The characteristic equation, $z^{mw} - A(z) = 0$, has mw zeros in the region $|z| \leq 1$.

Further, one of these zeros is at $z = 1$.

We know from (B-8) that the denominator of $S(z)$ is a polynomial of degree nv (as $nv \geq mw$ for a concentrator). Let the zeros of the CE, $z^{mw} - A(z) = 0$, in increasing order of magnitude be $z_1, z_2, \dots, z_{mw}, \dots, z_{nv}$. Thus, $|z_i| < 1$, $i = 1, \dots, mw - 1$, $z_{mw} = 1$, and $|z_i| > 1$, $i = mw + 1, \dots, nv$. Also, let the zeros of the numerator be α_i , $i = 1, \dots, mw$.

Thus, $S(z)$ can be written as

$$S(z) = A(z) \frac{C \prod_{j=1}^{mw} (z - \alpha_j)}{\prod_{j=1}^{nv} (z - z_j)} \quad (\text{B-10})$$

where C is a constant which will be determined later on. Now since $S(z)$ is a *p.g.f.*, we know from (B-6) that it is analytic in the region $|z| \leq 1$. This means that there can be no poles for $S(z)$ inside and on the unit disk. This in turn implies that the zeros of the numerator and the denominator in (B-9) lying in and on the unit disk should cancel out, i.e., $\alpha_i = z_i \forall i = 1, \dots, mw$. Thus we get

$$S(z) = A(z) \frac{C}{\prod_{j=mw+1}^{nv} (z - z_j)} \quad (\text{B-11})$$

We can determine the value of C using the property in (B-2). Therefore, $S(1)=1$

and $A(1)=1$ implies

$$C = \prod_{j=mw+1}^{nv} (1 - z_j) \quad (\text{B-12})$$

Finally, combining (B-11) and (B-12) we get;

$$S(z) = A(z) \prod_{j=mw+1}^{nv} \frac{1 - z_j}{z - z_j} \quad (\text{B-13})$$

Renaming the z_j s lying outside the unit disk as z_1, \dots, z_{nv-mw} and using $S(z) = A(z) \cdot Q(z)$ we get

$$Q(z) = \prod_{j=1}^{nv-mw} \frac{1 - z_j}{z - z_j} \quad (\text{B-14})$$

Note that here $|z_j| > 1$ as we have reordered the z_j s.

B-3 Steady-State Probability Distribution

The expressions for $S(z)$ and $Q(z)$ as given in (B-13) and (B-14) respectively can be used to obtain closed form expressions for the steady state probabilities $\pi_i \triangleq \Pr(S = i)$, $i = 0, 1, 2, \dots$. Breaking the right hand side of (B-14) into partial fractions we get

$$Q(z) = \prod_{j=1}^{nv-mw} (1 - z_j) \sum_{j=1}^{nv-mw} \frac{C_j}{z - z_j} \quad (\text{B-15})$$

where

$$C_j = \frac{1}{\prod_{k \neq j} (z_j - z_k)} \quad (\text{B-16})$$

Thus, for $S(z)$ we get

$$\sum_{i=0}^{\infty} \pi_i z^i = A(z) \prod_{j=1}^{nv-mw} (1 - z_j) \sum_{j=1}^{nv-mw} \frac{C_j}{z - z_j} \quad (\text{B-17})$$

Substituting $A(z) = \sum_{i=0}^{nv} a_i z^i$ and comparing the coefficient of z^k on both sides of (B-17) we get the following closed form expressions for the steady-state probabilities $\pi_k \triangleq \Pr(S = k)$:

$$\pi_k = \prod_{i=1}^{nv-mw} (1 - z_i) \sum_{j=1}^{nv-mw} -C_j \sum_{l=0}^k a_l z_j^{-(k-l+1)} \quad k = 0, 1, 2, \dots \quad (\text{B-18})$$

The expression is simplified for $k = 0$ to

$$\pi_0 = \prod_{i=1}^{nv-mw} (1 - z_i) \sum_{j=1}^{nv-mw} \frac{-C_j a_0}{z_j} \quad (\text{B-19})$$

The roots of the CE, z_j s, are easily obtained by using numerical methods like Newton-Raphson and then these formulas can be used to obtain the steady-state probability distributions for S and Q .

BIBLIOGRAPHY

- [1] F. A. Tobagi, “Fast packet switch architectures for broadband integrated services digital networks,” in *Proceedings of IEEE*, vol. 78, no. 1, Jan. 1990, pp. 133–167.
- [2] E. Gündüzhan and A. Y. Oruç, “Complexity and optimal design of packet switching concentrators,” *IEEE/ACM Transactions on Networking*, submitted for publication.
- [3] H. S. Kim, “Design and performance of Multinet switch: A multistage ATM switch architecture with partially shared buffers,” *IEEE/ACM Transactions on Networking*, vol. 2, no. 6, pp. 571–580, Dec. 1994.
- [4] M. H. Guo and R. S. Chang, “Multicast ATM switches: Survey and performance evaluation,” *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 2, pp. 98–131, Apr. 1998.
- [5] C. Y. Lee and A. Y. Oruç, “Design of efficient and easily routable generalized connectors,” University of Maryland, College Park, MD, Tech. Rep. UMIACS-TR-92-22 (CS-TR-2846), 1992.

- [6] M. G. Hluchyj and M. J. Karol, "Queuing in high-performance packet switching," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1587–1597, Dec. 1988.
- [7] D. X. Chen and J. W. Mark, "SCOQ: A fast packet switch with shared concentration and output queuing," *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, pp. 142–151, Feb. 1993.
- [8] K. Y. Eng, M. J. Karol, and Y. S. Yeh, "A growable packet (ATM) switch architecture: Design principles and application," *IEEE Transactions on Communications*, vol. 40, no. 2, pp. 423–430, Feb. 1992.
- [9] J. N. Giacomelli, *et al.*, "Sunshine: A high-performance self-routing broadband packet switch architecture," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1289–1298, Oct. 1991.
- [10] A. Huang and S. Knauer, "Starlite: A wideband digital switch," in *Proceedings of IEEE GLOBECOM'84*, Nov. 1984, pp. 121–125.
- [11] W. Wang and F. A. Tobagi, "The Christmas-tree switch: An output queuing space division fast packet switch based on interleaving distribution and concentration function," in *Proceedings of IEEE INFOCOM'91*, vol. 1, Bal Harbour, FL, USA, Apr. 1991, pp. 163–170.
- [12] Y. S. Yeh, M. G. Hluchyj, and A. S. Acampora, "The Knockout switch: A simple, modular architecture for high-performance packet switching," *IEEE*

- Journal on Selected Areas in Communications*, vol. SAC-5, no. 5, pp. 1273–1283, Oct. 1987.
- [13] W. M. Guo and A. Y. Oruç, “Regular sparse crossbar concentrators,” *IEEE Transactions on Computers*, vol. 47, no. 3, pp. 363–368, Mar. 1998.
- [14] A. Y. Oruç and H. M. Huang, “Crosspoint complexity of sparse crossbar concentrators,” *IEEE Transactions on Information Theory*, vol. 42, no. 5, pp. 1466–1471, Sept. 1996.
- [15] Y. W. Leung, “Design and analysis of a packet concentrator,” *IEICE Transactions on Communications*, vol. E-83B, no. 5, pp. 1115–1121, May 2000.
- [16] S. Y. R. Li, H. Li, and G. M. Koo, “Fast knockout algorithm for self-route concentration,” *Computer Communications*, vol. 22, no. 17, pp. 1574–1584, Oct. 1999.
- [17] R. Kannan, R. Bartos, K. Y. Lee, and H. F. Jordan, “SXmin: A self-routing high-performance ATM packet switch based on group-knockout principle,” *IEEE Transactions on Communications*, vol. 45, no. 6, pp. 710–722, June 1997.
- [18] K. L. E. Law and A. Leon-Garcia, “A large scalable ATM multicast switch,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 5, pp. 844–854, June 1997.

- [19] A. E. Kamal, “Efficient solution of multiple server queues with applications to the modeling of ATM concentrators,” in *Proceedings of IEEE INFOCOM’96*, vol. 1, San Francisco, CA, USA, Mar. 1996, pp. 248–254.
- [20] Y. S. Lin, C. B. Shung, and J. C. Chen, “Design of knockout concentrators,” in *IEE Proceedings on Communications*, vol. 145, no. 3, June 1998.
- [21] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968, pp. 530–531.
- [22] H. Bruneel and B. G. Kim, *Discrete-Time Models for Communication Systems Including ATM*. Boston: Kluwer Academic Publishers, 1993.
- [23] M. L. Chaudhry and N. K. Kim. (2001, July) A complete and simple solution for the discrete-time multi-server queue with bulk arrivals and deterministic service times. Working paper. [Online]. Available: [http://osl7.kaist.ac.kr/lab/Geom\(X\)Dc.pdf](http://osl7.kaist.ac.kr/lab/Geom(X)Dc.pdf)
- [24] S. K. Bose, *An introduction to queueing systems*. New York: Kluwer Academic/Plenum Publishers, 2002.
- [25] M.-H. Guo and R.-S. Chang, “Multicast ATM switches based on input cell scheduling,” *IEICE Transactions on Communications*, vol. E82-B, no. 4, pp. 600–607, Apr. 1999.
- [26] H. Takagi, *Analysis of Polling Systems*, ser. MIT Press Series in Computer Systems, H. Schwetman, Ed. MIT Press, 1986.

- [27] K. Chang and D. Sandhu, “Pseudo-conservation laws in cyclic-server, multi-queue systems with a class of limited service policies,” in *Proceedings of IEEE INFOCOM’90*, vol. 1, San Francisco, CA, USA, June 1990, pp. 260–267.

