# AN ATM SWITCH WITH DISTRIBUTED CELL SCHEDULING

*M. P. C. Arantes* [1], *S. Motoyama* [2]
[1] magda@dt.feec.unicamp.br
Faculdade de Valinhos
[2] motoyama@dt.fee.unicamp.br
DT – FEEC – Unicamp

***ABSTRACT** - An ATM switch with distributed cell scheduling is proposed in this paper. By using a crossbar switching structure with large buffers at input ports and small buffers at each crosspoint, the cell scheduling is distributed at input and at crosspoint buffers. The discrimination of incoming cells into service classes at input buffers, and the use of a modified virtual output queuing (VOQ) technique provide, to the proposed switch, facility to satisfy easily the QoS and a throughput of 100%. An example of performance analysis based on priority service classes is carried out and the results show a very promising ATM switch.*

## I. INTRODUCTION

Several high-speed ATM switching structures have been proposed in the literature [1]-[10]. A switching structure can be classified according to the buffer location, which affects the performance of the switch.

The switching structures with buffers at output ports or those that combine buffers at both input and at output ports present a theoretical throughput of 100%, but at expenses of either switching fabrics, or high speed memories, or internal speed-up. Therefore they are not attractive for high-speed backbone network applications, where the ATM switches have been intensively adopted.

An another alternative is to use buffers at the input ports, but this type of structures presents the well-known head of line blocking (HOLB) problem [1]. To overcome this limitation many efficient cell-scheduling algorithms have been proposed [2]-[5], and in [8] it was shown that HOLB problem could be completely eliminated by using virtual output queuing. The input buffering structures have two important characteristics for backbone applications. The memory capacity is only per link and they do not need neither sophisticated switch fabrics nor internal speed-up. The main drawback of the input buffering structures is the need for very high processing cell schedulers, which can limit the switching speed.

Recently, a crossbar switching structure using large buffers at input ports and small buffers at each crosspoint has been proposed [10]. In [10] the switching structure, which combines the virtual output queuing at each input port and small buffer at each crosspoint, uses a cell-scheduling algorithm based on two phases. In the first phase, at each virtual output queue, a scheduler selects a local cell to be transmitted. In the second phase, at each output port, another scheduler selects the final cell to be transmitted. The structure proposed in [10] seems very interesting for backbone applications as the processing of schedulers can be distributed at input and crosspoint queues thus avoiding the main drawback of the input buffering structures.

In this paper a combined input and crosspoint-queued switching structure is used, but differently from [10], in our approach the cells are discriminated into service classes at input buffers and by using a modified virtual output queuing technique, it is proposed a highly efficient switch capable of satisfying easily the QoS of each class of service.

This paper has the following organization. In section 2, the proposed switching structure is described. The cell scheduling algorithm proposal and the analytical model are presented in section 3. In section 4, the switch performance analysis is carried out. Finally, in section 5 the main conclusions are presented.

## II. THE PROPOSED STRUCTURE

A novel structure is presented in Fig. 1. In each input port a set of N buffers is provided, one buffer per output port. Each of N buffers at each input port consists of logical separate queues for each service class. At each input port, after header translation and header swapping each cell receives a time stamp used for scheduling purpose. Then a cell is discriminated according to output port and by service class and it is stored at one of virtual service class queues. At each input buffer an input scheduler ($\boldsymbol{IS_{ij}}$) is provided for the selection of a cell to be transmitted by using an appropriate scheduling algorithm. The selection being local, it means that the selected cell is the next to be transmitted in that buffer. Since an input scheduler is placed at each input buffer in all input links, and the scheduling algorithm can be run in parallel, an aggregate high processing schedulers can be achieved. Each input buffer has a separate line connecting to a buffer placed at the corresponding crosspoint (*XP*) and to an auxiliary buffer (*AB*) as is shown in Fig.1. The crosspoint buffer can accommodate only one cell and the auxiliary buffer can accommodate a local information (*LI*) from

each input buffer. The *LI* (time stamp and the service class information) is transmitted to the auxiliary buffer (*AB*) used by crosspoint scheduler (XS) placed at each output link. By using *LI*, XS runs the appropriate scheduling algorithm to select the next cell to be transmitted and authorizes the corresponding crosspoint buffer to transmit through a selected line. During the scheduling time a cell can be transmitted from input buffer to crosspoint buffer. Since each output port has a crosspoint scheduler and each one can run independently, an overall high capacity scheduling can be obtained.

## III. PERFORMANCE ANALYSIS MODEL

The type of scheduling algorithm to be adopted affects the performance of the proposed switching structure. As an example for performance analysis, it is used a non-preemptive priority scheme as the scheduling algorithm. The input scheduler always chooses the packet with higher priority, and if there are more then one cell with the same priority it chooses the cell that has the longest delay time.

The scheduling algorithm used to select the cell, which shall be transmitted from the buffers at input port to the crosspoint buffer is described, considering that the service classes $C_1$, $C_2$, $C_3$, $C_4$ and $C_5$ have descending priority order. The scheduler at input $i$ and output $j$ ($IS_{ij}$) examines first if $C_1$ class service has any cell to transmit. If any cell is waiting in that queue, the cell that has the longest delay time is chosen and is transmitted to the buffer at crosspoint $ij$ ($XP_{ij}$). At same time, $LI_{ij}$ is transmitted to the auxiliary buffer ($AB_j$). If there are not any cells waiting in the $C_1$ logical queue, then the $C_2$ logical queue is examined, and so on, until the $C_5$ logical queue had been examined. This procedure runs in parallel at each input scheduler.

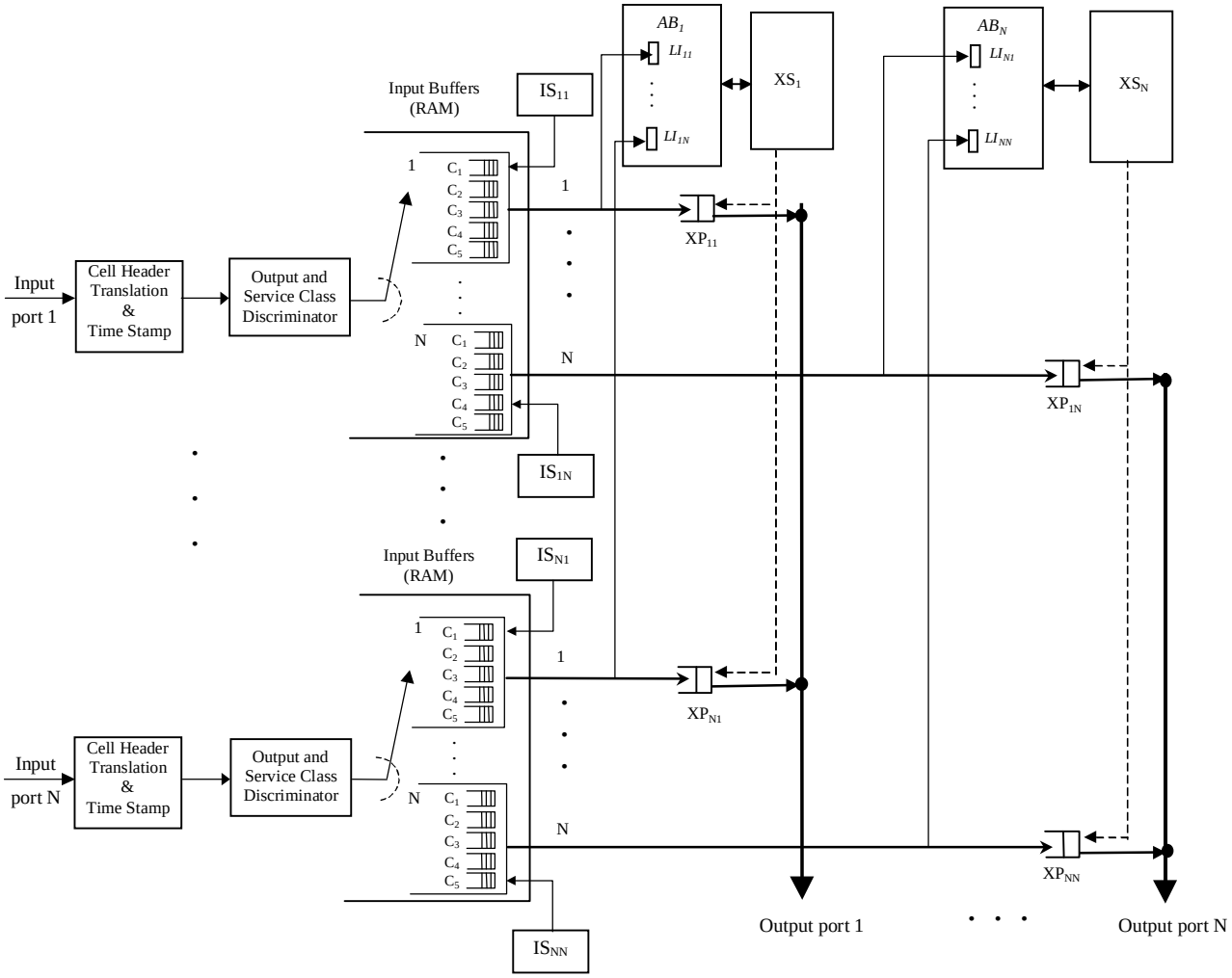


**Figure 2 1:** The proposed switching structure with large buffers at input port and a small buffer at each crosspoint.

The second scheduling algorithm is used for determining the cell that should be transmitted from the buffers at crossing points *($XP_{i,j}$ , i=1..N)* to output port *j*. The scheduler at output *j* (*$XS_j$*) uses the *$LI_{ij}$* to select a cell, obeying the descending priority order criteria. If more then one cell with the same priority are waiting, the one with the longest delay time is chosen to be transmitted at the next time slot. Since the above procedure is used in each time slot, the $C_5$ service class may not be served for many slots. But, the proposed scheduling algorithm is very efficient to attend time constraint services so that the quality of services (QoS) can be satisfied. All output schedulers are simultaneously doing the above procedure, so that a simple and very high speed switch can be implemented.

The cells of each service class are distributed among all output buffers and they are served according to non preemptive priority order. For analysis, we can consider the aggregate model as only one buffer for each service class and a server corresponding to an output line, as it is shown in Fig. 3.1.

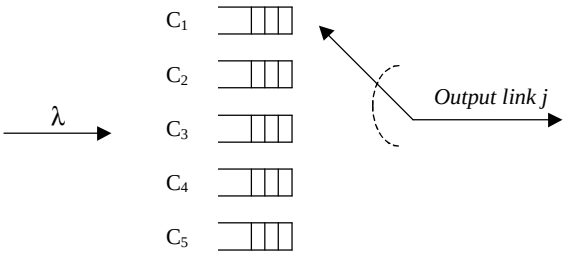


**Figure 3.1:** Queue aggregate model at *output link j*.

In this model, at each time slot, the server looks first for a cell to transmit in the ***$C_1$*** buffer that has the highest priority class. If there are not any cell waiting at ***$C_1$*** buffer, then the server goes to the next ***$C_2$*** buffer and so on, until ***$C_5$*** class is served.

For the analysis the followings assumptions are made. The number of input or output ports is *N* and the cell arrivals on the input ports obey independent and identical Bernoulli processes. The probability of a cell arrival in a time slot is *p*. $S_i$ is the traffic percentage of service class *i*, such that $\sum_i S_i = 1$. The probability of a cell arrival at input port being routed to a particular port is equal to *1/N*, therefore, the traffic at each aggregate buffer is $NpS_i\frac{1}{N}$. It is assumed that there are *r* service classes and a generic service class of priority *h* can assume *r* values, i.e. $h = 1, 2, 3, ..., r$ where *r* represents the lowest priority service class. The cells with a same priority are served in the FIFO (first in first out) discipline.

It is assumed that cell slots at input and outputs ports are not synchronized so that there is some residual time of service when a target cell arrives. It is also assumed that the cells are served in the same slots they arrived.

Using similar reasoning developed in [11] for priority queuing system with non-preemptive service discipline, the average cell waiting time $E\{W_h\}$ can be written as

$$E\{W_h\} = E\{T_0\} + \sum_{h=1}^{h} E\{T_k\} + \sum E\{T'_k\} \qquad \textbf{(1)}$$

where

$E\{T_0\}$ is the average time (residual time) to finish the transmission of a cell, when the target cell arrives;

$E\{T_k\}$ is the average time to serve all queued cells with the same or higher priorities $(h, h-1, h-2, ..., 1)$ when the target cell arrives, and

$E\{T'_k\}$ is the average time to serve all cells with higher priorities $(h-1, h-2, ..., 1)$ that arrives during the period $E\{W_h\}$ seconds and that will be served before the target cell.

$E\{T_k\}$ is given by

$$E\{T_k\} = E\{m_k\} \cdot T_{slot} \qquad \textbf{(2)}$$

where $E\{m_k\}$ is the average number of cells waiting for the service with higher priorities or the same priority but that arrives before the target cell and $T_{slot}$ is the time to transmit a cell.

From Little's formula we obtain

$$E\{m_k\} = \frac{S_k p}{T_{slot}} E\{W_k\} \qquad \textbf{(3)}$$

where $E\{W_k\}$ is the average waiting time of cells with priority *k*.

Thus,

$E\{T_k\} = S_k p E\{W_k\}$ and similarly,

$$E\{T'_k\} = S_k p E\{W_h\} \qquad \textbf{(4)}$$

Therefore, equation (1) can be solved for $E\{W_1\}$ then for $E\{W_2\}$ and by induction we obtain

$$E\{W_h\} = \frac{E\{T_0\}}{(1 - p\,\sigma_{h-1})(1 - p\,\sigma_h)} \qquad \textbf{(5)}$$

where, $\sigma_h = \sum_{k=1}^{h} S_k$ , $\sigma_0 = 0$ , $\sigma_r = 1$

For fixed slots size, $E\{T_0\}$ is given by

$$E\{T_0\} = \sum_{k=1}^{r} \frac{N-1}{N} \frac{p}{2} S_k T_{slot} = \frac{(N-1)p}{2N} T_{slot} \quad \textbf{(6)}$$

Thus,

$$E\{W_h\} = \frac{(N-1)p}{2N(1-p\sigma_{h-1})(1-p\sigma_h)} T_{slot} \quad \textbf{(7)}$$

For the case in which cells are served in subsequent slots, we have

$$E\{W_h\} = T_{slot}\left[1 + \frac{(N-1)p}{2N(1-p\sigma_{h-1})(1-p\sigma_h)}\right] \quad \textbf{(8)}$$

## IV. PERFORMANCE ANALYSIS

For the performance analysis, the case of the ATM service classes is considered. Therefore $C_1$, $C_2$, $C_3$, $C_4$ e $C_5$ stand for Constant Bit Rate (CBR), non real time Variable Bit Rate (nrtVBR), real time Variable Bit Rate (rtVBR), Available Bit Rate (ABR) and Unspecified Bit Rate (ABR), respectively.

Fig. 4.1, 4.2 and 4.3 show the performance of proposed switching structure using Eq. 8 above. In these figures, the different situations of load distribution among the service classes are presented for the 16x16-size switch.

In Fig. 4.1, it is considered load percentages of 40%, 20%, 20%, 10% and 10% for CBR, rtVBR, nrtVBR, ABR, and UBR respectively. It is observed that only the UBR service class has a long average waiting time for load below 90%. All the other service classes have reasonable waiting time. For the higher priorities services, CBR and rtVBR the waiting time is smaller than a three slots time for any load situation.

In Fig. 4.2(a) and Fig. 4.2(b), it was assumed that the network traffic is equally distributed among service classes. In this case, the cell delay times of higher priority service classes still keep small while the cell delay times for smaller priority service classes decrease.

In Fig. 4.3, it was considered that the lowest priority service classes (ABR and UBR) have 70% of the network load. In this case, the cell delay times for the higher priority classes CBR, rtVBR, nrtVBR and ABR are very small, and only UBR traffic, the smallest priority class, has considerable delay for load above 85%.

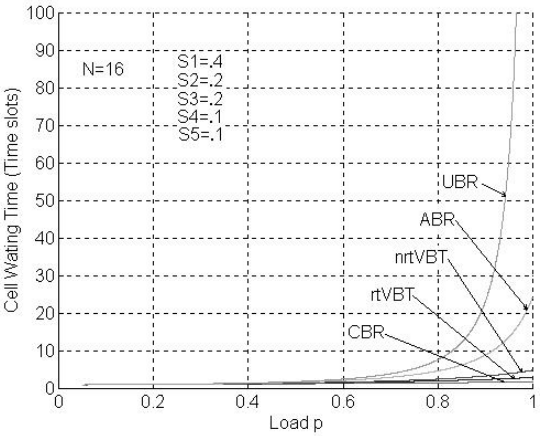


**Figure 4.1 (a)**

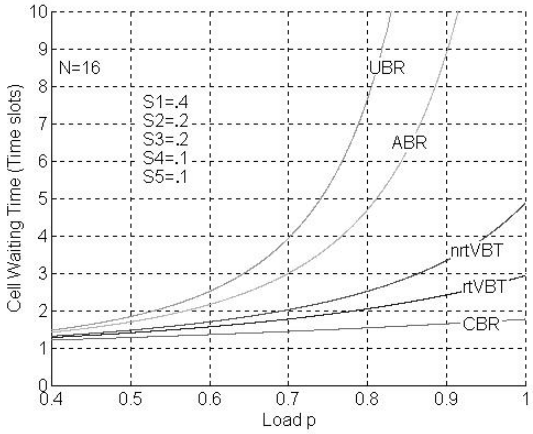


**Figure 4.1(b)**

**Figure 4.1:** Cell Waiting Time for a 16x16 switch with 80% of traffic in the higher priority service classes (CBR, rtVBR and nrtVBR).

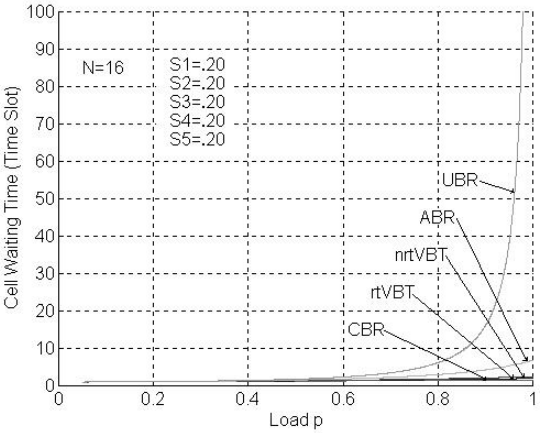


**Figure 4.2 (a)**

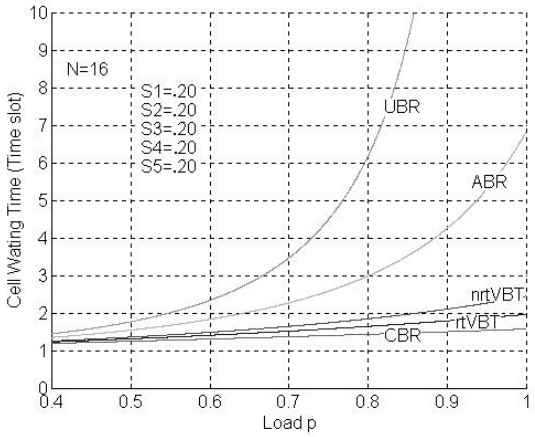


**Figure 4.2 (b)**

**Figure 4.2:** Cell Waiting Time for a 16x16 switch with the traffic equally distributed among service classes.

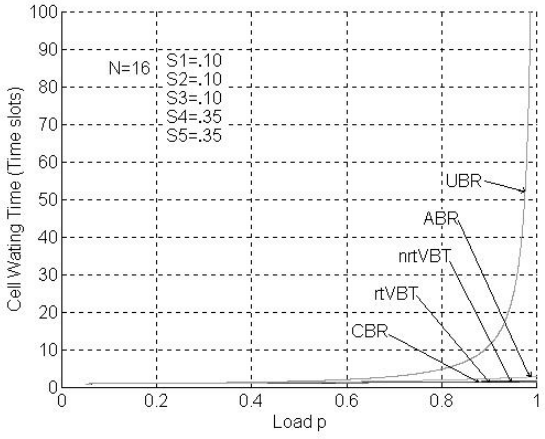


**Figure 4.3 (a)**

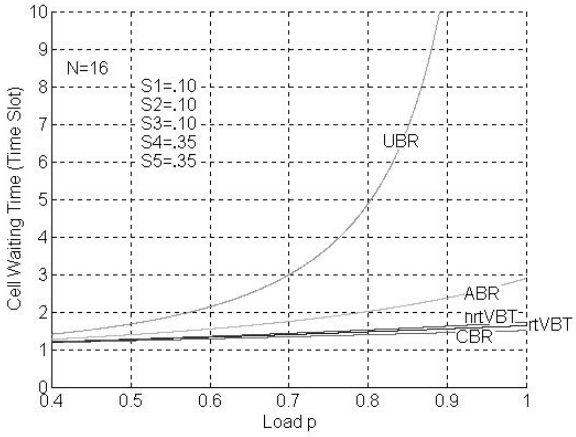


**Figure 4.3 (b)**

**Figure 4.3:** Cell Waiting Time for a 16x16 switch with 70% of traffic in the lower priority service classes (ABR and UBR).

## V. CONCLUSIONS

In this paper, a switching structure with large buffers at input ports and small buffers at each crosspoint and a cell scheduling based on distributed concept was proposed.

By discriminating the incoming cells into service classes and by using a modified virtual output queuing (VOQ), the cell scheduling can be distributed at input and at crosspoint buffers.

The proposed switch has throughput equal to 100% and using the proposed algorithm the cell delay time for each class of service can be controlled to satisfy easily the QoS. In conditions of networking traffic around 80% in higher priority classes (40% of CBR, 20% of rtVBR and of 20% nrtVBR), and a load about 90%, the maximum delay is equal to three times $T_{slot}$ ($T_{slot}$ = time to transmit a cell) for CBR and rtVBR cells; and it is smaller than $5T_{slot}$ for nrtVBR cells. The delay times are significant for ABR and UBR cells in situation of load above 80%. In addition, in condition of 70% network traffic in lower priority classes, it was observed that the CBR, VBR and ABR classes have small time delay (smaller than $3T_{slot}$) for any network load situation, and only the UBR class has considerable delay time when the load is superior to 85%.

## VI. REFERENCES


[1] M. Karol, M. Hluchyj e S. P. Morgan, "Input versus output queuing in a space division switch", IEEE Trans. Commun., vol. COM-35, pp.1347-1356, Dec. 1987.

[2] S. Motoyama, D. W. Petr, e V. S. Frost, "Scheduling cells in an input-queued switch", Electron. Lett., vol. 31, nº 14, pp. 1127-1128, July 1995

[3] N. McKeown, P. Varaiya , e J. Walrand, "Achieving 100% throughput in an input-queued switch", Electron. Lett., vol. 29, nº 25, pp. 2174-2175, December 1993

[4] N. McKeown, V. Anantharam, e J. Walrand, "Achieving 100% throughput in an input-queued switch", in Proc. IEEE INFOCOM'96, San Francisco, CA, Mar. 1996

[5] S. Motoyama, L. M. Ono, e M. C. Macigno, "An interative Cell Scheduling Algorithm for ATM Input-Queued Switch with Service Class Priority" in IEEE Communications Letters, vol. 03, nº 11, pp. 323-325, November 1999.

[6] K. Y. Eng, M. J. Karol e Y. S. Yeh, "A Growable Packet (ATM) Switch Architeture: Design Principles and Applications", IEEE Trans. Comm., Vol. 40, nº 2, pp. 423-430, February 1992

[7] Genda K., Y. Doi, K. Endo, e N. Yamanaka, "A Very -High-Speed ATM Switch Architecture Using Internal

Speed-up Technique", in NTT Review., vol. 9, nº 2, pp.20-27, March 1997

[8] Doi, Y., Yamanaka, N. “A high-speed ATM switch with input and cross-point buffers”, IEICE Trans. COMMUN., vol.E76-B, no.3, pp310-314, March 1993

[9] S. Motoyama, “Simple high speed ATM switch with service class priority”, Electron. Lett., vol. 36, nº 6, pp. 590-591, March 2000.

[10] Nabeshima, M. “Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch”, IEICE Trans. COMMUN., vol. E83-B, no.3, pp737-741, March 2000

[11] J. L. Hammond, P. J. P. O'Relly, "Performance Analysis of Local Computer Networks", Addison-Wesley Publishing Company, Chap. 3, pp. 98-104, 1986.