Design of knockout concentrators

Y.-S. Lin
C.B. Shung
J.-C. Chen

Indexing terms: Concentrators, Switching

Abstract: The knockout switch architecture has been found attractive for large-scale switch implementations because of its satisfactory cell loss performance, with constant output buffer speed-up independent of switch dimension. The per port hardware complexity of a knockout concentrator, however, does grow linearly with the switch dimension. In the paper, several approaches are investigated to reduce the hardware complexity of the knockout concentrator while retaining the cell loss performance. A bufferless hierarchical concentrator architecture with reduced hardware complexity is derived. The concentrator complexity can be further reduced by introducing buffers in the concentrator, and the trade-off is analysed. Furthermore, output grouping may be applied in the buffered hierarchical concentrator to reduce the per port complexity. Two large-scale switch design examples are derived using the proposed design approaches, producing a complexity reduction ranging from 1.2% to 89.7%.

1 Introduction

Large-scale asynchronous transfer mode (ATM) switches are widely recognized as an important component in building the Information Superhighway. ATM switches with hundreds of input and output ports and with giga-bit link rate have been built in laboratories. However, economical rather than technical factors, such as cost, reliability and manufacturability, often dictate the pace of network deployment and service availability. It is therefore highly desirable to design a switch architecture to be modular and regular for easy expansion, and to have a manageable hardware complexity with growing switch dimension. Such a switch architecture is referred to as 'scalable'. One example of the scalable switch architecture is the knockout switch [1]. As shown in Fig. 1, it consists of an $N \times N$ interconnection fabric and an $N$ to $L$ knockout concentrator, followed by an output buffer for each output port; $N$ is the number of ports and $L$ is the speed-up factor. The hardware complexity of an $N$ to $L$ concentrator can be measured by the number of comparisons required to produce $L$ winners from $N$ inputs, and is in the order of $N \times L$. There are three possible implementations of the $N$ to $L$ concentrator that require $(N \times L - L \times (L + 1)/2)$ [1], $(N \times L)$ [2], $(N - L) \times L$ [3] comparisons, respectively. Since $N >> L$, we simply take $N \times L$ as the complexity formula.

Fig. 1 $N \times N$ knockout switch with $N$ to $L$ knockout concentrators

An important result in [1], often referred to as the knockout principle, indicated that accepting at most $L$ simultaneously arrived cells per output port guarantees an overall cell loss ratio below a certain level comparable to other cell loss sources such as transmission errors, no matter how large the switch dimension. It was estimated that with a 0.9 uniformly distributed load and arbitrarily large switch dimension, $L = 8$ guaranteed a cell loss ratio of less than $10^{-6}$, and $L = 12$ guaranteed a cell loss ratio of less than $10^{-10}$ [1].

In this paper, we are concerned with the problem of reducing the hardware complexity of the knockout concentrator (and hence the knockout switch) without compromising the cell loss performance. This problem has been addressed by many researchers [2-10]. Our techniques focus on two themes: hierarchies and buffers. The idea of hierarchical concentrators has been extensively used in the literature. Large concentrators have been constructed using 32 to 8 concentrators [1]. The hierarchy has been decomposed by destination grouping [3] rather than input grouping [4], with a fixed speed-up factor of two at each level. A specially
mediate levels in the hierarchy. The Christmas-tree level. There is clearly a trade-off between the reduced (hence the concentrator complexity) of the following performance comparable to the knockout switch. buffering was proposed in other knockout-based switch concentrators by grouping the decomposed concentrators. We show that the concentrator is decomposed into specific instances of the concentrator can be done on the $N_{i+1}$ to $L$ concentrator. The resultant hierarchical knockout concentrator is a multi-stage connection of small concentrators with minimised overall complexity, in the class of bufferless input grouping hierarchical concentrators.

Given uniformly distributed input load $p$, $N$ and $L$ (where $L$ depends on the cell loss requirement), the optimum $n_i$ and $l_i$ can be derived with the cell loss probability formula in the Appendix (Section 9.1). Specifically, $n_i$, $l_i$ are chosen such that the decomposed concentrators are not the limiting factor of the overall cell loss performance. Fig. 3 depicts the required concentrator output size $l_i$ for a cell loss ratio of $10^{-10}$, given an aggregate concentrator load as $p_{ni} = n_i p N_i$ (in logarithmic scale). Those curves of $g > 1$ are for output-grouping cases (discussed in Section 4).

The number of comparisons $C$ of the hierarchical knockout concentrator can be derived as

$$C = \sum_{i=1}^{N_i} (n_i \times l_i)$$

$$= N_1 \left( l_1 + \frac{l_1}{n_1} \left( l_2 + \frac{l_2}{n_2} \left( l_3 + \cdots \right) \right) \right)$$

(1)

Compared to the undecomposed concentrator with $C_0 = NL$ comparisons,

$$\frac{C}{C_0} = \frac{l_1}{L} + \frac{l_1}{n_1} \left( \frac{l_2}{L} + \frac{l_2}{n_2} \left( \frac{l_3}{L} + \cdots \right) \right)$$

(2)

The normalised complexity ($C/C_0$) has a recursive factor $l/L + l/n$. If $n_i$ is greater than $L$, the second term is more significant and the factor increases as $n_i$ grows. If $n_i$ is less than $L$, the second term is more significant and the factor decreases as $n_i$ grows. Therefore, there is an optimum choice of $(n_i, l_i)$. By minimising the factor $l/L + l/n$, parameters of the hierarchical concentrators can be recursively determined at each level. In Fig. 4, $C/C_0$ is plotted versus $p_{ni}$ (which is proportional to $n_i$) and $N$. It can be seen that the complexity reduction is more significant for larger switches. For $N < 64$, the hierarchical approach results in a higher hardware complexity. This also suggests that when $N_i \leq 64$, no further decomposition should be applied.
The design procedure of a bufferless hierarchical concentrator is summarised below. Given \( p, N_1, L \), first find out the optimum \( n_1 \) from Fig. 4. Then use the \( g = 1 \) curve in Fig. 3 to find \( l_i \) such that the desired cell loss performance can be achieved. Then let \( N_2 = M_1 \times l_1 \) and repeat the first step until \( N_i \leq 64 \) for some \( i \). The resultant \( i \)-level hierarchical concentrator has a reduced hardware complexity, while achieving the comparable cell loss performance as the original concentrator. To be precise, the loss ratio is \( i \) times higher than the original concentrator. However, as illustrated in the design examples in Section 5, \( N = 1024 \) requires \( i = 3 \). It is acceptable to claim the loss performance as 'comparable' to the original.

Fig. 4

Fig. 5

Buffered hierarchical concentrators

If we place a buffer at the output of each \( i \)th level concentrator, as shown in Fig. 5, we can reduce the total input size at the \((i + 1)\) level to \( N_{i+1} = N/l_i \). As a result, the number of comparisons in the \((i + 1)\) level concentrators is reduced. However, such reduction is at the expense of the additional buffers. In this Section, we try to analyse this trade-off and derive the optimum design principles for buffered hierarchical concentrators. Since output buffering is employed, there is no HOL blocking problems as in other input-buffering approaches [4, 7].

For a given input load, the required buffer size for a buffered concentrator to achieve a desired cell loss performance can be found through discrete-time Markov chain analysis, as briefly described in the Appendix (Section 9.2). Fig. 6 plots the required per-port buffer size \( b_i \) versus the concentrator load for a cell loss performance of \( 10^{-10} \). Those curves with \( g > 1 \) are for output-grouping cases (discussed in Section 4).

The hardware complexity of the buffered concentrators consists of two parts: \( C = \sum_{i=1}^{N} M_i \times l_i \) comparisons and \( B = \sum_{i=1}^{N} b_i \) buffers. Since \( N_{i+1}/N_i = 1/n_i \) and \( M_i = N_{i+1} \), we have

\[
\begin{align*}
C &= l_i \times \frac{1}{n_i} \left( \frac{l_2}{L} + \frac{1}{n_2} \left( \frac{l_3}{L} + \ldots \right) \right) \\
B &= b_i \times \frac{1}{n_i} \left( \frac{b_{i+1}}{B_0} + \frac{B_0}{B_0} + \frac{B_0}{n_2} + \ldots \right)
\end{align*}
\]

where \( B_0 \) is the size of the output buffer at every output of the knockout switch.

Fig. 6

Fig. 7

Overhead of concentrator buffers normalised to the switch output buffer size \( B_0 \) versus concentrator load.

The normalised comparison cost for different \( N \) is also included to show the trade-off \( B/B_0 \) for all \( N \).

Normalised comparison cost of bufferless hierarchical concentrators of different dimension \( N \) at loss \( 10^{-10} \).

The goal at each level is to find \( n_i, l_i, b_i \) such that the overall hardware complexity \((C + B)\) is minimised. Design parameters can be recursively determined by
minimising \( l/L + 1/n_i \) and \( b_i N(B_i/P_i) + 1/n_i \). A smaller \( n_i \), which results in a smaller \( l_p \), is preferred for \( C_i C_0 \). For \( B_i B_0/n_i \) decreases as \( n_i \) (or \( P_i/N_i \)) grows, due to the buffer sharing effect. However, as \( N_i \) approaches \( N_1 \) (or \( P_i/N_i \) approaches 1), \( b_i/n_i \) actually grows with \( n_i \) because of the heavy load buffer sharing. Fig. 7 shows that there is a \( P_i/n_i \) providing minimum buffer complexity for all \( N_i \), but this \( P_i/n_i \) does not provide minimum comparison complexity. \( (B_i/B_0) \) depends on \( b_i/B_0 P_i/n_i \), and thus is independent of \( N_i \).

There is a complexity trade-off between comparisons and buffers in choosing \( n_i \). Of course, the actual choice needs to take into account the relative hardware cost of one comparison and a unit buffer size. Note that, in Fig. 7, since the comparison complexity is reduced in the buffered concentrator, \( C_i C_0 \) is less than unity even for small \( N_i \). In other words, the hierarchical idea in buffered concentrators can be applied to a wider range of switch dimensions than in the bufferless case.

### 4 Buffered hierarchical concentrator with output grouping

In this Section, we investigate the effect of output grouping on the buffered hierarchical concentrator design. A buffered concentrator jointly designed for \( g \) (the output grouping factor) output ports is shown in Fig. 8. Note that the architecture is very similar to the buffered concentrator for a single output shown in Fig. 5, except that the buffer output links are increased from 1 to \( g \). Indeed, the choice of \( (n_i', l_i') \) for the buffered hierarchical concentrator with output grouping can be made using the same design procedures described above.

![Buffered hierarchical concentrator with output grouping every \( g \) output port shares a buffered hierarchical concentrator](image)

Fig. 8 shows the required concentrator output size \( l_i' \), versus the concentrator load, for achieving the desired cell loss performance under different output grouping factors. Fig. 6 shows the required buffer size \( b_i' \), versus the concentrator load. Fig. 3 shows that, due to statistical multiplexing, the total concentrator output size is reduced by output grouping, i.e. \( l_i' < g \times l_i \). Fig. 6 shows that, due to the buffer sharing effect, the total buffer size is also reduced by output grouping, i.e. \( b_i' < g \times b_i \). Moreover, the averaged output buffer size is reduced, or \( B_0 < g \times B_0 \) for the same reason. Thus, output grouping can reduce both the number of comparisons and the buffer size at the same time. As stated previously [9], the complexity reduction by output grouping becomes more significant as \( g \) grows. However, a higher \( g \) requires a higher bandwidth for the buffers. To achieve an overall minimised hardware complexity, output grouping should be applied as much as possible until constrained by the buffer bandwidth.

When applied to the buffered hierarchical concentrator architecture, the output grouping factor \( g \) also affects other design parameters. It is obvious from Fig. 8 that \( g \) sets a lower bound for \( n_i' \). If the derived \( l_i' \) is less than \( g \), the \( i \)'th level degenerates to the buffer-less case. (In this case, the output links of the concentrator can be reduced from \( g \) to \( l_i' \) and no buffering is required. However we still assume the concentrator has \( g \) output links to simplify the complexity derivation.) As the output size \( l_i' \) of every level never gets higher than the final \( L' \), the maximum bandwidth requirement of all concentrator buffers never exceeds the output buffer bandwidth of the growable switch. Therefore, with the grouping factor \( g \), the buffered hierarchical concentrator with output grouping may have a memory bandwidth requirement of less than the growable switch.

For completeness, the hardware complexity of the buffered hierarchical concentrator with output grouping is listed below:

\[
B' = \sum_{i=1}^{N} M_i \times b_i' \quad (6)
\]

Since \( N_i / N_i' = g / n_i \) and \( N_i+1 = g \times M_i \), we have

\[
C' = \frac{C_i}{g \times C_0} = \frac{l_i}{g L} + \frac{l_i'/L}{n_i} \left( \frac{l_2}{g L} + \frac{l_i'/L}{n_2} \left( \frac{l_3}{g L} + \ldots \right) \right) \quad (7)
\]

\[
B' = \frac{b_i N}{g B_0 n_i} + \frac{1}{n_i} \left( \frac{b_2 N}{g B_0 n_2} + \frac{1}{n_2} \left( \ldots \right) \right) \quad (8)
\]

where \( C_i \) and \( B_0 \) are the per port comparison and buffer complexities of the knockout switch, respectively. Figs. 9 and 10 show the complexity trade-off curves for two particular output grouping factors, \( g = 2 \) and 16, respectively. As expected, the larger the output grouping factor, the more hardware reduction can be achieved.

![Trade-off between normalised concentrator buffer overhead and comparison cost for \( g = 2 \), \( B_i B_0 \) for all \( N \) - (fine)](image)

148

g = 2 and \( g = 16 \), respectively. Based on the desired cell loss performance, the knockout switch has the configuration of \( (N, L, B^0) = (N, 12, 104) \). For the growable switch, \( L' = 15 \) and 42 for \( g = 2 \) and 16, respectively [9]. The output buffer sizes \( B_0^g \) = 120 and 224 for \( g = 2 \) and 16, respectively, are derived in a similar way as described in Appendix 9.2.

When the trade-off between comparisons and buffers is considered, different weights based on practical implementation costs should be applied. The overall cost function can be expressed as \( C + \alpha B \). Considering that one comparison can be implemented with 16 gates [1] and one ATM cell requires 424 bit memory storage, if one logic gate is assumed to have the same size of one memory bit, then \( \alpha_1 = 424/104 \approx 26.5 \). If one logic gate is twice as large as one memory bit, \( \alpha_2 \approx 13.25 \).

The design parameters for \( N = 128 \) are derived in the Appendix (Section 9.3). Similar procedure can be applied for \( N = 1024 \). The derived design parameters and the cost figures are summarised in Tables 1 and 2 for \( N = 128 \) and 1024, respectively. Since the number of comparisons grows linearly with switch dimension \( N \), while the buffer requirements stays almost constant, the comparison part contributes more to the overall cost as \( N \) grows. As the proposed bufferless hierarchical approach selects exact output sizes of concentrators for given cell loss requirement, the comparison costs are optimised. Thus, better complexity reduction is achieved for larger \( N \). For example, the bufferless

### Table 1: Design parameters and complexities of different hierarchical concentrators for \( N = 128 \) with loss \( 10^{-10} \) at load = 90%

<table>
<thead>
<tr>
<th>Switch configuration</th>
<th>( M \times (n_0, l_0, b_0) )</th>
<th>( C )</th>
<th>( B ) (cells)</th>
<th>( C + \alpha_1 B ) (% of KO)</th>
<th>( C + \alpha_2 B ) (% of KO)</th>
<th>Maximum buffer bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knockout (KO)</td>
<td>1 \times (128, 12, 104)</td>
<td>1536</td>
<td>104</td>
<td>4292 (100%)</td>
<td>2914 (100%)</td>
<td>12</td>
</tr>
<tr>
<td>Bufferless</td>
<td>4 \times (32, 8, 0)</td>
<td>1408</td>
<td>104</td>
<td>4194 (97.0%)</td>
<td>2786 (96.6%)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1 \times (32, 12, 104)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffered-1</td>
<td>8 \times (16, 7, 7)</td>
<td>896</td>
<td>104</td>
<td>5136 (119.7%)</td>
<td>3016 (103.5%)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1 \times (8, 8, 104)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffered-2</td>
<td>4 \times (32, 8, 9)</td>
<td>1024</td>
<td>140</td>
<td>4734 (110.3%)</td>
<td>2879 (98.8%)</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1 \times (4, 4, 104)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Growable</td>
<td>1 \times (128, 15, 120)</td>
<td>960</td>
<td>60</td>
<td>2550 (59.4%)</td>
<td>1755 (60.2%)</td>
<td>15</td>
</tr>
<tr>
<td>Buffered with</td>
<td>4 \times (32, 10, 9)</td>
<td>640</td>
<td>78</td>
<td>2707 (63.0%)</td>
<td>1673.5 (57.4%)</td>
<td>10</td>
</tr>
<tr>
<td>output grouping</td>
<td>1 \times (8, 8, 120)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The knockout and growable concentrators are included for comparisons.

### Table 2: Design parameters and complexities of different hierarchical concentrators for \( N = 1024 \) with loss \( 10^{-10} \) at load = 90%

<table>
<thead>
<tr>
<th>Switch configuration</th>
<th>( M \times (n_0, l_0, b_0) )</th>
<th>( C )</th>
<th>( B ) (cells)</th>
<th>( C + \alpha_1 B ) (% of KO)</th>
<th>( C + \alpha_2 B ) (% of KO)</th>
<th>Maximum buffer bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Knockout (KO)</td>
<td>1 \times (1024, 12, 104)</td>
<td>12288</td>
<td>104</td>
<td>15044 (100%)</td>
<td>13666 (100%)</td>
<td>12</td>
</tr>
<tr>
<td>Bufferless</td>
<td>32 \times (32, 5, 0)</td>
<td>6784</td>
<td>104</td>
<td>9540 (63.4%)</td>
<td>8162 (59.7%)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>4 \times (16, 6, 5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 \times (32, 12, 104)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffered</td>
<td>16 \times (64, 6, 5)</td>
<td>6336</td>
<td>184</td>
<td>11212 (74.5%)</td>
<td>8774 (64.2%)</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>1 \times (16, 12, 104)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Growable</td>
<td>1 \times (1024, 42, 224)</td>
<td>2888</td>
<td>14</td>
<td>2880.3 (23.3%)</td>
<td>2774.1 (20.3%)</td>
<td>42</td>
</tr>
<tr>
<td>Buffered with</td>
<td>8 \times (128, 15, 0)</td>
<td>1192</td>
<td>15,875</td>
<td>1612.7 (13.1%)</td>
<td>1402.3 (10.3%)</td>
<td>32</td>
</tr>
<tr>
<td>output grouping</td>
<td>2 \times (64, 29, 15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 \times (32, 32, 224)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The knockout and growable concentrators are included for comparisons.
approach provides a complexity reduction under 5% for \( N = 128 \) and above 36% for \( N = 1024 \).

The buffered hierarchical approach achieves almost no complexity reduction for \( N = 128 \). However, it provides a trade-off between comparisons and buffers, as illustrated in the buffered-1 and buffered-2 configurations in Table 1. Buffered-1 has a lower comparison complexity, whereas buffered-2 has a lower buffer complexity. In addition, the bandwidth requirement on buffer memories is reduced from \( L = 12 \) of the knockout switch to \( L = 8 \) for both buffered configurations. As for \( N = 1024 \), although the optimum configuration provides 35.8% complexity but no buffer bandwidth reduction, it is possible to choose another sub-optimal configuration which requires a lower buffer bandwidth at the cost of higher comparison complexity. Such a trade-off allows flexible implementation options under different technology constraints.

Output grouping reduces both the comparison and buffer complexities through multiplexing the traffic of \( g \) ports. The proposed hierarchical buffered concentrator with output grouping further explores the trade-off between comparisons and buffers, and thus provides better complexity reduction and more flexible implementation than the growable switch. The improvement of the proposed approach over the growable switch is more significant as \( g \) grows. This is illustrated by the examples of \( g = 2 \), in which both approaches have comparable complexities, and \( g = 16 \), in which the proposed approach has about half the complexity of the growable switch. Although not explored in this paper, the output size of concentrators can be liberated from the fixed \( g \) to an optimised \( g \) at every level to achieve further complexity reduction.

6 Conclusions

The knockout switch is an attractive choice for large-scale switch implementations for its scalable property. We have studied a bufferless hierarchical concentrator architecture, and proposed a recursive procedure to determine the parameters of intermediate concentrators and reduce the overall complexity. By introducing buffers at the output of the intermediate concentrators, we can reduce the concentrator output, and thus further reduce the complexity. We have analysed the trade-off between the reduced concentrator complexity and the increased buffer complexity.

Further complexity reduction can be realised through output grouping. We have shown that output grouping can be easily incorporated in our hierarchical concentrator architecture. Output grouping can reduce not only the buffer size, but also the concentrator complexity. However, the output grouping factor is constrained by the feasible buffer memory bandwidth. Our approach provides a trade-off among comparison complexity, buffer complexity and buffer bandwidth requirements. Using the above design techniques, we have derived two knockout switch design examples.

For \( N = 128 \), we found that a two-level concentrator architecture was satisfactory, and the complexity reduced by 4.4%, 1.2%, and 42.6% for the bufferless, buffered, and buffered with output grouping (\( g = 2 \)) approaches, respectively. For \( N = 1024 \), we found that a two- or three-level concentrator architecture achieved the lowest complexity for the bufferless, buffered, and buffered with output grouping (\( g = 16 \)) approaches. The complexity reduction is 40.3%, 35.8%, and 89.7%, respectively.

These design examples demonstrate that the proposed design techniques are effective in reducing the hardware complexity of the knockout concentrator, and provide a trade-off between comparisons and buffers for flexible implementation under technology constraints. Although a uniformly distributed load is assumed for the analysis, the proposed design approaches are also expected to work for other traffic conditions.

7 Acknowledgments

This work was supported by the National Science Council of Republic of China under grant NSC-84-2215-E-009-046.

8 References


9 Appendix

9.1 Bufferless loss probability

Let \( \rho \) be the switch input load and \( N \) be the switch dimension. For the \( i \)th level, let \( n_i \) inputs, \( l_i \) concentrators, let \( P_{in}(k) \) and \( P_{out}(k) \) be the probabilities that there are \( k \) active cells arriving at the \( n_i \) inputs and that \( k \) out of the \( l_i \) outputs are delivering active cells, respectively. Let \( PL(n_i, l_i) \) be cell loss probability of the \( i \)th level concentrators of input size \( n_i \) and output size \( l_i \). We have:

\[
P_{in}(k) = C^n_{k} \left( \frac{\rho}{N_i} \right)^k \left( 1 - \frac{\rho}{N_i} \right)^{n_i-k}
\]

\[
P_{out}(k) = \begin{cases} 
P_{in}(k) & 0 \leq k < l_i \\
\sum_{j=k}^{l_i} P_{in}(j) & k = l_i \end{cases}
\]

\[
PL(n_i, l_i) = \frac{\sum_{j=l_i+1}^{n_i} (j - l_i) \times P_{in}(j)}{n_i}
\]
Let $P_i(k)$ be the probability that there are $k$ active cells among all $N_i$ links at the $i$th level, where

$$P_i(k) = C_k^N \left( \frac{\rho}{N_i} \right)^k \left( 1 - \frac{\rho}{N_i} \right)^{N_i-k}$$

(12)

Alternatively, $P_i(k)$ can also be derived with $P_{i-1,\text{out}}(k)$:

$$P_i(k) = \sum_{\text{all combinations}} \left[ \prod_{j=1}^{M_i-1} P_{i-1,\text{out}}(k_j) \right]$$

(13)

If we assume the cell loss (or load reduction) in the prior levels is negligible, then eqn. 12 shall have the same distribution as eqn. 13. This supports the recursion of the hierarchical decomposition.

### 9.2 Buffered loss probability

Let $PB_i(k)$ be the probability that the $i$th level buffer has $k$ cells in it at time $t$. The state transition of $PB_i$ from time $t$ to $t+1$ is governed by the state transition matrix represented by

$$PB_i(k^{t+1})$$

$$= \sum_{h^t, v^t+1 = \text{max}(h^t+j-0, 0)} \sum_{j=1}^{l_i} (PB_i(k^t) \times P_{i,\text{out}}(j))$$

$$\approx \sum_{h^t, v^t+1 = \text{max}(h^t+j-0, 0)} \sum_{j=1}^{n_i} (PB_i(k^t) \times P_{i,\text{in}}(j))$$

(14)

Its eigenvector is the steady-state buffer occupation distribution $PB_i$. The cell loss probability of a buffer with a size of $k$ cells is customarily approximated by $PB_i(k)$.

### 9.3 Derivation of parameters for $N = 128$

#### 9.3.1 Bufferless

From the $N = 128$ curve in Fig. 4, the minimum cost $\rho_{n_1}$ is around 0.24, which results in an optimum $n_1 = 128 \times 0.24/0.9 = 34.1$. We choose $n_1 = 32$ and consider Fig. 3 with the concentrator load $\rho_{n_1} = 0.9 \times 32/128 = 0.225$. The required concentrator output size is found and rounded to integer as $l_1 = 8$. The number of the second level inputs is $N_2 = 128/32 \times 8 = 32$, which is less than 64. Therefore, a direct implementation of $(n_2, l_2, b_2) = (32, 12, 104)$ is selected.

#### 9.3.2 Buffered

From the $N = 128$ curve in Fig. 7, the minimum cost $\rho_{n_2}$ are around 0.07 and 0.4 for comparison and buffer, respectively. Therefore, the overall optimum $n_1$ is between 9.96 and 56.9. If we choose $n_1 = 16$, we can find $l_1 = 7$ and $b_1 = 7$ from Figs. 3 and 6, respectively. As $N_2 = 8$ is smaller than $L = 12$, a direct implementation $(n_2, l_2, b_2) = (8, 8, 104)$ is selected for the second level. In a similar way, if we choose $n_1 = 32$, we have $(n_1, l_1, b_1) = (32, 8, 9)$ and $(n_2, l_2, b_2) = (8, 8, 120)

#### 9.3.3 Buffered with output grouping

As we assumed $g = 2$ for $N = 128$, we can find from Fig. 9 that the optimum $\rho_{n_2}$ is between 0.1 and 0.5. The possible choice of $n_1$ is thus limited to $n_1 = 16, 32, \text{or} 64$. In a way similar to the buffered case, but using Figs. 3 and 6, we found the optimum design parameters are $(n_1, l_1, b_1) = (32, 10, 9)$ and $(n_2, l_2, b_2) = (8, 8, 120)$.