# University of Castilla–La Mancha


A publication of the

**Computing Systems Department**

## An Alternative for Building High-Radix Switches: Application for Special Traffic Patterns*

by

J.A. Villar, F.J. Andújar, J.L. Sánchez, F.J. Alfaro, J. Duato

Technical Report #**DIAB-11-02-2** February 2011

(*) This work was supported by the Spanish MEC and MICINN as well as European Comission FEDER funds, under Grants "Consolider Ingenio–2010 CSD2006-00046" and "TIN2009–14475–C04", respectively; it was also partly supported by Junta de Comunidades de Castilla–La Mancha under grants "PCC08–0078–9856" and "POII10–0289–3724".

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS
ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE CASTILLA-LA MANCHA
CAMPUS UNIVERSITARIO s/n
02071, ALBACETE, ESPAÑA
Tlf. +34.967.599200, Fax +34.967.599224

# An Alternative for Building High-Radix Switches: Application for Special Traffic Patterns

Juan A. Villar and Francisco J. Andújar
Instituto de Investigación en Informática
Campus Universitario s/n
02071 – Albacete, España
{juanan,fandujar}@dsi.uclm.es

José L. Sánchez and Francisco J. Alfaro
Departamento de Sistemas Informáticos
Escuela Superior de Ingeniería Informática
02071 – Albacete, España
{falfaro, jsanchez}@dsi.uclm.es

José Duato
Dpto. de Ingeniería de Sistemas y Computadores
Camino de Vera, s/n
Universidad Politécnica de Valencia
46022 – Valencia, España
jduato@gap.upv.es

February 9, 2011

# Contents

# 1 Introduction

Interconnection networks are a key component for a wide range of multiprocessor systems, ranging from large supercomputers to multicore chips. High performance networks are essential in these systems, where high reliability in communications, high information transfer rates and very low latencies are critical. Often, the interconnection network is the subsystem that a more custom design requires. For instance, Tianhe-1A supercomputer [top10], number one in the November 2010 Top500 list, is composed of standard Intel and NVIDIA processors and a fancy new interconnection network. This custom interconnect design removes the interconnect bottleneck and significantly contributes to the high global performance of Tianhe-1A.

Interconnection network design is determined by the available technology. Recent advances on the technology have substantially improved the performance of the basic network components: links and switches. The latter are responsible for most of the interconnection network performance, and so they are the subject of major research. One of the main parameters characterizing network switches is the number of ports, which has a strong influence on cost, consumption and performance in the whole system.

Given a multiprocessor system with a large number of connected elements, increasing the number of switch ports results in a decrease in the number of switches and network links. As the cost of the network is proportional to the number of switches, it is clear that cost decreases by using switches with higher number of ports. Moreover, total consumption of the network is also considerably reduced as it is directly proportional to the number of switches in the network.

Regarding performance, in terms of latency, for example, it is clear that the use of switches with more ports involves a reduction in the average time to transfer data over the network. In particular, using fewer switches to connect the same number of elements reduces the number of hops and the number of possible packet collisions in the network, and thus the time to reach their destinations. Furthermore, having less switches, the total processing time of the packets in the switches along their paths is also reduced.

Thus, the design of switches with a high number of ports is an attractive option to improve the performance and reduce the cost of the interconnection network, specially for large multiprocessor systems. However, there are some problems to design such switches. One of them is related to the complexity of the switch logic. The switch becomes more complex as radix increases, taking up to a significant percentage of total system power [WPM03]. The balance between cost and efficiency is not easy to work through, requiring a deep study regarding this trade-off. On the one hand, the size of some switch structures grows quadratically with the number of ports. That is the case of, for example, the aggregate buffer requirements as identified in [GAG$^+$03], or the schedulers as stated in [MAM$^+$05]. Moreover, traditional flow control policies are also affected by switch radix in two aspects [MG07]: the round trip time drastically increases, and the memories for storing flow control credits are linearly dependent on the round trip time. On the other hand, pin count will slowly increase next decade, according to the ITRS [ITR10], and therefore switch ports number will slightly increase. Moreover, there are difficulties to apply some improvement techniques when the number of ports is high. For instance, Virtual Output Queuing (VOQ) implementation becomes unfeasible in practice for switches with large number of ports. To overcome these problems, different solutions have been proposed, but actually, they are postponing the problem for coming switch generations.

In any case, switch size constraints are mainly determined by the current integration scale and package pin count. To go beyond the integration scale bounds, an alternative solution for building high-radix switches is the combination of several low-radix switches. This solution has the advantage of obtaining higher number of ports. Moreover, some difficulties mentioned above lose importance.

The main idea is to implement $m'$-port switches from several smaller $m$-port switches. For instance, a $m'$-port switch consisting of two identical $m$-port switches ($m'/2 < m < m'$) can internally interconnect each other using $m - m'/2$ ports, using the remaining ports for external connections. Note that this strategy will remain valid as integration scale keeps evolving.

An important consequence of this strategy for building larger switches is that the resulting switch will no longer be homogeneous. Switch performance will vary depending on the internal configuration. The internal switches interconnection can become a potential bottleneck if they have to support most of the traffic handled in the switch. Therefore, it is essential to minimize the impact of this bottleneck, otherwise the latency on the network will be increased. Thus, the switch-level connection pattern[1] becomes an important design decision in the construction of this kind of switches. An arbitrary pattern may produce a significant performance degradation. Consequently, it is necessary to determine the most convenient pattern in order to obtain the best switch performance.

In this paper, we present a formalization of this kind of high-radix switches and propose a methodology for configuring them in an optimal manner when they are used to build large switch–based interconnection networks. To show how this methodology works, it is applied to a particular interconnection network.

The report is organized as follows: Section 2 describes the *combined switches*, and in Section 3 we introduce the notation used later. Section 4 formally defines and characterizes the *twin switches*. In Section 5, we propose our methodology for searching the optimal configuration of *twin switches*, and show how it works in a particular case. Section 8 gives a brief review to existing proposals on high-radix switches. Additionally, in the Appendix A we have included basic concepts on multistage interconnection networks.

[1] We distinguish between *network-level connection pattern* and *switch-level connection pattern*. The former is the traditional interconnection pattern connecting switch-based networks (e.g., *butterfly* permutation in multistage interconnection networks); and the latter refers to how every high-radix switch ports are mapped to the ports of the internal switches.

# 2 High-Radix Switches by Combining Low-Radix Switches

As mentioned above, it is possible to build high-radix switches by combining several low-radix switches. This strategy makes possible to eventually overtake integration technology and dramatically shorten the time-to-market. Note that this will remain valid as integration technology continues evolving.

This strategy opens a series of new issues that must be addressed so that it would be implemented in an efficient manner. In this section, we define the combined switches and briefly overview the issues characterizing them. Then in the next sections, we formally analyze them in depth.

## 2.1 Combined Switches

In this section we define the combined switches giving a general definition. Then we turn our attention to a particular subclass of this kind of switches, which will be used in order to show the characteristics and evaluate the performance of combined switches as a high-radix switch alternative.

***Definition 1.1*** *A Combined switch, or simply $C$-switch, is a switch formed by several smaller interconnected switches (internal switches). The ports being offered by a $C$-switch are obtained from free ports of its internal switches after they are interconnected.*

This is a general definition because it does not specify either the number of internal switches or their radix. Therefore any switch obtained by combining other lower switches is included in this category. However, there exist some difficulties to build $C$-switches having many internal switches and a high heterogeneity degree.

In order to keep low the internal latency, a full-connected subnetwork interconnecting all the internal switches is preferred. As the number of internal switches increases, the number of ports dedicated to subnetwork connections grows as fast as the number of ports devoted to external communications decreases, so this way of building high-radix switches would lose interest. Therefore, it seems reasonable that the number of internal switches may not be too large.

On the other hand, a simpler $C$-switch internal design can be achieved if all the internal switches are equal. Although this aspect is not as restrictive as the number of internal switches, it would be also recommendable that all the internal switches have the same radix.

An interesting case is that where $C$-switches are built from only two identical internal switches. This subclass of $C$-switches still offers an important increase in the number of ports while the interconnection between the two internal switches is the simplest one.

***Definition 1.2*** *A Twin switch, or simply $T$-switch, is a switch formed by two identical smaller interconnected switches. The ports being offered by a $T$-switch are obtained from free ports of its two internal switches after they are interconnected.*

Considering that the two internal switches and the $T$-switch have radixes $m$ and $n$, respectively, Figure 1(a) shows a basic diagram of a $T$-switch, where internal switches are denoted as $\alpha$ and $\beta$. Although $T$-switches seem to be simple, there are significant challenges in its design. Two of them stand out especially: (1) to obtain the appropriate switch-level connection pattern of internal subnetwork, (2) to determine the adequate number of ports used to interconnect switches $\alpha$ and $\beta$.

Switch-level connection pattern has an important influence on packet latency. Time to cross the $T$-switch will be minimal when only one internal switch ($\alpha$ or $\beta$) is used (Figure 1(b)). The bad case is obtained when both internal switches are used for every path crossing the $T$-switch (Figure 1(c)). To obtain the best case is not trivial and an in-depth study is required, in which several factors,

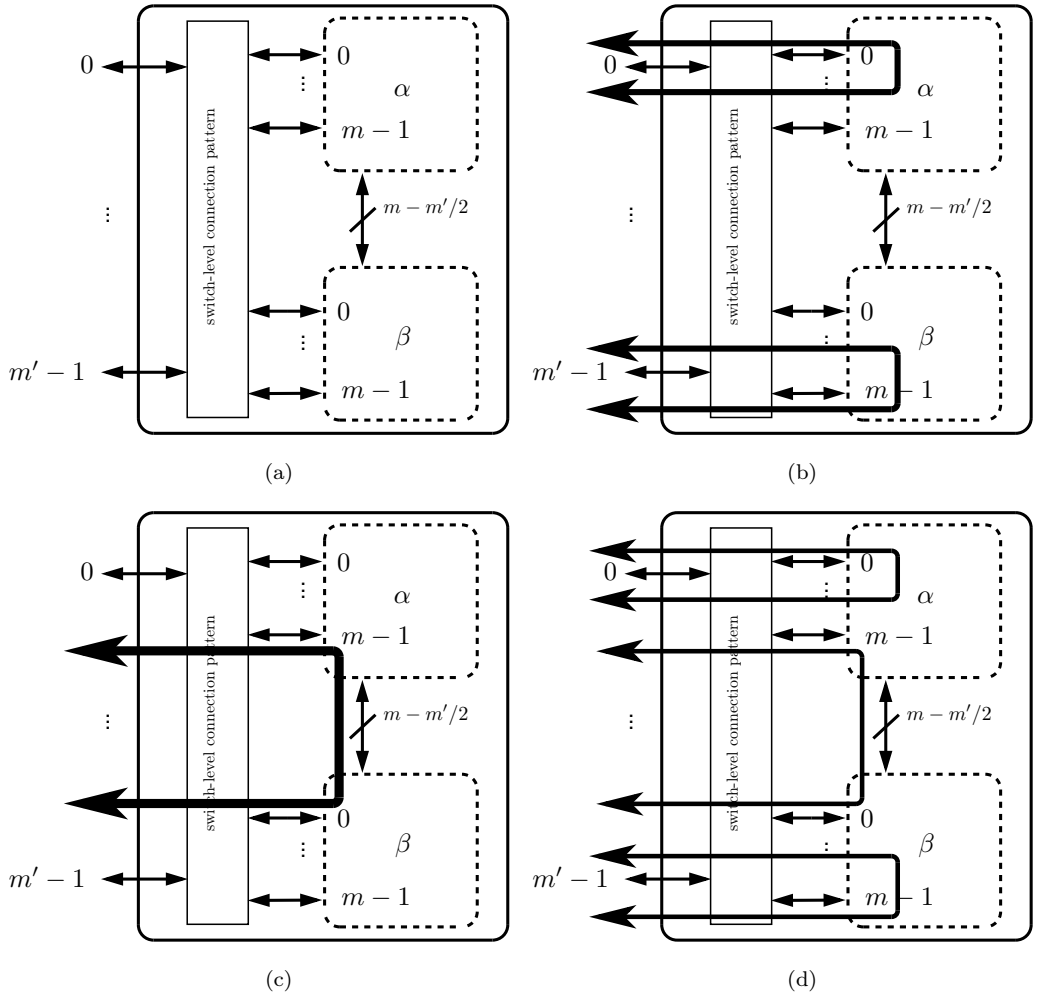



Figure 1: (a) Basic block diagram of a $T$-switch and several cases: (b) Optimal, (c) Bad, and (d) Common.

e.g., network topology, routing and traffic pattern must be considered. From Section 2.2 we show in a formal way how the optimal switch-level connection pattern can be obtained if these factors are considered.

Regarding the second challenge, the number of internal ports must be that which avoids the creation of the internal bottleneck between $\alpha$ and $\beta$. Obviously, the number of internal ports and the switch-level connection pattern have a clear interdependence.

The situations shown in Figure 1(b) and Figure 1(c) are appropriate for illustrating the $T$-switch design challenges, but a more common situation is that shown in Figure 1(d), where both kinds of cases coexist. In such situations, the main objective in the switch-level connection pattern design of $T$-switches is to minimize the use of the ports interconnecting the internal switches.

Therefore, since in some cases communication will require the use of both internal switches (i.e. a path passing through a $T$-switch will use both $\alpha$ and $\beta$), we have to avoid interconnection between $\alpha$ and $\beta$ becomes a bottleneck. Moreover, the adequate number of ports of each internal switch to connect with each other must be determined. It is clear that the greater the number of ports used to interconnect the internal switches the lower the probability of this interconnection becomes a bottleneck. However, as mentioned above, as the number of ports devoted to interconnect $\alpha$ and $\beta$ increases the $T$-switch radix decreases. Note that we are assuming that all the ports provide the same bandwidth (otherwise instead of using the number of ports, the aggregate bandwidth should be

used). Consequently, a trade-off between both aspects must be found.

In summary, internal configuration of $T$-switches, and in general $C$-switches, becomes a key aspect in the design of this kind of high-radix switches. In the following Section, we present a general methodology to obtain the best configuration of this kind of switches when they are used in large interconnection networks. It is obvious that optimal configuration of a $C$-switch depends on the conditions under which it is used, that is, network type, topology, routing algorithm, traffic pattern, etc. To show how the methodology works, we apply it to a particular subclass of interconnection network.

## 2.2 Combined Switches Configuration Methodology

Our methodology to determine the optimal switch-level connection pattern of $C$-switches consists of the following main steps:

1. Network paths analysis. The purpose of this step is to determine the connections required in each $C$-switch at network level and the amount of times all these connections are used taking into account all the possible paths used by the packets.

2. Switch classification. Depending on the network characteristics and load conditions, few or many different $C$-switch configurations could be obtained. In this step, $C$-switches are grouped according to their connection requirements, and so, several types of $C$-switch will be distinguished.

   As result of the previous phase, it can occur that some of the possible connections in the $C$-switches support one, or more paths, and however there may be connections that are never established.

   In a fat-tree topology, for instance, $C$-switches in different stages may require different switch-level connection patterns, and the same may even occur with $C$-switches in the same stage. When a simple traffic pattern and balanced routing algorithm are used, it is likely all the $C$-switches in the network require the same switch-level connection pattern.

3. Switch configuration. From connection requirements and given the number of internal switches forming the $C$-switch, this last step consists in finding the optimal configuration for each class of $C$-switch. That is, we must find the optimal switch-level connection pattern of each class, trying to minimize the use of the interconnection between internal switches.

Summing up, given the network topology, the routing algorithm and the traffic pattern, we can determine the paths generated in the network, the $C$-switches used by each path, and the connections required in each $C$-switch. If the two ports involved in a $C$-switch connection are connected to the same internal switch, the paths using that connection will only use one internal switch when passing through the $C$-switch. Therefore, the objective is to get a switch-level connection pattern, where the most of the connections satisfy this condition. In general, $C$-switches with different connection requirements will have different internal configurations, thus the methodology proposes to search for the optimal switch-level connection pattern for each $C$-switch, or group of $C$-switches requiring the same connections, separately. Finally, if possible, the same switch-level connection pattern for all the $C$-switches will be found.

The first two steps of this methodology are independent of the internal $C$-switch structure. Obviously, the third step depends on the structure of the $C$-switch.

Note that although the methodology is general and can be applied considering different $C$-switches and interconnection networks, the $C$-switch configuration depends on the particular network properties. Therefore, in order to apply this methodology, network type, topology, routing algorithm and traffic pattern must be specified.

### 2.3 Study Conditions

As above mentioned, switch-based interconnection networks cover a wide range of network configurations. To present our formal analysis of the $C$-switches behavior and how they can be configured in an optimal way, we have chosen a very representative case. On the one hand, we consider $T$-switches due to the reasons outlined in Section 2.1. On the other hand, and since fat-trees are one of the most common topologies today in the largest supercomputers on the Top500 list, we focus on bidirectional interconnection networks (BMINs).

## 3 Notation

We have assumed the following notation throughout this paper:

- $N$ is the total number of terminals.
- $k$ is the switch arity, or number of ports that connect to terminals/switches in the previous stage and switches in the next stage (if available). Hence, the total number of ports of a $k\times k$ switch is $2k$. The ports faced to the previous stage are numbered from 0 to $k-1$, and the ports connected with the switches in the next stage are labeled from $k$ to $2k-1$.
- Every switch port has an associated global identifier inside the stage, $L = l_{n-1}\dots l_0, 0 \leq l_i < n$, apart from the internal identifier inside the switch. Both identifiers are related by the connection pattern between stages.
- $n$ is the total number of stages, where $n = \log_k N$.
- $h$ is the terminal identifier ($0 \leq h < N$). It consists of a string of $n$ digits $h_{n-1}\dots h_1 h_0$ ($0 \leq h_i < k$). $\mathcal{H}$ is the set whose members are the terminals of the MIN, verifying $card(\mathcal{H}) = N$.
- $\langle s, o\rangle$ is a tuple that identifies uniquely a switch, where $s$ refers to the stage ($0 \leq s < n$), and $o = o_{n-2}, \dots, o_1, o_0$ indicates the position of the switch inside the stage, where $0 \leq o_i < k$ and $0 \leq i < n-1$.

## 4 Twin Switches

We fully characterize the $T$-switches by means of the following definitions and propositions.

***Definition 1.3*** *Let $\mathcal{U}$ be the set of ports on a $k\times k$ switch. Hence,*

$$\mathcal{U} = \{i \in \mathbb{N},\ 0 \leq i < 2k\}$$

***Definition 1.4*** *Let $\mathcal{B}$ be the set of ports on a $k\times k$ switch connecting to switches from the previous stage (or the input ports in a MIN). Hence,*

$$\mathcal{B} = \{i \in \mathcal{U},\ 0 \leq i < k\}$$

***Definition 1.5*** *Let $\mathcal{F}$ be the set of ports on a $k\times k$ switch that connect to switches on the next stage in a MIN network. Hence,*

$$\mathcal{F} = \{i \in \mathcal{U},\ k \leq i < 2k\}$$

Figure 2 shows the detailed organization of $T$-switches. From Definitions 1.3, 1.4, and 1.5 it is obvious that $card(\mathcal{U}) = 2k$ and $card(\mathcal{B}) = card(\mathcal{F}) = k$, where $card$ is the cardinality of sets.

Most of the internal switch ports are dedicated for external communications, meanwhile a concrete number of them are responsible for intra communications between internal switches.

- $\mathcal{B}^{\alpha} \cup \mathcal{B}^{\beta} = \mathcal{B}$
- $\mathcal{F}^{\alpha} \cup \mathcal{F}^{\beta} = \mathcal{F}$
- $\mathcal{B}^{\alpha} \cap \mathcal{B}^{\beta} = \emptyset$
- $\mathcal{F}^{\alpha} \cap \mathcal{F}^{\beta} = \emptyset$

According to the above, $T$-switches can be re-defined as follows:

***Definition 1.6*** *A $k \times k$ $T$-switch is a bidirectional switch formed by two identical smaller interconnected switches. the $2k$ ports being offered by this $T$-switch are obtained from the $k$ free ports of each internal switch after they are interconnected by $r$ ports.*

***Definition 1.7*** *Let $\mathcal{P}^i$ be the set of ports on the switch $i$, where $i \in \{\alpha, \beta\}$. Moreover, $\mathcal{P}^i$ is divided into three disjoint subsets $\mathcal{J}^i$ to interconnect the internal switches, $\mathcal{B}^i$, and $\mathcal{F}^i$ for the $T$-switch ports. In a more formal way:*

- $\mathcal{P}^i = \mathcal{J}^i \cup \mathcal{B}^i \cup \mathcal{F}^i$
- $\mathcal{J}^i \neq \emptyset$
- $\mathcal{B}^i \cap \mathcal{F}^i = \emptyset$
- $\mathcal{B}^i \cap \mathcal{J}^i = \emptyset$
- $\mathcal{F}^i \cap \mathcal{J}^i = \emptyset$

According to this, it is clear to derive that:

- $\text{card}(\mathcal{J}^{\alpha}) = \text{card}(\mathcal{J}^{\beta})$
- $\text{card}(\mathcal{P}^{\alpha}) = \text{card}(\mathcal{P}^{\beta})$

***Proposition 1.1*** *In a $T$-switch consisting of two internal switches $\alpha$ and $\beta$, it is verified that*

- *$card(\mathcal{B}^{\alpha}) = card(\mathcal{F}^{\beta})$*
- *$card(\mathcal{B}^{\beta}) = card(\mathcal{F}^{\alpha})$*

**Proof:** According to Definition 1.7 we know how the switch ports of $\alpha$ and $\beta$ are configured:

- $\mathcal{B} = \mathcal{B}^{\alpha} \cup \mathcal{B}^{\beta}$
- $\mathcal{B}^{\alpha} \cap \mathcal{B}^{\beta} = \emptyset$

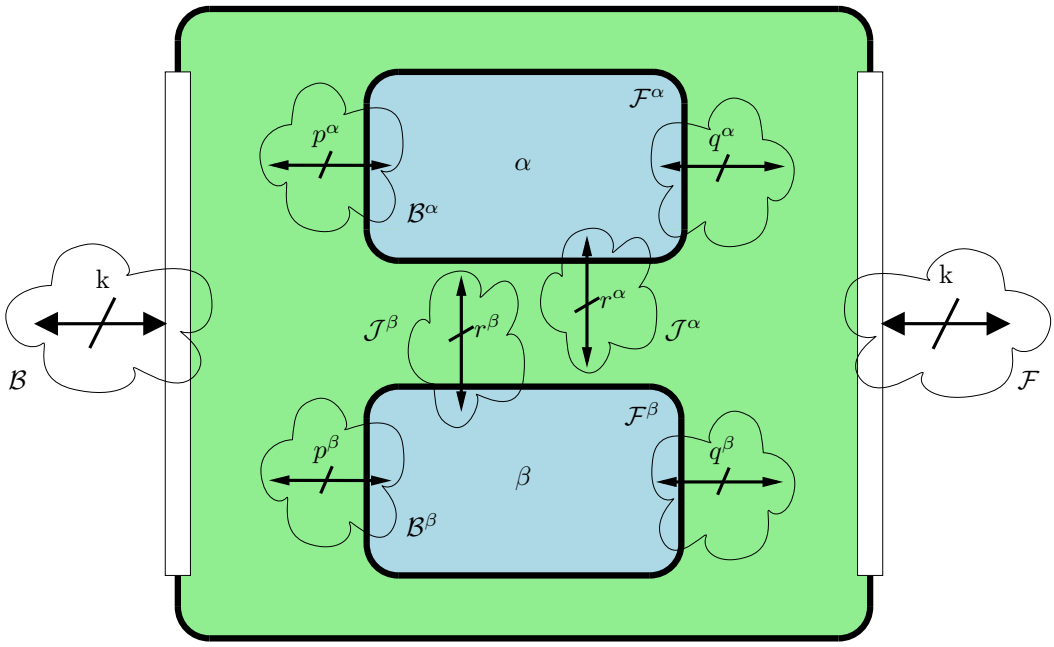



Figure 2: $T$-switch.

- $\mathcal{F} = \mathcal{F}^\alpha \cup \mathcal{F}^\beta$
- $\mathcal{F}^\alpha \cap \mathcal{F}^\beta = \emptyset$

since the sets are disjoint, the expression can be rewritten as

$$card(\mathcal{B}) = card(\mathcal{B}^\alpha) + card(\mathcal{B}^\beta) = k \tag{1}$$
$$card(\mathcal{F}) = card(\mathcal{F}^\alpha) + card(\mathcal{F}^\beta) = k \tag{2}$$

clearing equations

$$card(\mathcal{B}^\beta) = k - card(\mathcal{B}^\alpha) \tag{3}$$
$$card(\mathcal{F}^\alpha) = k - card(\mathcal{F}^\beta) \tag{4}$$

and also

$$card(\mathcal{B}^\alpha) = k - card(\mathcal{B}^\beta) \tag{5}$$
$$card(\mathcal{F}^\beta) = k - card(\mathcal{F}^\alpha) \tag{6}$$

On the other hand, the Definition 1.7 describes how the switch ports of $\alpha$ and $\beta$ are distributed,

$$\mathcal{P}^\alpha = \mathcal{J}^\alpha \cup \mathcal{B}^\alpha \cup \mathcal{F}^\alpha$$
$$\mathcal{P}^\beta = \mathcal{J}^\beta \cup \mathcal{B}^\beta \cup \mathcal{F}^\beta$$

and also

$$\mathcal{B}^\alpha \cap \mathcal{F}^\alpha = \emptyset,\ \mathcal{B}^\alpha \cap \mathcal{J}^\alpha = \emptyset, \text{ y } \mathcal{F}^\alpha \cap \mathcal{J}^\alpha = \emptyset$$
$$\mathcal{B}^\beta \cap \mathcal{F}^\beta = \emptyset,\ \mathcal{B}^\beta \cap \mathcal{J}^\beta = \emptyset, \text{ y } \mathcal{F}^\beta \cap \mathcal{J}^\beta = \emptyset$$

since $card(\mathcal{P}^\alpha) = card(\mathcal{P}^\beta)$, it is verified that

$$card(\mathcal{J}^\alpha) + card(\mathcal{B}^\alpha) + card(\mathcal{F}^\alpha) = card(\mathcal{J}^\beta) + card(\mathcal{B}^\beta) + card(\mathcal{F}^\beta)$$

considering $card(\mathcal{J}^\alpha) = card(\mathcal{J}^\beta)$, and the expressions 3 and 4, we know that

$$\begin{aligned} card(\mathcal{B}^\alpha) + k - card(\mathcal{F}^\beta) &= k - card(\mathcal{B}^\alpha) + card(\mathcal{F}^\beta) \\ 2 * card(\mathcal{B}^\alpha) &= 2 * card(\mathcal{F}^\beta) \\ card(\mathcal{B}^\alpha) &= card(\mathcal{F}^\beta) \end{aligned}$$

this demonstrates the first clause of the Proposition.

In a similar procedure, but using the equations 5 and 6, the second clause of the Proposition is proved, that is,

$$card(\mathcal{B}^\beta) = card(\mathcal{F}^\alpha)$$

□

As it is suggested by the Figure 2, we assume that:

- $\text{card}(\mathcal{B}^\alpha) = p^\alpha$.
- $\text{card}(\mathcal{F}^\alpha) = q^\alpha$.
- $\text{card}(\mathcal{B}^\beta) = p^\beta$.
- $\text{card}(\mathcal{F}^\beta) = q^\beta$.

thus from Proposition 1.1, it is possible to conclude that $p^\alpha = q^\beta \quad$ and $\quad q^\alpha = p^\beta$.

Accordingly, and in order to use a simpler notation and without loss of accuracy in the final solution, or rigor in the procedure, the following simplifications are assumed:

$$card(\mathcal{B}^\alpha) = card(\mathcal{F}^\beta) = p. \tag{7}$$
$$card(\mathcal{B}^\beta) = card(\mathcal{F}^\alpha) = q. \tag{8}$$

***Definition 1.8*** *Let $\mathcal{C}^i \subset \mathcal{U}$ be the configuration of an internal switch $i$ of a $T$-switch, where $i$ defines the set of ports on the switch $i$, which are a subset of the ports of the $T$switch, where $i \in \{\alpha, \beta\}$. Hence,*

$$\mathcal{C}^i = \mathcal{B}^i \cup \mathcal{F}^i$$

***Proposition 1.2*** *The number of elements in $\mathcal{C}^i$ is $k$, where $i \in \{\alpha, \beta\}$. In other words $card(\mathcal{C}^i) = k$.*

**Proof:** The cardinal of $\mathcal{C}^\alpha$ is $card(\mathcal{C}^\alpha) = card(\mathcal{B}^\alpha) + card(\mathcal{F}^\alpha)$. From the Proposition 1.1, we know that $\text{card}(\mathcal{B}^\beta) = \text{card}(\mathcal{F}^\alpha)$. Thus $card(\mathcal{C}^\alpha) = card(\mathcal{B}^\alpha) + card(\mathcal{B}^\beta)$.

As defined above in Definition 1.7, $\mathcal{B}^\alpha$ and $\mathcal{B}^\beta$ are disjoint sets and $\mathcal{B} = \mathcal{B}^\alpha \cup \mathcal{B}^\beta$. Therefore, $card(\mathcal{B}) = card(\mathcal{B}^\alpha) + card(\mathcal{B}^\beta)$. By substituting in the above equation, we get $card(\mathcal{C}^\alpha) = card(\mathcal{B}) = k$.

Starting from $\mathcal{C}^\beta$ and applying the same process, we will also reach the conclusion that $card(\mathcal{C}^\beta) = k$. □

***Definition 1.9*** *Let $\mathcal{V}$ be the set of all possible configurations of an internal switch $i$ in a $T$-switch, where $i \in \{\alpha, \beta\}$. Hence,*

$$\mathcal{V} = \{\mathcal{C}^i \subset \mathcal{U} \mid card(\mathcal{C}^i) = k\}$$

It is important to note that $\mathcal{U}$ is a set, while $\mathcal{V}$ is a set whose members are sets. That is, any configuration $\mathcal{C}^i$ is contained in $\mathcal{U}$, and belongs to $\mathcal{V}$. Hence, $\mathcal{C}^i \subset \mathcal{U}$, and $\mathcal{C}^i \in \mathcal{V}$.

Hereafter, the configuration of an internal switch will be denoted by $\mathcal{C}$, when it does not matter if the switch configuration refers to the switches $\alpha$ or $\beta$. The superscript will be only used when necessary to distinguish between $\alpha$ and $\beta$.

***Definition 1.10*** *Let $\mathcal{T}$ be the configuration of $T$-switch, which is determined by a pair of configurations belonging to $\mathcal{V}$. Hence,*

$$\mathcal{T} = \{\mathcal{C}^\alpha,\ \mathcal{C}^\beta \in \mathcal{V} \mid \mathcal{C}^\beta = (\mathcal{C}^\alpha)^C\}$$

***Definition 1.11*** *Let $\gamma$ be a pattern of connections that are applied to a $T$-switch. Then we define $\mathcal{S}^\gamma$ as the subset of $\mathcal{V}$, whose configurations minimize the number of connections using the ports $\mathcal{J}$ of the internal switches. In other words:*

$$\mathcal{S}^\gamma = \{\mathcal{C} \in \mathcal{V} \ \mid \mathcal{C} \text{ minimizes the use of ports} \mathcal{J}\}$$

***Proposition 1.3*** *Let $\mathcal{T}$ be a configuration of a $T$-switch. The configuration $\mathcal{T}$ is an optimal configuration for a specific pattern of connections $\gamma$ if the configuration of its internal switches belongs to $\mathcal{S}^\gamma$. In other words:*

$$\text{If } \mathcal{T} = \{\mathcal{C}^\alpha,\ \mathcal{C}^\beta \mid \mathcal{C}^\alpha,\ \mathcal{C}^\beta \in \mathcal{S}^\gamma\},\ \text{then } \mathcal{T} \text{ is optimal}$$

**Proof:** The proof is trivial. The configurations belonging to $\mathcal{S}^\gamma$ minimize the number of connections that use the internal link. Consequently, $\mathcal{T}$ is optimal since there are other configurations that minimize the total connections. □

The $T$-MINs are MIN networks built using $T$-switches. These networks can be both unidirectional and bidirectional. Our study in this report focuses only on the last ones, however the study of the unidirectional are quite similar.

***Definition 1.12*** *A $T$-MIN interconnection network is a MIN network in which all the switches are $T$-switches.*

***Definition 1.13*** *A $T$-BMIN interconnection network is a BMIN network in which all the switches are $T$-switches.*

## 4.1 Internal connections of Twin switches

A $k \times k$ $T$-switch, like any other same size full-duplex switch allows a set of connections between their ports. In particular, we have

- Forward and backward connections. There are $k \times k$ possible combinations that imply pass through the $T$-switch (in forward and backward directions). We denote by $CC(\langle s, o\rangle)$ (cross connections) the number of different connections between $k$ input ports and $k$ output ports, so $CC(\langle s, o\rangle) = k^2$. If necessary to make a difference between forward and backward directions we will denote by $CC_f(\langle s, o\rangle)$ and $CC_b(\langle s, o\rangle)$, respectively. Hence,

$$CC_f(\langle s, o\rangle) = CC_b(\langle s, o\rangle) = k^2$$

- Turnaround connections. Only $k$ ports take part to establish this type of connection. We denote by $TC(\langle s, o\rangle)$ (turnaround connections) the number of different connections between such ports. It is assumed that there is no turnaround connection between a port and itself. Hence,

$$TC(\langle s, o\rangle) = k(k-1)$$

For the two connection types, considering the internal organization of the $T$-switch, some of them imply to go across the internal links $\mathcal{J}$ that interconnect the switches $\alpha$ and $\beta$. We denote by $CC_I(\langle s, o\rangle)$ the number of paths that go across the $\langle s, o\rangle$ switch by using the switches $\alpha$ and $\beta$. Similarly, We denote by $TC_I(\langle s, o\rangle)$ the number of paths that turn around the $\langle s, o\rangle$ switch by using the switches $\alpha$ and $\beta$. When necessary, we will differentiate the forward from backward direction by $CC_{If}(\langle s, o\rangle)$ and $CC_{Ib}(\langle s, o\rangle)$, respectively. In Figure 2, and taking into account the expressions 7 and 8 it is deduced that:

$$CC_I(\langle s, o\rangle) = p{\times}p + q{\times}q = p^2 + (k-p)^2 = 2p^2 + k^2 - 2kp$$
$$TC_I(\langle s, o\rangle) = p{\times}q + q{\times}p = 2p(k-p) = 2(kp - p^2)$$

and also

$$CC_{If}(\langle s, o\rangle) = p{\times}p + q{\times}q = p^2 + (k-p)^2 = 2p^2 + k^2 - 2kp$$
$$CC_{Ib}(\langle s, o\rangle) = p{\times}p + q{\times}q = p^2 + (k-p)^2 = 2p^2 + k^2 - 2kp$$

The total number of times that the internal links connecting the internal switches $\alpha$ and $\beta$ on a $\langle s, o\rangle$ $T$-switch would be used, is denoted by $C_i(\langle s, o\rangle)$, and it is obtained from the $CC_I(\langle s, o\rangle)$ ($CC_{If}(\langle s, o\rangle)$ and $CC_{Ib}(\langle s, o\rangle)$ if applicable) and $TC_I(\langle s, o\rangle)$ in each case are obtained under the initial conditions.

In the previous, we saw the expressions $CC_I(\langle s, o\rangle)$ and $TC_I(\langle s, o\rangle)$ that have been obtained considering the isolated $T$-switch, without taking into account the entire network; nor the paths that are routed by the concrete routing algorithm; nor the characteristics of the traffic that determine the paths. Once all these aspects are considered, it is possible to determine what happens in individual cases and therefore we can get the configuration of all network $T$-switches to minimize the number of crosses by the internal links [2] that interconnect the switches $\alpha$ and $\beta$.

In such cases, the number of paths that pass through switches can be calculated using these expressions or not (depending on the characteristics of each case), all the paths that pass through or turn on each $\langle s, o\rangle$ switch, and which of them make it through the switches $\alpha$ and $\beta$.

To obtain these expressions we need to know when a path reaches the switch $\langle s, o\rangle$. Considering that is a BMIN network topology of $N$ nodes and $n$ stages, a switch $\langle s, o\rangle$ will be achieved:

- from a switch located in a previous stage or from one terminal node to reach a later stage. In this case the path goes across the switch. We denote by $C_f(\langle s, o\rangle)$ (a.k.a. forward crosses) the number of paths that go across the switch $\langle s, o\rangle$, $0 \leq s < n-1$, in the forward direction. This is true for switches belonging to all stages except the last.

- from a switch located in a later stage to arrive at an previous stage or a terminal node. In this case the path goes across the switch. We denote by $C_b(\langle s, o\rangle)$ (a.k.a. backward crosses) the number of paths that go across the switch $\langle s, o\rangle$, $0 \leq s < n-1$, in the backward direction. This is also true for switches belonging to all stages except the last.

- from a switch located in a previous stage or a terminal node to reach another different switch in the same stage or different terminal node. In this case there is a twist on the switch itself. We denote by $T(\langle s, o\rangle)$ (a.k.a. turnaround connection) the number of paths that turn around the switch $\langle s, o\rangle$. Unlike earlier, this is true in all switches.

[2] In what follows, we will also use the term "internal link" to refer to the ports $\mathcal{J}$ that connect the switches $\alpha$ and $\beta$.

Sometimes it will be convenient to consider $C_f(\langle s, o\rangle)$ and $C_b(\langle s, o\rangle)$ together. Thus, we also introduce $C(\langle s, o\rangle)$ (total crosses) as $C(\langle s, o\rangle) = C_f(\langle s, o\rangle) + C_b(\langle s, o\rangle)$.

The above expressions are switch-level expressions, but they do not distinguish between individual ports. However, for this study it is necessary to know which connections are established between individual ports and how many times. This information will determine which of them may be made without using the internal switches $\alpha$ and $\beta$.

Therefore, we will also consider counters similar to those introduced above, but at connection level. But we will distinguish among those going in the forward, downward, or turn-around connections. To register the number of occurrences of each event will be used, $C_f(\langle s, o\rangle, l, l')$, $C_b(\langle s, o\rangle, l, l')$ and $T(\langle s, o\rangle, l, l')$, respectively. In short:

- $C_f(\langle s, o\rangle, l, l')$ records the number of times it is used the connection between the ports $l$ and $l'$, with $0 \leq l < k$ and $k \leq l' < 2k$.
- $C_b(\langle s, o\rangle, l, l')$ records the number of times it is used the connection between the ports $l$ and $l'$, with $k \leq l < 2k$ and $0 \leq l' < k$.
- $T(\langle s, o\rangle, l, l')$ records the number of times it is used the connection between the ports $l$ and $l'$, with $0 \leq l, l' < k$ and $l \neq l'$. The sum of the first two are denoted by $C(\langle s, o\rangle, l, l')$.

Adding the first two we obtain $C(\langle s, o\rangle, l, l')$.

# 5 Reachable Nodes from a BMIN Switch

Notice that although the methodology is general and it can be applied considering different $C$-switches and interconnection networks, the optimal $T$-switch configuration depends on the particular network properties. In this case we have considered the following network properties: BMINs $k$–ary $n$–tree with $N$ terminals and $k$×$k$ $T$-switches ($k \geq 4$), DESTRO routing algorithm, complement and perfect-shuffle traffic patterns. Therefore, in order to apply the switch configuration methodology we must specify the network topology, the routing algorithm, and the network load.

We have chosen the BMIN $k$–ary $n$–tree network [DYN03], a subclass of fat-trees which are one of the most common topologies today in the largest supercomputers on the Top500 list. The $k$–ary $n$–tree network topology belongs to the family of fat–trees and it is derived from a concrete class of MINs: the $k$–ary $n$–butterflies (or $k$–ary $n$-flies) [Lei92]. A $k$–ary $n$–fly MIN is obtained by applying the $\beta_i^k$ permutation, $0 \leq i < n$, to obtain the network-level connection patterns between stages. The $k$–ary $n$–tree connect $N$ nodes using $nk^{n-1}$ switches. Two switches $\langle s, o_{n-2} \dots o_0 \rangle$ and $\langle s', {o'}_{n-2} \dots {o'}_0 \rangle$ are connected with a link if $s' = s + 1$ and $o_i = {o'}_i \ \forall i \neq s$. Moreover, there is a link between the switch $\langle 0, o_{n-2} \dots o_0 \rangle$ and the terminal $h = h_{n-1} \dots h_0$ if $o_i = h_{i+1}, 0 \leq i < n-1$.

The routing algorithm is DESTRO [GGG+07]. It is a deterministic routing algorithm for fat-trees. It is based on using at each switch the ascending output port given by the destination component of the packet that is being routed corresponding to the switch stage. This routing algorithm is able to evenly balance network traffic and reduce to the minimum the number of paths that share each link, and as a consequence, it reduces network contention.

Complement and perfect-shuffle traffic patterns are assumed as network workload because they are frequently used in many studies about interconnection networks.

Under these conditions, we formally demonstrate what is the best configuration of the $T$-switches. The following three sections correspond to the steps in the methodology.

For a further treatment, it is interesting to know the reachable nodes from a particular BMIN switch. We have distinguished two cases: (1) in the first case we only take into account the topological capabilities of butterfly BMINs; (2) however in the second case, we have additionally considered a routing algorithm. In both cases, we introduce some definitions and derived propositions. Before every definition, we give an example for a sake of clarity.

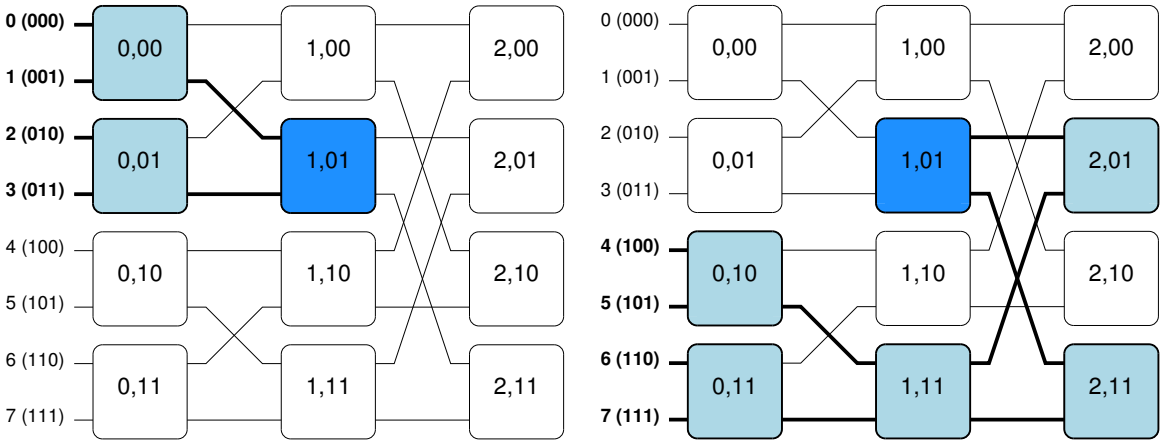
In all the examples, we assume a 2-ary 3-tree BMIN with $N = 8$ nodes, and consider the $\langle 1, 01 \rangle$ (dark blue) as the reference switch. On the other hand, we highlight in light blue the switches in the middle of the reference switch and the reachable nodes.

## 5.1 Reachable nodes from a BMIN switch considering the network topology

***Example 1.1*** *Topologically speaking, a path can reach the nodes {0,1,2,3} in the descending phase, and the nodes {4,5,6,7} in the ascending phase, from the* $\langle 1, 01 \rangle$ *switch, as it can be seen in the Figures 3(a) and 3(b), respectively.*

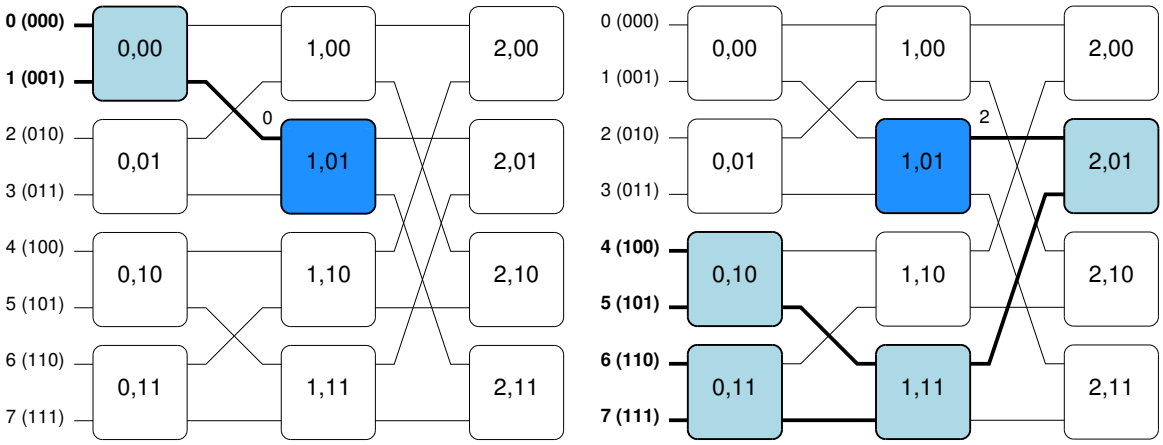
***Example 1.2*** *Topologically speaking, a path can reach the nodes {0,1} in the descending phase by using the output port 0, and the nodes {4,5,6,7} in the ascending phase by using the output port 2, from the* $\langle 1, 01 \rangle$ *switch, as it can be seen in the figures 4(a) and 4(b), respectively.*

In a more formal way, given a $k$-ary $n$-tree BMIN network with $N$ nodes, we introduce the following definitions:




(a) Descending phase. (b) Ascending phase.

Figure 3: Reachable nodes from a switch considering the topology.




(a) Descending phase by the output port 0. (b) Ascending phase by the output port 2.

Figure 4: Reachable nodes through a port from a switch considering the topology.

***Definition 1.14*** *Let $N_b^t(\langle s, o\rangle)$ be the network node set that are topologically reachable from the switch $\langle s, o\rangle$ by a path in the descending phase, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$ for $0 \leq s < n$. Hence,*

$$N_b^t(\langle s, o\rangle) = \{\ (h_{n-1} \dots h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1]\ \}$$

***Definition 1.15*** *Let $N_f^t(\langle s, o\rangle)$ be the network node set that are topologically reachable from the switch $\langle s, o\rangle$ by a path in the ascending phase, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$ for $0 \leq s < n-1$. Hence,*

$$N_f^t(\langle s, o\rangle) = (N_b^t(\langle s, o\rangle))^C = \{(h_{n-1} \dots h_0) : \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1}\}$$

*where $C$ refers to set complement operation.*

Similarly, we also define the reachable node set by the output port $l$ in the switch $\langle s, o\rangle$.

***Definition 1.16*** *Let $N_b^t(\langle s,o\rangle,l)$ be the network node set that are topologically reachable by the output port $l$ from the switch $\langle s,o\rangle$ by a path in the descending phase, where $\langle s,o\rangle = \langle s,(o_{n-2}\dots o_0)\rangle$ for $0 \le s < n$ and $0 \le l < k$. Hence,*

$$N_b^t(\langle s,o\rangle,l) = \{(h_{n-1}\dots h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1,n-1]\ y\ h_s = l\}$$

***Definition 1.17*** *Let $N_f^t(\langle s,o\rangle,l)$ be the network node set that are topologically reachable by using the output port $l$ from the switch $\langle s,o\rangle$ by a path in the ascending phase, where $\langle s,o\rangle = \langle s,(o_{n-2}\dots o_0)\rangle$ for $0 \le s < n-1$ and $k \le l < 2k$. Hence*

$$N_f^t(\langle s,o\rangle,l) = \{(h_{n-1}\dots h_0) :\ \exists i \in [s+1,n-1] \mid h_i \neq o_{i-1}\}$$

Taking the previous definitions as the starting point, the following propositions can be considered:

***Proposition 1.4*** *The total number of topologically reachable network nodes from the switch $\langle s,o\rangle$ by a path in the descending phase is $D_b^t(\langle s,o\rangle) = k^{s+1}$.*

**Proof:** Every node identifier can be written as a sequence of $n$ digits:

$$(h_{n-1}\dots h_{s+1}h_sh_{s-1}\dots h_0)$$

and according to Definition 1.14, the sequence can be rewritten as:

$$(o_{n-2}\dots o_{s+1}o_sh_sh_{s-1}\dots h_0)$$

that is, the $n-1-s$ most significant digits $o_{n-2}\dots o_s$ are the same for all the set members. The $s+1$ remaining digits $h_s\dots h_0$ may be different and distinguish the set nodes between them. Every $h_i$ would take values inside the range [0, k], that is, $k$ different values. Therefore, the number of different sequences, or the number of set nodes is $k^{s+1}$. □

***Proposition 1.5*** *The total number of topologically reachable network nodes from the switch $\langle s,o\rangle$ by a path in the ascending phase is $D_f^t(\langle s,o\rangle) = k^n - k^{s+1}$.*

**Proof:** According to Definition 1.15, $N_f^t(\langle s,o\rangle) = (N_b^t(\langle s,o\rangle))^C$. Therefore,

$$\begin{aligned} D_f^t(\langle s,o\rangle) &= card(N_f^t(\langle s,o\rangle)) = card((N_b^t(\langle s,o\rangle))^C) = \\ &= card(\mathcal{H}) - card(N_b^t(\langle s,o\rangle)) = N - card(N_b^t(\langle s,o\rangle)) = N - k^{s+1} = k^n - k^{s+1} \end{aligned}$$

□

***Proposition 1.6*** *The total number of topologically reachable network nodes by using the output port $l$ from the switch $\langle s,o\rangle$ by a path in the descending phase, where $0 \le s < n$ and $0 \le l < k$, is $D_b^t(\langle s,o\rangle,l) = k^s$.*

**Proof:** If $s > 0$, the topologically reachable network nodes by using the output port $l$ from the switch $\langle s,o\rangle$ by a path in the descending phase are the same as the reachable nodes from the switch $\langle s-1,o'\rangle$, which $\langle s,o\rangle$ is connected to through its port $l$, hence

$$D_b^t(\langle s,o\rangle,l) = D_b^t(\langle s-1,o'\rangle) = k^{(s-1)+1} = k^s$$

If $s = 0$, only one node is connected through the port $l$, and $D_b^t(\langle 0,o\rangle,l) = k^s$ is also verified because $k^0 = 1$. □

***Proposition 1.7*** *The total number of topologically reachable network nodes by using the output port $l$ from the switch $\langle s, o\rangle$ by a path in the ascending phase, where $0 \leq s < n-1$ and $k \leq l < 2k$, is $D_f^t(\langle s, o\rangle, l) = k^n - k^{s+1}$.*

**Proof:** According to Definitions 1.15 and 1.17, the sets $N_f^t(\langle s, o\rangle)$ and $N_f^t(\langle s, o\rangle, l)$ have the same members. Hence,

$$D_f^t(\langle s, o\rangle, l) = D_f^t(\langle s, o\rangle) = k^n - k^{s+1}$$

□

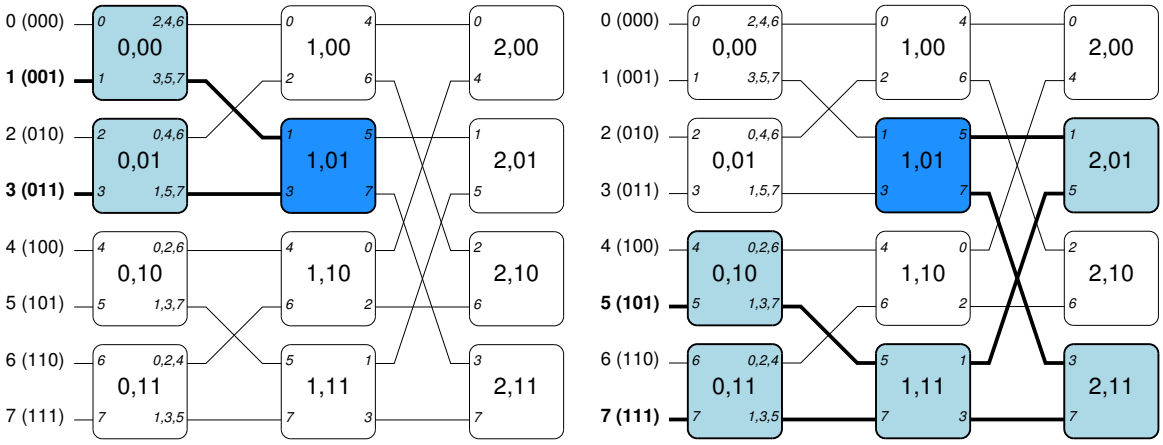
## 5.2 Reachable nodes from a BMIN switch considering the network topology and the routing algorithm

Similar to Section 5.1, we also give some examples to understand the following definitions and propositions. The reference switch and the reachable nodes are highlighted in dark and light blue, respectively. Furthermore, every output port is labeled with the reachable network nodes identifier, which are calculated by a routing algorithm likewise the described algorithm in Section A.7.1.

Considering the routing algorithm, we have to take into account that given a switch, i.e., $\langle 1, 01\rangle$, only a concrete destinations are reachable:

***Example 1.3*** *From switch $\langle 1, 01\rangle$, the reachable destinations are {1,3} in the descending phase, and {5,7} in the ascending phase, as it can be seen in the figures 5(a) and 5(b), respectively.*

For the reason stated just before starting this example, nodes 0 and 2 are not reachable because the routing algorithm prevents (in every hop) paths for reaching the switch $\langle 1, 01\rangle$.

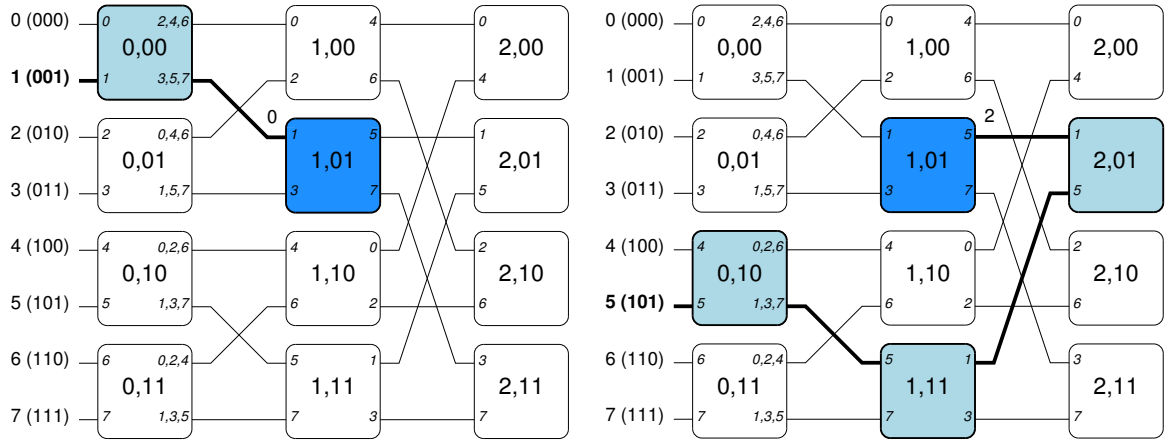



(a) Descending phase. (b) Ascending phase.

Figure 5: Reachable nodes from the switch considering the routing algorithm.

***Example 1.4*** *Let us take the switch $\langle 1, 01\rangle$ as the reference point. Through the port 0, a path can only reach the node {1} in the descending phase; but through the port 2, the path can only reach the node {5} in the ascending phase. This can be observed in the figures 6(a) and 6(b), respectively.*

In a more formal way, given a $k$-ary $n$-tree BMIN network with $N$ nodes and considering **DESTRO** (Section A.7.1) as the routing algorithm, $R$, we introduce the definitions below:




(a) Descending phase by the output port 0.

(b) Sentido ascendente by the output port 2.

Figure 6: Reachable nodes through a port from a switch considering the routing algorithm.

***Definition 1.18*** *Let $N_b^R(\langle s, o\rangle)$ be the reachable network nodes from the switch $\langle s, o\rangle$ by a path in the descending phase, considering $R$, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$, $0 \leq s < n$. Hence*

$$N_b^R(\langle s, o\rangle) = \{(h_{n-1} \dots h_0) : \ h_i = o_{i-1} \ \forall i \in [s+1, n-1],\ h_i = o_i \ \forall i \in [0, s-1]\}$$

***Definition 1.19*** *Let $N_f^R(\langle s, o\rangle)$ be the reachable network nodes from the switch $\langle s, o\rangle$ by a path in the ascending phase, considering $R$, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$, $0 \leq s < n-1$. Hence,*

$$N_f^R(\langle s, o\rangle) = \{(h_{n-1} \dots h_0) : \ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1} \text{ and } h_i = o_i \ \forall i \in [0, s-1]\}$$

Similarly, we also define the reachable node set by the output port $l$ in the switch $\langle s, o\rangle$.

***Definition 1.20*** *Let $N_b^R(\langle s, o\rangle, l)$ be the reachable network node set by using the output port $l$ from the switch $\langle s, o\rangle$ by a path in the descending phase, considering $R$, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$ with $0 \leq s < n$ and $0 \leq l < k$. Hence,*

$$N_b^R(\langle s, o\rangle, l) = \{(h_{n-1} \dots h_0) : \ h_i = o_{i-1} \ \forall i \in [s+1, n-1],\ h_s = l, \ and h_i = o_i \ \forall i \in [0, s-1]\}$$

***Definition 1.21*** *Let $N_f^R(\langle s, o\rangle, l)$ be the reachable network node set by using the output port $l$ from the switch $\langle s, o\rangle$ by a path in the ascending phase, considering $R$, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$ with $0 \leq s < n-1$ and $k \leq l < 2k$. Hence*

$$N_f^R(\langle s, o\rangle, l) = \{(h_{n-1} \dots h_0) : \ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1}, \text{ and } h_s = l - k, \ y \ h_i = o_i \ \forall i \in [0, s-1]\}$$

Taking the previous definitions as the starting point, the following propositions can be considered:

***Proposition 1.8*** *The number of reachable network nodes from the switch $\langle s, o\rangle$ by a path in the descending phase, considering $R$, is $D_b^R(\langle s, o\rangle) = k$.*

**Proof:** The network node identifier is a sequence of $n$ digits:

$$(h_{n-1} \dots h_{s+1} h_s h_{s-1} \dots h_0)$$

and according to Definition 1.18, the sequence can be rewritten as:

$$(o_{n-2} \dots o_{s+1} o_s h_s o_{s-1} \dots o_0)$$

All the digits are the same as those of the switch $\langle s, o \rangle$, with the exception of the digit $h_s$. Therefore, the digit $h_s$ makes the difference of the node identifiers in the set. Since $h_s$ could take $k$ values ($0 \leq h_s < k$), there are $k$ different members belonging to the set $N_b^R(\langle s, o \rangle)$, and consequently $D_b^R(\langle s, o \rangle) = k$. □

***Proposition 1.9*** *The number of reachable network nodes from the switch $\langle s, o \rangle$ by a path in the ascending phase, considering R, is $D_f^R(\langle s, o \rangle) = k^{n-s} - k$.*

**Proof:** Every node identifier is a sequence of $n$ digits with the format below:

$$(h_{n-1} \dots h_{s+2} h_{s+1} h_s h_{s-1} \dots h_0)$$

and according to Definition 1.19, such a sequence can be rewritten as:

$$(h_{n-1} \dots h_{s+2} h_{s+1} h_s o_{s-1} \dots o_0)$$

The Definition 1.19 states the $n - s - 1$ most significant digits, $h_{n-2} \dots h_s$ provide $k^{n-s-1}$ different combinations of valid sequences. However, there is only one combination that does not verify the Definition 1.19. Specifically, it is that which has $h_{n-1} \dots h_{s+1} = o_{n-2} \dots o_s$. In that way, the number of combinations for the $n - s - 1$ left most significant digits is $k^{n-s-1} - 1$.

On the other hand, the $s$ least significant digits, $h_{s-1} \dots h_0$, are fixed by Definition 1.19.

Finally, the digit $h_s$ would take one value of $k$ possibilities. Therefore, the number of possible combinations for node identifiers, that is, the members of the set $N_f^R(\langle s, o \rangle)$ is $(k^{n-s-1} - 1) \times k = k^{n-s} - k$. □

***Proposition 1.10*** *The number of reachable network nodes by using the output port l from the switch $\langle s, o \rangle$ by a path in the descending phase, considering R, where $0 \leq l < k$ and $0 \leq s < n$, is $D_b^R(\langle s, o \rangle, l) = 1$.*

**Proof:** According to Proposition 1.8, the number of reachable network nodes from the switch $\langle s, o \rangle$ by a path in the descending phase is $k$, and because $R$ is balanced, those nodes are uniformly distributed between the $k$ output ports, and therefore, $D_b^R(\langle s, o \rangle, l) = D_b^R(\langle s, o \rangle)/k = k/k = 1$. □

***Proposition 1.11*** *The number of reachable network nodes through the output port l from the switch $\langle s, o \rangle$ by a path in the ascending phase, considering R, where $k \leq l < 2k$ and $0 \leq s < n - 1$ is $D_f^R(\langle s, o \rangle, l) = k^{n-s-1} - 1$.*

**Proof:** According to Proposition 1.9, the number of reachable nodes from the switch $\langle s, o \rangle$ by a path in the ascending phase is $k^{n-s} - k$, and since $R$ is balanced, those nodes are uniformly distributed between the $k$ output ports, and therefore $D_f^R(\langle s, o \rangle, l) = D_f^R(\langle s, o \rangle)/k = (k^{n-s} - k)/k = k^{n-s-1} - 1$. □

***Proposition 1.12*** *Given an input port $l$, $k \leq l < 2k$, an output port $l'$, $0 \leq l' < k$, and both belong to a switch $\langle s, o\rangle$, where $\langle s, o\rangle = \langle s, (o_{n-2} \dots o_0)\rangle$ with $0 \leq s < n-1$, considering $R$, there exist paths in the descending phase that arrive at $\langle s, o\rangle$ by $l$ and leave it through $l'$ if and only if $l' = l - k$ is verified.*

**Proof:** Let us suppose that a path in the descending phase arrives at the switch $\langle s, o\rangle$, and let $h'$ be the destination node of that path. The path would leave the switch through the output port $l'$ in the switch $\langle s, o\rangle$.

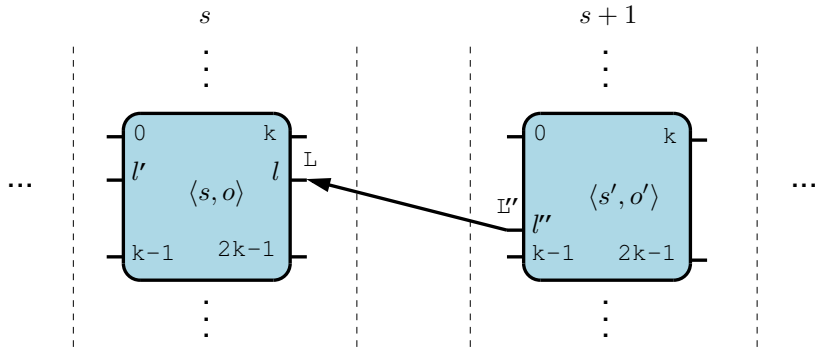



Figure 7: Associated channel between two adjacent switches.

Since $R$ is the routing algorithm, the Definition 1.20 determines the reachable nodes by a path in the descending phase in $\langle s, o\rangle$ by using $l'$. According to Proposition 1.10, such a set has only one member, this is, the destination node $h'$:

$$N_b^R(\langle s, o\rangle, l') = \{\ h'\ \} = \{\ (o_{n-2} \dots o_{s+1} o_s l' o_{s-1} \dots o_0)\ \}$$

Since the path is in the descending phase, it comes from one of the switches placed in the upwards stage $\langle s', o'\rangle$, with $s' = s + 1$. If a path passes through two switches, each one belonging to different stages, both switches are connected by a channel. For example, in the Figure 7, the switches $\langle s, o\rangle$ and $\langle s', o'\rangle$ are connected through the ports $l$ and $l''$. Taking into account the $k$-ary $n$-tree topology definition, it is known the switch identifiers verify $o_i = {o'}_i\ \forall i \neq s$, in other words:

$$\langle s', o'\rangle = \langle s+1, (o_{n-2} \dots o_{s+1} o'_s o_{s-1} \dots o_0)\rangle$$

It should be notice all the values that $o'$ can take are known except ${o'}_s$.

On the other hand, the Definition 1.20 specifies the reachable network node set in the descending phase from the switch $\langle s', o'\rangle$ through the output port $l''$. Such a set has only one member: the destination node $h'$:

$$N_b^R(\langle s', o'\rangle, l'') = \{h'\} = \{\ (o_{n-2} \dots o_{s+1} l'' {o'}_s o_{s-1} \dots o_0)\ \}$$

Since the sets $N_b^R(\langle s, o\rangle, l')$ and $N_b^R(\langle s', o'\rangle, l'')$ have the same members, $\{h'\}$,

$$N_b^R(\langle s, o\rangle, l') = \{\ (o_{n-2} \dots o_{s+1} o_s l' o_{s-1} \dots o_0)\ \}$$
$$N_b^R(\langle s', o'\rangle, l'') = \{\ (o_{n-2} \dots o_{s+1} l'' {o'}_s o_{s-1} \dots o_0)\ \}$$

that is checked only if:

$$o'_s = l'$$
$$o_s = l''$$

Let us analyze the relation between $l$ and $l''$ with the connection pattern. The pattern was defined in Section A.2 in terms of global ports $L$ and $L''$. The number of internal port $l$, $k \leq l < 2k$, is associated with $L = l_{n-1} \dots l_0$ by the connection pattern as:

$$L = k \times o + (l - k) = k \times (o_{n-2} \dots o_0) + (l - k) = o_{n-2} \dots o_0 (l - k)$$

The multiplication of $o = o_{n-2} \dots o_0$ (in base $k$) by $k$ is calculated by shifting $o = o_{n-2} \dots o_0$ left one position, and assigning the digit $o_0 = 0$. Then the addition operation sets the digit $o_0$ to $l - k$ $(0 \leq l - k < k)$.

Similarly, the global port number $L'' = l''_{n-1} \dots l''_0$ is related with the internal port $l''$, $0 \leq l'' < k$, by the expression below:

$$L'' = k \times o' + l'' = k \times (o'_{n-2} \dots o'_0) + l'' = o'_{n-2} \dots o'_0 l''$$

Then again, the butterfly permutation associates the ports $L$ and $L''$ as follows:

$$\beta^k_{s+1}(L) = L''$$
$$\beta^k_{s+1}(o_{n-2} \dots o_{s+1} o_s o_{s-1} \dots o_0 (l - k)) = o'_{n-2} \dots o'_{s+1} o'_s o'_{s-1} \dots o'_0 l''$$

taking into account that $o'_s = l'$ and $l'' = o_s$, it is obtained in $L''$

$$\beta^k_{s+1}(o_{n-2} \dots o_{s+1} o_s o_{s-1} \dots o_0 (l - k)) = o'_{n-2} \dots o'_{s+1} l' o'_{s-1} \dots o'_0 o_s \tag{9}$$

how the switches $\langle s, o \rangle$ and $\langle s', o' \rangle$ are related

$$\langle s', o' \rangle = \langle s + 1, (o_{n-2} \dots o_{s+1} o'_s o_{s-1} \dots o_0) \rangle$$

and it can be substituted in the Expression 9, remaining then

$$\beta^k_{s+1}(o_{n-2} \dots o_{s+1} o_s o_{s-1} \dots o_0 (l - k)) = o_{n-2} \dots o_{s+1} l' o_{s-1} \dots o_0 o_s$$

at long last, the butterfly permutation is applied

$$o_{n-2} \dots o_{s+1} (l - k) o_{s-1} \dots o_0 o_s = o_{n-2} \dots o_{s+1} l' o_{s-1} \dots o_0 o_s \tag{10}$$

The equality 10 is verified when $l' = l - k$. □

It should be noted this section has covered the first step, which is independent of the traffic pattern, of our methodology. The two other steps take into consideration the traffic pattern: complement and perfect-shuffle traffic pattern. We briefly describe both steps bellow:

- This is the second step. Depending on the network characteristics and load conditions, few or many different $C$-switch configurations could be obtained. In this step, $C$-switches are grouped according to their connection requirements, and so, several types of $C$-switch will be distinguished.

  As result of the previous phase, it can occur that some of the possible connections in the $C$-switches support one, or more paths, and however there may be connections that are never established.

  In a fat-tree topology, for instance, $C$-switches in different stages may require different switch-level connection patterns, and the same may even occur with $C$-switches in the same stage. When a simple traffic pattern and balanced routing algorithm are used, it is likely all the $C$-switches in the network require the same switch-level connection pattern.

- This is the third step. From connection requirements and given the number of internal switches forming the $C$-switch, this last step consists in finding the optimal configuration for each class of $C$-switch. That is, we must find the optimal switch-level connection pattern of each class, trying to minimize the use of the interconnection between internal switches.

Due to the length of these steps, we have included them in two separate sections. Hence, the Section 6 and Section 7 deal with the complement traffic pattern and perfect-shuffle, respectively.

# 6 Applying the Methodology for Complement Traffic

This section performs the search of optimal $T$-switch configuration using the same methodology. On this occasion, the network is evaluated under complement traffic pattern.

## 6.1 Network Paths Analysis

The generated paths with the complement traffic pattern are studied in this section. Figure 8 shows graphically the paths generated by this traffic pattern in a 2–ary 3–tree network. There are so many different paths as end nodes, and the internal switch connections are determined by the specific routing algorithm. As it is shown, all the paths reach the switches of the last stage (Proposition 1.16).

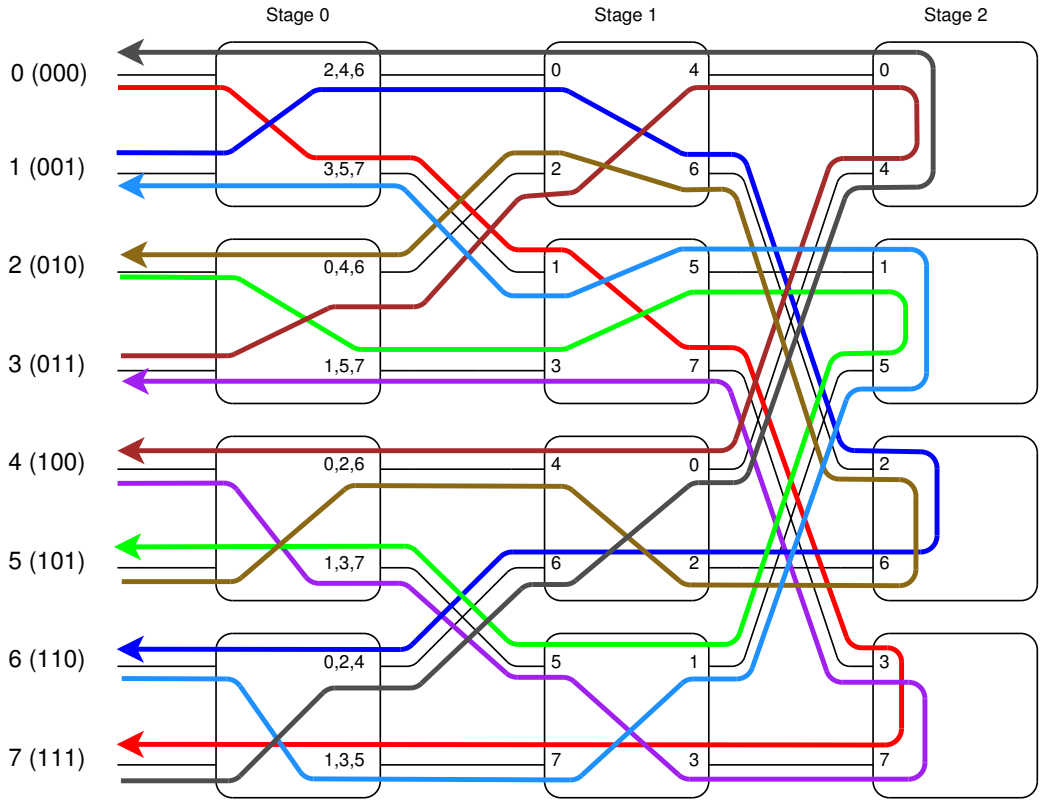



Figure 8: Generated paths under complement traffic pattern in a 2–ary 3–tree network.

To avoid continually repeating the same premises in every definition, they are now indicated and then omitted from the propositions. In this way, the statements remain more clear and simple. Specifically, the premises are as follows:

- The network topology is a $T$–BMIN $k$–ary $n$–tree with $N$ end nodes.
- The network load is generated by the complement traffic pattern.
- The routing algorithm is that defined in Section A.7.1. It is a deterministic in the ascending phase and self–routing in the descending phase.

### 6.1.1 Ascending phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o \rangle$ in the ascending phase, $0 \leq s < n$, are described below.

***Proposition 1.13*** *Given the ports $l$ and $l'$ of the switch $\langle s,o\rangle$, $0 \leq s < n-1$, $0 \leq l < k$ and $k \leq l' < 2k$, at the most there is one path passing through $\langle s,o\rangle$ in the ascending phase by using $l$ and $l'$, where $l' = \overline{l} + k$. Hence,*

$$C_f(\langle s,o\rangle, l, l') = \begin{cases} 1, & \text{if and only if } l' = \overline{l} + k \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path passes through the switch $\langle s,o\rangle$ in the ascending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle s,o\rangle, l)$, the destination node, $h'$, belongs to the set $N_f^R(\langle s,o\rangle, l')$ and the identifier $h'$ is obtained by applying the complement function to $h$. Therefore,

$$C_f(\langle s,o\rangle, l, l') = \text{card}((N_b^t(\langle s,o\rangle, l))^\pi \cap N_f^R(\langle s,o\rangle, l'))$$

$$\begin{aligned}(N_b^t(\langle s,o\rangle, l))^\pi \cap N_f^R(\langle s,o\rangle, l') = \ldots = \\ = (\{(h_{n-1}\ldots h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1] \text{ and } h_s = l\ \})^\pi \cap \\ \cap \{(h_{n-1}\ldots h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1},\ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l' - k\}\end{aligned}$$

Then, applying the complement function to the elements of the set of sources $N_b^t(\langle s,o\rangle, l)$, we obtain

$$\begin{aligned}(N_b^t(\langle s,o\rangle, l))^\pi \cap N_f^R(\langle s,o\rangle, l') = \\ = \{(h_{n-1}\ldots h_0) :\ h_i = \overline{o_{i-1}}\ \forall i \in [s+1, n-1] \text{ and } h_s = \overline{l}\} \cap \\ \cap \{(h_{n-1}\ldots h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1},\ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l' - k\}\end{aligned}$$

According to first set, $h_i = \overline{o_{i-1}}\ \forall i \in [s+1, n-1]$, and from second set, at least one $h_i$ is not equal to $o_{i-1}$. The first condition satisfies the second one at the same time, so all the $h_i$ are different from $o_{i-1}$. The first condition is the most restrictive. Hence, the following set is obtained

$$\begin{aligned}(N_b^t(\langle s,o\rangle, l))^\pi \cap N_f^R(\langle s,o\rangle, l') = \{(h_{n-1}\ldots h_0) :\ h_i = \overline{o_{i-1}}\ \forall i \in [s+1, n-1],\ h_i = o_i\ \forall i \in [0, s-1], \\ h_s = \overline{l},\ \text{ and } h_s = l' - k\}\end{aligned}$$

that is to say

$$(N_b^t(\langle s,o\rangle, l))^\pi \cap N_f^R(\langle s,o\rangle, l') = \{(\overline{o_{n-2}}\ldots\overline{o_s}\overline{l}o_{s-1}\ldots o_0) \text{ if and only if } h_s = l' - k = \overline{l}\}$$

As all the digits of every node identifier are fixed, at the most there is a path in the ascending phase that passes through the switch $\langle s,o\rangle$ by using the ports $l$ and $l'$. Moreover, $h_s = l' - k = \overline{l}$ means that the unique path exists if $l' - k = \overline{l}$, in other words, if $l' = \overline{l} + k$. Since otherwise the result set would be empty. □

***Proposition 1.14*** *The number of paths in ascending phase passing through the switch $\langle s,o\rangle$, $0 \leq s < n-1$, is $k$. Hence,*

$$C_f(\langle s,o\rangle) = k$$

**Proof:** According to Proposition 1.13 we know that there are paths that pass through the switch $\langle s,o\rangle$ by using the ports $l$ and $l'$, where $0 \leq l < k$ and $k \leq l' < 2k$, if and only if $l' = \overline{l} + k$. Moreover, there is only one path for each pair $l$ and $l'$. Therefore, the total number of paths passing through the switch $\langle s,o\rangle$ in the ascending phase is obtained by calculating the number of pairs $l$ and $l'$ that fulfill the condition $l' = \overline{l} + k$.

The port $l$ takes values between 0 and $k-1$, so $l' = \overline{l} + k = 2k - l - 1$ takes values between $k$ and $2k-1$, i.e.,, inside the range where $l'$ is defined. Therefore, there are $k$ valid pairs of ports $l$ and $l'$ that are used by the paths in the ascending phase for passing through the switch $\langle s,o\rangle$. □

***Proposition 1.15*** *There is no path passing through, in the ascending phase, the switches of the last stage. Hence,*

$$C_f(\langle n-1, o\rangle) = C_f(\langle n-1, o\rangle, l, l') = 0$$

**Proof:** By the definition of the BMIN topology, there are no forward connections in the switches of the last stage. □

### 6.1.2 Turnaround phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o\rangle$ in the turnaround phase, $0 \leq s < n$, are described below.

***Proposition 1.16*** *All the paths turn around in the switches of the last stage.*

**Proof:** Given the nodes $h$ and $h'$, where $0 \leq h, h' < N$, there is a path between both if and only if $h' = \overline{h}$. For all the paths, the stage where the turnaround connection is established, is calculated by applying the function FirstDifference() (Definition 1.26 in Section A.5)

$$\text{FirstDifference}(h, h') = \text{FirstDifference}((h_{n-1} \dots h_0), (\overline{h_{n-1}} \dots \overline{h_0})) = n-1$$

Therefore, all the paths reach a switch of the last stage, and as a consequence, all the turnaround connections are produced that stage. □

***Proposition 1.17*** *Given the ports $l$ and $l'$ of the switch $\langle n-1, o\rangle$, where $0 \leq l < k$ and $0 \leq l' < k$, at the most there is one path that turns in the switch using input port $l$ and output port $l'$, such as $l' = \overline{l}$. Hence,*

$$T(\langle n-1, o\rangle, l, l') = \begin{cases} 1, & \textit{if and only if } l' = \overline{l} \\ 0, & \textit{otherwise} \end{cases}$$

**Proof:** A path is turned around in the switch $\langle n-1, o\rangle$ by the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle n-1, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle n-1, o\rangle, l')$ and the identifier $h'$ is obtained by applying the complement function to $h$. Hence,

$$T(\langle n-1, o\rangle, l, l') = \text{card}((N_b^t(\langle n-1, o\rangle, l))^\pi \cap N_b^R(\langle n-1, o\rangle, l'))$$

$$\begin{aligned}&(N_b^t(\langle n-1, o\rangle, l))^\pi \cap N_b^R(\langle n-1, o\rangle, l') = \\ &\quad = (\{(h_{n-1} \dots h_0) :\ h_{n-1} = l\})^\pi \ \cap\ \{(h_{n-1} \dots h_0) :\ h_i = o_i\ \forall i \in [0, n-2] \text{ and } h_{n-1} = l'\}\end{aligned}$$

Then the complement function is applied over the source set $N_b^t(\langle n-1, o\rangle, l)$. The fixed digits of $h$ are complemented and the value of the other digits are indifferent. We obtain

$$\begin{aligned}&(N_b^t(\langle n-1, o\rangle, l))^\pi \cap N_b^R(\langle n-1, o\rangle, l') = \\ &\quad = \{(h_{n-1} \dots h_0) :\ h_{n-1} = \overline{l}\} \ \cap\ \{(h_{n-1} \dots h_0) :\ h_i = o_i\ \forall i \in [0, n-2] \text{ and } h_{n-1} = l'\} \\ &\quad = \{(h_{n-1} \dots h_0) :\ h_i = o_i\ \forall i \in [0, n-2],\ h_{n-1} = \overline{l} \text{ and } h_{n-1} = l'\}\end{aligned}$$

that is to say

$$(N_b^t(\langle n-1, o\rangle, l))^\pi \cap N_b^R(\langle n-1, o\rangle, l') = \{\ (h_{n-1} \dots h_0) \text{ tal que } h_{n-1} = l' = \overline{l}\ \}$$

All the digits $h_i$ of all the node identifiers in the set are fixed. The $n-1$ least significant digits are defined by the switch and the most significant digit, $h_{n-1}$, is defined by the output port $l'$. So, at the most there is a path that turns around in the switch $\langle n-1, o\rangle$ using input port $l$ and output port $l'$. Moreover, $h_{n-1} = l' = \overline{l}$ means that path exists if $l' = \overline{l}$. Since otherwise the result set would be empty. □

***Proposition 1.18*** *The number of paths that turn around in a switch of the last stage is k. Hence,*

$$T(\langle n-1, o\rangle) = k$$

**Proof:** According to Proposition 1.17 there is a path that turns around in the switch $\langle n-1, o\rangle$ using input port $l$ and output port $l'$, $0 \leq l, l' < k$, whenever $l' = \overline{l}$.

There are $k$ different pairs of ports $l$ and $l'$ that satisfy this condition, and therefore the total number of paths that turn around in the switch $\langle n-1, o\rangle$ is $k$. □

### 6.1.3 Descending phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o\rangle$ in the descending phase, $0 \leq s < n$, are described below.

***Proposition 1.19*** *There is no path passing through, in descending phase, the switches of the last stage. Hence,*

$$C_b(\langle n-1, o\rangle) = C_b(\langle n-1, o\rangle, l, l') = 0$$

**Proof:** By the definition of the BMIN topology, there are no backward connections in the switches of the last stage. □

***Proposition 1.20*** *Given the ports $l$ and $l'$ of the switch $\langle s, o\rangle$, where $0 \leq s < n-1$, $k \leq l < 2k$ and $0 \leq l' < k$, at the most there is one path in the descending phase that passes through the switch by using the ports $l$ and $l'$.*

**Proof:** A path goes through the switch $\langle s, o\rangle$ in the descending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_f^t(\langle s, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle s, o\rangle, l')$ and the identifier $h'$ is obtained applying the complement function to $h$. Therefore,

$$C_b(\langle s, o\rangle, l, l') = \mathrm{card}((N_f^t(\langle s, o\rangle, l))^\pi \cap N_b^R(\langle s, o\rangle, l'))$$

$$\begin{aligned}
&(N_f^t(\langle s, o\rangle, l))^\pi \cap N_b^R(\langle s, o\rangle, l') = \\
&\quad = (\{(h_{n-1} \ldots h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1}\})^\pi \cap \\
&\qquad \cap \{(h_{n-1} \ldots h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1], h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l'\}
\end{aligned}$$

Then, the complement function is applied on the elements of the source set $N_f^t(\langle s, o\rangle, l)$. The members of this set are all the nodes, except the nodes that have $h_i = o_{i-1}, \forall i \in [s+1, n-1]$, as there exist any $h_i \neq o_{i-1}$. When it is complemented, the result set has all the nodes, except the nodes with $h_i = \overline{o_{i-1}}, \forall i \in [s+1, n-1]$. In other words, the members of the set have at least one $h_i$ that is not equal to $\overline{o_{i-1}}$:

$$\begin{aligned}
&(N_f^t(\langle s, o\rangle, l))^\pi \cap N_b^R(\langle s, o\rangle, l') = \\
&\quad = \{(h_{n-1} \ldots h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq \overline{o_{i-1}}\} \cap \\
&\qquad \cap \{(h_{n-1} \ldots h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1], h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l'\}
\end{aligned}$$

The first set imposes that there exists at least one $h_i$, where $i \in [s+1, n-1]$, different to $\overline{o_{i-1}}$, whereas the second set imposes that all the $h_i$ to be equal to $o_{i-1}$ in the interval $[s+1, n-1]$. The second condition also satisfies the first one at the same time, because all the digits $h_i$ are different to $\overline{o_{i-1}}$. Therefore the second condition is more restrictive than the first one. Thus, the following set is obtained:

$$(N_f^t(\langle s, o\rangle, l))^\pi \cap N_b^R(\langle s, o\rangle, l') = \\ = \{(h_{n-1} \dots h_0) : \ h_i = o_{i-1} \ \forall i \in [s+1, n-1], \ h_i = o_i \ \forall i \in [0, s-1], \text{ and } h_s = l'\}$$

that is to say

$$(N_f^t(\langle s, o\rangle, l))^\pi \cap N_b^R(\langle s, o\rangle, l') = \{(o_{n-2} \dots o_s l' o_{s-1} \dots o_0)\}$$

All the node identifier digits that meet the conditions are fixed, so at the most there is one pair of nodes $h$ and $h'$ that passes through the switch $\langle s, o\rangle$ by using the ports $l$ and $l'$ in descending phase.

It must be noticed that the obtained set only indicates the possible destination of the path and the output port of the switch, but it gives no information of the input port $l$, as it was the case of the paths in the ascending phase (Proposition 1.13). □

***Proposition 1.21*** *Given the ports $l$ and $l'$ of the switch $\langle s, o\rangle$, where $0 \le s < n-1$, $k \le l < 2k$ and $0 \le l' < k$, there is one path that uses these ports in the ascending phase if and only if $l' = l - k$.*

**Proof:** According to Proposition 1.20 at the most there is one path in the descending phase that uses the output port $l'$. Moreover, according to Proposition 1.12 $l' = l - k$ is always satisfied in descending phase. Hence,

$$C_b(\langle s, o\rangle, l, l') = \begin{cases} 1, & \text{if and only if } l' = l - k \\ 0, & \text{otherwise} \end{cases}$$

□

***Proposition 1.22*** *The number of paths passing through the switch $\langle s, o\rangle$ in descending phase, $0 \le s < n-1$, is $k$. Hence,*

$$C_b(\langle s, o\rangle) = k$$

**Proof:** According to Proposition 1.21 there exist paths that pass through the switch $\langle s, o\rangle$ by using the ports $l$ and $l'$, where $k \le l < 2k$ and $0 \le l' < k$, if and only if $l' = l - k$. Moreover, according to Proposition 1.20, there is only one path if $l' = l - k$. Therefore, the total number of paths in the descending phase that pass through the switch $\langle s, o\rangle$ is equal to the number of pairs of ports $l$ and $l'$ that satisfy the condition $l' = l - k$.

The port $l$ takes values between $k$ and $2k-1$, and as a consequence $l' = l - k$ takes values between 0 and $k-1$, i.e.,, in the range of $l'$. So there are $k$ valid pairs of ports $l$ and $l'$ that are used by other paths in descending phase to pass through the switch $\langle s, o\rangle$. □

***Proposition 1.23*** *The number of paths passing through the switch $\langle s, o\rangle$, $0 \le s < n-1$, is $2k$. Hence,*

$$C(\langle s, o\rangle) = 2k$$

**Proof:** The result is directly inferred from the Propositions 1.18 and 1.22.

$$C(\langle s, o\rangle) = C_f(\langle s, o\rangle) + C_b(\langle s, o\rangle) = k + k = 2k$$

□

To sum up, Table 1 outlines the expressions obtained in the previous propositions.

Table 1: Number of paths passing through the switch $\langle s, o \rangle$ in then ascending, turnaround and descending phases under complement traffic.

$$C_f(\langle s,o\rangle) = \begin{cases} k, & \text{if } s \in [0, n-2] \\ 0, & \text{if } s = n-1 \end{cases}$$

$$T(\langle s,o\rangle) = \begin{cases} 0, & \text{if } s \in [0, n-2] \\ k, & \text{if } s = n-1 \end{cases}$$

$$C_b(\langle s,o\rangle) = \begin{cases} k, & \text{if } s \in [0, n-2] \\ 0, & \text{if } s = n-1 \end{cases}$$

$$C(\langle s,o\rangle) = \begin{cases} 2k, & \text{if } s \in [0, n-2] \\ 0, & \text{if } s = n-1 \end{cases}$$

$$\underset{0\le l<k \quad k\le l'<2k}{C_f(\langle s,o\rangle, l, l')} = \begin{cases} 1, & \text{if } s \in [0, n-2] \text{ and } l' = \overline{l} + k \\ 0, & \text{otherwise} \end{cases}$$

$$\underset{0\le l<k \quad 0\le l'<k}{T(\langle s,o\rangle, l, l')} = \begin{cases} 1, & \text{if } s = n-1 \text{ and } l = \overline{l'} \\ 0, & \text{otherwise} \end{cases}$$

$$\underset{k\le l<2k \quad 0\le l'<k}{C_b(\langle s,o\rangle, l, l')} = \begin{cases} 1, & \text{if } s \in [0, n-2] \text{ and } l' = l - k \\ 0, & \text{otherwise} \end{cases}$$

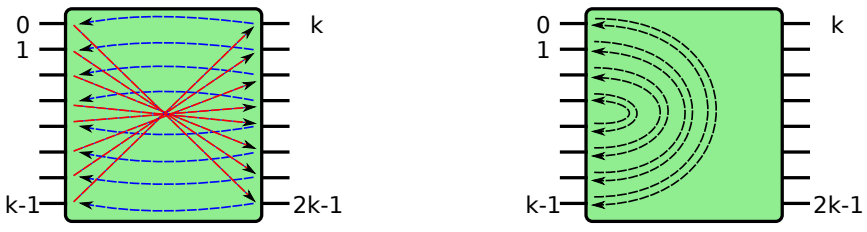
## 6.2 Switch Classification

According to the expressions in Table 1, when the network topology is a $N$ end nodes $T$–BMIN $k$–ary $n$–tree multistage network, the generated traffic is based on the complement traffic pattern ($\pi$) and the paths are determined by the routing algorithm defined in Section A.7.1, two types of switches are identified according to the connections required in the switches:

**Type** $\pi a$ All the switches, except those belonging to the last stage, fall into this class. Forward and backward connections are required in these switches. Forward connections are established from the input port $l$ to the output port $l' = 2k - l - 1$, $0 \le l < k$; and backward connections are established from input port $l$ to output port $l' = l - k$, $k \le l < 2k$.

**Type** $\pi b$ The switches of the last stage belong to this class, because they only require turnaround connections. Connection are established from the input port $l$ to the output port $l' = k - l - 1$, where $0 \le l, l' < k$, $l \ne l'$.

Figure 9 shows the internal connections for a 8×8 switch.




(a) Type $\pi a$. (b) Type $\pi b$.

Figure 9: Connections required for each type of switch.

## 6.3 Switch Configuration

To reach the most appropriate internal configuration of $T$-switches under complement traffic pattern, the methodology indicated in Section 2.2 is applied.

### 6.3.1 Type $\pi a$ configuration of switch

According to our methodology, firstly we identify the set of optimal configurations to forward and backward connections, separately. Based on these two sets, we derive the global optimal configuration for the $T$-switch.

#### 6.3.1.1 Optimal switch configuration to forward connections

From Proposition 1.13, there uniquely exist forward connections between the ports $l$, $0 \leq l < k$ and $l'$, $k \leq l' < 2k$, if and only if $l' = \bar{l} + k$. It is clear that if all pairs of ports satisfying this condition are connected to the same internal switch (i.e., $\alpha$ or $\beta$), the resulting configuration will minimize the number of forward connections that go across the internal link in a $T$-switch.

Next, to establish the relation between such ports, a binary relation $\mathcal{R}_f$ is defined. Furthermore, several definitions with respect to it are introduced to help obtaining later the set of optimal configurations of forward connections in $T$-switch.

***Definition 1.22*** *The binary relation $\mathcal{R}_f$ on a set $\mathcal{U}$ is defined as follows:*

$$\mathcal{R}_f = \{(l, l') \in \mathcal{U}^2 \mid l + l' = 2k - 1\}$$

Note that expression $l + l' = 2k - 1$ is equivalent to $l' = \bar{l} + k$ because:

$$\begin{aligned} l' &= \bar{l} + k \\ l' &= k - l - 1 + k \\ l' &= 2k - 1 - l \\ l + l' &= 2k - 1 \end{aligned}$$

***Proposition 1.24*** *$\mathcal{R}_f$ is a symmetric relation.*

**Proof:** If $\mathcal{R}_f$ is symmetric then it is verified that $\forall l, l' \in \mathcal{U}, (l, l') \in \mathcal{R}_f \Rightarrow (l', l) \in \mathcal{R}_f$. The demonstration is trivial by the equality $l + l' = l' + l = 2k - 1$. □

***Proposition 1.25*** *Let $l, l', l''$ be three ports such that $l, l', l'' \in \mathcal{U}$, and $(l, l') \in \mathcal{R}_f$. It is verified that $(l, l'') \in \mathcal{R}_f$ if and only if $l' = l''$.*

**Proof:** The demonstration is trivial. There exists only one $l'$ satisfying $l + l' = 2k - 1$. □

In other words, this proposition indicates that each port $l$ is associated to a unique port $l'$ by $\mathcal{R}_f$.

***Proposition 1.26*** *Let $\mathcal{S}_f^{\pi a}$ be the set of configurations that minimize the number of paths in the ascending phase that go across the internal link in a T-switch of type $\pi a$. Hence,*

$$\mathcal{S}_f^{\pi a} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_f\}$$

*and there are no forward connections that go across the internal link.*

**Proof:** Firstly, it is proved that $\mathcal{S}_f^{\pi a}$ is not empty and then its members are optimal configurations.

Let $\mathcal{C} \in \mathcal{S}_f^{\pi a}$ be a configuration. According to the definition of $\mathcal{S}_f^{\pi a}$, for a port $l \in \mathcal{C}$ there exists another port $l' \in \mathcal{C}$ such that $(l, l') \in \mathcal{R}_f$. This in turn implies that there exists a port $l'' \in \mathcal{C}$ such that $(l', l'') \in \mathcal{R}_f$; and another port $l''' \in \mathcal{C}$ such that $(l'', l''') \in \mathcal{R}_f$; and so on.

Nevertheless, the Propositions 1.24 and 1.25 demonstrate that $\mathcal{R}_f$ is symmetric, and if there exist a port $l''$ such that $(l', l'') \in \mathcal{R}_f$ then the reason is because $l'' = l$. Therefore, the inclusion of a port $l$ in $\mathcal{C}$ implies the inclusion of a unique port $l'$ satisfying $(l, l') \in \mathcal{R}_f$.

Moreover, since $card(\mathcal{C}) = k$ (Proposition 1.2), $k$ is even (initial hypothesis); and all the $k$ ports are grouped in pairs $(l, l')$ which verify the binary relation $\mathcal{R}_f$, there would be always configurations belonging to $\mathcal{S}_f^{\pi a}$. Hence $\mathcal{S}_f^{\pi a} \neq \emptyset$.

On the other hand, there are only connections between two ports $l$ and $l'$, if $(l, l') \in \mathcal{R}_f$. If $\forall l \in \mathcal{C}$, there exists a port $l'$ such that $(l, l') \in \mathcal{R}_f$, then all the connections are established between ports belonging to the same internal switch. Therefore, there exist no connections that use the internal link; and the members of $\mathcal{S}_f^{\pi a}$ are optimal configurations.

Finally, we show by reductio ad absurdum that the members of $\mathcal{S}_f^{\pi a}$ are the unique optimal configurations. Let $\mathcal{C}'$ be a configuration that minimizes the use of the internal link and $\mathcal{C}' \notin \mathcal{S}_f^{\pi a}$. This means there exists a port $l$ that is not related to another port $l'$.

$$\mathcal{C}' \notin \mathcal{S}_f^{\pi a} \ \Rightarrow\ \exists l \in \mathcal{C}' \mid \forall l' \in \mathcal{C}' \ \ (l, l') \notin \mathcal{R}_f$$

If the configuration $\mathcal{C}'$ is used to build a $T$-switch (it should be reminded $\mathcal{C}'$ determines the ports connected to each internal switch) the port $l$ must accomplish the forward connections to a port that is not allocated inside the same internal switch. Consequently, those connections have to use the internal link, and the configuration $\mathcal{C}'$ does not minimize the use of the internal link, since there are configurations better than $\mathcal{C}'$ (because they do not go across the internal link), which belong to $\mathcal{S}_f^{\pi a}$.

Therefore, the members of $\mathcal{S}_f^{\pi a}$ are configurations that minimize the number of connections that use the internal link. □

Example 1.5 shows two optimal configurations for a $T$-switch of type $\pi a$ considering the forward connections.

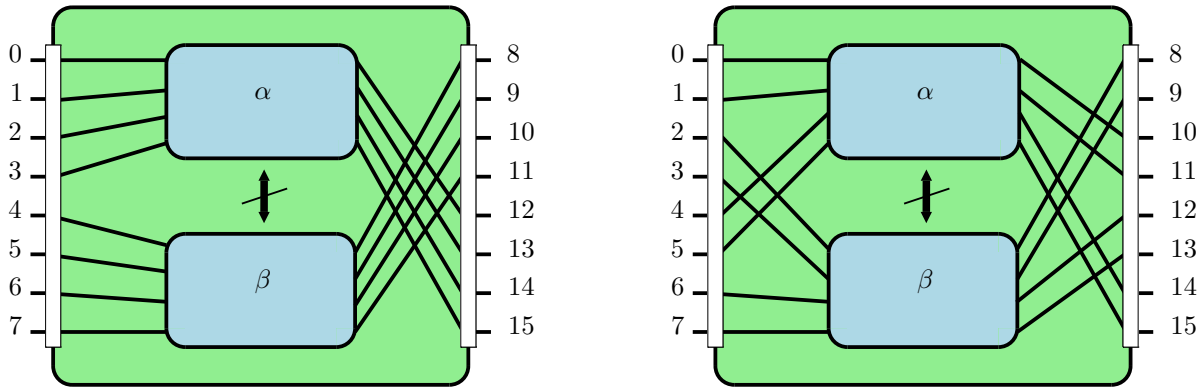
***Example 1.5*** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two configurations for a 8×8 T-switch*

$$\mathcal{T}_1 = \left\{\mathcal{C}_1^{\alpha}, \mathcal{C}_1^{\beta} \in \mathcal{V} \mid \mathcal{C}_1^{\alpha} = \{0, 1, 2, 3, 12, 13, 14, 15\},\ \mathcal{C}_1^{\beta} = (\mathcal{C}_1^{\alpha})^C = \{4, 5, 6, 7, 8, 9, 10, 11\}\right\}$$
$$\mathcal{T}_2 = \left\{\mathcal{C}_2^{\alpha}, \mathcal{C}_2^{\beta} \in \mathcal{V} \mid \mathcal{C}_2^{\alpha} = \{0, 1, 4, 5, 10, 11, 14, 15\},\ \mathcal{C}_2^{\beta} = (\mathcal{C}_2^{\alpha})^C = \{2, 3, 6, 7, 8, 9, 12, 13\}\right\}$$

*Both $\mathcal{T}_1$ and $\mathcal{T}_2$ are optimal configurations for the forward connections in a T-switch of type $\pi a$ because they verify*

$$\mathcal{C}_1^{\alpha},\ \mathcal{C}_1^{\beta},\ \mathcal{C}_2^{\alpha},\ \mathcal{C}_2^{\beta} \in \mathcal{S}_f^{\pi a}$$

*In the Figure 10, the connections for the configurations $\mathcal{T}_1$ and $\mathcal{T}_2$ are shown.*




(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$

Figure 10: Optimal switch configurations for a $8 \times 8$ $T$-switch of type $\pi a$ considering forward connections.

#### 6.3.1.2 Optimal switch configuration to backward connections

By Proposition 1.12, there exist backward connections between a port $l$, $k \leq l < 2k$, and another port $l'$, $0 \leq l' < k$, if and only if $l' = l - k$. Following a similar procedure to previous one in Section 6.3.1.1, the binary relation $\mathcal{R}_b$ is defined, and its corresponding Propositions, from which the set of optimal configurations for backward connections in a $T$-switch is derived. These configurations have also the property of being optimum for whatever traffic pattern generated in the network.

***Definition 1.23*** *The binary relation $\mathcal{R}_b$ on a set $\mathcal{U}$ is defined as follows*

$$\mathcal{R}_b = \{(l, l') \in \mathcal{U}^2 \mid |l - l'| = k \}$$

Note that expression $|l - l'| = k$ comes from $l' = l - k$ because

$$l' = l - k \Rightarrow \begin{cases} l' - l = -k \\ l - l' = k \end{cases} \Rightarrow \begin{cases} |l' - l| = |-k| \\ |l - l'| = |k| \end{cases} \Rightarrow |l' - l| = |l - l'| = k$$

***Proposition 1.27*** *$\mathcal{R}_b$ is a symmetric relation.*

**Proof:** If $\mathcal{R}_b$ is symmetric, then it will verify that $\forall l, l' \in \mathcal{U}, (l, l') \in \mathcal{R}_b \Rightarrow (l', l) \in \mathcal{R}_b$. The demonstration is trivial because

$$|l - l'| = |-(l - l')| = |l' - l| = k$$

consequently $\mathcal{R}_b$ is symmetric. □

***Proposition 1.28*** *Let $l, l', l''$ be three ports such that $l, l', l'' \in \mathcal{U}$ and $(l, l') \in \mathcal{R}_b$. It is verified that $(l, l'') \in \mathcal{R}_b$ if and only if $l' = l''$.*

**Proof:** If $|l - l'| = k$ then

$$|l - l'| = k \Rightarrow \begin{cases} l - l' = k & \Rightarrow \quad l' = l - k \\ & \text{or} \\ l - l' = -k & \Rightarrow \quad l' = l + k \end{cases}$$

Although there are two possible values for $l'$, only one is a valid port. It is clear that if $0 \leq l < k$, then $l' = l - k < 0$ is not a valid port because it is out of bounds of valid ports; otherwise, if $k \leq l < 2k$, then $l' = l + k \geq 2k$ will be an invalid port. In any case, there will be one port $l' \in \mathcal{U}$ such that $(l, l') \in \mathcal{R}_b$.

On the other hand, if $l' = l''$, then it will be trivial to demonstrate that $(l, l'') \in \mathcal{R}_b$. □

The Propositions 1.29 and 1.30 that are enunciated and demonstrated bellow, are not only applicable under complement traffic, but under any traffic pattern, since they are only derived from the network topology and routing algorithm.

***Proposition 1.29*** *Let $\mathcal{S}_b$ be the set of configurations that minimize the use of the internal link in a $T$-switch considering the backward connections, then*

$$\mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

*and there are no backward connections that go across the internal link.*

**Proof:** Let $\mathcal{C}$ be a configuration such that $\mathcal{C} \in \mathcal{S}_b$. According to the definition of $\mathcal{S}_b$, for a port $l \in \mathcal{C}$ there exists another port $l' \in \mathcal{C}$ such that $(l, l') \in \mathcal{R}_b$. This in turn implies that there exists a port $l'' \in \mathcal{C}$ such that $(l', l'') \in \mathcal{R}_b$; and another port $l''' \in \mathcal{C}$ such that $(l'', l''') \in \mathcal{R}_b$; and so on.

Nevertheless, according to Propositions 1.27 and 1.28, the inclusion of port $l$ in $\mathcal{C}$ implies the inclusion of a unique port $l'$ satisfying $(l, l') \in \mathcal{R}_b$.

Moreover, since $card(\mathcal{C}) = k$ (Proposition 1.2), $k$ is even (initial hypothesis); and all the $k$ ports are grouped in pairs $(l', l'')$ which verify the binary relation $\mathcal{R}_b$, there would be always configurations belonging to $\mathcal{S}_b$. Hence, $\mathcal{S}_b \neq \emptyset$.

On the other hand, there are only connections between two ports $l$ and $l'$, if $(l, l') \in \mathcal{R}_b$. If $\forall l \in \mathcal{C}$, there exists a port $l'$ such that $(l, l') \in \mathcal{R}_b$, then all the connections are established between port belonging to the same internal switch. Consequently, there exist no connections that use the internal link; and the members of $\mathcal{S}_b$ are optimal configurations. □

***Proposition 1.30*** *If there exist backward connections for every pair of ports $l$ and $l'$, such that $(l, l') \in \mathcal{R}_b$, then the members belonging to $\mathcal{S}_b$ are the unique optimal configurations. Hence,*

$$\forall (l, l') \in \mathcal{R}_b,\ C_b(\langle s, o\rangle, l, l') > 0 \Rightarrow \forall \mathcal{C} \in \mathcal{V},\ \textit{if } \mathcal{C} \notin \mathcal{S}_b,\ \textit{then } \mathcal{C} \textit{ is not optimum}$$

**Proof:** Assuming the existence of backward connections for all the pairs of ports $l$ and $l'$ that verify $(l, l') \in \mathcal{R}_b$, let us suppose that there exists a configuration $\mathcal{C}'$ that minimizes the use of the internal link, and $\mathcal{C}' \notin \mathcal{S}_b$. If $\mathcal{C}' \notin \mathcal{S}_b$, then there will exist a port $l$ that is not related to $l'$.

$$\mathcal{C}' \notin \mathcal{S}_b \ \Rightarrow\ \exists l \in \mathcal{C}' \mid \forall l' \in \mathcal{C}',\ (l, l') \notin \mathcal{R}_b$$

If the configuration $\mathcal{C}'$ is used to build a $T$-switch (it should be reminded $\mathcal{C}'$ determines the ports connected to each internal switch) the port $l$ must accomplish the backward connections to a port that is not allocated inside the same internal switch, since all the ports establish backward connections. Consequently, those connections have to use the internal link, and the configuration $\mathcal{C}'$ does not minimize the use of the internal link, since there are configurations better than $\mathcal{C}'$ (because they do not go across the internal link), which belong to $\mathcal{S}_b$.

Therefore, the members of $\mathcal{S}_b$ are configurations that minimize the number of connections that use the internal link. □

Example 1.6 shows two optimal configurations for a $T$-switch of type $\pi a$ considering the backward connections.

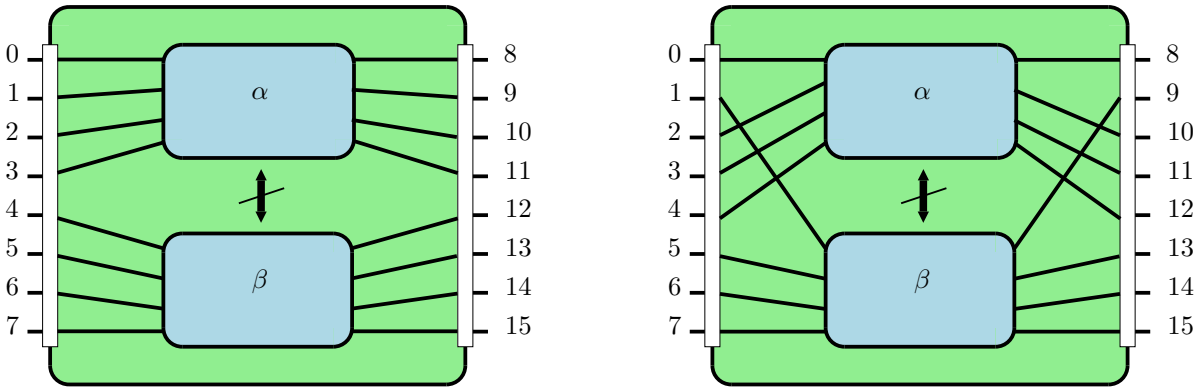
***Example 1.6*** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two configurations for a 8×8 $T$-switch*

$$\mathcal{T}_1 = \left\{\mathcal{C}_1^\alpha, \mathcal{C}_1^\beta \in \mathcal{V} \mid \mathcal{C}_1^\alpha = \{0,1,2,3,8,9,10,11\},\ \mathcal{C}_1^\beta = (\mathcal{C}_1^\alpha)^C = \{4,5,6,7,12,13,14,15\}\right\}$$
$$\mathcal{T}_2 = \left\{\mathcal{C}_2^\alpha, \mathcal{C}_2^\beta \in \mathcal{V} \mid \mathcal{C}_2^\alpha = \{0,2,3,4,8,10,11,12\},\ \mathcal{C}_2^\beta = (\mathcal{C}_2^\alpha)^C = \{1,5,6,7,9,13,14,15\}\right\}$$

*Both $\mathcal{T}_1$ and $\mathcal{T}_2$ are optimal configurations for the backward connections in a $T$-switch because they verify*

$$\mathcal{C}_1^\alpha,\ \mathcal{C}_1^\beta,\ \mathcal{C}_2^\alpha,\ \mathcal{C}_2^\beta \in \mathcal{S}_b$$

*The Figures 11a and 11b illustrate the connections for the configurations $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively.*




(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$

Figure 11: Optimal switch configurations for a 8×8 $T$-switch considering backward connections.

***Proposition 1.31*** *Let $\mathcal{S}_b^{\pi a}$ be the set of configurations that minimize the use of the internal link in a $T$-switch of type $\pi a$, considering the backward connections, then*

$$\mathcal{S}_b^{\pi a} = \mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_b\}$$

*and there are no backward connections that go across the internal link.*

**Proof:** From Proposition 1.29, members of $\mathcal{S}_b$ are optimal configurations considering backward connections. Moreover, in a $T$-switch of type $\pi a$, where there exist backward connections between each pair of ports $l, l'$ such that $(l,l') \in \mathcal{R}_b$, the configurations belonging to $\mathcal{S}_b$ are the unique optimal configurations by the Proposition 1.30. □

#### 6.3.1.3 Optimal switch configuration to all the connections

To obtain the set of configurations that minimize the use of the internal link, considering at the same time the forward and backward connections, we apply the set intersection of $\mathcal{S}_f^{\pi a}$ and $\mathcal{S}_b^{\pi a}$. Hence,

$$\begin{aligned}\mathcal{S}^{\pi a} &= \mathcal{S}_f^{\pi a} \cap \mathcal{S}_b^{\pi a} = \\ &= \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_f\} \cap \{\mathcal{C}' \in \mathcal{V} \mid \forall l \in \mathcal{C}',\ \exists l' \in \mathcal{C}' \mid (l,l') \in \mathcal{R}_b\} = \\ &= \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l', l'' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_f, (l,l'') \in \mathcal{R}_b\}\end{aligned}$$

This is true when $k$ is a multiple of 4. Otherwise, the resulting set would be empty ($\mathcal{S}^{\pi a} = \emptyset$). For this reason, we have defined two situations: (a) $k$ is a multiple of 4, (b) $k$ is not a multiple of 4.

To distinguish between both situations, the set of optimal configurations that minimize the use of the internal link would be represented by $\mathcal{S}_4^{\pi a}$ in the situation (a); and by $\mathcal{S}_{\not 4}^{\pi a}$ in the situation (b).

***Proposition 1.32*** *Let $l_0, l_1, l_2$ be three ports such that $l_0, l_1, l_2 \in \mathcal{U}$ and $(l_0, l_1) \in \mathcal{R}_f$ and $(l_0, l_2) \in \mathcal{R}_b$. Then there exists a port $l_3 \in \mathcal{U}$ such that $(l_2, l_3) \in \mathcal{R}_f$ and $(l_1, l_3) \in \mathcal{R}_b$.*

**Proof:** If $(l_0, l_1) \in \mathcal{R}_f$ and $(l_0, l_2) \in \mathcal{R}_b$, then by the definitions and propositions of $\mathcal{R}_f$ and $\mathcal{R}_b$ it is known that

$$l_0 + l_1 = 2k - 1 \tag{12}$$
$$|l_0 - l_2| = k \tag{13}$$

Moreover, there exist $l_3, l_4 \in \mathcal{U}$ such that $(l_1, l_3) \in \mathcal{R}_b$ and $(l_2, l_4) \in \mathcal{R}_f$. Therefore,

$$|l_1 - l_3| = k \tag{14}$$
$$l_2 + l_4 = 2k - 1 \tag{15}$$

To avoid problems with the absolute values of the above expressions, we have considered two separate cases:

A. $0 \leq l_0 < k$.

If $k \leq l_1, l_2 < 2k$, then $0 \leq l_3, l_4 < k$. So it is verified that $l_0 < l_2$ and $l_3 < l_1$, and the expressions 13 and 14 can be substituted for

$$l_0 - l_2 = -k$$
$$l_1 - l_3 = k$$

obtaining

$$l_2 = l_0 + k$$
$$l_1 = l_3 + k$$

Also, replacing the value of $l_1$ in the expression 12

$$l_0 + l_3 + k = 2k - 1 - l_0$$
$$l_3 = k - 1 - l_0 \tag{16}$$

and the value of $l_2$ in the expression 15

$$l_0 + k + l_4 = 2k - 1$$
$$l_4 = k - 1 - l_0 \tag{17}$$

By the expressions 16 and 17, it is deduced that $l_3 = l_4$.

B. $k \leq l_0 < 2k$.

In this case, if $0 \leq l_1, l_2 < k$, then $k \leq l_3, l_4 < 2k$. It is verified $l_2 < l_0$ and $l_1 < l_3$, and the expressions 13 and 14 can be substituted for

$$l_0 - l_2 = k$$
$$l_1 - l_3 = -k$$

obtaining

$$l_2 = l_0 - k \tag{18}$$
$$l_1 = l_3 - k \tag{19}$$

Also, replacing the value of $l_1$ in the expression 12

$$\begin{aligned} l_0 + l_3 - k &= 2k - 1 - l_0 \\ l_3 &= 3k - 1 - l_0 \end{aligned} \tag{20}$$

and the value of $l_2$ in the expression 15

$$\begin{aligned} l_0 - k + l_4 &= 2k - 1 \\ l_4 &= 3k - 1 - l_0 \end{aligned} \tag{21}$$

By expressions 20 and 21, it is deduced that $l_3 = l_4$.

Therefore, $l_3$ and $l_4$ are the same port, and so $(l_2, l_3) \in \mathcal{R}_f$ and $(l_1, l_3) \in \mathcal{R}_b$. □

According to Proposition 1.32, once the binary relations $\mathcal{R}_f$ and $\mathcal{R}_b$ are simultaneously considered, the switch ports are associated by groups of 4.

***Proposition 1.33*** *Let $\mathcal{S}_4^{\pi a}$ be the set of configurations that minimize the use of the internal link in a $T$-switch of type $\pi a$, considering all the connections and being $k$ a multiple of 4, then*

$$\mathcal{S}_4^{\pi a} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l', l'' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_f, (l, l'') \in \mathcal{R}_b\}$$

*and there exist no connections that use the internal link.*

**Proof:** Notice that $\mathcal{S}_4^{\pi a}$ is the intersection of the sets $\mathcal{S}_f^{\pi a}$ and $\mathcal{S}_b^{\pi a}$:

$$\mathcal{S}_4^{\pi a} = \mathcal{S}_f^{\pi a} \cap \mathcal{S}_b^{\pi a} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l', l'' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_f, (l, l'') \in \mathcal{R}_b\}$$

Let $\mathcal{C}$ a configuration such that $\mathcal{C} \in \mathcal{S}_4^{\pi a}$. According to definition of $\mathcal{S}_4^{\pi a}$, for the port $l \in \mathcal{C}$, there exist two ports $l', l'' \in \mathcal{C}$ such that $(l, l') \in \mathcal{R}_f$ and $(l, l'') \in \mathcal{R}_b$. As $l$ and $l'$ belong to $\mathcal{C}$, in turn, $l'$ and $l''$ are related to other two ports.

By the Propositions of $\mathcal{R}_f$ and $\mathcal{R}_b$, it is verified that $(l', l) \in \mathcal{R}_f$ and $(l'', l) \in \mathcal{R}_b$; and by Proposition 1.32 there exists a fourth port $l'''$ such that $(l', l''') \in \mathcal{R}_b$ and $(l'', l''') \in \mathcal{R}_f$. Consequently, if $\mathcal{C}$ verifies the definition of $\mathcal{S}_4^{\pi a}$, $l'''$ must be included in $\mathcal{C}$. In other words, to obtain a configuration that minimizes the number of connections that go across the internal link, the switch ports must be grouped (i.e.,, connected to the same internal switch) in fours groups.

Since $card(\mathcal{C}) = k$ and $k$ is a multiple of 4, then $\mathcal{S}_4^{\pi a} \neq \emptyset$ is verified.

For there to be connections between the ports $l$ and $l'$, it must be met $(l, l') \in \mathcal{R}_f$ or $(l, l') \in \mathcal{R}_b$. If for all the ports $l \in \mathcal{C}$, there exist $l', l'' \in \mathcal{C}$ such that $(l, l') \in \mathcal{R}_f$ and $(l, l'') \in \mathcal{R}_b$, all connections are established between ports belonging to the same internal switch. Therefore, no connections are established going across the internal link, and the configuration in $\mathcal{S}_4^{\pi a}$ are optimum.

Finally, we will prove by *reduction ad absurdum* that the members of $\mathcal{S}_4^{\pi a}$ are the unique optimal configurations. Let us suppose a configuration $\mathcal{C}'$ that minimizes the use of the internal link such that $\mathcal{C}' \notin \mathcal{S}_4^{\pi a}$. This means there exists a port $l$, which is not related to another port $l'$ by neither a forward nor backward connection.

$$\mathcal{C}' \notin \mathcal{S}_4^{\pi a} \ \Rightarrow\ \exists l \in \mathcal{C}' \mid \forall l', l'' \in \mathcal{C}', (l, l') \notin \mathcal{R}_f \ \textit{or}\ (l, l'') \notin \mathcal{R}_b$$

If $\mathcal{C}'$ is used to build a $T$-switch (note that $\mathcal{C}'$ determines the ports connected to each internal switch), then $l$ will participate in at least one of the connections (i.e.,, $\mathcal{R}_f$ or $\mathcal{R}_b$) with a port, which is not allocated at the same internal switch. Consequently, those connections must use the internal

link and $\mathcal{C}'$ does not minimize the use of such link, because there exist better configurations, which belong to $\mathcal{S}_4^{\pi a}$ and do not use the internal link.

Therefore, the configurations belonging to $\mathcal{S}_4^{\pi a}$ minimize the number of connections using the internal link if $k$ is a multiple of 4. □

Example 1.7 shows an optimal configuration for $T$-switch considering forward and backward connections at the same time, and being $k$ a multiple of 4.

***Example 1.7*** *Let $\mathcal{T}$ be a configuration of a 8×8 $T$-switch*

$$\mathcal{T} = \Big\{\mathcal{C}^\alpha, \mathcal{C}^\beta \in \mathcal{V} \mid \mathcal{C}^\alpha = \{0,1,6,7,8,9,14,15\},\ \mathcal{C}^\beta = (\mathcal{C}^\alpha)^C = \{2,3,4,5,10,11,12,13\}\Big\}$$

*$\mathcal{T}$ is an optimal configuration for a $T$-switch of type $\pi a$, because it is verified that*

$$\mathcal{C}^\alpha,\ \mathcal{C}^\beta \in \mathcal{S}_4^{\pi a}$$

*and $k$ is a multiple of 4.*

*In Figure 12, the $\mathcal{T}$ configuration connections are shown.*

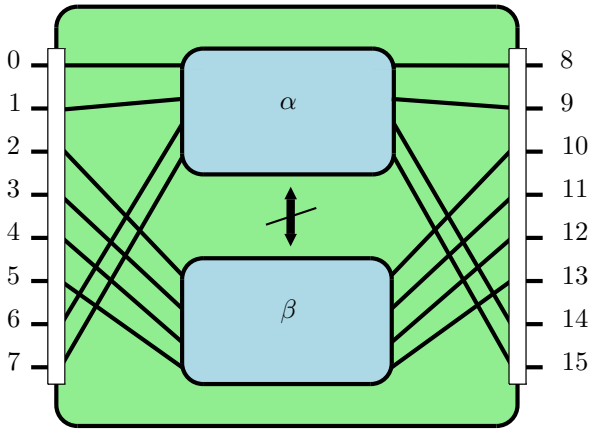



Figure 12: Optimal switch configuration for a 8×8 $T$-switch of type $\pi a$.

***Proposition 1.34*** *Let $\mathcal{S}_4^{\pi a}$ be the set of configurations that minimize the use of the internal link in a $T$-switch of type $\pi a$, considering all the connections, and $k$ is even, but not a multiple of 4, then*

$$\mathcal{S}_4^{\pi a} = \{\mathcal{C} \in \mathcal{V} \mid \exists j, j' \in \mathcal{C} \mid (j,j') \in \mathcal{R}_f \text{ or } (j,j') \in \mathcal{R}_b, \\ \forall l \in (\mathcal{C} - \{j,j'\})\ \ \exists l', l'' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_f, (l,l'') \in \mathcal{R}_b\}$$

*and there exist two connections using the internal link.*

**Proof:** If $k$ is even, but is not a multiple of 4, then $k = 4i+2,\ i \in \mathbb{N}^*$. So, a $k$×$k$ $T$-switch has $8i+4$ ports and the internal switches (i.e.,, $\alpha$ and $\beta$) have $4i+2$ ports each one to connect to the $T$-switch ports. That is to say,

$$card(\mathcal{U}) = 8i + 4$$
$$card(\mathcal{C}) = 4i + 2$$

Let us assume that the $(4i+2)\times(4i+2)$ $T$-switch ports are grouped by two groups of $4i\times4i$ and $2\times2$ ports, respectively. In other words, two internal switches: one of $4i\times4i$ ports and another of $2\times2$ ports. It is possible to obtain optimal configurations for the $4i\times4i$ $T$-switch by means of $\mathcal{S}_4^{\pi a}$, and it is easy to deduce the best configurations for the $2\times2$ $T$-switch because of its low number of ports.

To do this, the set $\mathcal{U}$ of ports is splitted up into $\mathcal{U}_{4i}$ and $\mathcal{U}_2$, which represent the $4i\times4i$ and $2\times2$ $T$-switches, respectively. To avoid the connections between ports belonging to $\mathcal{U}_2$ and $\mathcal{U}_{4i}$, the following sets are defined:

$$\mathcal{U}_2 = \{l_0, l_1, l_2, l_3 \in \mathcal{U} \mid (l_0, l_3),\ (l_2, l_1) \in \mathcal{R}_f, (l_0, l_2),\ (l_1, l_3) \in \mathcal{R}_b\} \tag{22}$$

$$\mathcal{U}_{4i} = \mathcal{U} - \mathcal{U}_2 \tag{23}$$

Then, optimal configurations are obtained as follows:

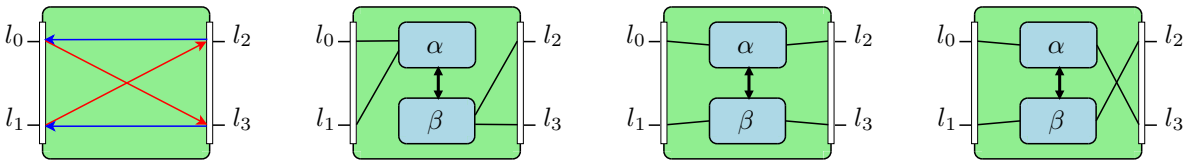
The set of optimal configurations for a $4i \times 4i$ $T$-switch is obtained by applying the Proposition 1.33:

$$\mathcal{S}_{4i}^{\pi a} = \{\mathcal{C}' \subset \mathcal{U}_{4i} \mid card(\mathcal{C}') = 4i, \forall l \in \mathcal{C}', \exists l', l'' \in \mathcal{C}' \mid (l, l') \in \mathcal{R}_f, (l, l'') \in \mathcal{R}_b\}$$

and there exist no connections that use the internal link.

Taking the $2\times2$ $T$-switch into consideration, Figure 13(a) depicts the established connections between the ports belonging to $\mathcal{U}_2$. There exist three [3] possible configurations for the $T$-switch, which are shown in the Figure 13. They are the following

$$\mathcal{T}_1 = \{\{l_0, l_1\}, \{l_2, l_3\}\}$$
$$\mathcal{T}_2 = \{\{l_0, l_2\}, \{l_1, l_3\}\}$$
$$\mathcal{T}_3 = \{\{l_0, l_3\}, \{l_1, l_2\}\}$$




(a) Internal connections. (b) $\mathcal{T}_1$ configuration. (c) $\mathcal{T}_2$ configuration. (d) $\mathcal{T}_3$ configuration.

Figure 13: Internal connections in a $2\times2$ $T$-switch and possible configurations.

Defined the relations between ports belonging to $\mathcal{U}_2$, we have noticed that all the connections in $\mathcal{T}_1$ use the internal link, being 4 the total quantity. In $\mathcal{T}_2$, the ports establishing backward connections are connected to the same internal switch. Meanwhile, the ports establishing forward connections are in $\mathcal{T}_3$, which are connected to the same internal switch.

Therefore, the forward and backward connections in $\mathcal{T}_2$ and $\mathcal{T}_3$ use the internal link, and there are 2 connections in both cases. That is to say, to obtain an optimal configuration, the ports that establish either forward or backward connections have to be allocated to the same internal switch. In such a way, an optimal configuration for those 4 ports is

$$\mathcal{S}_2^{\pi a} = \{\{j, j'\}\ j, j' \in \mathcal{U}_2 \mid (j, j') \in \mathcal{R}_b \text{ or } (j, j') \in \mathcal{R}_f\}$$

and there exist two connections that use the internal link.

[3] There are only three different configurations based on behavior. It is clear that there are other three different configurations from those shown, but with the same performance.

If the two calculated sets are joined, then

$$\begin{aligned}\mathcal{S}_{\not 4}^{\pi a} &= \mathcal{S}_{4i}^{\pi a} \cup \mathcal{S}_{2}^{\pi a} = \\ &= \{\{j,j'\}, j, j' \in \mathcal{U}_2 \mid (j,j') \in \mathcal{R}_b \text{ or } (j,j') \in \mathcal{R}_f\} \cup \\ &\quad \cup \{\mathcal{C}' \subset \mathcal{U}_{4i} \mid card(\mathcal{C}') = 4i, \forall l \in \mathcal{C}', \exists l', l'' \in \mathcal{C}' \mid (l,l') \in \mathcal{R}_f, (l,l'') \in \mathcal{R}_b\} \\ &= \{\mathcal{C}'' \subset (\mathcal{U}_{4i} \cup \mathcal{U}_2) = \mathcal{U} \mid \exists j, j' \in \mathcal{C}'' \mid ((j,j') \in \mathcal{R}_f \text{ or } (j,j') \in \mathcal{R}_b), \\ &\quad \forall l \in (\mathcal{C}'' - \{j,j'\})\ \exists l', l'' \in \mathcal{C}'' \mid (l,l') \in \mathcal{R}_f, (l,l'') \in \mathcal{R}_b\}\end{aligned}$$

Since $\mathcal{U}_{4i} = \mathcal{U} - \mathcal{U}_2$, the sets $\mathcal{U}_2$ and $\mathcal{U}_{4i}$ are disjoint, so the cardinal of the union of both sets is $4i+2$. Since, $\mathcal{C}'' \subset \mathcal{U}$ and $card(\mathcal{C}'') = 4i+2$, it is verified that $\mathcal{C}'' \in \mathcal{V}$. Hence,

$$\mathcal{S}_{\not 4}^{\pi a} = \{\mathcal{C} \in \mathcal{V} \mid \exists j, j' \in \mathcal{C} \mid ((j,j') \in \mathcal{R}_f \text{ or } (j,j') \in \mathcal{R}_b), \quad \forall l \in (\mathcal{C} - \{j,j'\})\ \exists l', l'' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_f, (l,l'') \in \mathcal{R}_b\}$$

and there exist two connections that use the internal link.

Moreover, the configurations in $\mathcal{S}_{\not 4}^{\pi a}$ are optimum. As it was seen in the proof of Proposition 1.33, the ports must be grouped in fours groups, in such a way that all connections would be established with port of the same internal switch. Consequently, if $k$ is not a multiple of 4, then it will be impossible to make groups for all the ports, so it will be no possible to obtain a configuration without having connections through the internal link.

By a similar reasoning, it would be impossible to setup a configuration with only one connection through the internal link. The explanation to this fact comes of the existence of connections between the four ports that are related (Figure 13(a)). It should be noticed that it is not possible to separate the ports, in such a way, there would be only one connection using the internal link.

Therefore, the configurations of $\mathcal{S}_{\not 4}^{\pi a}$ are optimum, since there is no other configuration with fewer connections through the internal link than thos belonging to $\mathcal{S}_{\not 4}^{\pi a}$. □

In the Example 1.8, the optimal configuration for $T$-switches, considering both forward and backward connections and being $k$ a non–multiple of 4, is showed.

***Example 1.8*** *Let us assume a* $6 \times 6$ *T-switch. There exist three 4–port groups, which establish connections between them. Specifically, the ports* 0, 5, 6, 11 *form a group; the ports* 1, 4, 7, 10 *form another; finally, the ports* 2, 3, 8, 9 *form the last group. Additionally, let* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *be two configurations of the assumed switch,*

$$\mathcal{T}_1 = \left\{\mathcal{C}_1^{\alpha}, \mathcal{C}_1^{\beta} \in \mathcal{V} \mid \mathcal{C}_1^{\alpha} = \{0,1,5,6,7,11\},\ \mathcal{C}_1^{\beta} = (\mathcal{C}_1^{\alpha})^C = \{2.3,4,8,9,10\}\right\}$$
$$\mathcal{T}_2 = \left\{\mathcal{C}_2^{\alpha}, \mathcal{C}_2^{\beta} \in \mathcal{V} \mid \mathcal{C}_2^{\alpha} = \{0,1,5,6,10,11\},\ \mathcal{C}_2^{\beta} = (\mathcal{C}_2^{\alpha})^C = \{2,3,4,7,8,9\}\right\}$$

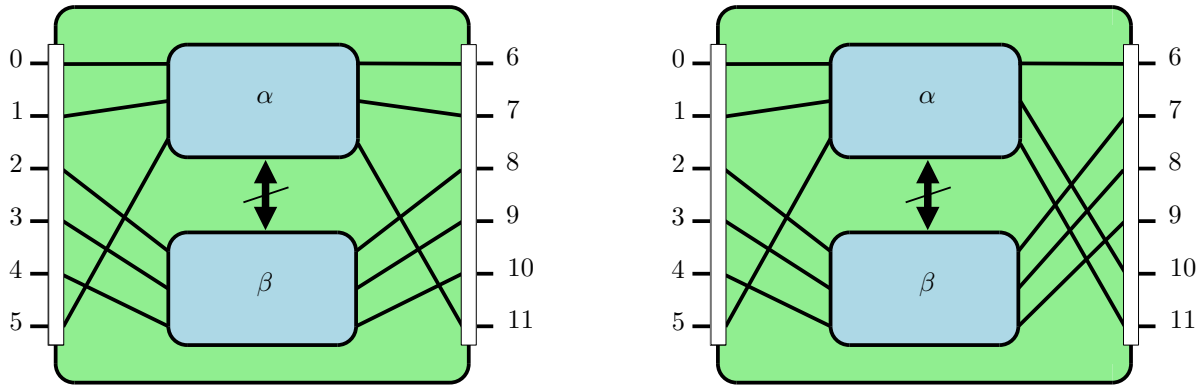
*Both* $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are optimum for the T-switch of type* $\pi a$ *because it is verified:*

$$\mathcal{C}_1^{\alpha},\ \mathcal{C}_1^{\beta},\ \mathcal{C}_2^{\alpha},\ \mathcal{C}_2^{\beta} \in \mathcal{S}_{\not 4}^{\pi a}$$

*and* $k$ *is not a multiple of* 4.

*In this Example it is noticeable in both cases the fours groups (i.e.,* 1, 4, 7, 10 *ports) is split into the two internal switches. Figure 14 illustrates the connections of the configurations* $\mathcal{T}_1$ *and* $\mathcal{T}_2$.

□




(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$

Figure 14: Optimal switch configuration for a 6×6 $T$-switch of type $\pi a$.

### 6.3.2 Type $\pi b$ configuration of switch

Since turnaround connections occur exclusively in the last–stage switches, only the half number of switch ports is used, i.e., $k$ ports. As the number of internal ports is also $k$, one obvius and optimal configuration for these switches is that using all the ports of one internal switch to connect to switches in the $n-2$ stage, that is, both $(p=k, q=0)$ and $(p=0, q=k)$ alternatives offer the best results: the internal link connecting the $\alpha$ and $\beta$ switches is never used.

Those alternatives are valid for the last–stage switches in a BMIN, but in special cases, other configurations would provide a similar behaviour, while reducing the number of different configurations (i.e., the number of types of switch).

To be more comprehensive, it is now introduced an analysis like the ones that have been done in the previous sections. In a $T$-switch of type $\pi b$, there only exist turnaround connections between the ports $l, l'$, $0 \leq l, l' < k$ that verify $l' = \bar{l}$. Below, the binary relation $\mathcal{R}_t$ and its propositions are defined, from which the optimal configuration set for the $T$-switch of type $\pi b$ is obtained.

***Definition 1.24*** *The binary relation $\mathcal{R}_t$ on a set $\mathcal{U}$ is defined as follows*

$$\mathcal{R}_t = \{(l, l') \in \mathcal{U}^2 \mid l + l' = k - 1\}$$

Note that the expression $l + l' = k - 1$ is derived from $l' = \bar{l}$

$$\begin{aligned} l' &= \bar{l} \\ l' &= k - l - 1 \\ l + l' &= k - 1 \end{aligned}$$

***Proposition 1.35*** *Let $\mathcal{S}^{\pi b}$ be the set of configurations that minimize the use of the internal link in a $T$-switch of type $\pi b$. Hence,*

$$\mathcal{S}^{\pi b} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C}, 0 \leq l < k,\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_t\}$$

*and there exist no connections that use the internal link.*

**Proof:** The demonstration consists in proving that $\mathcal{S}^{\pi b}$ is not empty, but it is trivial. For instance, $\mathcal{C} = \mathcal{B}$ belongs to $\mathcal{S}^{\pi b}$, as it includes every backward port, all turnaround connections belong to $\mathcal{C}$,

and it is verified that for every port $l$, there exists another port $l'$ such that $(l, l') \in \mathcal{R}_t$. Therefore, $\mathcal{S}^{\pi b} \neq \emptyset$.

On the other hand, there exist connections between the ports $l$ and $l'$ if $(l, l') \in \mathcal{R}_t$. If $\forall l \in \mathcal{C}$, there exists a port $l'$ such that $(l, l') \in \mathcal{R}_t$, then all the connections are established between ports that belong to same internal switch. Consequently, there are no connections that use the internal link, and the configurations in $\mathcal{R}_t$ are optimum.

Finally, we demonstrate by reductio ad absurdum that the configurations in $\mathcal{S}^{\pi b}$ are the unique optimal configurations. Let us consider a configuration, $\mathcal{C}'$, that minimizes the use of internal link and $\mathcal{C}' \notin \mathcal{S}^{\pi b}$. That means there exists a port $l \in \mathcal{C}'$ which is not related to another port $l' \in \mathcal{C}'$.

$$\mathcal{C}' \notin \mathcal{S}_t^{\pi b} \;\Rightarrow\; \exists l \in \mathcal{C}', 0 \leq l < k, \mid \forall l' \in \mathcal{C}' \;\; (l, l') \notin \mathcal{R}_t$$

In this case, if the configuration $\mathcal{C}'$ is used to setup a $T$-switch (remind that $\mathcal{C}'$ determines the ports, which are connected to each internal switch), the port $l$ will be used to establish a turnaround connection with another port belonging to a different internal switch. Therefore, those connections have to go across the internal link and $\mathcal{C}'$ would not minimize the use of the internal link, since there would be better configurations such the ones belonging to $\mathcal{S}^{\pi b}$, which really do not need to use the internal link.

Consequently, the configurations in $\mathcal{S}^{\pi b}$ minimize the number of connections using the internal link. □

***Example 1.9*** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two configurations for a 8×8 $T$-switch.*

$$\mathcal{T}_1 = \left\{ \mathcal{C}_1^{\alpha}, \mathcal{C}_1^{\beta} \in \mathcal{V} \mid \{0,1,6,7\} \subset \mathcal{C}_1^{\alpha},\ \{2,3,4,5\} \subset \mathcal{C}_1^{\beta} \right\}$$
$$\mathcal{T}_2 = \left\{ \mathcal{C}_2^{\alpha}, \mathcal{C}_2^{\beta} \in \mathcal{V} \mid \{0,7\} \subset \mathcal{C}_1^{\alpha},\ \{1,2,3,4,5,6\} \subset \mathcal{C}_2^{\beta} \right\}$$

*Both $\mathcal{T}_1$ and $\mathcal{T}_2$ are optimum for backward connections in a $T$-switch of type $\pi b$, because it is verified*

$$\mathcal{C}_1^{\alpha},\ \mathcal{C}_1^{\beta},\ \mathcal{C}_2^{\alpha},\ \mathcal{C}_2^{\beta} \in \mathcal{S}^{\pi b}$$

*Figure 15 depicts the connections for the configurations $\mathcal{T}_1$ and $\mathcal{T}_2$. Both figure and example obviate the forward ports, as their configuration is irrelevant for these cases.*

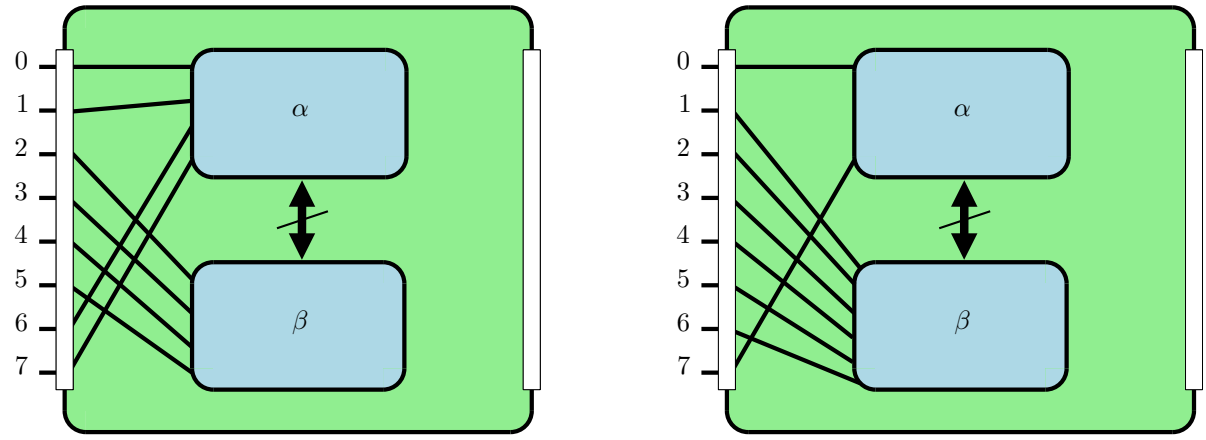
### 6.3.3 Type $\pi$ configuration of switch

Under complement traffic pattern and when $k$ is a multiple of 4, it is possible to find optimal configurations for $T$-switches considering forward, turnaround and backward connections.

Indeed, if $k$ is a multiple of 4, it will be possible to obtain optimal configurations for a $T$-switch of type $\pi b$ from $\mathcal{S}_4^{\pi a}$. As discussed below, any configuration $\mathcal{C}$ belonging to $\mathcal{S}_4^{\pi a}$ also belongs to $\mathcal{S}^{\pi b}$. However, the reverse statement is not always true because if the configuration $\mathcal{C}'$ is that which has all the backward ports belonging to $\mathcal{F}$ and verifying $\mathcal{C}' \in \mathcal{S}^{\pi b}$, but $\mathcal{C}' \notin \mathcal{S}_4^{\pi a}$. Hence,

$$\begin{aligned} \mathcal{C} \in \mathcal{S}_4^{\pi a} &\;\Rightarrow\; \mathcal{C} \in \mathcal{S}^{\pi b} \\ \mathcal{C} \in \mathcal{S}^{\pi b} &\;\not\Rightarrow\; \mathcal{C} \in \mathcal{S}_4^{\pi a} \end{aligned}$$

Consequently, $\mathcal{S}_4^{\pi a} \subset \mathcal{S}^{\pi b}$ is verified, since $\mathcal{S}_4^{\pi a}$ is a subset of $\mathcal{S}^{\pi b}$, the configurations in $\mathcal{S}_4^{\pi a}$ are also optimum for a $T$-switch of type $\pi b$.




(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$

Figure 15: Optimal switch configurations for a 8×8 $T$-switch of type $\pi b$.

***Proposition 1.36*** *Let $l_0, l_1, l_2$ be ports such that $l_0, l_1, l_2 \in \mathcal{U}$, $0 \leq l_0, l_1, l\,2 < k$, and verify $(l_0, l_1) \in \mathcal{R}_f$ and $(l_0, l_2) \in \mathcal{R}_b$. There exists a port $l_3 \in \mathcal{U}$ such that $(l_2, l_3) \in \mathcal{R}_f$, $(l_1, l_3) \in \mathcal{R}_b$ and $(l_0, l_3) \in \mathcal{R}_t$.*

**Proof:** Proposition 1.32 proved there uniquely exists one port $l_3$ such that $(l_2, l_3) \in \mathcal{R}_f$ and $(l_1, l_3) \in \mathcal{R}_b$. Furthermore, it was obtained that:

$$l_3 = k - 1 - l_0, \text{ if } 0 \leq l < k \text{ (expression 16)}$$
$$l_3 = 3k - 1 - l_0, \text{ if } k \leq l < 2k \text{ (expression 20)}$$

For the range of $l$ in which we are focused on, $0 \leq l < k$, we have that $l_3 = k - 1 - l_0$. That is to say, $l_3 + l_0 = k - 1$. Therefore, $(l_0, l_3) \in \mathcal{R}_t$. □

***Proposition 1.37*** *The configurations of the $\mathcal{S}_4^{\pi a}$ set minimize the use of the internal link in a $T$-switch of type $\pi b$, if $k$ is a multiple of 4.*

**Proof:** The configurations of $\mathcal{S}_4^{\pi a}$ should be grouped in fours groups, because if for a port $l$ there are other two ports $l', l''$ such that $(l, l') \in \mathcal{R}_f$ and $(l, l'') \in \mathcal{R}_b$, it will be compulsory the inclusion of a fourth port $l'''$ in that configuration such that $(l', l''') \in \mathcal{R}_b$ and $(l'', l''') \in \mathcal{R}_f$. Moreover, by the Proposition 1.36 it is known that if $0 \leq l < k$, then it verified that $(l, l''') \in \mathcal{R}_t$.

Consequently, the configurations $\mathcal{C} \in \mathcal{S}_4^{\pi a}$ also verify that $\forall l \in \mathcal{C},\ 0 \leq l < k,\ \exists l' \mid (l, l') \in \mathcal{R}_t$, and they minimize the number of connections that use the internal link when $k$ is multiple of 4. □

Thus, from the set $\mathcal{S}_4^{\pi a}$, it is possible to obtain one optimal configuration for the $T$-switches in the network if $k$ is a multiple of 4 and the complement pattern traffic is considered. In this case, the configuration shown in Example 1.7 is also optimum for the $T$-switches located on the last stage of the network.

It should be noticed that it is impossible to establish a unique and optimum configuration for all the switches in the network when $k$ is not a multiple of 4, because $\mathcal{S}_4^{\pi a}$ implies the existence of a port that participates in the backward connection, or in the forward connection, at the same switch, and therefore, there is not the fourth port that is needed to verify the conditions of $\mathcal{R}_t$.

# 7 Applying the Methodology for Perfect-Shuffle Traffic

This section performs the search of optimal $T$-switch configuration using the same methodology as in the complement case. On this occasion, the network is evaluated under perfect–shuffle traffic pattern.

## 7.1 Network Paths Analysis

The generated paths with the perfect–shuffle traffic pattern are studied in this section. Figure 16 shows graphically the paths generated by this traffic pattern in a 2–ary 3–tree. There are so many different paths as end nodes, and the internal switch connections are determined by the specific routing algorithm. As it is shown in the case of complement traffic, there is also a noticeable symmetry in the paths. The paths still reach the switches of the last stage, but some of them turning around before the last stage.

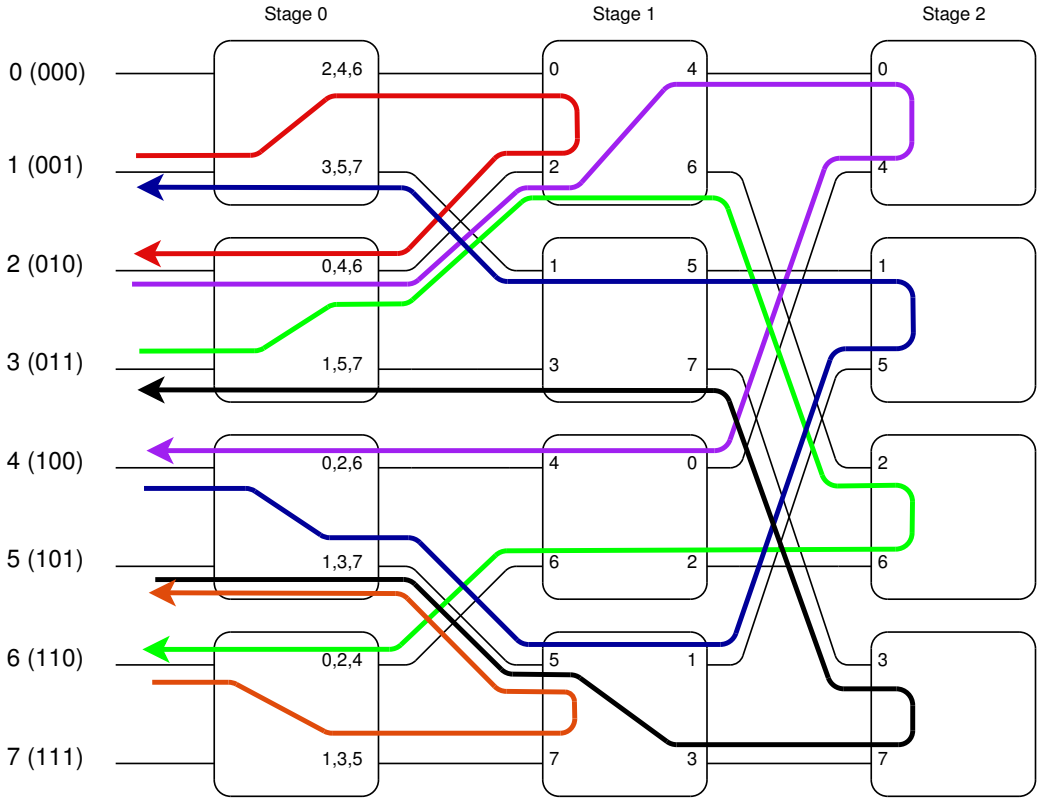



Figure 16: Generated paths under perfect–shuffle traffic pattern in a 2–ary 3–tree network.

To avoid continually repeating the same premises in every definition, they are now indicated and then omitted from the propositions. In this way, the statements remain more clear and simple. Specifically, the premises are as follows:

- The network topology is a $T$–BMIN $k$–ary $n$–tree with $N$ end nodes.
- The network load is generated by the perfect–shuffle traffic pattern.
- The routing algorithm is that defined in Section A.7.1. It is deterministic in the ascending and descending phase (self–routing).

### 7.1.1 Ascending phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o\rangle$ in the ascending phase, $0 \leq s < n$, are described below. Since there are differences between what happens in every switch, the propositions will be organized by the stage.

***Proposition 1.38*** *Given the ports $l$ and $l'$ of the switch $\langle 0, o\rangle$, $0 \leq l < k$ and $k \leq l' < 2k$, at the most there is one path passing through $\langle 0, o\rangle$ in the ascending phase by using $l$ and $l'$, and only if $l' - k = o_{n-2}$ and $\exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1}$; or $l' - k = o_{n-2}$ and $\nexists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1}$ and $l \neq o_{n-2}$. Hence,*

$$C_f(\langle 0, o\rangle, l, l') = \begin{cases} 1, & \text{if } l' - k = o_{n-2} \text{ and } \exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1} \\ 1, & \text{if } l' - k = o_{n-2} \text{ and } \nexists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1} \text{ and } l \neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path passes through the switch $\langle 0, o\rangle$ in the ascending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle 0, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_f^R(\langle 0, o\rangle, l')$, and the identifier $h'$ is obtained by applying the perfect–shuffle function to $h$. Therefore,

$$C_f(\langle 0, o\rangle, l, l') = \text{card}((N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_f^R(\langle 0, o\rangle, l'))$$

$$(N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_f^R(\langle 0, o\rangle, l') = (\{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1] \text{ and } h_0 = l\ \})^\sigma \cap \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [1, n-1] \mid h_i \neq o_{i-1} \text{ and } h_0 = l' - k\ \}$$

Then applying the perfect–shuffle function to the elements of the source set $N_b^t(\langle 0, o\rangle, l)$, we obtain

$$\begin{aligned} &(N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_f^R(\langle 0, o\rangle, l') = \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-2}\ \forall i \in [2, n-1],\ \ h_1 = l, h_0 = o_{n-2}\ \} \cap \\ &\quad\quad \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [1, n-1]\ \ h_i \neq o_{i-1} \text{ and } h_0 = l' - k\ \} \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-2}\ \forall i \in [2, n-1],\ h_1 = l,\ \exists i \in [1, n-1] \mid h_i \neq o_{i-1}, \\ &\quad\quad h_0 = o_{n-2} = l' - k\ \} \\ &\quad = \{\ (o_{n-3}, \ldots, o_0, l, o_{n-2}) \text{ if } o_{n-2} = l' - k \text{ and } \exists i \in [1, n-1] \mid o_{i-2} \neq o_{i-1}\ \} \end{aligned}$$

that is

$$\begin{aligned} &(N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_f^R(\langle 0, o\rangle, l') = \\ &\qquad \{\ (o_{n-3}, \ldots, o_0, l, o_{n-2}) \text{ if } o_{n-2} = l' - k \text{ and } (\exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1} \text{ ó } l \neq o_{n-2})\ \} \end{aligned}$$

As all the digits of every node identifier are fixed, at the most there is one path in the ascending phase that passes through the switch by using the ports $l$ and $l'$. Moreover, $l' - k = o_{n-2}$ means that in the switches of the first stage, $s = 0$, the path always has the output port $l' = o_{n-2} + k$ in the ascending phase apart from the input port $l$.

On the other hand, there must exist a $i \in [1, n-1]$ such as $h_i \neq o_{i-1}$. As all $h_i$ of the interval $[1, n-1]$ are known, they can be substituted in the last condition. So, either, there is a $i \in [2, n-1]$ such as $o_{i-2} \neq o_{i-1}$; or $l \neq o_0$. If the first one is satisfied then the second one will not be necessary, taking $l$ an arbitrary value of the interval $[0, k-1]$. Otherwise, $o_{n-2} = \ldots = o_1 = o_0$, and only if $l \neq o_0$ there would be a path between $l$ and $l'$, which is verified because $o_0 = o_{n-2}$ and therefore $l \neq o_{n-2}$. □

***Proposition 1.39*** *The number of paths in the ascending phase passing through a switch of the first stage is $k$ if $\exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1}$. Otherwise, it is $k-1$. Hence,*

$$C_f(\langle 0, o\rangle) = \begin{cases} k, & \text{if } \exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1} \\ k-1, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.38 we know that all the paths that pass through the switch in ascending phase use the output port $l' = o_{n-2} + k$. In case of existing a $i \in [2, n-1]$ such as $o_{i-2} \neq o_{i-1}$, all the $k$ values for the input port $l$ are valid and therefore there would be $k \times 1 = k$ paths passing through the switch by using the input port $l$. If $l \neq o_{n-2}$, only $k-1$ input ports are valid and therefore there would be $(k-1) \times 1 = k-1$ paths. □

***Proposition 1.40*** *Given the ports $l$ and $l'$ of the switch $\langle s, o\rangle$, where $1 \leq s < n-1$, $0 \leq l < k$ and $k \leq l' < 2k$, at the most there is one path passing through the switch in the ascending path by using $l$ and $l'$, and only if $o_0 = o_{n-2}$ and $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$; or $o_0 = o_{n-2}$ and $\nexists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$ and $l \neq o_s$. Hence,*

$$C_f(\langle s, o\rangle, l, l') = \begin{cases} 1, & \text{if } o_0 = o_{n-2} \text{ and } \exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \\ 1, & \text{if } o_0 = o_{n-2} \text{ and } \nexists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \text{ and } l \neq o_s \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path passes through the switch $\langle s, o\rangle$ in the ascending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle s, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_f^R(\langle s, o\rangle, l')$, and the identifier $h'$ is obtained by applying the perfect–shuffle function to $h$. Therefore,

$$C_f(\langle s, o\rangle, l, l') = card((N_b^t(\langle s, o\rangle, l))^\sigma \cap N_f^R(\langle s, o\rangle, l'))$$

$$\begin{aligned} &(N_b^t(\langle s, o\rangle, l))^\sigma \cap N_f^R(\langle s, o\rangle, l') = \\ &\quad = (\{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1] \text{ and } h_s = l\ \})^\sigma \cap \\ &\qquad \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1},\ \ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l' - k\ \} \end{aligned}$$

then the *perfect–shuffle* function is applied over the source set $N_b^t(\langle s, o\rangle, l)$

$$\begin{aligned} &(N_b^t(\langle s, o\rangle, l))^\sigma \cap N_f^R(\langle s, o\rangle, l') = \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-2}\ \forall i \in [s+2, n-1], h_{s+1} = l \text{ and } h_0 = o_{n-2}\ \} \cap \\ &\qquad \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1},\ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l' - k\ \} \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-2}\ \forall i \in [s+2, n-1],\ h_{s+1} = l,\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1}, \\ &\qquad h_s = l' - k,\ h_i = o_i\ \forall i \in [1, s-1] \text{ and } h_0 = o_{n-2} = o_0\ \} \\ &\quad = \{\ (o_{n-3}, \ldots, l, l' - k, o_{s-1}, \ldots, o_1, o_{n-2}) :\ o_{n-2} = o_0\ y\ \exists i \in [s+1, n-1] \mid h_{i-2} \neq o_{i-1}\ \} \end{aligned}$$

that is

$$\begin{aligned} (N_b^t(\langle s, o\rangle, l))^\sigma \cap N_f^R(\langle s, o\rangle, l') = \{\ &(o_{n-3}, \ldots, l, l' - k, o_{s-1}, \ldots, o_1, o_{n-2}) \text{ if } o_{n-2} = o_0 \text{ and} \\ &(\exists i \in [s+1, n-1] \mid o_{i-1} \neq o_{i-2} \text{ or } l \neq o_s)\ \} \end{aligned}$$

As all digits of every node identifier are fixed, at the most there is one path in the ascending phase that passes through the switch $\langle s, o\rangle$ by using $l$ and $l'$, based on two conditions:

a) $o_0 = o_{n-2}$.

b) $\exists i \in [s+1, n-1] \mid o_{i-2} \neq o_{i-1}$.

As all $h_i$ of the interval $[s-1, n-1]$ are known, they can be substituted in the last condition b). Then it is observed, either, there exist a $i \in [s+2, n-1]$ such as $o_{i-2} \neq o_{i-1}$, or $l \neq o_s$, that means, if the first one is satisfied then it will not necessary to accomplish the second one, so $l$ will take an arbitrary value of the interval $[0, k-1]$. Otherwise, if the first condition is not verified, $l \neq o_s$, there would be one path between $l$ and $l'$ uniquely if $l \neq o_s$. □

***Proposition 1.41*** *The number of paths in the ascending phase passing through the switch* $\langle s, o\rangle$, $1 \leq s < n-1$, *is* $k^2$ *if* $o_0 = o_{n-2}$ *and* $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$*; or* $k^2 - k$ *if* $o_0 = o_{n-2}$ *and* $\nexists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$. *Hence,*

$$C_f(\langle s, o\rangle) = \begin{cases} k^2, & \text{if } o_0 = o_{n-2} \text{ and } \exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \\ k^2 - k, & \text{if } o_0 = o_{n-2} \text{ and } \nexists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** When $o_0 = o_{n-2}$ we know there are paths in the ascending phase passing through the switch according to Proposition 1.40. Moreover, if there exists a $i \in [s+2, n-1]$ such as $o_{i-2} \neq o_{i-1}$, then both $l$ and $l'$ will take $k$ values, being $k \times k = k^2$ paths possible. If $l$ is not equal to $o_s$, then $l$ will take $k-1$ values so only $(k-1) \times k = k^2 - k$ paths would be possible. □

***Proposition 1.42*** *There is no path passing through, in the ascending phase, the switches of the last stage. Hence,*

$$C_f(\langle n-1, o\rangle) = C_f(\langle n-1, o\rangle, l, l') = 0$$

**Proof:** By the definition of the BMIN topology, there are no forward connections in the switches of the last stage. □

### 7.1.2 Turnaround phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o\rangle$ in the turnaround phase, $0 \leq s < n$, are described below.

***Proposition 1.43*** *There is no path passing through the switches of the first stage in the turnaround phase. Hence,*

$$T(\langle 0, o\rangle) = T(\langle 0, o\rangle, l, l') = 0$$

**Proof:** A path is turned around in the switch $\langle 0, o\rangle$ by using the input port $l$ and the output port $l'$ ($l \neq l'$) if the source node, $h$, belongs to the set $N_b^t(\langle 0, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle 0, o\rangle, l')$ where $l \neq l'$, and the identifier $h'$ is obtained by applying the perfect–shuffle function to $h$. Hence,

$$T(\langle 0, o\rangle) = \text{card}((N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_b^R(\langle 0, o\rangle, l'))$$

$$(N_b^t(\langle 0, o\rangle, l))^\sigma \cap N_b^R(\langle 0, o\rangle, l') = (\{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1] \text{ and } h_0 = l\ \})^\sigma \cap \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1] \text{ and } h_0 = l'\ \}$$

Then, the perfect–shuffle function is applied over the source set $N_b^t(\langle 0,o\rangle,l)$

$$
\begin{aligned}
&(N_b^t(\langle 0,o\rangle,l))^\sigma \cap N_b^R(\langle 0,o\rangle,l') = \\
&\quad = \{(h_{n-1},\ldots,h_0):\ h_i = o_{i-2}\ \forall i \in [2,n-1],\ h_1 = l,\ h_0 = o_{n-2}\ \} \cap \\
&\qquad \{(h_{n-1},\ldots,h_0):\ h_i = o_{i-1}\ \forall i \in [1,n-1] \text{ and } h_0 = l'\ \} \\
&\quad = \{(h_{n-1},\ldots,h_0):\ h_i = o_{i-2}\ \forall i \in [2,n-1],\ h_i = o_{i-1}\ \forall i \in [1,n-1],\ h_1 = l,\ h_0 = o_{n-2},\ h_0 = l'\ \} \\
&\quad = \{(o_{n-2},\ldots,o_2,l,l') \text{ if } o_{i-1} = o_{i-2}\ \forall i \in [2,n-1],\ l = o_0 \text{ and } l' = o_{n-2}\ \}
\end{aligned}
$$

According to the conditions that determine the digits of the set node identifiers we know that $l' = o_{n-2} = o_{n-3} = o_{n-4} = \ldots = o_1 = o_0 = l$, that is, $l = l'$. However, it is supposed that $l \neq l'$. Therefore there is no node belonging to the set, which verifies such conditions, in other words, the result set is empty. Therefore, it is demonstrated there is no path turning around in the switches of the first stage. □

***Proposition 1.44*** *Given the ports $l$ and $l'$ of the switch $\langle s,o\rangle$, where $1 \leq s < n-1$, $0 \leq l,l' < k$ and $l \neq l'$, at the most there is one path that turns around in the switch by using $l$ and $l'$ only if $o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1]$ and $o_0 = o_{n-2}$ and $l = o_s$. Hence,*

$$T(\langle s,o\rangle,l,l') = \begin{cases} 1, & \text{if } o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1] \text{ and } o_0 = o_{n-2} \text{ and } l = o_s \text{ and } l' \neq l \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path is turned around in the switch $\langle s,o\rangle$ by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle s,o\rangle,l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle s,o\rangle,l')$, where $l \neq l'$, and the identifier $h'$ is obtained by applying the perfect–shuffle function to $h$. Hence,

$$T(\langle s,o\rangle,l,l') = \text{card}((N_b^t(\langle s,o\rangle,l))^\sigma \cap N_b^R(\langle s,o\rangle,l'))$$

$$
\begin{aligned}
&(N_b^t(\langle s,o\rangle,l))^\sigma \cap N_b^R(\langle s,o\rangle,l') = \\
&\quad = (\{(h_{n-1},\ldots,h_0):\ h_i = o_{i-1}\ \forall i \in [s+1,n-1] \text{ and } h_s = l\ \})^\sigma \cap \\
&\qquad \{(h_{n-1},\ldots,h_0):\ h_i = o_{i-1}\ \forall i \in [s+1,n-1],\ h_i = o_i\ \forall i \in [0,s-1] \text{ and } h_s = l'\ \}
\end{aligned}
$$

Then the perfect–shuffle function is applied over the source set $N_b^t(\langle s,o\rangle,l)$

$$
\begin{aligned}
&(N_b^t(\langle s,o\rangle,l))^\sigma \cap N_b^R(\langle s,o\rangle,l') = \\
&\quad = \{\ (h_{n-1},\ldots,h_0):\ h_i = o_{i-2}\ \forall i \in [s+2,n-1],\ h_{s+1} = l \text{ and } h_0 = o_{n-2}\ \} \cap \\
&\qquad \{\ (h_{n-1},\ldots,h_0):\ h_i = o_{i-1}\ \forall i \in [s+1,n-1],\ h_i = o_i\ \forall i \in [0,s-1] \text{ and } h_s = l'\ \} \\
&\quad = \{\ (h_{n-1},\ldots,h_0):\ h_i = o_{i-1}\ \forall i \in [s+1,n-1],\ h_i = o_{i-2}\ \forall i \in [s+2,n-1], \\
&\qquad h_{s+1} = l,\ h_s = l',\ h_i = o_i\ \forall i \in [0,s-1] \text{ and } h_0 = o_{n-2}\ \} \\
&\quad = \{\ (o_{n-2},\ldots,o_s,l',o_{s-1},\ldots,o_0) \text{ if } o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1],\ o_0 = o_{n-2} \text{ and } l = o_s\}
\end{aligned}
$$

Two restrictions have to be verified so that there would exist paths passing through a switch in the turnaround phase. Firstly, $o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1]$, and secondly, $o_{n-2} = o_0$. The unique valid port is $l = o_s$, while there is no restriction on the output port. Therefore, every pairs ($l$,$l'$) has only one path under these restrictions. □

***Proposition 1.45*** *Only if $o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1]$ and $o_0 = o_{n-2}$, at the most there are $k-1$ paths passing through the switch $\langle s,o\rangle$, $1 \leq s < n-1$, in the turnaround phase. Hence,*

$$T(\langle s,o\rangle) = \begin{cases} k-1, & \text{if } o_{i-1} = o_{i-2}\ \forall i \in [s+2,n-1] \text{ and } o_0 = o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.44 all the paths that turn around the switch $\langle s, o\rangle$ use the same input port, $l = o_0$, and any of the remaining output ports. Since a port is never used as input and output port in one connection, then there are $k-1$ paths available to pass through the switch. □

***Proposition 1.46*** *Given the ports $l$ and $l'$ of the switch $\langle n-1, o\rangle$, where $0 \leq l, l' < k$ and $l \neq l'$, at the most there is one path that passing through the switch in the turnaround phase by using the input port $l$ and the output port $l'$, only if $l = o_0$. Hence,*

$$T(\langle n-1, o\rangle, l, l') = \begin{cases} 1, & \text{if } l = o_0 \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path goes through the switch $\langle n-1, o\rangle$ by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_b^t(\langle n-1, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle n-1, o\rangle, l')$, where $l \neq l'$, and the identifier $h'$ is obtained applying the perfect–shuffle function to $h$. Therefore,

$$T(\langle n-1, o\rangle, l, l') = \text{card}((N_b^t(\langle n-1, o\rangle, l))^\sigma \cap N_b^R(\langle n-1, o\rangle, l'))$$

$$\begin{aligned}&(N_b^t(\langle n-1, o\rangle, l))^\sigma \cap N_b^R(\langle n-1, o\rangle, l') = \\ &\quad = (\{\ (h_{n-1}, \ldots, h_0) :\ h_{n-1} = l\ \})^\sigma\ \cap\ \{\ (h_{n-1}, \ldots, h_0) :\ h_{n-1} = l' \text{ and } h_i = o_i\ \forall i \in [0, n-2]\ \}\end{aligned}$$

then the perfect–shuffle function is applied over the source set $N_b^t(\langle n-1, o\rangle, l)$

$$\begin{aligned}&(N_b^t(\langle n-1, o\rangle, l))^\sigma \cap N_b^R(\langle n-1, o\rangle, l') = \\ &\quad = (\{(h_{n-1}, \ldots, h_0) :\ h_0 = l\ \} \cap \{\ (h_{n-1}, \ldots, h_0) :\ h_{n-1} = l' \text{ and } h_i = o_i\ \forall i \in [0, n-2]\ \} \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_{n-1} = l', h_i = o_i\ \forall i \in [0, n-2] \text{ and } h_0 = l\ \} \\ &\quad = \{\ (l', o_{n-2}, \ldots, o_0) :\ \text{ if } o_0 = l\}\end{aligned}$$

As all the digits of the node identifier are fixed, at the most there is one path that turns around in the switch $\langle n-1, o\rangle$ by using $l$ and $l'$. Moreover, the input port is $l = o_0$. □

***Proposition 1.47*** *The number of paths that turn around in a switch of the last stage is $k-1$. Hence,*

$$T(\langle n-1, o\rangle) = k-1$$

**Proof:** According to Proposition 1.46 all the paths passing through the switch of the last stage in the turnaround phase use the same input port, $l = o_0$, but they must use one of the remaining $k-1$ ports. Since there is a unique port that verifies $l = o_0$, there exist $k-1$ paths in the turnaround phase in the switch $\langle n-1, o\rangle$. □

### 7.1.3 Descending phase of the paths

Some propositions related to the paths passing through the switch $\langle s, o\rangle$ in the descending phase, $0 \leq s < n$, are described below.

***Proposition 1.48*** *There is no path passing through, in descending phase, the switches of the last stage. Hence,*

$$C_b(\langle n-1, o\rangle) = C_b(\langle n-1, o\rangle, l, l') = 0$$

**Proof:** By the definition of the BMIN topology, there are no backward connections in the switches of the last stage. □

***Proposition 1.49*** *Given the ports $l$ and $l'$ of the switch $\langle s, o\rangle$, where $1 \leq s < n-1$, $k \leq l < 2k$ and $0 \leq l' < k$, at the most there is one path in the descending phase that passes through the switch by using the ports $l$ and $l'$, only if $o_0 \neq o_{n-2}$ or $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$. Hence,*

$$C_b(\langle s, o\rangle, l, l') = \begin{cases} 1, & \text{if } o_0 \neq o_{n-2} \text{ or } \exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path goes through the switch $\langle s, o\rangle$ in the descending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_f^t(\langle s, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle s, o\rangle, l')$, and the node identifier $h'$ is obtained applying the perfect–shuffle function to $h$. Therefore,

$$C_b(\langle s, o\rangle, l, l') = \text{card}((N_f^t(\langle s, o\rangle, l))^\sigma \cap N_b^R(\langle s, o\rangle, l'))$$

$$\begin{aligned}
&(N_f^t(\langle s, o\rangle, l))^\sigma \cap N_b^R(\langle s, o\rangle, l') = \\
&\quad = (\{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [s+1, n-1] \mid h_i \neq o_{i-1}\})^\sigma \cap \\
&\qquad \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1],\ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l'\ \}
\end{aligned}$$

Then the function perfect–shuffle is applied over the source set $N_f^t(\langle s, o\rangle, l)$

$$\begin{aligned}
&(N_f^t(\langle s, o\rangle, l))^\sigma \cap N_b^R(\langle s, o\rangle, l') = \\
&\quad = \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [s+2, n-1] \mid h_i \neq o_{i-2} \text{ or } h_0 \neq o_{n-2}\ \} \cap \\
&\qquad \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1],\ h_i = o_i\ \forall i \in [0, s-1] \text{ and } h_s = l'\ \} \\
&\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [s+1, n-1],\ h_i = o_i\ \forall i \in [0, s-1],\ h_s = l' \text{ and} \\
&\qquad \exists i \in [s+2, n-1] \mid h_i \neq o_{i-2} \text{ or } h_0 \neq o_{n-2}\ \} \\
&\quad = \{\ (o_{n-2}, \ldots, o_s, l', o_{s-1}, \ldots, o_0) \text{ if } \exists i \in [s+2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ or } o_0 \neq o_{n-2}\ \}
\end{aligned}$$

All the node identifier digits that meet the conditions are fixed, so at the most there is one path in descending phase that passes through the switch by using $l$ and $l'$.

It must be verified that there exist a $i \in [s+2, n-1]$ such as $o_{i-1} \neq o_{i-2}$, or $o_0 \neq o_{n-2}$. Then, $l'$ holds the same position as the digit $h_s$, which is free of restrictions, so $l'$ would take any value of the interval $[0, k-1]$. □

***Proposition 1.50*** *Given the ports $l$ and $l'$ of the switch $\langle s, o\rangle$, where $1 \leq s < n-1$, $k \leq l < 2k$ and $0 \leq l' < k$, there exists one path in the descending phase that passes through the switch by using the ports $l$ and $l'$ if and only if $o_0 \neq o_{n-2}$ or $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$ and $l' = l - k$. Hence,*

$$C_b(\langle s, o\rangle, l, l') = \begin{cases} 1, & \text{if } o_0 \neq o_{n-2} \text{ or} \exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \text{ and } l' = l - k \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.49, at the most there is one path passing through the switch $\langle s, o\rangle$ in the descending phase with the output port $l'$ only if $o_0 \neq o_{n-2}$ or $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$. On the other hand, by the Proposition 1.12 it is known there exists a descending path between the ports $l$ and $l'$ only if $l' = l - k$. □

***Proposition 1.51*** *The number of paths passing through the switch $\langle s, o\rangle$ in descending phase, $1 \leq s < n-1$, is $k$ at the most.*

$$C_b(\langle s, o\rangle) = \begin{cases} k, & \text{if } o_0 \neq o_{n-2} \text{ or } \exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.50, there is one path passing through the switch $\langle s, o\rangle$ in the descending phase if either there exist a $i \in [s+2, n-1]$ such as $o_{i-1} \neq o_{i-2}$, or $o_0 \neq o_{n-2}$ is verified, and also $l' = l - k$.

The port $l$ takes values in the interval $[k, 2k-1]$, while the port $l' = l - k$ does the same in $[0, k-1]$ that coincides with the range of $l'$. So there are $k$ available pairs $(l, l')$ that will be used by the paths to pass through the switch $\langle s, o\rangle$ in the descending phase. □

***Proposition 1.52*** *Given the ports $l$ and $l'$ of the switch $\langle 0, o\rangle$, where $k \leq l < 2k$ and $0 \leq l' < k$, at the most there is one path in descending phase that passes through the switch by using $l$ and $l'$, only if $\exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ or $\nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ and $l' \neq o_{n-2}$. Hence,*

$$C_b(\langle 0, o\rangle, l, l') = \begin{cases} 1, & \text{if } \exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ and } l' = l - k \\ 1, & \text{if } \nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ and } l' = l - k \text{ and } l' \neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** A path goes through the switch $\langle 0, o\rangle$ in the descending phase by using the ports $l$ and $l'$ if the source node, $h$, belongs to the set $N_f^t(\langle 0, o\rangle, l)$, the destination node, $h'$, belongs to the set $N_b^R(\langle 0, o\rangle, l')$ and the identifier $h'$ is obtained applying the perfect–shuffle function to $h$. Therefore,

$$C_b(\langle 0, o\rangle, l, l') = \text{card}((N_f^t(\langle 0, o\rangle, l))^\sigma \cap N_b^R(\langle 0, o\rangle, l'))$$

$$\begin{aligned} &(N_f^t(\langle 0, o\rangle, l))^\sigma \cap N_b^R(\langle 0, o\rangle, l') = \\ &\quad = (\{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [1, n-1] \mid h_i \neq o_{i-1}\})^\sigma \cap \\ &\qquad \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1] \text{ and } h_0 = l'\ \} \end{aligned}$$

Then the function perfect–shuffle is applied over the source set $N_f^t(\langle 0, o\rangle, l)$.

$$\begin{aligned} &(N_f^t(\langle 0, o\rangle, l))^\sigma \cap N_b^R(\langle 0, o\rangle, l') = \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ \exists i \in [2, n-1] \mid h_i \neq o_{i-2} \text{ or } h_0 \neq o_{n-2}\} \cap \\ &\qquad \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1] \text{ and } h_0 = l'\ \} \\ &\quad = \{\ (h_{n-1}, \ldots, h_0) :\ h_i = o_{i-1}\ \forall i \in [1, n-1], h_s = l',\ \exists i \in [2, n-1] \mid h_i \neq o_{i-2} \text{ or } h_0 \neq o_{n-2}\ \} \\ &\quad = \{\ (o_{n-2}, \ldots, o_0, l') \text{ if } \exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ or } l' \neq o_{n-2}\ \} \end{aligned}$$

All the node identifier digits that meet the conditions are fixed, so at the most there is one path in descending phase that passes through the switch by using $l$ and $l'$. It must be verified that there exist a $i \in [2, n-1]$ such as $o_{i-1} \neq o_{i-2}$, or $l' \neq o_{n-2}$. If the first condition is verified $l'$ will take a value of the interval $[0, k-1]$. Otherwise, $l'$ will be different from $o_{n-2}$. □

***Proposition 1.53*** *Given the ports $l$ and $l'$ of the switch $\langle 0, o\rangle$, where $k \leq l < 2k$ and $0 \leq l' < k$ at the most there exists one path passing through the switch in the descending phase by using the input port $l$ and output port $l'$, only if $\exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ or $\nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ and $l' \neq o_{n-2}$ and $l' = l - k$. Hence,*

$$C_b(\langle 0, o\rangle, l, l') = \begin{cases} 1, & \text{if } \exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ and } l' = l - k \\ 1, & \text{if } \nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2} \text{ and } l' = l - k \text{ and } l' \neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.52, at the most there exists one path in the descending phase with output port $l'$ only if $\exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ or $\nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ and $l' \neq o_{n-2}$. On the other hand, by Proposition 1.12 it is known there exists a path in the descending phase between the ports $l$ and $l'$ only if $l' = l - k$. □

***Proposition 1.54*** *The number of paths passing through the switch $\langle 0, o \rangle$ in descending phase is $k$ if $\exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1}$; otherwise it is $k-1$. Hence,*

$$C_b(\langle 0, o \rangle) = \begin{cases} k, & \text{if } \exists i \in [2, n-1] \mid o_{i-2} \neq o_{i-1} \\ k-1, & \text{otherwise} \end{cases}$$

**Proof:** According to Proposition 1.52 there is one path that passes through the switch $\langle 0, o \rangle$ in descending phase if there exists an $i \in [2, n-1]$ such as $o_{i-1} \neq o_{i-2}$, or if $l' \neq o_{n-2}$ and also $l' = l - k$.

The port $l$ takes values in the interval $[k, 2k-1]$, while the port $l' = l - k$ does the same in $[0, k-1]$ that coincides with the range of $l'$. So there are $k$ available pairs $(l, l')$ that will be used by the paths to pass through the switch $\langle s, o \rangle$ in the descending phase.

If $\exists i \in [2, n-1] | o_{i-1} \neq o_{i-2}$, then there will be one path in the descending phase by every port $l'$, so the total number of paths will be $k$. Otherwise, if $l' \neq o_{n-2}$ then only $k-1$ values will be valid for $l$ and therefore there will be $k-1$ paths. □

To sum up, Table 2 outlines the expressions obtained in the previous propositions.

## 7.2 Switch Classification

According to the expressions in Table 2, when the network topology is a $N$ end nodes $T$–BMIN $k$–ary $n$–tree multistage network, the generated traffic is based on the perfect–shuffle traffic pattern ($\sigma$) and the paths are determined by the routing algorithm defined in Section A.7.1, six types of switches are identified according to the connections required in the switches.

In the following we introduce the six types of switches, dividing them in three subgroups based on the stage, where the switch is situated. Also, we show the connections for each type using a 4×4 switch as an example. We assume that its identifier $\langle s, o \rangle$ is such as $o_s = 1$.

### 7.2.1 First stage ($s = 0$)

There is no path passing through the switches in the first stage in the turnaround phase according to Proposition 1.43. That means connections in these switches are established by the paths in the ascending and descending phase. The condition $\exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$ determines the number of paths passing through the switch for both phases.

#### 7.2.1.1 Type $\sigma a$

**Condition:** $\exists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$

For this type, according to Proposition 1.39 it is known there are $k$ paths in the ascending phase, and $k$ paths in the descending phase by the Proposition 1.54. Moreover, Propositions 1.38 and 1.53 determine which connections are taken based on the relationship between ports and switches. Figure 17(a) illustrates these paths.

Table 2: Number of paths passing through the switch $\langle s, o\rangle$ in then ascending, turnaround and descending phases under perfect–shuffle traffic.

$$C_f(\langle 0,o\rangle,l,l') \underset{0\le l<k,\ k\le l'<2k}{=} \begin{cases} 1, & \text{if } l'-k=o_{n-2} \text{ and } \exists i\in[2,n-1] \mid o_{i-2}\neq o_{i-1} \\ 1, & \text{if } l'-k=o_{n-2} \text{ and } \nexists i\in[2,n-1] \mid o_{i-2}\neq o_{i-1} \text{ and } l\neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

$$C_f(\langle 0,o\rangle) = \begin{cases} k, & \text{if } \exists i\in[2,n-1] \mid o_{i-2}\neq o_{i-1} \\ k-1, & \text{otherwise} \end{cases}$$

$$C_f(\langle s,o\rangle,l,l') \underset{1\le s<n-1,\ 0\le l<k,\ k\le l'<2k}{=} \begin{cases} 1, & \text{if } o_0=o_{n-2} \text{ and } \exists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \\ 1, & \text{if } o_0=o_{n-2} \text{ and } \nexists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \text{ and } l\neq o_s \\ 0, & \text{otherwise} \end{cases}$$

$$C_f(\langle s,o\rangle) \underset{1\le s<n-1}{=} \begin{cases} k^2, & \text{if } o_0=o_{n-2} \text{ and } \exists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \\ k^2-k, & \text{if } o_0=o_{n-2} \text{ and } \nexists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \\ 0, & \text{otherwise} \end{cases}$$

$$C_f(\langle n-1,o\rangle,l,l') \underset{0\le l<k,\ k\le l'<2k}{=} C_f(\langle n-1,o\rangle) = 0$$

$$T(\langle 0,o\rangle,l,l') \underset{0\le l,l'<k,\ l\neq l'}{=} T(\langle 0,o\rangle) = 0$$

$$T(\langle s,o\rangle,l,l') \underset{1\le s<n-1,\ 0\le l,l'<k,\ l\neq l'}{=} \begin{cases} 1, & \text{if } o_{i-1}=o_{i-2}\ \forall i\in[s+2,n-1] \text{ and } o_0=o_{n-2} \text{ and } l=o_s \\ 0, & \text{otherwise} \end{cases}$$

$$T(\langle s,o\rangle) \underset{1\le s<n-1}{=} \begin{cases} k-1, & \text{if } o_{i-1}=o_{i-2}\ \forall i\in[s+2,n-1] \text{ and } o_0=o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

$$T(\langle n-1,o\rangle,l,l') \underset{0\le l,l'<k,\ l\neq l'}{=} \begin{cases} 1, & \text{if } l=o_0 \\ 0, & \text{otherwise} \end{cases}$$

$$T(\langle n-1,o\rangle) = k-1$$

$$C_b(\langle n-1,o\rangle,l,l') \underset{k\le l<2k,\ 0\le l'<k}{=} C_b(\langle n-1,o\rangle) = 0$$

$$C_b(\langle s,o\rangle,l,l') \underset{1\le s<n-1,\ k\le l<2k,\ 0\le l'<k}{=} \begin{cases} 1, & \text{if } \exists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \text{ or } o_0\neq o_{n-2} \text{ and } l'=l-k \\ 0, & \text{otherwise} \end{cases}$$

$$C_b(\langle s,o\rangle) \underset{1\le s<n-1}{=} \begin{cases} k, & \text{if } \exists i\in[s+2,n-1] \mid o_{i-2}\neq o_{i-1} \text{ or } o_0\neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

$$C_b(\langle 0,o\rangle,l,l') \underset{1\le s<n-1,\ k\le l<2k,0\le l'<k}{=} \begin{cases} 1, & \text{if } \exists i\in[2,n-1] \mid o_{i-1}\neq o_{i-2} \text{ and } l'=l-k \\ 1, & \text{if } \nexists i\in[2,n-1] \mid o_{i-1}\neq o_{i-2} \text{ and } l'=l-k \text{ and } l'\neq o_{n-2} \\ 0, & \text{otherwise} \end{cases}$$

$$C_b(\langle 0,o\rangle) = \begin{cases} k, & \text{if } \exists i\in[2,n-1] \mid o_{i-2}\neq o_{i-1} \\ k-1, & \text{otherwise} \end{cases}$$

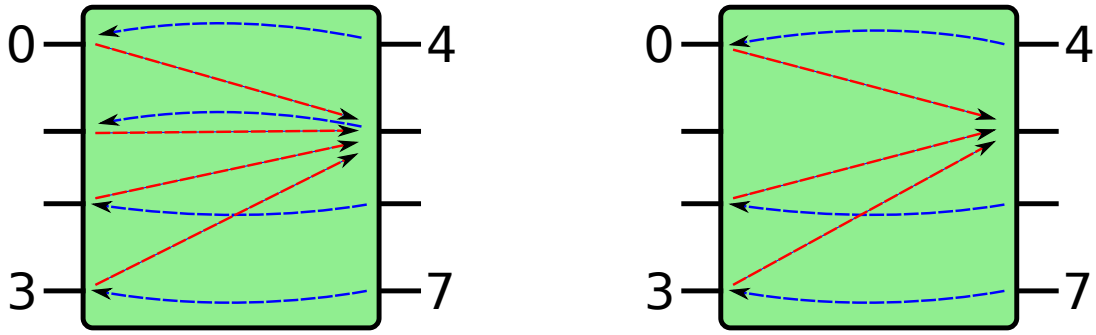
#### 7.2.1.2 Type $\sigma b$

**Condition:** $\nexists i \in [2, n-1] \mid o_{i-1} \neq o_{i-2}$

This type is similar to Type $\sigma a$, but removing two connections:

- The connection whose input port $o_{n-2}$ and it is established by a path in the ascending phase.
- The connection whose output port is $o_{n-2}$ and it is established by a path in the descending phase.

Figure 17(b) show a switch of Type $\sigma b$.




(a) Type $\sigma a$. (b) Type $\sigma b$.

Figure 17: Connections required for each type of switch in the first stage.

### 7.2.2 Intermediate stages ($1 \leq s < n-1$)

Because of Propositions 1.41, 1.45 and 1.51 that quantify the number of paths in the ascending, turnaround and descending phase, respectively, passing through a switch in the intermediate stage, we can extract the conditions that imply the existence of connections in these switches. Such conditions are the following:

1. $o_0 = o_{n-2}$
2. $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$, taking into account this confirms $o_{i-1} = o_{i-2} \ \ \forall i \in [s+2, n-1]$, because one is the negation of another.

So the existence of connections is given by analysing such conditions:

1. If $o_0 \neq o_{n-2}$, there are not paths neither in the ascending phase nor turnaround phase, but there are paths in the descending phase.
2. If $o_0 = o_{n-2}$ is verified then it will be necessary an extra checkout.
   (a) If $\exists i \in [s+2, n-1] \ \ o_{i-2} \neq o_{i-1}$ there are $k^2$ paths in the ascending phase and $k$ paths in the descending phase. There is no paths in the turnaround phase.
   (b) Otherwise, $o_{i-2} = o_{i-1} \ \forall i \in [s+2, n-1]$ would be verified, therefore, there would be $k^2 - k$ paths in the ascending phase and $k-1$ paths in the turnaround phase. However, there would not be paths passing through the switch in the descending phase.

Overall, we have identified the following types of switches:

#### 7.2.2.1 Type $\sigma c$

**Conditions** $o_0 = o_{n-2}$ and $\exists i \in [s+2, n-1] \mid o_{i-2} \neq o_{i-1}$

According to Proposition 1.40, there is one path in the ascending phase for every pairs of ports $l$ and $l'$, where $0 \leq l < k$ and $k \leq l' < 2k$, that is, $k^2$ different pairs are available (Proposition 1.41). However, according to Proposition 1.51 there are $k$ paths passing through the switch in the descending phase, and the required connections are determined by Proposition 1.50.

Figure 18(a) illustrates the established connections by the switch belonging to this type for the example switch.

#### 7.2.2.2 Type $\sigma d$

**Conditions** $o_0 = o_{n-2}$ and $o_{i-2} = o_{i-1}$ $\forall i \in [s+2, n-1]$

According to Proposition 1.41, there are $k^2 - k$ paths in the ascending phase existing one path between every pairs of ports $(l, l')$, where $0 \leq l < k$, $l \neq o_s$ and $k \leq l' < 2k$. Additionally, the Proposition 1.45 states there are $k-1$ paths in the turnaround phase by using the input port $l = o_s$ and the output port one of the remaining ports $l''$, $0 \leq l'' < k$ where $l \neq l''$.

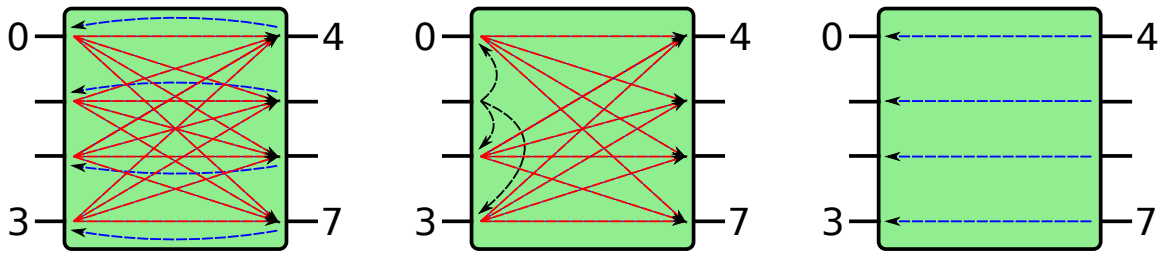
In the example switch, Figure 18(b) depicts the setup connections by a switch belonging to this type.

#### 7.2.2.3 Type $\sigma e$

**Conditions** $o_0 \neq o_{n-2}$

According to Proposition 1.51, there are $k$ paths in the descending phase, and the established connections are determined based on Proposition 1.50.

Similarly to previous types, the setup connections are indicated in the example switch in the Figure 18(c).




(a) Type $\sigma c$. (b) Type $\sigma d$. (c) Type $\sigma e$.

Figure 18: Connections required for the switches in the intermediate stages.

### 7.2.3 Last stage ($s = n - 1$)

#### 7.2.3.1 Type $\sigma f$

In the last stage switches, it is known there are only paths in the turnaround phase according to Propositions 1.42, 1.48 and 1.47. Specifically, there are a total of $k - 1$ paths in every switch. Additionally, the Proposition 1.46 states the connections have the same input port $o_0$ and, the output

port, is one the remaining ports.

Figure 19 illustrates the established connections by the switch belonging to this type for the example switch.

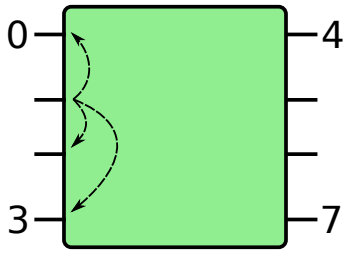



Figure 19: Connections required for the switches in the last stage.

## 7.3 Switch Configuration

To find out the most appropriate configuration of $T$-switches in this case, in which perfect–shuffle traffic pattern is considered, the methodology described in Section 2.2 is applied. However, in some cases, the optimal configurations of $T$-switches are derived considering all the types of connections at the same time instead considering separately each kind of connection. In these cases, the search of the configuration is simplified to a *function optimization problem.*

### 7.3.1 Type $\sigma a$ configuration of switch

***Proposition 1.55*** *Let $\mathcal{S}^{\sigma a}$ be the set of configurations that minimize the use of the internal link considering a $T$-switch of type $\sigma a$, then*

$$\mathcal{S}^{\sigma a} = \mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

*and there are $\dfrac{k}{2}$ connections using the internal link.*

**Proof:** According to Proposition 1.29, all configurations in $\mathcal{S}_b$ are optimal considering backward connections, and there are no backward connections using the internal link.

Let $b$ be the number of pairs of ports $(l, l') \in \mathcal{R}_b$ such that $l, l'$ belong to a configuration $\mathcal{C}$. At most there exist $k/2$ pairs of related ports such that $card(\mathcal{C}) = k$, that is, $0 \leq b \leq k/2$.

So, the number of backward connections that use the internal link depends on $b$. In a $T$-switch of type $\sigma a$, there exists a connection between each pair of ports $(l, l') \in \mathcal{R}_b$. Therefore, the total number of connections is $k$. Each pair of ports, which are not related, implies that there are two backward connections passing through the internal link. Consequently, there are $2(\frac{k}{2} - b) = k - 2b$ backward connections using the internal link. In particular, the configurations belonging to $\mathcal{S}^{\sigma a}$, considering all ports are in pairs ($b = k/2$) there are no connections using the internal link.

Since each pair $(l, l') \in \mathcal{R}_b$ implies having one port connected to downwards stage (or endnode) and another port connected to upwards stage, for all configuration in $\mathcal{C}$, where $\mathcal{B}$ and $\mathcal{F}$ are subsets of ports in $\mathcal{C}$ connecting to previous and next stage, respectively, then

$$card(\mathcal{B}) \geq b \tag{25}$$

$$card(\mathcal{F}) \geq b \tag{26}$$

On the other hand, every port ports connected to a previous stage establishes a forward connection with the port $o_{n-2} + k$. Thus, the number of forward connections using the internal link is equal to the number of ports connecting to previous stage and not belonging to $\mathcal{C}$, which includes the port $o_{n-2} + k$. That is to say, the number of forward connections using the internal link is equal to the cardinal of $\mathcal{F}$.

It is possible to prove that configurations in $\mathcal{S}^{\sigma a}$ verify that $card(\mathcal{B}) = card(\mathcal{F}) = k/2$. Therefore, the number of forward connections that use the internal link of $T$-switch, and also the total number of connections, since there exist no backward connections using the internal link, is $k/2$.

In the following, we demonstrate by reductio ad absurdum that configurations belonging to $\mathcal{S}^{\sigma a}$ are uniquely optimal. Let us suppose that $\mathcal{C}'$ minimizes the use of the internal link and $\mathcal{C}' \notin \mathcal{S}^{\sigma a}$. This means there exists a port $l$ that is not related to another port $l'$.

$$\mathcal{C}' \notin \mathcal{S}^{\sigma a} \;\Rightarrow\; \exists l \in \mathcal{C}' \mid \forall l' \in \mathcal{C}',\ (l, l') \notin \mathcal{R}_b$$

As there is at least one port $l$ that is not matched (is single), according to the binary relation $\mathcal{R}_b$, then

$$0 \leq b < k/2 \tag{27}$$

Two cases are possible, depending on whether or not the port $o_{n-2}$ belongs to $\mathcal{C}'$. If the port $o_{n-2}$ belongs to $\mathcal{C}'$, the number of connections using the internal link is $k - 2b + card(\mathcal{F}')$. By the expression 26

$$k - 2b + card(\mathcal{F}') \geq k - 2b + b = k - b$$

and the expression 27,

$$k - b > k - \frac{k}{2} = \frac{k}{2}$$

That is to say, the number of connections that use the internal link is greater than $k/2$, and for this reason $\mathcal{C}'$ does not minimize the use of the internal link, because there are $k/2$ connections belonging to $\mathcal{S}^{\sigma a}$ using the internal link. This is in contradict with the initial hypothesis: $\mathcal{C}'$ minimizes the use of the internal link.

On the other hand, if considering the port $o_{n-2}$ does not belong to $\mathcal{C}'$, the number of connections that use the internal link will be $k - 2b + card(\mathcal{B}')$. The same conclusion is reached by expression 25: $\mathcal{C}'$ is not optimal.

Therefore, the configurations belonging to $\mathcal{S}^{\sigma a}$ minimize the number of connections that use the internal link, and they are the unique ones. □

Example 1.10 shows two optimal $T$-switch configurations of type $\sigma a$.

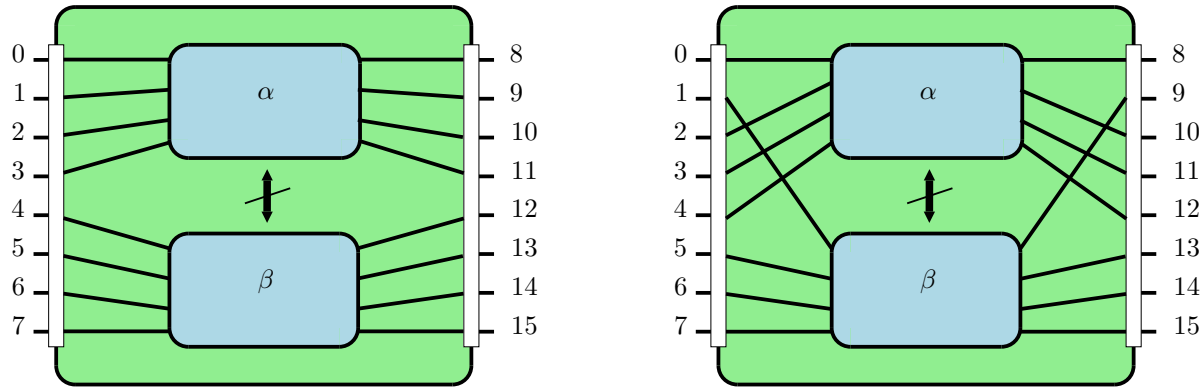
***Example 1.10*** *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two configurations for a 8×8 $T$-switch, where*

$$\mathcal{T}_1 = \left\{\mathcal{C}_1^{\alpha}, \mathcal{C}_1^{\beta} \in \mathcal{V} \mid \mathcal{C}_1^{\alpha} = \{0,1,2,3,8,9,10,11\},\ \mathcal{C}_1^{\beta} = (\mathcal{C}_1^{\alpha})^C = \{4,5,6,7,12,13,14,15\}\right\}$$
$$\mathcal{T}_2 = \left\{\mathcal{C}_2^{\alpha}, \mathcal{C}_2^{\beta} \in \mathcal{V} \mid \mathcal{C}_2^{\alpha} = \{0,2,3,4,8,10,11,12\},\ \mathcal{C}_2^{\beta} = (\mathcal{C}_2^{\alpha})^C = \{1,5,6,7,9,13,14,15\}\right\}$$

*Both $\mathcal{T}_1$ and $\mathcal{T}_2$ are optimal configurations to backward connections for type $\sigma a$ configuration of $T$-switch, since it is verified:*

$$\mathcal{C}_1^{\alpha},\ \mathcal{C}_1^{\beta},\ \mathcal{C}_2^{\alpha},\ \mathcal{C}_2^{\beta} \in \mathcal{S}^{\sigma a}$$

*Figura 20 shows the connections of $\mathcal{T}_1$ and $\mathcal{T}_2$ configurations.*




(a) $\mathcal{T}_1$ (b) $\mathcal{T}_2$

Figure 20: Possible optimal configurations for a 8×8 $T$-switch of type $\sigma a$.

### 7.3.2 Type $\sigma b$ configuration of switch

***Proposition 1.56*** *Let $\mathcal{S}^{\sigma b}$ be the set of configurations that minimize the use of the internal link of a $T$-switch of type $\sigma b$, then*

$$\mathcal{S}^{\sigma b} = \mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

*and there exist $\dfrac{k}{2}$ connections using the internal link.*

**Proof:** According to Proposition 1.29, configurations of $\mathcal{S}_b$ are optimal considering the backward connnections, and there are no backward connections using the internal link.

On the other hand, all the nodes connected to a previous stage (or terminal) establish a forward connection to the port $o_{n-2} + k$, except the port $o_{n-2}$. The number of forward connections using the internal link is the number of ports connected to the previous stage, which do not belong to the configuration $\mathcal{C}^i$ such that $o_{n-2} + k \in \mathcal{C}^i$. That is, the number of forward connections using the internal link is equal to the cardinal of $\mathcal{F}^i$. Moreover, if $o_{n-2} \notin \mathcal{C}^i$, the number of connections is $card(\mathcal{F}^i) - 1$, since the port $o_{n-2}$ does not establish a forward connection.

It is possible to prove that configurations in $\mathcal{S}^{\sigma b}$ verify $card(\mathcal{B}) = card(\mathcal{F}) = k/2$. Therefore, the number of forward connections using the internal link is $k/2$, because the port $o_{n-2}$ belongs to the same configuration as the port $o_{n-2} + k$, since $(o_{n-2}, o_{n-2} + k) \in \mathcal{R}_b$.

Hence, the total number of connections using the internal link is $k/2$.

Next, we demonstrate by reductio ad absurdum that configurations belonging to $\mathcal{S}^{\sigma b}$ are uniquely optimal. Also it is considered $b$ being the number of pairs of ports $(l, l') \in \mathcal{R}_b$, where $l, l'$ belong to a configuration $\mathcal{C}$ such that $0 \leq b \leq k/2$.

Let's suppose that another configuration $\mathcal{C}'$ minimizes the use of the internal link, while it verifies $\mathcal{C}' \notin \mathcal{S}^{\sigma b}$. This means that there exists a port $l$ that is not related to another port $l'$.

$$\mathcal{C}' \notin \mathcal{S}^{\sigma b} \ \Rightarrow\ \exists l \in \mathcal{C}' \mid \forall l' \in \mathcal{C}',\ (l, l') \notin \mathcal{R}_b$$

As there is at least one port $l$ is not matched, then

$$0 \leq b < k/2 \tag{28}$$

To obtain the number of connections that use the internal link and belong to $\mathcal{C}'$, two different cases are considered: a) ports $o_{n-2}$ and $o_{n-2}+k$ belong to the same configuration ($o_{n-2}, o_{n-2}+k \in \mathcal{C}'$

or $o_{n-2}, o_{n-2}+k \in (\mathcal{C}')^C$); b) such ports belong to different configurations ($o_{n-2} \in \mathcal{C}', o_{n-2}+k \in (\mathcal{C}')^C$ or $o_{n-2} \in (\mathcal{C}')^C, o_{n-2}+k \in \mathcal{C}'$).

If $o_{n-2}$ and $o_{n-2}+k$ belong to the same configuration, then

- The unique pair of ports $(l, l') \in \mathcal{R}_b$ that does not establish any backward connection is $(o_{n-2}, o_{n-2}+k)$. Since both ports belong to the same configuration, the number of backward connections using the internal link is $k-2b$.
- The number of forward connections using the internal link is
  - $card(\mathcal{F}')$, if $o_{n-2}+k \in \mathcal{C}'$
  - $card(\mathcal{B}')$, if $o_{n-2}+k \notin \mathcal{C}'$

Similar to Proposition 1.55 proof, it can be proved that $\mathcal{C}'$ is not optimal in this case, and then configurations belonging to $\mathcal{S}^{\sigma b}$ are optimal.

If $o_{n-2}$ and $o_{n-2}+k$ do not belong to the same configuration, then

- The unique pair of ports $(l, l') \in \mathcal{R}_b$ that does not establish any backward connection is $(o_{n-2}, o_{n-2}+k)$. Since both ports belong to different configurations, the number of backward connections using the internal link is $k-2b-1$.
- The number of forward connections that use the internal link is
  - $card(\mathcal{F}')-1$, if $o_{n-2}+k \in \mathcal{C}'$
  - $card(\mathcal{B}')-1$, if $o_{n-2}+k \notin \mathcal{C}'$

On the other hand, each pair of ports $(l, l') \in \mathcal{R}_b$ involves there is a port connected to the previous stage and another port connected to the next stage. In this case, $o_{n-2}$ and $o_{n-2}+k$ would be in different configurations, so:

$$card(\mathcal{F}')-1 \geq b \quad \text{if } o_{n-2}+k \in \mathcal{C}' \tag{29}$$
$$card(\mathcal{B}')-1 \geq b \quad \text{if } o_{n-2}+k \notin \mathcal{C}' \tag{30}$$

If the port $o_{n-2}$ belongs to $\mathcal{C}'$, the number of connections using the internal link will be $k-2b-1+card(\mathcal{F})-1$. By the expression 29:

$$k-2b-1+card(\mathcal{F})-1 \geq k-2b-1+b = k-b-1$$

and by the expression 28

$$k-b-1 > k-\frac{k}{2}-1 = \frac{k}{2}-1$$

As we work with integers, it must be

$$k-b-1 \geq \frac{k}{2}$$

In this case, it is possible that configuration $\mathcal{C}'$ would minimize the use of the internal link, but in no case the number of connections using the internal link will be less than in the configurations belonging to $\mathcal{S}^{\sigma b}$.

It is also possible to conclude that $\mathcal{C}'$ can be optimal, but never better than a configuration belonging to $\mathcal{S}^{\sigma b}$, considering that $o_{n-2}$ does not belong to $\mathcal{C}'$, and according to expression 30.

Consequently, configurations belonging to $\mathcal{S}^{\sigma b}$ minimize the number of connections using the internal link. □

It must be noticed the set $\mathcal{S}^{\sigma b}$ contents optimal configurations for a $T$-switch of type $\sigma b$ with independence of the switch identifier, but there also exist other configurations that minimize the use of the internal link, which depend on the switch identifier.

The configurations shown in Example 1.10 are equally optimal in this case.

### 7.3.3 Type $\sigma c$ configuration of switch

According to our methodology, the sets of optimal configurations are separately calculated for each of the connections (i.e., forward and backward). From these sets, the configurations of $T$-switch are derived taking into account such results.

#### 7.3.3.1 Optimal configuration considering forward connections

***Proposition 1.57*** *Let $\mathcal{S}_f^{\sigma c}$ be the set of configurations that minimize the use of the internal link of a $T$-switch of type $\sigma c$ considering the forward connections, then*

$$\mathcal{S}_f^{\sigma c} = \{\mathcal{C} \in \mathcal{V} \mid card(\mathcal{B}) = card(\mathcal{F}) = k/2\}$$

*and there exist $\frac{1}{2}k^2$ connections using the internal link.*

**Proof:** In this case, the port $l$, $0 \leq l < k$, establishes a forward connection to each port $l'$, $k \leq l' < 2k$. That is, all the ports $l$ behavior identically with independence of their identifier within the switch.

Therefore, the optimal value of $p$ must be calculated to obtain the optimal configurations.

First, the value of $C_I(\langle s, o\rangle)$ must be calculated.

$$C_I(\langle s, o\rangle) = \frac{C_f(\langle s, o\rangle) \times CC_{If}(\langle s, o\rangle)}{k^2} = 2p^2 + k^2 - 2kp$$

To obtain the value of $p$ that minimizes $C_I(\langle s, o\rangle)$ it is necessary to calculate the derivative of $C_I(\langle s, o\rangle)$ with respect to $p$.

$$C'_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C_I(\langle s, o\rangle) = 4p - 2k$$

Then, the derivative $C'_I(\langle s, o\rangle)$ is equal to zero to determine the critical points (e.g., maximum and minimum).

$$\begin{aligned} C'_I(\langle s, o\rangle) &= 0 \\ 4p - 2k &= 0 \\ p &= \frac{k}{2} \end{aligned}$$

To know if $p = k/2$ is a critical point, the second derivative of $C_I(\langle s, o\rangle)$ with respect to $p$ must be used at the point $p = k/2$. If it takes a negative value, or positive value, the function will have a maximum, or minimum, at the point $p = k/2$, respectively. Otherwise, the point $p = k/2$ will be a possible inflection point.

$$C''_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C'_I(\langle s, o\rangle) = 4$$

As $C_I''(\langle s,o\rangle)$ is always positive, the function $C_I(\langle s,o\rangle)$ has a minimum at $p = k/2$. Then, substituting the value of $p$ in the function $C_I(\langle s,o\rangle)$:

$$C_I(\langle s,o\rangle) = 2\left(\frac{k}{2}\right)^2 k + k^2 - 2\frac{k}{2}k = \frac{1}{2}k^2$$

That is to say, a configuration is optimal if $p = q = k/2$ and the number of paths passing through the internal link of the $T$-switch is $\frac{1}{2}k^2$. □

Example 1.11 shows an optimal configuration for a $T$-switch of type $\sigma c$, considering forward connections.

***Example 1.11*** *Let $\mathcal{T}$ be a configuration for a 8×8 $T$-switch, where*

$$\mathcal{T} = \left\{\mathcal{C}^\alpha, \mathcal{C}^\beta \in \mathcal{V} \mid \mathcal{C}^\alpha = \{0,1,2,3,8,9,12,13\},\ \mathcal{C}^\beta = (\mathcal{C}^\alpha)^C = \{4,5,6,7,10,11,14,15\}\right\}$$

*$\mathcal{T}$ is an optimal configuration to forward connections for $T$-switches of type $\sigma c$, since it is verified:*

$$\mathcal{C}^\alpha,\ \mathcal{C}^\beta \in \mathcal{S}_f^{\sigma c}$$

*Figure 21 shows the connections of the $\mathcal{T}$ configuration.*

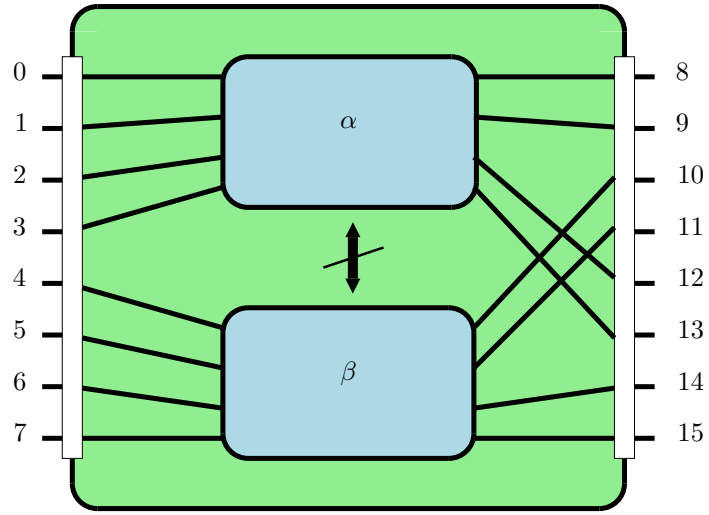



Figure 21: Possible optimal configuration for a $8 \times 8$ $T$-switch of type $\sigma c$ considering forward connections.

#### 7.3.3.2 Optimal configuration considering backward connections

***Proposition 1.58*** *Let $\mathcal{S}_b^{\sigma c}$ be the set of configurations that minimize the use of the internal link for a $T$-switch of type $\sigma c$ considering backward connections, then*

$$\mathcal{S}_b^{\sigma c} = \mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l,l') \in \mathcal{R}_b\}$$

*and there are no backward connections using the internal link.*

**Proof:** According to Proposition 1.29, configurations in $\mathcal{S}_b$ are optimal considering backward connections. Since there exist backward connections in a $T$-switch of type $\sigma c$, which are established between the pair of ports $l, l'$ such that $(l, l') \in \mathcal{R}_b$, the configurations of $\mathcal{S}_b$ are the uniquely optimal configurations according to Proposition 1.30. □

The configurations shown in the Example 1.10 are equally optimal in this case.

#### 7.3.3.3 Optimal configuration considering all connections

***Proposition 1.59*** *Let $\mathcal{S}^{\sigma c}$ be the set of configurations that minimize the use of the internal link for a $T$-switch of type $\sigma c$, then*

$$\mathcal{S}^{\sigma c} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

*and there exist $\frac{1}{2}k^2$ connections using the internal link.*

**Proof:** From definitions of sets $\mathcal{S}_f^{\sigma c}$ and $\mathcal{S}_b^{\sigma c}$, it is proved that $\mathcal{S}^{\sigma c} = \mathcal{S}_b^{\sigma c}$, and configurations belonging to $\mathcal{S}^{\sigma c}$ are optimal, and there exist $\frac{1}{2}k^2$ connections using the internal link. □

The configurations shown in the Example 1.10 are equally optimal in this case.

### 7.3.4 Type $\sigma d$ configuration of switch

***Proposition 1.60*** *Let $\mathcal{S}^{\sigma d}$ be the set of configurations that minimize the use of the internal link for a $T$-switch of type $\sigma d$, then*

$$\mathcal{S}^{\sigma d} = \{\mathcal{C} \in \mathcal{V} \mid card(\mathcal{B}) = card(\mathcal{F}) = k/2\}$$

*and there exist $\frac{1}{2}k^2$ connections using the internal link.*

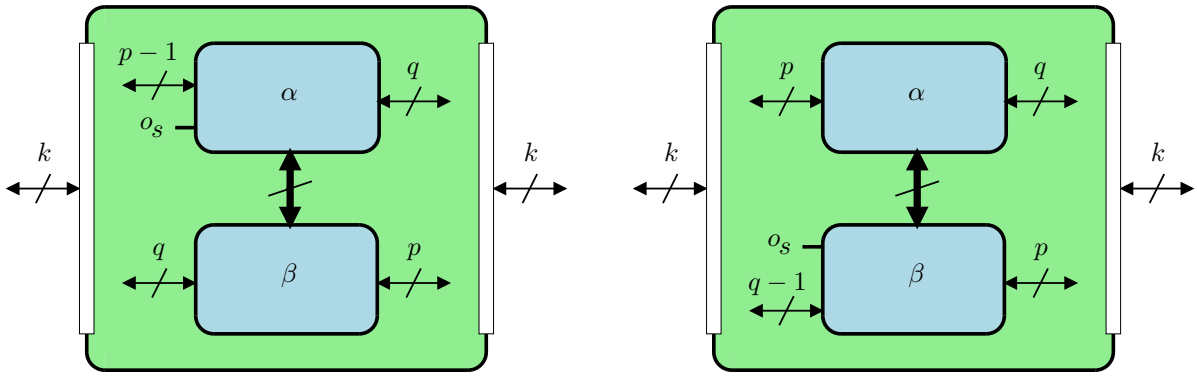
**Proof:** In this case, a port $l$, $0 \leq l < k$, $l \neq o_s$, establishes a forward connection with each port $l'$, $k \leq l' < 2k$. The port $o_s$ establishes a turnaround connection with the remaining ports connected to switches at the previous stage.

In other words, every port $l$ have identical behavior with independence of its identifier inside the switch, except $l = o_s$. Again, it is possible to recognize the set of optimal configurations that minimize the use of the internal link by tuning the value of $p$, but as the port $o_s$ does not behave like other ports, two different cases have been identified: a) when the port $o_s$ is one of the $p$ ports connected to a previous–stage switch; b) when the port $o_s$ is one of the $q$ ports connected to a previous–stage switch. Figure 22 highlights both cases.

#### 7.3.4.1 The port $o_s$ is one of the $p$ backward ports

Firstly, the value of $C_I(\langle s, o\rangle)$ must be obtained.

$$\begin{aligned} C_I(\langle s, o\rangle) &= (p-1)p + q^2 + q \\ &= (p-1)p + (k-p)^2 + (k-p) \\ &= 2p^2 - 2p(k+1) + k^2 + k \end{aligned}$$




(a) $o_s$ is one of the $p$ (backward) ports. (b) $o_s$ is one of the $q$ (backward) ports.

Figure 22: Identified cases for a $T$-switch of type $\sigma d$.

To obtain the value of $p$ that minimizes $C_I(\langle s, o\rangle)$, it is necessary to realize the second derivative of $C_I(\langle s, o\rangle)$ with respect to $p$.

$$C'_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C_I(\langle s, o\rangle) = 4p - 2(k+1)$$

Then, the second derivative $C'_I(\langle s, o\rangle)$ is equal to zero to determine the critical points.

$$\begin{aligned} C'_I(\langle s, o\rangle) &= 0 \\ 4p - 2(k+1) &= 0 \\ p &= \frac{k+1}{2} \end{aligned}$$

To know if $p = \frac{k+1}{2}$ is a critical point, the second derivative of $C_I(\langle s, o\rangle)$ with respect to $p$ must be used at the point $p = \frac{k+1}{2}$. If it takes a negative value, or positive value, the function will have a maximum, or minimum, at the point $p = \frac{k+1}{2}$, respectively. Otherwise, the point $p = \frac{k+1}{2}$ will be a possible inflection point.

$$C''_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C'_I(\langle s, o\rangle) = 4$$

As $C''_I(\langle s, o\rangle)$ takes only positive values, the function $C_I(\langle s, o\rangle)$ has a minimum at $p = \frac{k+1}{2}$. However, $k$ is even, therefore, $\frac{k+1}{2}$ is not an integer number, so it is necessary to check out the values of $C_I(\langle s, o\rangle)$ at two nearby points of $\frac{k+1}{2}$.

- If $p = \left\lfloor \frac{k+1}{2} \right\rfloor = \frac{k}{2}$, then $C_I(\langle s, o\rangle) = 2\left(\frac{k}{2}\right)^2 - 2\left(\frac{k}{2}\right)(k+1) + k^2 + k = \frac{1}{2}k^2$
- If $p = \left\lceil \frac{k+1}{2} \right\rceil = \frac{k}{2} + 1$, then: $C_I(\langle s, o\rangle) = 2\left(\frac{k}{2}+1\right)^2 - 2\left(\frac{k}{2}+1\right)(k+1) + k^2 + k = \frac{1}{2}k^2$

Both $p = \frac{k}{2}$ and $p = \frac{k}{2} + 1$ allow to find out optimal configurations, being $\frac{1}{2}k^2$ the number of paths pass through the internal link of the switch.

#### 7.3.4.2 The port $o_s$ is one of the $q$ backward ports

Firstly, the value of $C_I(\langle s, o\rangle)$ must be obtained.

$$\begin{aligned} C_I(\langle s, o\rangle) &= p^2 + q(q-1) + p = \\ &= p^2 + (k-p)(k-p-1) + p = \\ &= 2p^2 + 2p(1-k) + k^2 - k \end{aligned}$$

To obtain the value of $p$ that minimizes $C_I(\langle s, o\rangle)$, it is necessary to realize the second derivative of $C_I(\langle s, o\rangle)$ with respect to $p$.

$$C'_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C_I(\langle s, o\rangle) = 4p + 2(1-k)$$

Then, the second derivative $C'_I(\langle s, o\rangle)$ is equal to zero to determine the critical points.

$$\begin{aligned} C'_I(\langle s, o\rangle) &= 0 \\ 4p + 2(1-k) &= 0 \\ p &= \frac{k-1}{2} \end{aligned}$$

To know if $p = \frac{k-1}{2}$ is a critical point, the second derivative of $C_I(\langle s, o\rangle)$ with respect to $p$ must be used at the point $p = \frac{k-1}{2}$. If it takes a negative value, or positive, the function will have a maximum, or minimum, at the point $p = \frac{k-1}{2}$, respectively. Otherwise, the point $p = \frac{k-1}{2}$ may be an inflection point.

$$C''_I(\langle s, o\rangle) = \frac{\partial}{\partial p} C'_I(\langle s, o\rangle) = 4$$

As $C''_I(\langle s, o\rangle)$ is always positive, the function $C_I(\langle s, o\rangle)$ has a minimum at $p = \frac{k-1}{2}$. However, $k$ is even, consequently, $\frac{k-1}{2}$ is not an integer number, so it is necessary to check out the values of $C_I(\langle s, o\rangle)$ at two nearby points of $\frac{k-1}{2}$.

- If $p = \left\lfloor \frac{k-1}{2} \right\rfloor = \frac{k}{2} - 1$, then $C_I(\langle s, o\rangle) = 2\left(\frac{k}{2} - 1\right)^2 - 2\left(\frac{k}{2} - 1\right)(1-k) + k^2 - k = \frac{1}{2}k^2$
- If $p = \left\lceil \frac{k-1}{2} \right\rceil = \frac{k}{2}$, then $C_I(\langle s, o\rangle) = 2\left(\frac{k}{2}\right)^2 - 2\left(\frac{k}{2}\right)(1-k) + k^2 - k = \frac{1}{2}k^2$

Both $p = \frac{k}{2} - 1$ and $p = \frac{k}{2}$ allow to determine optimal configurations, being $\frac{1}{2}k^2$ the number of paths passing through the internal link of the switch.

#### 7.3.4.3 Summing up

- $p = \frac{k}{2}$ and $p = \frac{k}{2} + 1$ allow to determine optimal configurations if $o_s$ is one of the $p$ backward ports of the $T$-switch.
- $p = \frac{k}{2} - 1$ and $p = \frac{k}{2}$ allow to determine optimal configurations if $o_s$ is one of the $q$ backward ports of the $T$-switch.

Therefore, it is possible to find out optimal configurations if $p = \frac{k}{2}$ with independence of the identifier of the $T$-switch of type $\sigma d$, and the number of paths passing through the internal link is $\frac{1}{2}k^2$. □

It must be noticed that the set $\mathcal{S}^{\sigma d}$ includes optimal configurations for a $T$-switch of type $\sigma d$ with independence of its switch identifier, but $\mathcal{S}^{\sigma d}$ does not contain all the optimal configurations because there exist other configurations that minimize the use of the internal link depending on the switch identifier.

The configuration shown in the Example 1.11 are equally optimal in this case.

### 7.3.5 Type $\sigma e$ configuration of switch

***Proposition 1.61*** *Let $\mathcal{S}^{\sigma e}$ be the set of configurations that minimize the use of the internal link for a $T$-switch of type $\sigma e$, then*

$$\mathcal{S}^{\sigma e} = \mathcal{S}_b = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

*and there are no backward connections using the internal link.*

**Proof:** According to Proposition 1.29, in a $T$-switch of type $\sigma e$ backward connections are only established, so all the configurations of $\mathcal{S}_b$ are optimal. Moreover, since these connections exist between all the pairs of ports $l, l'$ such that $(l, l') \in \mathcal{R}_b$, the configurations in $\mathcal{S}_b$ are the unique optimal configurations according to the Proposition 1.30. □

The configurations shown in the Example 1.10 are equally optimal in this case.

### 7.3.6 Type $\sigma f$ configuration of switch

Since turnaround connections are exclusively established in the last–stage switches, half the switch ports are only used, that is, $k$ ports. Moreover, as the number of ports of an internal switch is $k$, an obvious and optimal configuration is that using only the ports of one of the internal switches to connect to $(n-2)$–stage switches. Two choices are possible: $(p = k, q = 0)$ and $(p = 0, q = k)$ are optimal configurations; the internal link connecting the internal $\alpha$ and $\beta$ switches, are never used. In a more formal way:

***Proposition 1.62*** *Let $\mathcal{S}^{\sigma f}$ be the set of configurations that minimize the use of the internal link for a $T$-switch of type $\sigma f$, then*

$$\mathcal{S}^{\sigma f} = \{\mathcal{C} \in \mathcal{V} \mid \mathcal{B} = \emptyset\ o\ \mathcal{F} = \emptyset\}$$

*and there are no connections using the internal link.*

**Proof:** If $\mathcal{B} = \emptyset$ then $card(\mathcal{B}) = 0$, and according to Definition 1.8 and Proposition 1.2 ($card(\mathcal{C}) = k$), it is verified $card(\mathcal{F}) = k$. This means that all the forward ports are connected to $\mathcal{C}$, and then all the backward ports are connected to $\mathcal{C}^C$. As to obtain a configuration $\mathcal{T}$ for a $T$-switch it is necessary to use $\mathcal{C}$ and $\mathcal{C}^C$, all the backward links for a $T$-switch are derived from all the links in a single internal switch.

Consequently, if $\mathcal{F} = \emptyset$, then $card(\mathcal{F}) = 0$, and according to Definition 1.8 and Proposition 1.2, $card(\mathcal{B}) = k$. Similar to the previous case, all the backward links of the $T$-switch are derived from a single internal switch.

It is obvious that if all the backward links for a $T$-switch are obtained with a unique internal switch, no turnaround connection will use the internal link existing in a $T$-switch. □

### 7.3.7 $s$–stage switch configuration, $0 \leq s < n-1$

***Proposition 1.63*** *Let $\mathcal{S}^{\sigma}$ be the set of configurations that minimize the use of the internal link for a $s$–stage $T$-switch, where $0 \leq s < n-1$, then*

$$\mathcal{S}^{\sigma} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

**Proof:** According to propositions 1.55, 1.56, 1.59 and 1.61,

$$\mathcal{S}^{\sigma a} = \mathcal{S}^{\sigma b} = \mathcal{S}^{\sigma c} = \mathcal{S}^{\sigma e} = \{\mathcal{C} \in \mathcal{V} \mid \forall l \in \mathcal{C},\ \exists l' \in \mathcal{C} \mid (l, l') \in \mathcal{R}_b\}$$

and by Proposition 1.60,

$$\mathcal{S}^{\sigma d} = \{\mathcal{C} \in \mathcal{V} \mid card(\mathcal{B}) = card(\mathcal{F}) = k/2\}$$

It is possible to demonstrate that the sets $\mathcal{S}^{\sigma a}$, $\mathcal{S}^{\sigma b}$, $\mathcal{S}^{\sigma c}$, and $\mathcal{S}^{\sigma e}$ are subsets of $\mathcal{S}^{\sigma d}$. Therefore, this set provides configurations that optimize the use of the internal link for every switch $\langle s, o \rangle$, where $0 \leq s < n-1$. Moreover, these configurations are the unique optimal in this case: there exist other configurations that minimize the use of the internal link in an isolated way, but they are optimal for all switches $\langle s, o \rangle$, where $0 \leq s < n-1$. □

### 7.3.8 Configuration of switch

# 8 Related Work

In this section we review existing proposals of high-radix switches in the literature, which are mainly focused on solving the scaling problems from traditional switch designs.

The switch YARC is the switch high-radix used by the Cray BlackWidow [SAKD06]. BlackWidow uses a folded-Clos topology. YARC is a hierarchical switch with the internal organization defined in [KDTG05], which states that increasing the level of the switch is the most efficient strategy to increase the bandwidth of the switch. The authors claim that for a specific number of signals in the chip is preferable to form more links (giving fewer signals per link) that implement fewer links at the expense of allocating more signals to each link.

The paper makes clear that the traditional designs of switches with fewer ports can not be adapted to the new switches with many ports, which refer as high-radix, because the traditional centralized organization does not scale properly. The report proposes a new hierarchical switch architecture that improves the performance of traditional switches with few number ports. The new architecture distributes and simplifies the control logic and reduces the communication lines within the chip. The result is a viable switch, but with a very low yield, due to the problem of head of line blocking. Adding buffers in the crossing points at the crossbar can eliminate the HOL blocking decoupling the input and output of the switch.

The *folded-Clos* topology doubles in cost to the butterfly multistage topology with the same capacity and higher latency than a butterfly bidirectional multistage network. This is because in the butterfly multistage packets are sent through intermediate stages of the network before being routed to their final destination. The topology *flattened butterfly* [KDA07] is an alternative to the folded-Clos topology. Improvements in the signal technology have allowed longer cable lengths. Based on the conventional butterfly multistage topology, the switches of the intermediate stages are replaced by high-radix switches and they are connected with new longer cables. As a result, it reduces the number of jumps for intermediate switches, which decreases the latency. In addition, fewer switches means less cost.

Overlooking the advantages of high-radix switches, *Dragonfly* [KDSA08] topology proposes to increase the effective radix of the switch using a set of switches interconnected by a subnet. The set operates as a virtual switch within a hierarchical network. The hierarchy has three levels: the lower level which nodes connect to the virtual switches at the intermediate level, there is a local subnet switches interconnecting the intermediate level, and finally there is a global subnet to interconnect virtual switches.

The three topologies considered that high-radix switches are homogeneous elements. All exploit the number of available ports, but none discusses the possible variation of network performance based on which ports are used to connect the different components of the network.

The *Partitioned Crossbar Input Queued (PCIQ)* switch [MFD+06] is a more recent proposal for the internal organization of high-radix switches. It is based on replacing the central crossbar by several *internal* crossbars improving the readability of the buffers without increasing the hardware cost. Each crossbar has a round-robin arbiter, which has a linear cost and logarithmic response time, as the radix of the switch increases.

Moreover, the switch implements RECN [GFD+06] as congestion control mechanism. Thus, the switch completely eliminates the HOL blocking at both switch and network levels, thus the maximum productivity is not consistent with traffic. This architecture has a lower cost than the hierarchical architecture proposed in [KDTG05].

Additionally, for instance, a high-radix switch based network is required in order to exploit the computing power of a system made of Merrimac stream processors [DHE+03]. Also new communication technologies like Proximity Communication (PxC) from Sun Microsystems are tied to high-radix architectural designs [EGF+08].

Regarding the alternative for building high-radix switches using low-radix switches, the Oracles's Sun Blade 6048 Infiniband QDR Switched Network Express Module (NEM) [Sun10] has already implemented this strategy, offering the ability to connect up to 12 dual-node blades in a single shelf. Each NEM provides 12 connections from each of the 36-port Mellanox's InfiniScale IV switches. A total of 24 connections are used to communicate with the two compute nodes on each of the dual-node server blades, with 9 ports used to connect the two switches together. The 30 remaining ports (15 per switch chip) are used as links to either other NEMs, or to external switches.

To the best of our knowledge, there are currently no formal studies published on determination of switch-level connection pattern.

This way to get high-radix switches requires a thorough study to find out the best switch-level connection pattern that we have done in this report and we are presenting from Section 2.2.

# References

[aa89] H.J. Siegel at al. Using the multistage cube network topology in parallel supercomputers. In *Proceedings of the IEEE, vol. 77, pp. 1932–1953*, December 1989.

[DeH90] André DeHon. Technical report: Fat–tree routing for transit. Technical report, 1990.

[DHE+03] W. J. Dally, P. Hanrahan, M. Erez, T. J. Knight, F. Labonte, J-H A., N. Jayasena, U. J. Kapasi, A. Das, J. Gummaraju, and I. Buck. Merrimac: Supercomputing with streams. In *SC'03*, Phoenix, Arizona, November 2003.

[DYN03] José Duato, Sudhakar Yalamanchili, and Lionel Ni. *Interconnection networks. An engineering approach.* Morgan Kaufmann Publishers Inc., 2003.

[ea91] E. Bakker et al. Linear interval routing algorithms review 2., 1991.

[EGF+08] Hans Eberle, Pedro J. Garcia, José Flich, José Duato, Robert Drost, Nils Gura, David Hopkins, and Wladek Olesinski. High-radix crossbar switches enabled by proximity communication. In *SC '08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, pages 1–12, Piscataway, NJ, USA, 2008. IEEE Press.

[GAG+03] M. Gusat, F. Abel, F. Gramsamer, R. Luijten, C. Minkenberg, and M. Verhappen. Stability degree of switches with finite buffers and non-negligible round-trip time. *International Conference on Computer, Communication and Networking*, 27(5–6):243–252, 2003.

[GFD+06] P. J. García, J. Flich, J. Duato, I. Johnson, F. J. Quiles, and F. Naven. Efficient, scalable congestion management for interconnection networks. *IEEE Micro*, 26, 2006(5):52–66, September 2006.

[GGG+07] Crispín Gómez, Francisco Gilabert, María E. Gómez, Pedro Lopez, and José Duato. Deterministic versus adaptive routing in fat-trees. Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[GL73] L. Rodney Goke and G. J. Lipovski. Banyan networks for partitioning multiprocessor systems. In *ISCA '73: Proceedings of the 1st annual symposium on Computer architecture*, pages 21–28, New York, NY, USA, 1973. ACM.

[GLD05] M. E. Gomez, P. Lopez, and J. Duato. A memory-effective routing strategy for regular interconnection networks. In *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Papers*, page 41.2, Washington, DC, USA, 2005. IEEE Computer Society.

[ITR10] International Technology Roadmap for Semiconductors: 2010 Update, 2010. `www.itrs.net/Links/2010ITRS/Home2010.htm`.

[KDA07] John Kim, William J. Dally, and Dennis Abts. Flattened butterfly: a cost-efficient topology for high-radix networks. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 126–137, New York, NY, USA, 2007. ACM.

[KDSA08] John Kim, Wiliam J. Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable Dragonfly topology. In *ISCA '08: Proceedings of the 35th Annual International Symposium on Computer Architecture*, pages 77–88, Washington, DC, USA, 2008. IEEE Computer Society.

[KDTG05] John Kim, William J. Dally, Brian Towles, and Amit K. Gupta. Microarchitecture of a high-radix router. *SIGARCH Comput. Archit. News*, 33(2):420–431, 2005.

[KS83] C. Kruskal and M. Snir. The performance of multistage interconnection networks for multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, December 1983.

[Lei85] C. E. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing. *IEEE Transactions on Computers*, 34(10):892–901, 1985.

[Lei92] F.T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, 1992.

[LM88] Charles Leiserson and Bruce M. Maggs. Communication-efficient parallel algorithms for distributed random-access machines. *Algorithmica*, 3:53–77, 1988.

[lWF80] Chuan lin Wu and Tse-Yun Feng. On a class of multistage interconnection networks. *IEEE Trans. Computers*, 29(8):694–702, 1980.

[MAM+05] Cyriel Minkenberg, Francois Abel, Peter Muller, Raj Krishnamurthy, Mitchell Gusat, and B. Roe Hemenway. Control path implementation for a low-latency optical HPC switch. In *Proceedings of the 13th Symposium on High Performance Interconnects*, HOTI'05, pages 29–35, Washington, DC, USA, 2005. IEEE Computer Society.

[MFD+06] G. Mora, J. Flich, J. Duato, P. López, E. Baydal, and O. Lysne. Towards an efficient switch architecture for high-radix switches. In *ANCS '06: Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems*, pages 11–20, New York, NY, USA, 2006. ACM.

[MG07] Cyriel Minkenberg and Mitchell Gusat. Speculative flow control for high-radix datacenter interconnect routers. *Parallel and Distributed Processing Symposium, International*, 0:1–10, 2007.

[NGM97] L.M. Ni, Y. Gui, and S. Moore. Performance evaluation of switch-based wormhole networks. 8(5):462–474, May 1997.

[Pat81] J.H. Patel. Performance of processor-memory interconnections for multiprocessors. *IEEE Transactions on Computers*, C-30(10):771–780, October 1981.

[SAKD06] Steve Scott, Dennis Abts, John Kim, and William J. Dally. The BlackWidow high-radix Clos network. *SIGARCH Comput. Archit. News*, 34(2):16–28, 2006.

[SJS08] Frank Olaf Sem-Jacobsen and Tor Skeie. Maintaining quality of service with dynamic fault tolerance in fat-trees. In *HiPC*, pages 451–464, 2008.

[Sun10] Sun datacenter Infiniband switch 36, Sun datacenter Infiniband switch 72, Sun datacenter Infiniband switch 648: Architecture and deployment, April 2010.

[top10] Top500 Supercomputer Site, 2010. `www.top500.org`.

[WPM03] Hangsheng Wang, Li-Shiuan Peh, and Sharad Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 36, Washington, DC, USA, 2003. IEEE Computer Society.

# A Multistage Interconnection Networks

This appendix introduces multistage interconnection networks and presents some of their basic aspects, which will be used in the rest of this report. Fat–trees networks receive an special attention because they have been chosen for carrying out this study, due to they are one of the most used interconnection network in the supercomputers market.

## A.1 Multistage interconnection networks

Multistage Interconnection Networks (MINS) connect input devices to output devices through a number of switch stages, where each switch is a crossbar network. The number of stages and the connections patterns between two adjacent stages determine the routing capability of the networks (Figure 23). Other characteristics considered in MINs are the number of switches and the switch radix, the number of stages, message average latency, path diversity, routing algorithm, or link direction (i.e., unidirectional, bidirectional) [KS83].

In practice, all the switches are identical, thus amortizing the design cost. When switches have the same number of input and output ports, MINs also have the same number of input and output ports. Since there is a one–to–one correspondence between inputs and outputs, the connections between stages are also called permutations.

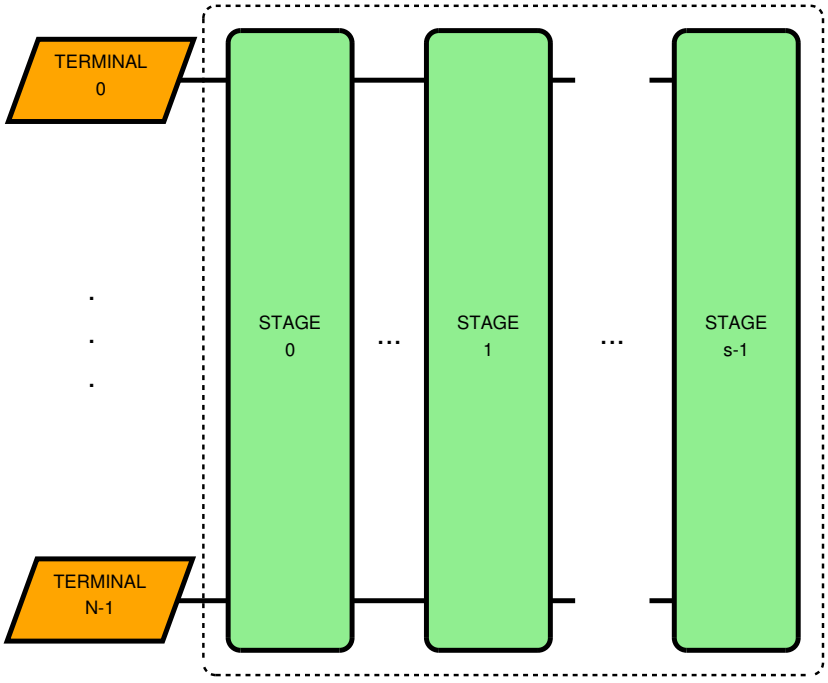



Figure 23: A multistage network with $N$ terminals and $s$ stages.

Depending on the availability of paths to establish new connections, MINs have been traditionally divided into three classes [DYN03]:

1. *Blocking.* A connection between a free input/output pair is not always possible because of conflicts with the existing connections. Typically, there is a unique path between every input/output pair, thus minimizing the number of switches and stages. However, it is also possible to provide multiple paths to reduce conflicts and increase fault tolerance. These MINs have been often implemented because of its simple design and easy control. These blocking

networks are also known as multipath networks. Omega network is an example of blocking network (Figure 24a).

2. *Nonblocking.* Any input port can be connected to any free output port without affecting the existing connections. Nonblocking networks have the same functionality as a crossbar. They require multiple paths between every input and output, which in turn leads to extra stages. Therefore, all permutations are supported. However, they are expensive and some require more complex control logic. The best–known example is the Clos network (Figure 24b).

3. *Rearrangeable.* Similarly to nonblocking networks, any input port can be connected to any free output port. However, the existing connections may require rearrangement of paths. These networks also require multiple paths between every input and output, but the number of paths and the cost are smaller than in the case of nonblocking networks. The best-known example of a rearrangeable network is the Beneš network (Figure 24c).

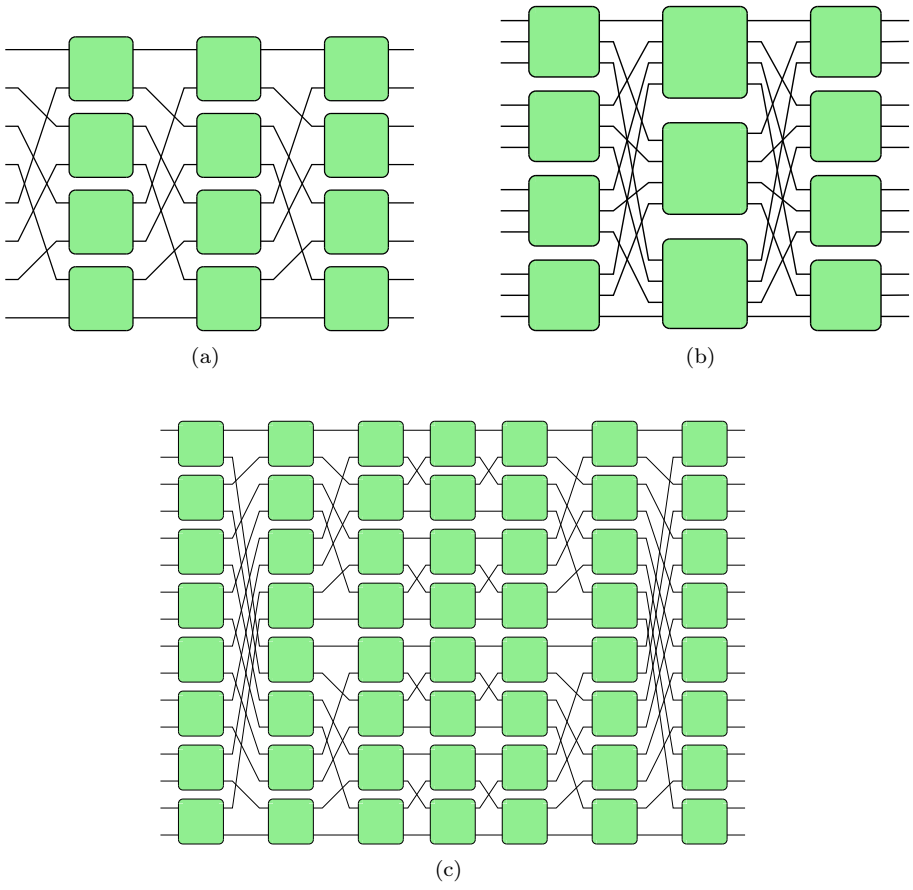



Figure 24: MINs: (a) Omega 8×8, (b) 3–stage Clos, (c) Beněs 8×8.

Other aspects that make the difference between MINs are the required number of stages and the permutation used to connect two adjacent stages. Given a connection pattern, the number of stages depends on the number of switch ports. Some commonly known permutations are perfect–shuffle, bit–reversal or butterfly (Section A.3).

Many of the known MINs, such as Omega, flip, cube, butterfly, and baseline, belong to the class of Delta networks [Pat81] and have been shown to be topologically and functionally equivalent [lWF80]. A good survey of those MINs can be found in [aa89].

## A.2 Preliminary definitions

In this section some basic concepts of MINs are introduced to make easier the description below. The notation used is based on that used in other studies about these networks, and it aims to provide thoroughness to the study to be held later.

In what follows we take into account the following considerations:

- The input/output terminals are the computing nodes.
- All the switches have the same number of ports.

### A.2.1 Notation

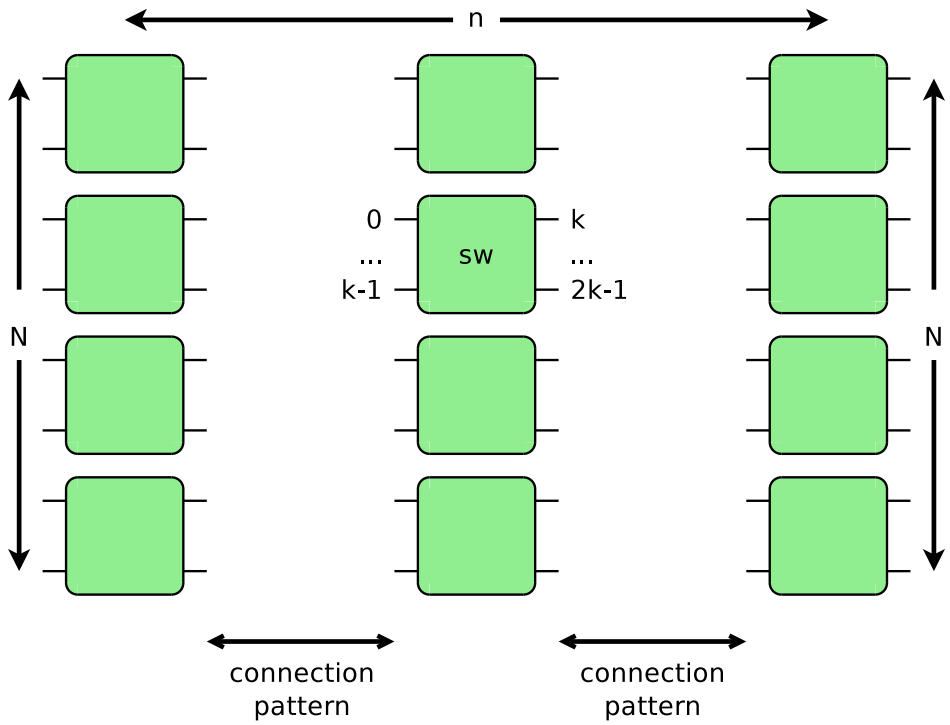



Figure 25: Assumed notation for a MIN with $N$ terminals and $k \times k$ switches.

We have assumed the following notation (Figure 25):

- $N$ is the total number of terminals (or processing nodes).
- $k$ is the switch arity, or number of ports that connect to terminals/switches in the previous stage and switches in the next stage (if available). Hence, the total number of ports of a $k \times k$ switch is $2k$. The ports faced to the previous stage are numbered from 0 to $k-1$, and the ports connected with the switches in the next stage are labeled from $k$ to $2k-1$.
- Every switch port has an associated global identifier inside the stage, $L = l_{n-1} \ldots l_0, 0 \leq l_i < n$, apart from the internal identifier inside the switch. Both identifiers are related by the connection pattern between stages.
- $n$ is the total number of stages, where $n = \log_k N$.
- $h$ is the terminal identifier ($0 \leq h < N$). It consists of a string of $n$ digits ($h_{n-1} \ldots h_1 h_0$), $0 \leq h_i < k$. $\mathcal{H}$ is the set whose members are the terminals of the MIN, verifying card($\mathcal{H}$)=$N$, where card is the cardinality of sets.

- $\langle s, o \rangle$ is a tuple that identifies uniquely a switch, where $s$ refers to the stage ($0 \leq s < n$), and $o = o_{n-2}, \ldots, o_1, o_0$ indicates the position of the switch inside the stage, where $0 \leq o_i < k$ and $0 \leq i < n-1$.

## A.3 Connection pattern

Several connection patterns have been proposed to interconnect two adjacent stages of the MIN and the processing nodes with the first and/or the last stage. These patterns correspond to certain well–known permutations. Some of those permutations are defined below.

- **Perfect shuffle**

  The *perfect k–shuffle* permutation $\sigma^k$ is defined by

  $$\sigma^k(x_{n-1}x_{n-2}\ldots x_1x_0) = x_{n-2}x_{n-3}\ldots x_1x_0x_{n-1}$$

  It performs a cyclic shifting of the digits $x_i$, $0 \leq i < n$, to the left for one position. Every $x_i$ consists of $k$ bits.

  The *inverse perfect shuffle* does the opposite to the *perfect shuffle* permutation,

  $$\sigma^{k^{-1}}(x_{n-1}x_{n-2}\ldots x_1x_0) = x_0x_{n-1}\ldots x_1$$

- **Digit reversal**

  The *digit reversal* permutation $\rho^k$ is defined by

  $$\rho^k(x_{n-1}x_{n-2}\ldots x_1x_0) = x_0x_1\ldots x_{n-2}x_{n-1}$$

  This permutation is usually referred to as *bit reversal*. It performs a swapping between digits $x_i$ and $x_{n-1-i}$ where $0 \leq i < n$. Every $x_i$ consists of $k$ bits.

- **Butterfly**

  The $i$th $k$–ary *butterfly* permutation $\beta_i^k$ where $0 \leq i < n$, is defined by

  $$\beta_i^k(x_{n-1}x_{n-2}\ldots x_1x_0) = x_{n-1}\ldots x_{i+1}x_0x_{i-1}\ldots x_1x_i$$

  It interchanges the least significant digit with the $i$th digit. Every $x_i$ consists of $k$ bits. Note that $\beta_0^k$ is also called identity permutation.

Additionally, the connection pattern is used to name the MINs. So, we will referred to butterfly MINs, perfect–shuffle MINs and bit-reversal MINs using butterfly, perfect–shuffle and bit–reversal permutations, respectively.

## A.4 Unidirectional MINs

In an unidirectional MIN, the channels and switches are unidirectional. Half the terminals are located at one side, and the another half at the opposite side. The MIN is in the middle. Communication is allowed only in one direction. Figure 26 shows an unidirectional switch with $k$ input ports and $k$ output ports.

All paths in an unidirectional MIN go across all the stages. So all paths have the same length. Figure 27 illustrates the unidirectional MIN topology for $N = 8$ nodes built with 2×2 switches.

Banyan networks are a class of MINs with the property that there is a unique path between any pair of source and destination [GL73]. A Delta network is a subclass of banyan networks, which

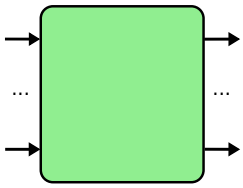



Figure 26: Unidirectional switch.

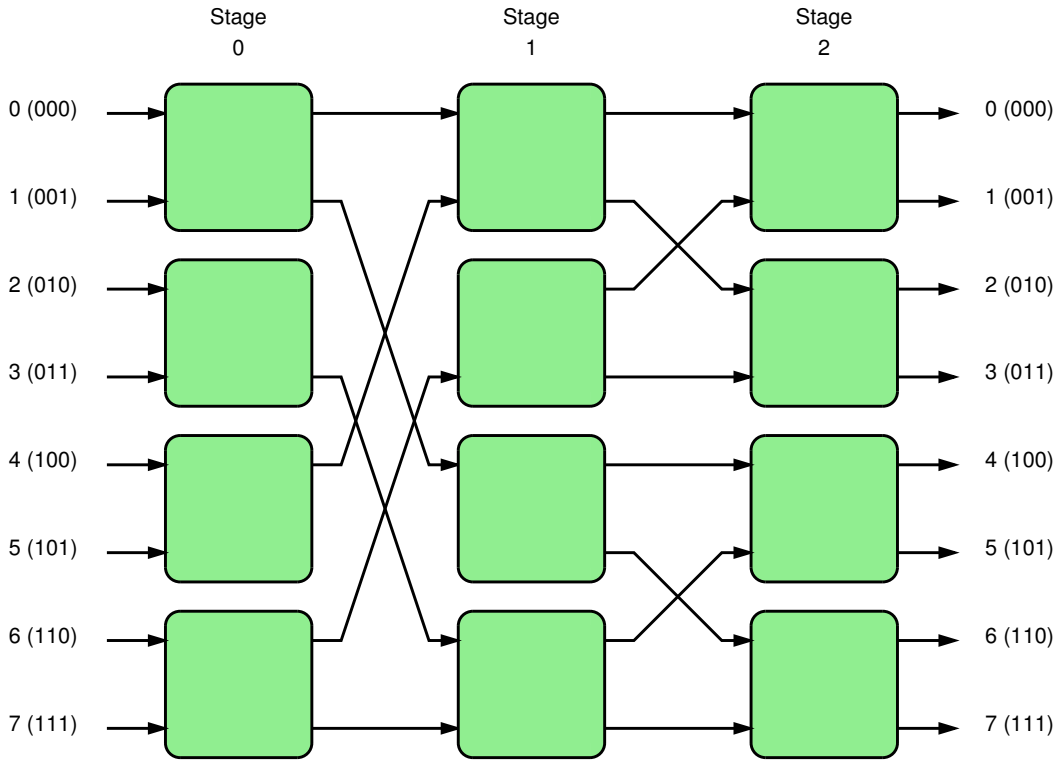



Figure 27: Unidirectional MIN built with 2×2 switches.

is constructed from identical $k \times k$ switches in $n$ stages, where each stage contains $N/k$ switches. Many of the known MINs, such as Omega, flip, cube, butterfly, and baseline, belong to the class of Delta networks [Pat81] and have been shown to be topologically and functionally equivalent [lWF80]. A good survey of those MINs can be found in [aa89].

The most popular routing algorithm in unidirectional MINs is self–routing.

### A.4.1 Self–routing algorithm

The self–routing algorithm is a deterministic routing algorithm for MIN networks. The routing decision is based on the destination address. The paths are established in a distributed way by using routing tags [Pat81].

The routing function determines the output port taking into account which digit is the least significant one at the $i^{th}$ stage. In unidirectional MINs with $k \times k$ switches, if the value of the digit is $i$ $(0 \leq i < k)$, the packet will be forwarded by the output port $k + i$.

For a $k \times k$ switch, there are $k$ output ports. If the value of the corresponding routing tag is $i$ $(0 \leq i < k)$, the corresponding packet will be forwarded via port $i$. For an $n$–stage MIN, the routing tag is $T = t_{n-1} \ldots t_1 t_0$, where $t_i$ controls the switch at the $i^{th}$ stage.

Figure 28 shows the paths followed by one packet from node 0010 to node 1010, and another one from node 0111 to node 1101 in a $N = 16$ butterfly unidirectional MIN with 16×16 switches. The routing tags are 0101 and 1011, respectively. In the first case, the packet is forwarded by the output port 3 in the stages $s_0$ and $s_2$. However, it is routed by the output port 2 in the stages $s_1$ and $s_3$.

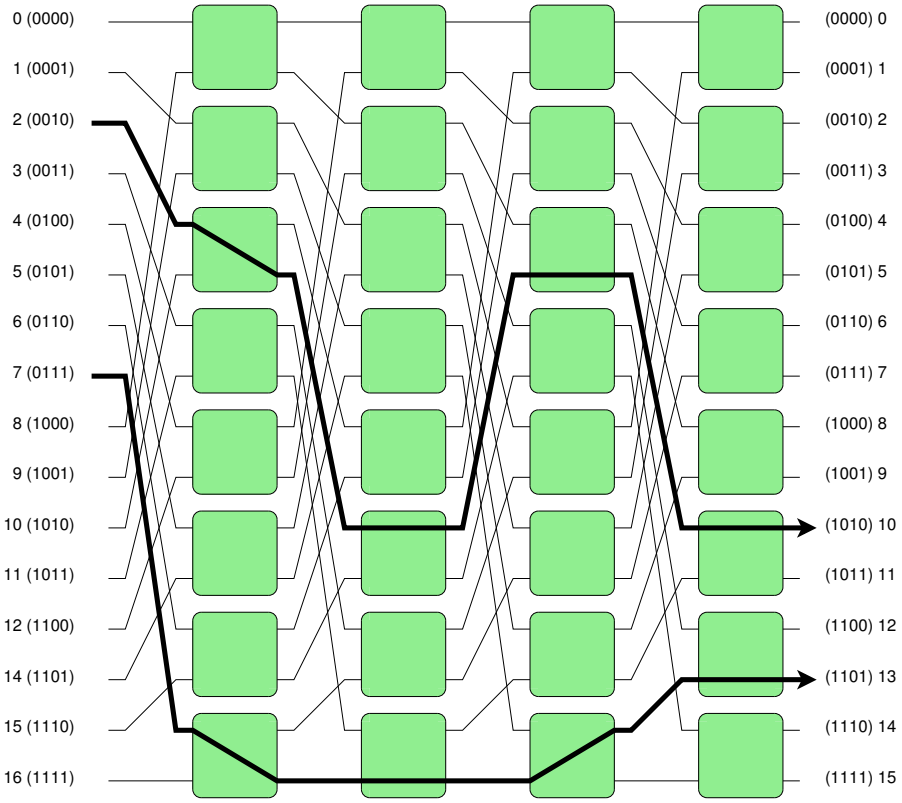
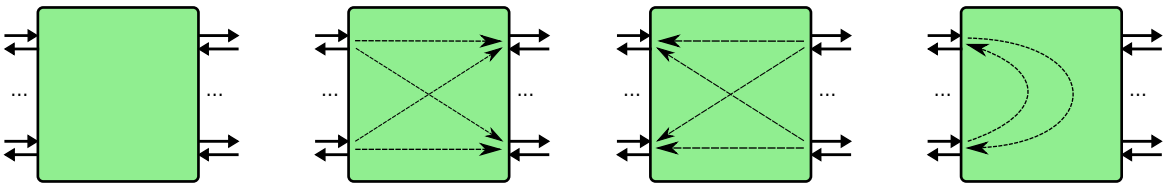



Figure 28: Paths selected by the self–routing algorithm in a $N = 16$ butterfly MIN.

## A.5 Bidirectional MINs

Bidirectional MINs (BMINs) are composed of bidirectional switches and switch ports are connected by bidirectional channels (Figure 29(a)). This means that packets can be transmitted simultaneously in opposite directions between neighboring switches. A bidirectional channel is built by joining two unidirectional channels, in opposite directions. In such a way, the bidirectional channel can transmit two packets at the same time in opposite directions between neighboring switches. The bidirectional switch performs three types of internal connections: forward, turnaround and backward connections. Figures 29(b,c,d)) depict the three possible internal connections. As turnaround connections between ports at the same side of a switch are possible, paths have different lengths.



(a) Bidirectional switch. (b) Forward connection. (c) Backward connection. (d) Turnaround connection.

Figure 29: Internal connections in a bidirectional switch.

The network terminals are not directly connected to the switches placed at the stage $j = \log_k N - 1$. Figure 30 shows a $N = 8$ BMIN built with 2×2 switches. Notice that switches of the last stage are not connected to the terminals as it occurs in unidirectional MINs.

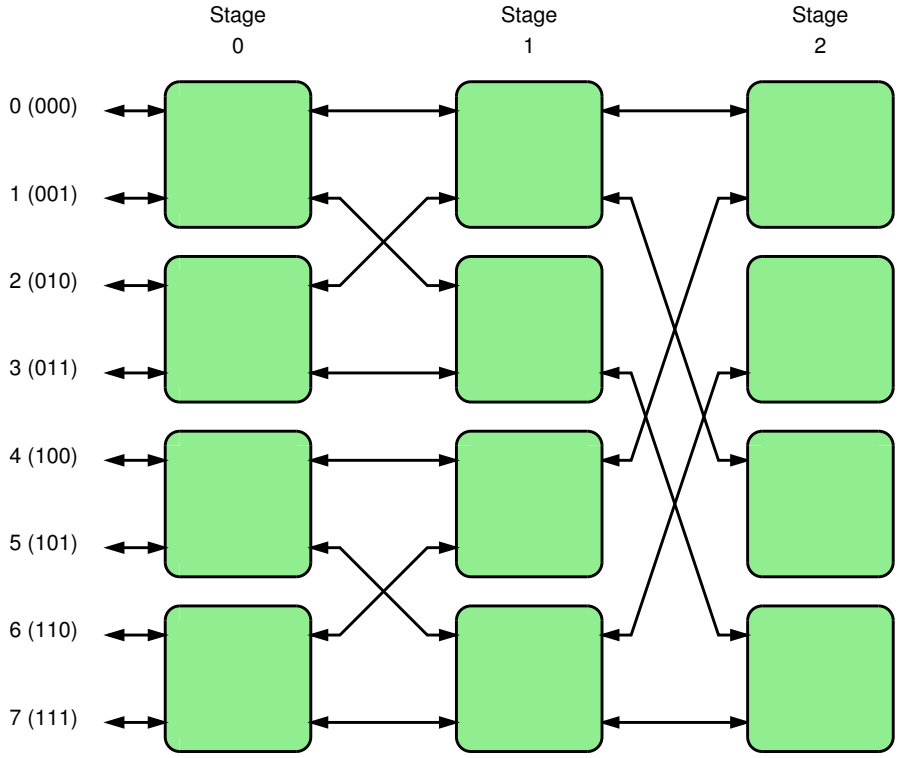



Figure 30: BMIN de $N = 8$ y $k = 2$.

Paths are established in BMINs by crossing stages in the forward direction, then establishing a turnaround connection, and finally crossing stages in the backward direction to the destination terminal. This is usually referred to as turnaround routing (Section A.5.1).

In a BMIN, the routing algorithm could select several minimal paths to send a packet from the source node $h$ to the destination node $h'$. Firstly, a path goes across stages in the forward direction. Each switch can select any of its $k$ output ports. However, once the turnaround connection is crossed, a single path is available up to the destination node.

According to the definition of the turnaround routing algorithm in the Section A.5.1, the turnaround connection is done in any switch at the stage $t$ ($t$ = FirstDifference$(h, h')$). So, there are $k^t$ switches where all the possible paths between $h$ and $h'$ could do the turnaround connection.

Turnaround routing algorithm is adaptive in the forward phase because it selects any subpath between the source node $h$ and the $k^t$ switches placed at the stage $t$. This feature balances the load in the network, and gives fault tolerant characteristics.

On the other hand, after doing the turnaround connection, there is an unique path to the destination node $h'$. For this reason, the turnaround routing algorithm is deterministic in the descending, or backward, phase, like self–routing is. If no fault tolerance policy is defined, when a switch/channel fails, then the packets could not reach the destination node $h'$. In [SJS08], the authors propose a strategy to provide fault tolerance in the descending phase in fat–trees.

A butterfly BMIN with turnaround routing can be viewed as a fat tree [Lei85]. In a fat tree, processors are located at leaves, and internal vertices are switches (Section A.6).

### A.5.1 Turnaround–routing algorithm

To send a packet from a source node $h$ to a destination node $h'$, it is first sent forward to the least common ancestor of both nodes. Then, the packet is turnaround at stage $t$ (the concrete switch does not matter), and it is sent backward to the destination.

The existing path between a source–destination pair that is obtained by the turnaround routing algorithm can be formalized as follows [NGM97]:

***Definition 1.25*** *A turnaround routing path between any source and destination pair must meet the following conditions:*

- *The path consists of a sequence of forward connections, one turnaround connection in the stage $s$, and backward connections.*
- *The number of forward connections is equal to the number of backward connections.*
- *To prevents redundant communication from occurring, no connection is allowed to use the same switch port as input and output port.*

***Definition 1.26*** *Given $h = h_{n-1}h_{n-2}\dots h_0$ and $h' = h'_{n-1}h'_{n-2}\dots h'_0$ two nodes, the FirstDifference(h,h′) function returns the stage $s$, ($0 \leq s < n$), where the turnaround connection occurs.*

$$\textit{FirstDifference}(h, h') = s \text{ if and only if } h_s \neq h'_s \text{ and } h_j = h'_j,\ \forall j \in [s+1, n-1]$$

Note, the FirstDifference($h$,$h'$) function returns the position where the first (leftmost) different digit appears between $h$ and $h'$.

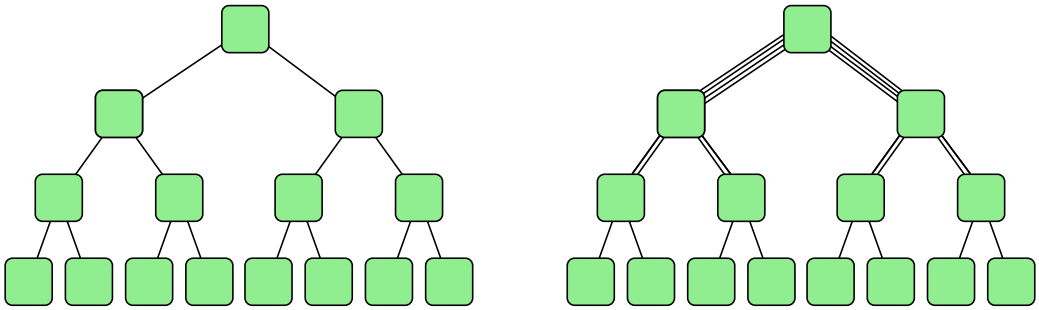
The turnaround routing is deadlock free and shortest path routing [NGM97]. As mentioned, in a BMIN there are multiple choices of the shortest path, which the turnaround routing may select, between a source and a destination. Specifically, if the turnaround connection is done at the stage $s$, there are $k^s$ valid paths of minimum length. After the packet is turnaround, there is only path to destination, for this reason, the turnaround routing is said to be adaptive and deterministic in the forward and backward direction, respectively. This feature makes possible to have load–balanced and to design fault tolerant networks.

## A.6 *Fat–tree* topology

A *fat–tree* is an indirect interconnection network based on a complete binary tree. Unlike the traditional notion of a tree, where all branches are similar, fat–trees are more like real trees in that they get thicker closer to the root [Lei85].

A binary tree retains the capacity of their channels while fat–tree increases the capacity of the channels as it approaches the root (Figure 31). The processing nodes are located at the leaves of the fat–tree. Each node of the fat–tree corresponds to a switch. Going upwards in the fat–tree, the channels capability increases, but complexity and hardware cost do proportionally. The capacity is determined by the amount of available hardware. This means that a fat–tree topology is also parameterizable in the channel bandwidth.

Routing packets in a fat–tree is quite easy, since there is a unique minimum path between every pair of computing nodes. A message going from node $h$ to node $h'$ goes up the tree to their least common ancestor and then back down according to the least significant bits of $h'$. Note that at any



(a) Binary tree. (b) Fat–tree.

Figure 31: Binary tree and fat-tree.

node of the tree, there are several choices for the routing of a packet. In such cases, the routing algorithm may select one of the channels to distribute the load minimizing the network congestion.

Fat–trees have many desirable properties. The universality theorem proposed by Leiserson states that for any given amount of communications hardware, a fat–tree build from that amount of hardware can simulate every other network built from the same amount of hardware, using only slightly more time (a polylogarithmic factor greater) [Lei85].

The number of switch ports (or switch radix) in the fat–tree increases as going up the tree to the root. This makes unfeasible the physical implementation of the switches. For this reason, Leiserson proposed alternative implementations using switches of fixed radix [LM88]. In Figure 32, the organization of a fat–tree is showed. It is possible to note how the channel capability increases further from the leaves.

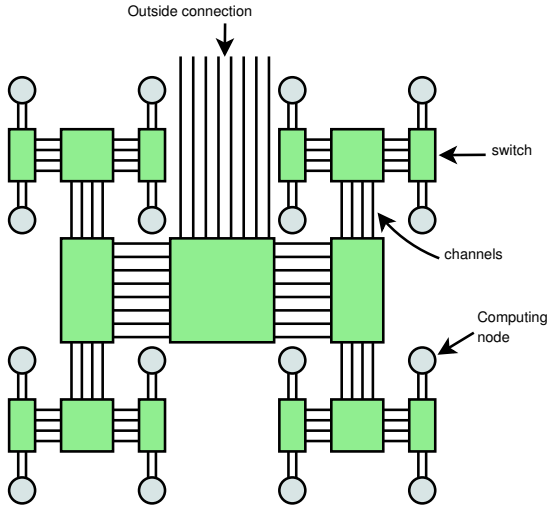



Figure 32: Organization of a fat–tree.

DeHon [DeH90] studied the implementation limitations such wiring, packing complexity and fault tolerant schemes.

The fat–trees are currently the preferred topology for supercomputers [4] like: TianHe-1A at NSC (China), Roadrunner at LANL (USA), and JUGENE at FZJ (Germany), among others.

[4] According to the November 2010 Top500 Supercomputing list at www.top500.org.

### A.6.1 $k$–ary $n$–tree topology

The $k$–ary $n$–tree network topology belongs to the family of fat–trees and it is derived from a concrete class of MINs: the $k$–ary $n$–butterflies (or $k$–ary $n$-flies) [Lei92]. A $k$–ary $n$–fly MIN is obtained by applying the $\beta_i^k$ permutation, $0 \leq i < n$, to obtain the connection patterns between stages.

The $k$–ary $n$–tree connect $N$ nodes using $nk^{n-1}$ switches. Two switches $\langle s, o_{n-2} \dots o_0 \rangle$ and $\langle s', o'_{n-2} \dots o'_0 \rangle$ are connected with a channel if $s' = s + 1$ and $o_i = o'_i \quad \forall\ i \neq s$. Moreover, there is a channel between the switch $\langle 0, o_{n-2} \dots o_0 \rangle$ and the processing node $h = h_{n-1} \dots h_0$ if $o_i = h_{i+1}, 0 \leq i < n-1$.

Figure 33 illustrates a 2–ary 4–tree MIN network for 16 processing nodes with 4 stages and 2×2 switches.

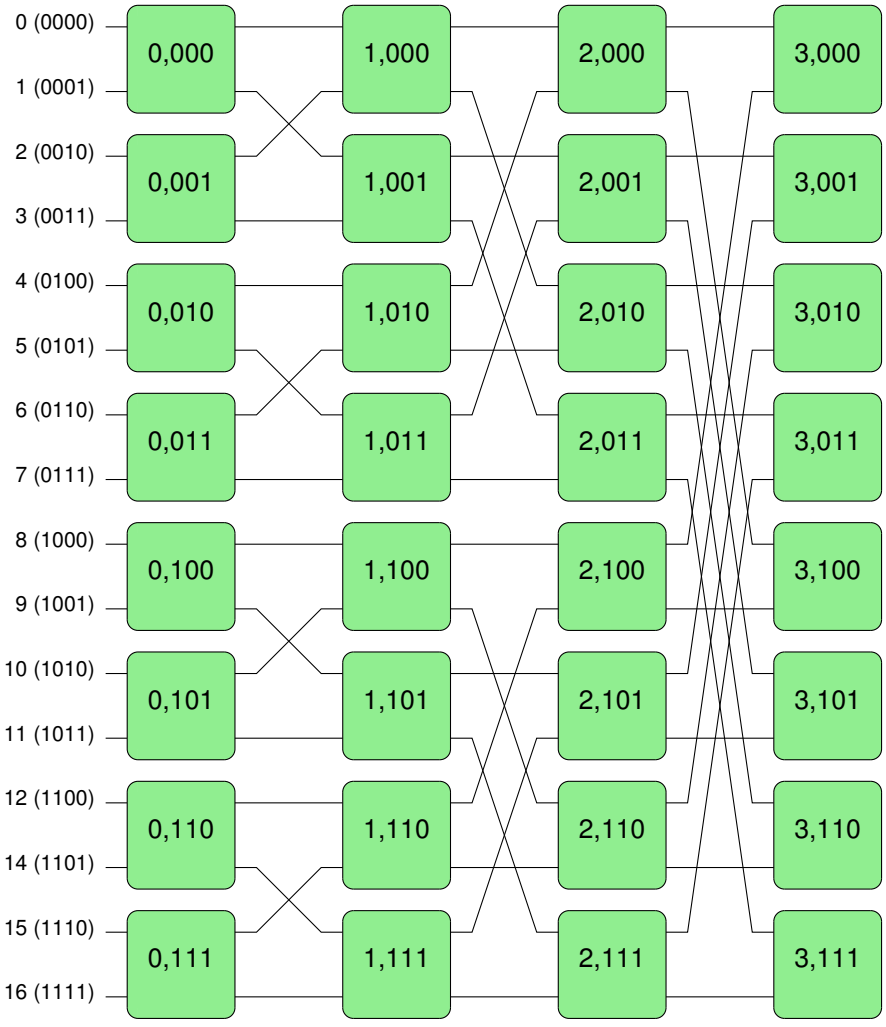



Figure 33: 2–ary 4–tree MIN network.

## A.7 Load–balanced routing algorithm

The routing algorithm is the mechanism that determines the path that a message follows on the network to reach its destination, from its source node. Usually, there are multiple paths that can carry a message to its destination; among these paths we can find a set of minimal length paths. A good routing strategy seems to be the one that just uses the minimal paths. There are also other

aspects that are usually taken into account when designing routing algorithms, which we have already mentioned led to complex and sophisticated design methodologies for these algorithms.

Taking into account this, the load–balanced routing algorithm concept is introduced. Based on this definition, it would be possible to identify which routing algorithms distribute the generated paths between the network elements in a balanced way.

***Definition 1.27*** *Given an interconection network $I = G(C, N)$, the routing algorithm $R$ is said to be balanced, or $R$ fully distributes the paths generated in $I$, if all the switch channels in $C$ belonging to $I$ are crossed by the same number of paths.*

***Definition 1.28*** *Given a multistage interconnection network $I$, the routing algorithm $R$ is said to be balanced, or $R$ fully distributes the paths generated in $I$, if all the channels of the switches belonging to a given stage, are crossed by the same number of paths.*

For MIN networks there are adaptive routing algorithms that balance partially or fully the paths generated. However, adaptive algorithms arise some difficulties (e.g., out–of–order delivery, more complex implementation) that make deterministic routing algorithms be interesting for some applications.

It is possible to design deterministic routing algorithms, which would balance the paths generated. A simple strategy consists in assigning the output ports in each switch to the difference paths passing through the switches by means of a function that spreads the paths (i.e., load network) between their ports.

For example, let us suppose a $k \times k$ switch that belongs to a concrete stage in a BMIN. From that switch it would be possible to arrive at $m$ destination nodes: $d_0, d_1, \ldots, d_{m-1}$ ($m$ is a multiple of $k$). The following functions would distribute the destinations between the $k$ output ports of the switch:

- allocation of consecutive destinations to the same port. Let $m' = m/k$ be the number of destination nodes assigned to each output port. Hence,

  To the port $l_0$ it assigns destinations $d_0, d_1, \ldots, d_{m'-1}$

  To the port $l_1$ it assigns destinations $d_{m'}, d_{m'+1}, \ldots, d_{2m'-1}$

  ...

  To the port $l_{k-1}$ it assigns destinations $d_{(k-1)m'}, d_{(k-1)m'+1}, \ldots, d_{km'-1}$

- cyclic allocation of consecutive destinations to consecutive ports. Let $m' = m/k$ be the number of destination nodes assigned to each output port. Hence,

  To the port $l_0$ it assigns destinations $d_0, d_k, \ldots, d_{(m'-1)k}$

  To the port $l_1$ it assigns destinations $d_1, d_{k+1}, \ldots, d_{(m'-1)k+1}$

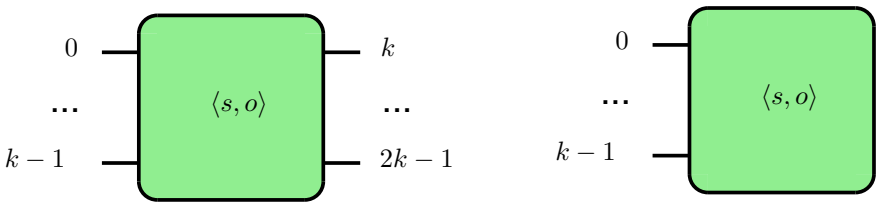
  ...

  To the port $l_{k-1}$ it assigns destinations $d_{k-1}, d_{2k-1}, \ldots, d_{m'k-1}$

The second one, for example, corresponds to the proposal in [GGG+07] and it is defined in the next section.

### A.7.1 DESTRO routing algorithm

As described in detail in Section A.2.1, and adapted to BMINs, the $2k$ ports of the switch $\langle s, o\rangle$ are distributed in two disjoint groups. On one hand, the $k$ ports that connect to switches that are located in the previous stage $s-1$, where $0 \leq s < n$, are labeled from 0 to $k-1$ (Figures 34(a) and 34(b)). On the other hand, the other $k$ ports that connect to switches that are located in the stage $s+1$, where $0 \leq s < n-1$, are labeled from $k$ to $2k-1$. Switches belonging to the last stage $n-1$ only use half of the ports, and they are labeled from 0 to $k-1$ (Figure 34(b)).




(a) If $0 \leq s < n-1$. (b) If $s = n-1$.

Figure 34: Numbering scheme for switch ports.

In an informal way, the port $l$ is obtained from the $k$–ary number, $h_s$, of the destination node. In switches with $2k$ ports, $h_s$ can represent two different port labels, according to the numbering scheme that has been assumed: $l = h_s$, where $0 \leq h_s < k$; and $l = h_s + k$, where $k \leq l < 2k$. To obtain the exact value of $l$, the routing function simply needs to know the direction of the path/route of a message (i.e., in forward or backward direction). Hence,

$$R(\langle s, o\rangle, h) = \begin{cases} l = h_s + k, & \text{if the message goes forward} \\ l = h_s, & \text{if the message goes backward} \end{cases}$$

In [GGG+07] the authors propose an implementation of the DESTRO deterministic routing algorithm for fat–trees by means of *Flexible Interval Routing*, (FIR) [GLD05]. FIR is an extension of *Interval Routing*, (IR) [ea91]. In IR, every port has two registers (*First Interval* and *Last Interval*) that define the beginning and end of routing interval, respectively. To send a message through a port, the destination address must be inside the routing interval. FIR adds an extra register (*Mask Register*) per port for defining which bits of the destination address must be compared to the routing interval. In order to guarantee deadlock freedom, FIR adds one more extra register (*Routing Restrictions Register*) per port that establishes the priority between several ports.

***Proposition 1.64*** *If the traffic pattern is uniform, DESTRO routing algorithm is balanced according to the Definition 1.28.*

**Proof:** When the traffic is uniformly distributed in the network, the probability of sending traffic from the switch $\langle s, o\rangle$ to a destination node $h$, $h \in \{N_b^R(\langle s, o\rangle) \cup N_f^R(\langle s, o\rangle)\}$ (see Definitions 1.18 and 1.19), is constant.

All the ports $l$, $0 \leq l < k$, of the $k^{n-1}$ switches at the stage $s$, $0 \leq s < n$, receive the same number of paths in backward direction according to Definition 1.20, because

$$card(N_b^R(\langle s, o_0\rangle, l_j)) = \cdots = card(N_b^R(\langle s, o_i\rangle, l_k)) = \cdots = card(N_b^R(\langle s, o_{(k^{n-1})}\rangle, l_m)) \quad \forall l_j, l_k, l_m \in [0, k-1]$$

Similarly, all the ports $l'$, $k \leq l' < 2k$, of the previous $k^{n-1}$ switches, receive the same number of paths in forward direction according to Definition 1.21, because

$$card(N_f^R(\langle s, o_0\rangle, l'_j)) = \cdots = card(N_f^R(\langle s, o_i\rangle, l'_k)) = \cdots = card(N_f^R(\langle s, o_{(k^{n-1})}\rangle, l'_m)) \quad \forall l'_j, l'_k, l'_m \in [k, k-1]$$

□