Design and Control of Next Generation Distribution Frames

Design and Control of Next Generation Distribution Frames

Davide Cuda, Paolo Giaccone, Massimo Montalto
Dipartimento di Elettronica, Politecnico di Torino, Italy

Abstract

In today's access networks, the permutation of circuits connecting the subscriber lines to Plain Old Telephone Service (POTS) and to Digital Subscriber Line Access Multiplexers (DSLAMs) occurs in the Main Distribution Frame (MDF) and is still manually configured. However, new market regulations and new policies adopted by operators require increasingly more frequent permutations, making the manual configuration activity particularly expensive. Very recently, Automated MDFs (AMDF) have been developed to provide inexpensive and almost real-time switching capability. Our study, based on more than 50 years of research activities on architectures for circuit switching, is focused on overcoming the limits of classical architectures. In fact, strictly non-blocking multistage networks are too expensive, as a consequence of the large number of ports they require (sometimes exceeding 100,000). In addition, rearrangeable multistage networks can temporarily interrupt active circuits, affecting the performance of ADSL subscriber lines.

As a possible solution to these problems, we propose the design of AMDFs based on Non-Interruptive Rearrangeable (NIR) networks. We show how to optimize the routing control to minimize the setup time of a circuit and to exploit output grouping. We believe that the solution described above is not only relevant for the theory of multistage interconnection networks, but also for the design and operation of large AMDFs.

Keywords: Multistage switching architectures, non-interruptive networks.

1. Introduction

In the access segment of a telecommunication network, as shown in Fig. 1, the Main Distribution Frame (MDF) is placed at the central office and provides many carriers with the possibility to reach the subscriber lines and connect them to the Digital Subscriber Line Access Multiplexers (DSLAMs) and to the Plain

☆The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n. 257740 (Network of Excellence "TREND")

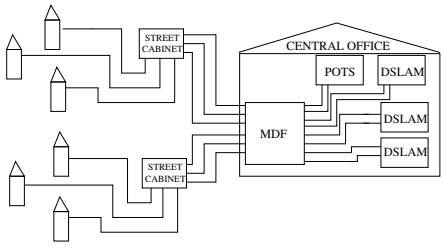

Figure 1: Basic scheme of the current access network.

Old Telephone Services (POTS). The MDF is basically a circuit switch, which has been manually operated so far because requests of setting up or tearing down connections were mostly occasional. However, new telecommunication market regulations and current efforts to reduce the energy consumption in the access network impose new operational requirements that can be met only by almost real-time switching operations.

On the one hand, over the last 15-20 years, because of the new telecommunication market regulations in North American and European countries, competition has been increasing in the context of the access network. The variety of Internet Service Provides (ISPs) coexisting in a central office and the consequent offer diversification have increased access network complexity. Migration of customers among carriers (churn) continues to grow at high rates due to market rivalry and networks evolution. According to [1] and [2], carriers experience on average a 30-35% yearly churn rate (i.e., percentage of customers changing their carrier each year). As a result, operating the access network, heavily influenced by truck rolls, is today an expensive, slow and fault-prone process.

On the other hand, access networks represent the main contribution of the overall energy consumption of the Internet [3, 4]. Indeed, since all the Asymmetric Digital Subscriber Line (ADSL) subscribers are connected uniformly to the linecards of many DSLAMs, all these linecards/DSLAMs are always powered on, independent of the activity of the ADSL users. Nowadays some large European ISPs [5] are devising and evaluating smart policies to concentrate active users during off-peak hours onto few linecards/DSLAMs and to power off all the other ADSL interfaces. Note that this process requires aggregating all the inactive ADSL subscribers on a few ports and implementing some (in-band or out-band) wake-on-ADSL mechanisms, still proprietary and not included in the ANSI/ITU ADSL standards. Thanks to the large variability in the traffic load [6], powering on/off single linecards or whole DSLAMs may lead to a relevant power saving, when such a process is operated daily, based on the activity of each subscriber line.

1.1. From traditional to next-generation MDFs

Fig. 2 shows the basic structure of an MDF, switching the subscriber lines to the different carriers. The Incumbent Local Exchange Carrier (ILEC) is the

main carrier, whereas the Competitive Local Exchange Carriers (CLEC) are the other (minor) carriers. As convention, we assume that subscriber lines are at the inputs (left ports) of the switch, and the carriers with their DSLAMs are at the outputs (right ports) of the switch. Note that the concept of input and output side is fictitious in this scenario, since the electric circuits allow the communication in both directions.

ILECs are traditionally responsible for managing the MDF. The design of MDFs follows some requirements dictated by the ILEC and leading to a port share-out proportional to the market presence. Quite often, the switch of a MDF is asymmetric, since the ports connected to the carriers are around 1.5 times the number of ports connected to the subscriber lines. In fact, the ILEC owns the MDF, which is dimensioned to support the connectivity to all the customer lines of the central office. Hence, the largest fraction of ports is assigned to the ILEC and the remaining ports are spread on different CLECs.

From the switching point of view, the main requirements to design a traditional MDF were the following: (i) non-blocking behavior, (ii) full connectivity between the input and output ports, (iii) large number of ports (1,000-10,000). These requirements have been met by engineering the cabling system and the patch panels, to reduce to a minimum manual operations.

Instead, automatic MDFs support a very large number of input/output ports; 10,000-100,000 ports are already available in commercial products [7, 8, 9, 10]. Different approaches are adopted for their design. One possible approach consists of adopting a robot that performs the connections directly on the patch panel [11].

The most common approach is to interconnect many basic switching modules according to the classical schemes adopted in multistage circuit switching networks. The basic switching modules are implemented with a particular technology, peculiar to the AMDF manufacturer. Commercial AMDFs employ electromechanical relays [7] or MEMS (Micro Electro-Mechanical Systems) [10, 12, 13] as crosspoints (i.e., basic switching devices), whereas new promising technologies appear to be NEMS (Nano Electro Mechanical Systems) [14] and TEMS (Transparent Embedded Magnetic Switch) [15]. Independently from the technology adopted, the switching architectures are based on classical rearrangeable Clos networks [16], since they allow good scalability in terms of cost, measured as the number of crosspoints (or basic modules), when the size of the switch is very large.

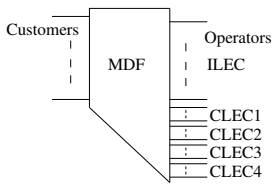

Figure 2: Example of asymmetric MDF connecting the subscriber lines to one ILEC and four CLEC carriers.

Due to the large number and high frequency of reconfigurations, next-generation AMDFs will have to support switching operations which are transparent for pre-existing circuits, i.e. new circuit requests must "not-interrupt" any other circuit that is already set up. In particular, such a *not-interruptive* (NIR) property is crucial for ADSL subscriber lines, since active ADSL lines are not robust to interruptions in the electrical continuity of the circuit. Indeed, in the case of an interruption, the ADSL connection enters a retrain phase to adapt to the dynamic channel conditions [17]; however, this process takes between 1 and 10 seconds, which is not acceptable for frequent (daily) reconfigurations. Thus, the NIR property represents a make-or-break requirement, and actual AMDF designs do not satisfy it when based on classical *rearrangeable non-blocking* (RNB) networks, in which pre-existing circuits can be temporary interrupted when adding a new circuit. Alternative architectures, based on *strictly non-blocking* (SNB) networks, satisfy the NIR requirement by construction, but their cost is almost twice the one of RNB networks. Thus SNB networks are a viable alternative only when used in small size switches.

Recently, the seminal work in [18] has shown the design of NIR networks with a cost very close to classical RNB networks, but the proposed architecture is not optimized for the particular scenarios considered in our work. Furthermore, some cheap electro-mechanical technologies adopted to implement each switching module (e.g. in case of relays) require a non-negligible amount of time (seconds or even minutes) to change their internal state and reconfigure. To avoid interruptions, the reconfiguration algorithm follows a particular sequence of module configurations, that cannot be parallelized. In NIR networks, the number of reconfiguration steps may grow with the size of the AMDF. Thus, for large networks the overall reconfiguration time can become order of minutes/hours, unacceptable for practical deployment.

Furthermore, the AMDF must connect each specific subscriber line to *any* port of a given carrier. A change of carrier requires connecting the user to any port of the carrier's DSLAMs; in case of daily user aggregation to save energy, the user must be connected to any port of a powered-on linecard. As a consequence, it is possible to define a *group* as the set of either (i) all the ports of a DSLAM linecard, or (ii) all the ports of a DSLAM, or (iii) all the ports of a carrier, depending on the scenario. Each switching connection is defined as a couple (input port, group). This fact can be exploited to reduce the final cost of the switch.

In this paper we show how to optimize jointly the design and the routing control of multistage switching networks, expressively designed for AMDFs. We have considered only NIR architectures, without considering RNB networks, which cannot cope with frequent reconfigurations, and SNB networks, whose cost is too high, being almost twice than NIR networks. The main contributions of our work are (i) the methodology to design, at minimal cost, NIR networks that support *output grouping* and *minimize reconfiguration times*, (ii) to propose the corresponding routing algorithm to connect new circuits and (iii) to compare numerically different design architectures.

Note also that NIR networks were also investigated in [19, 20], where authors

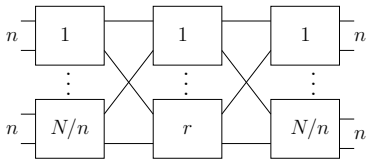

Figure 3: Three stage Clos network.

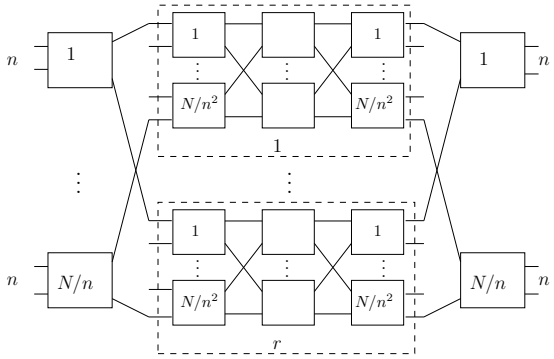

Figure 4: Recursive construction of a Clos network

propose the new class of $Log_2(N, 0, p)$ networks, built with $\sqrt{N}$ Banyan networks of size $N \times N$. Their final cost, in terms of crosspoints, grows as $\theta(N^{1.5} \log N)$ (since the cost of Banyan networks is $\theta(N \log N)$), whereas the cost of the NIR networks, proposed in [18] and considered in our work, grows as $\theta(N \log N)$, similarly to classical RNB networks.

The paper is organized as follows. In Sec. 2 we discuss the design of NIR networks and in Sec. 3 we propose the corresponding routing algorithms. In Sec. 4 we discuss the design of NIR networks that exploit output grouping. Finally, in Sec. 5 we evaluate some examples of design. A preliminary version of our work appeared in [21]; Sec. 4 and the results for output grouping in Sec. 5 extend [21].

2. Multi-stage switching networks

We consider the design of a $N \times N$ circuit switch that is fully-connected and non-blocking, hence it is always possible to connect any idle input to any idle output. We evaluate the cost of the switch in terms of crosspoints (C_X). A basic module of size $n \times m$ built with a crossbar has cost $C_X = nm$. When a new circuit must be established, the switching architecture can be [16] either *strictly non-blocking* (SNB), i.e. the path to connect the circuit is established without rerouting pre-existing active paths of former circuits, or *rearrangeable non-blocking* (RNB), i.e. the path may require to rearrange pre-existing paths.

Table 1: Synoptic view of candidate networks-(N, n) for AMDF.

Network archit.	Routing algorithm	C_X	NIR	Output grouping
SNB	trivial	$4nN - 2N + 2N^2/n - N^2/n^2$	yes	yes
RNB	Paull	$2nN + N^2/n$	no	yes
NIR1	Conf-NIR1	$2nN + 2N + N^2/n + N^2/n^2$	yes	no
NIR2	Conf-NIR2	$2nN + N^2/n + 4N + 2N^2/n^2$	yes	yes

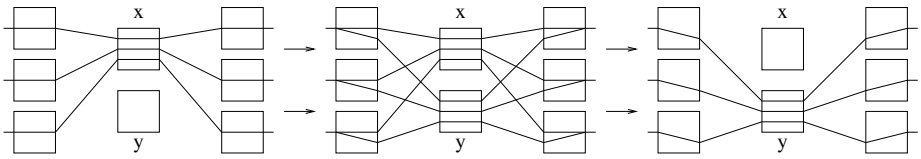

Figure 5: Example of operations to rearrange the circuits through x without any interruption by exploiting duplicated paths through y.

We consider a symmetric three-stage Clos network [16], depicted in Fig. 3, built with N/n modules of size $n \times r$ at the I stage, r modules of size $(N/n) \times (N/n)$ at the II stage and N/n modules of size $r \times n$ at the III stage. Depending on r, Clos network shows different properties. The condition $r = 2n-1$ guarantees SNB, with a trivial routing algorithm. Moreover, the condition $r = n$ guarantees RNB behavior, when the *Paull algorithm* (described in [16]) is adopted. The cost of RNB and SNB are reported in Table 1. Asymptotically, for $N \to \infty$, it is well known that the cost of SNB networks is almost twice the cost of RNB: $C_X(\text{SNB}) \approx 2C_X(\text{RNB})$.

When the number of ports is very large, it is possible to adopt a recursive contruction. Indeed, II-stage modules can be implemented with three stage Clos networks. To ensure the SNB or the RNB property, the conditions described above must be met at each recursive level. An example of a five stage Clos network is depicted in Fig. 4.

2.1. NIR rearrangeable Clos networks

Quite recently, [18] has defined the class of *non-interruptive rearrangeable* (NIR) networks, that are rearrangeable but, whenever adding a new circuit, pre-existing circuits do not experience any electrical interruption. The basic idea is to create duplicate paths through additional (with respect to RNB Clos networks) II-stage modules, following the sequence of operations shown in Fig. 5. Interestingly, the condition $r = n + 2$ is enough to build NIR Clos networks; thus, only two additional modules in the II stage (independently from the switch size) are sufficient to observe an end-to-end behavior similar to SNB. The NIR property is achieved thanks to duplicated paths through the two additional II-stage modules, in order to "backup" all the paths that must be rearranged and

Figure 6: Example of output and input divertability for a 4×4 basic module.

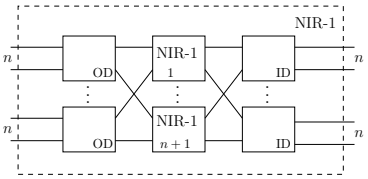

Figure 7: NIR1 network obtained with recursive construction.

avoid interruptions. We will denote as *NIR2* such class of NIR networks. The corresponding cost, computed in Table 1, grows asymptotically as an RNB network: $C_X(\text{NIR2-}(N,n)) \approx C_X(\text{RNB-}(N,n))$. The notation "network-$(N,n)$" refers to a $N \times N$ network with (N/n) modules at the I and III stage. In addition, [18] has shown that it is possible to further reduce the number of II-stage modules by setting $r = n+1$, i.e. with just one additional middle stage module, and adopting a more complex routing algorithm. This architecture will be later referred as *NIR1* and Table 1 shows that $C_X(\text{NIR1-}(N,n)) \approx C_X(\text{RNB-}(N,n))$. Note that [18] showed that $r \geq n+1$ is also a necessary condition for the NIR property.

To support the NIR construction, it is necessary to create double paths between I-stage modules and III-stage modules. Such modules must support the property of *output* and *input divertability*, respectively. A switching network is said to be output-divertable (input-divertable) if it is always possible to non-interruptively append a new path from the input (output) of a given existing path to some unused output (input) [18], as shown in Fig. 6. Note that a single crossbar switch of size $n \times m$ provides, thanks to the available nm crosspoints, both input and output divertability.

2.2. Recursive construction of large switches

To design networks with a very large number of ports, as for the AMDF considered in the CO scenario, a modular and scalable design can be achieved by a recursive construction. In this case, each II-stage module is implemented through another multistage network.

Fig. 7 an 8 show the basic recursive rule to design NIR1 and NIR2 networks [18], respectively. "ID" stands for input-divertable network, and "OD" stands for output-divertable network. In NIR1 networks, the II-stage modules must be NIR1 to guarantee NIR behavior. On the contrary, in NIR2 networks, II-stage modules are simply RNB networks. Note that the routing algorithms must be carefully chosen to ensure NIR behavior, as discussed in Sec. 3.

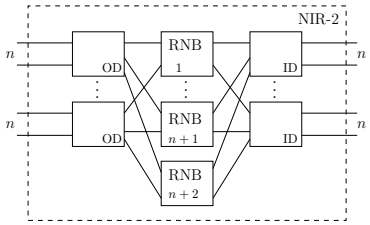

Figure 8: NIR2 network obtained with recursive construction.

3. Routing in NIR1 and NIR2 networks

We propose two algorithms, *Conf-NIR1* and *Conf-NIR2*, to route the circuits in NIR1 and NIR2 networks, respectively. Both of them (i) minimize the reconfiguration time, by exploiting parallel operations, (ii) avoid interruptions for pre-existing paths, by exploiting duplicated paths, and (iii) support output grouping. These algorithms are based on the algorithms proposed in [18] and work in an incremental way, by adding one circuit at the time.

We use a Paull matrix [16] to describe the operations of the algorithm. Let Ω_i be the set of modules of the i-stage and $|\Omega_i|$ be the corresponding number of modules. A Paull matrix describes in a compact way the active paths in a three stages Clos network. It is a matrix $M = [m_{ij}]$ of size $|\Omega_1| \times |\Omega_3|$, in which each row corresponds to a I-stage module and each column corresponds to a III-stage module; each element $m_{ij} \subseteq \Omega_2$ is the set all the II-stage modules used to connect $i \in \Omega_1$ to $j \in \Omega_3$. For example, if II-stage module $a \in m_{ij}$, then it exists a circuit whose path connects I-stage module i to III-stage module j through a, i.e. the ith input of a is connected to its jth output.

As an example, consider the Paull matrix describing the sequences of operations presented in Fig. 5. Here, the Paull matrix is a 3×3 matrix whose elements $m_{ij} \in \{\emptyset, x, y, (x, y)\}$, where $m_{ij} = (x, y)$ means that I-stage module i is connect to III-stage module j through both II-stage modules x and y; this notation allows the description of duplicated paths across x and y. The evolution of Paull matrix is the following:

$$\begin{bmatrix} x & \emptyset & \emptyset \\ \emptyset & x & \emptyset \\ \emptyset & \emptyset & x \end{bmatrix} \longrightarrow \begin{bmatrix} (x,y) & \emptyset & \emptyset \\ \emptyset & (x,y) & \emptyset \\ \emptyset & \emptyset & (x,y) \end{bmatrix} \longrightarrow \begin{bmatrix} y & \emptyset & \emptyset \\ \emptyset & y & \emptyset \\ \emptyset & \emptyset & y \end{bmatrix}$$

Initially, Paull matrix contains only II-stage module x. To rearrange without interrupting, the existing connection is duplicated through y by exploiting the output (input) divertability of I-stage (III-stage) modules. Finally, x is disconnected from the network and will be used to duplicate the paths for the new connections.

Before describing the algorithms to configure NIR1 and NIR2 networks, we describe the classical Paull algorithm to configure RNB networks [16], since it will be referred to later. The pseudocode of Paull algorithm when a new circuit

*** **Paull** ***

function **add-new-connection**($m_1 \in \Omega_1$, $m_3 \in \Omega_3$)

if exists ($a \in \Omega_2$ s.t. $m_1 \nrightarrow a \nrightarrow m_3$) // no need for rearrangement

  connect $m_1 \rightarrow a \rightarrow m_3$

else it exists ($a, b \in \Omega_2$ s.t. $m_1 \nrightarrow a$ & $m_1 \rightarrow b$ & $a \rightarrow m_3$ & $b \nrightarrow m_3$) // rearrange

  P =find-path(a, b, m_1, m_3) $[s_0]$

  detach($a \in P$), detach($b \in P$) $[s_1]$

  swap($a \in P, b \in P$) $[s_2]$

  connect $m_1 \rightarrow a \rightarrow m_3$ $[s_3]$

  attach(a), attach(b) $[s_4]$

end if

Figure 9: Pseudocode to add a new circuit from I-stage m_1 to III-stage m_3 in a RNB Clos network using the classical Paull algorithm.

must be added from a free input of I-stage module m_1 to a free output of III-stage module m_3 is provided in Fig. 9. Note that notation $x \rightarrow y$, being x and y modules in subsequent stages, refers to the presence of an active path between the two modules; notation $x \nrightarrow y$ means the opposite. When adding a new circuit, two different situations can occur. The circuit can be directly added if there exists a II-stage module a that is not already connected to m_1 and to m_3. Otherwise, it is necessary to rearrange the paths that are currently passing through *exactly* 2 specific II-stage modules, namely a and b. Indeed, there must exist two modules a and b such that m_1 (m_3) is not connect to a (b) while it is connected b (a). From a and b, it is possible to defined an ab-path as a sequence of elements in the Paull matrix in which a and b appear and adjacent elements in the sequence correspond to different rows or columns; this path includes all the active circuits that are rearranged to accommodate the new connection. The *key idea* is to swap a and b along the ab-path (i.e., to rearrange the corresponding circuits) so that, at the end, a can be used to satisfy the new connection.

At step s_0, the Paull algorithm computes ab-path P. At step s_1, a and b modules are disconnected from the network to allow the reconfiguration. Then, at step s_2 the connections belonging to P and passing through a and b are swapped. By construction, in s_3, a can be now used to connect m_1 to m_3 and accommodate the new connection. Finally, in step s_4, a and b modules are connected again through the network.

3.1. Circuit routing in a NIR1 network

Fig. 10 shows the pseudocode of the *Conf-NIR1* algorithm we propose to configure NIR1 networks, derived from [18]. When adding a new circuit, the algorithm checks whether the network must be rearranged, exactly as in the Paull algorithm. If it is not necessary to rearrange, the circuit is added by creating a path along I-stage m_1, II-stage a and III-stage m_3; no interruption occurs for pre-existing circuits. If it is necessary to rearrange the paths, the key idea is

*** **Conf-NIR1** ***

function **add-new-connection**($m_1 \in \Omega_1$, $m_3 \in \Omega_3$)
let $c \in \Omega_2$ be the backup module
if exists ($a \in \Omega_2$ s.t. $m_1 \nrightarrow a \nrightarrow m_3$) // no need for rearrangement
  connect $m_1 \to a \to m_3$
else it exists ($a, b \in \Omega_2$ s.t. $m_1 \nrightarrow a$ & $m_1 \to b$ & $a \to m_3$ & $b \nrightarrow m_3$) // rearrange
  P =find-path(a, b, m_1, m_3) $[s_0]$
  duplicate($a \in P \Rightarrow c$) $[s_1]$
  attach(c) $[s_2]$
  detach($a \in P$) $[s_3]$
  duplicate($b \in P \Rightarrow a$), connect $m_1 \to a \to m_3$ $[s_4]$
  attach(a) $[s_5]$
  detach($b \in P$) $[s_6]$
  duplicate($c \in P \Rightarrow b$) $[s_7]$
  attach(b) $[s_8]$
  detach(c) $[s_9]$
end if

Figure 10: Pseudocode to add a circuit from I-stage m_1 to III-stage m_3 in a NIR1 switch.

to swap a and b in the ab-path (i.e., to rearrange the corresponding circuits). However, the major difference from the Paull algorithm is that the sequence of operations allow now to use a "backup" module c to avoid disconnecting any already established connection.

Fig. 11 shows an example of the steps in the algorithm, described through the temporal evolution of Paull matrix, in the case a new circuit between the I-stage module and the III-stage module corresponding to the position highlighted by the circle. During step s_0, the ab-path is computed; for completeness Fig. 11 reports both ab-path P (dotted) and the ba-path.

In details, during s_1, all the paths passing through a and belonging to P are replicated onto c; this requires the divertability of I-stage and III-stage modules, since the paths must be duplicated to prevent interruption. In s_2, module c is connected to the I stage and to the III stage, and then all the paths through a belonging to P are removed, by configuring properly m_1 and m_3. During s_4, the paths crossing b and included in P are duplicated through a; by construction, this operation is always permitted. Concurrently, the new circuit between m_1 and m_3 is established with the path through a, which is available now. Afterwards, in s_5, module a is connected to the remaining network. In s_6, the paths through b in P are removed and then b is used to duplicate the paths through c. Finally, in s_8 module b is again connected to the network and, during the last step c, it is detached (i.e., disconnected from the remaining network) to make it available for the future.

We now evaluate the configuration time of the algorithm. Let δ be the maximum time to configure each single module; we assume that many paths

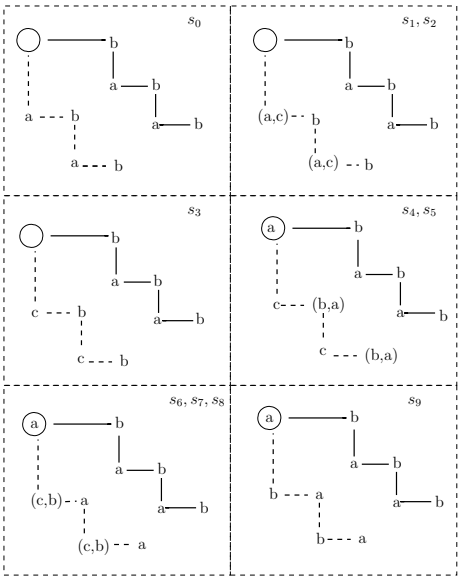

Figure 11: Example of evolution of the Paull matrix in Conf-NIR1 when rearranging the paths.

can be established/removed in parallel in the same module, i.e., connecting $x_1 \to y_1$ at the same time of $x_2 \to y_2$ takes at most δ. We can claim:

Property 1. *Let T_R be the overall configuration time for a NIR1-(N, n) network when adding a new circuit. Then: $T_R \leq 9\delta$.*

Proof. Note that each step $s_1 - s_9$ in Fig. 10 lasts no more than δ. Since all of them must be operated *sequentially* to prevent interruptions, $T_R \leq 9\delta$. □

For a three stage network, the latter property shows that T_R is independent from the overall switch size N, whereas for more stages the configuration time grows logarithmically with N. Indeed, in the case of recursive construction and $N = n^k$:

Property 2. *Let T_R be the overall configuration time for a NIR1-(n^k, n) network, built with $k-1$ recursive levels, when adding a new circuit. Then:*

$$T_R \leq 9\delta(k-1) = 9\delta\left(\frac{\log N}{\log n} - 1\right) \tag{1}$$

Proof. Referring to the pseudocode in Fig. 10, the *connect* operation in s_4 requires recursively applying *connect* to a smaller NIR1 network (labeled a in the pseudocode). Instead, all the other steps require at most some constant time $\leq \delta$. Hence, the configuration time is increased by (at most) 9δ at each level of recursion, thanks to Property 1. By construction, the number of factorization levels is $k-1$, being $k = \log_n N$, and (1) holds. □

*** **Conf-NIR2** ***	
function **add-new-connection**($m_1 \in \Omega_1$, $m_3 \in \Omega_3$)	
let $c, d \in \Omega_2$ be the backup modules	
if exists ($a \in \Omega_2$ s.t. $m_1 \nrightarrow a \nrightarrow m_3$) // no need for rearrangement	
connect $m_1 \rightarrow a \rightarrow m_3$	
else it exists ($a, b \in \Omega_2$ s.t. $m_1 \nrightarrow a$ & $m_1 \rightarrow b$ & $a \rightarrow m_3$ & $b \nrightarrow m_3$) // rearrange	
P =find-path(a, b, m_1, m_3)	$[s_0]$
duplicate($a \Rightarrow c$), duplicate($b \Rightarrow d$)	$[s_1]$
attach(c), attach(d)	$[s_2]$
detach(a), detach(b)	$[s_3]$
duplicate($b \in P \Rightarrow a$), duplicate($a \in P \Rightarrow b$)	$[s_4]$
connect $m_1 \rightarrow a \rightarrow m_3$	$[s_4]$
attach(a), attach(b)	$[s_5]$
detach(c), detach(d)	$[s_6]$
end if	

Figure 12: Pseudocode to add a circuit from I-stage m_1 to III-stage m_3 in a NIR2 switch.

As a conclusion, *the configuration time in large NIR1 networks grows logarithmically with the size of the network.*

3.2. Circuit routing in a NIR2 network

Fig. 12 shows the pseudocode to route a new circuit in a NIR2 network and Fig. 13 shows the Paull matrix evolution for an example. The *key idea* of Conf-NIR2 is simply to use two backup modules c and d in the II stage to duplicate the pre-existing paths, across a and b, that must be reconfigured. Now a and b can be detached, reconfigured in isolation and then reconnected again to the network. At the end, c and d are set free. Conf-NIR2 and Paull algorithm are very similar; indeed, steps $s_1 - s_4$ of Conf-NIR2 substitute $s_1 - s_2$ of Paull to exploit duplicated paths through c and d and avoid interruptions. Note that paths are duplicated exploiting the divertability of the I and III stages during s_1 and s_4.

Property 3. *Let T_R be the overall configuration time for a NIR2-(N, n) network when adding a new circuit. Then: $T_R \leq 6\delta$.*

Proof. Each step of Conf-NIR2, except for s_0, lasts at most δ. Each step s_i includes all the operations that can be parallelized; each step accounts for δ and the above property holds. □

In the case of recursive construction and $N = n^k$, we claim:

Property 4. *Let T_R be the overall configuration time for a NIR2-(n^k, n) network, built with $k-1$ recursive levels, when adding a new circuit. Then: $T_R \leq 6\delta$.*

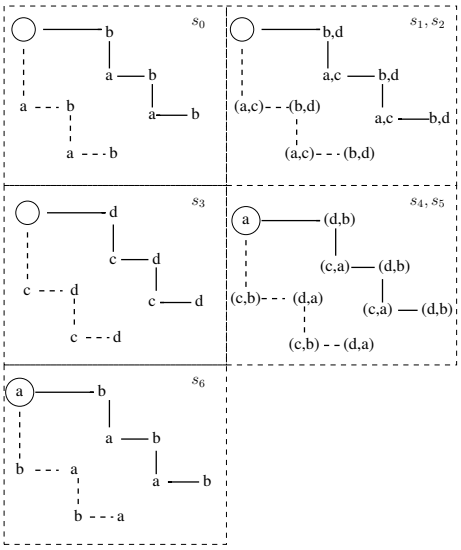

Figure 13: Example of evolution of the Paull matrix in Conf-NIR2 when rearranging the paths.

Proof. In the pseudocode of Fig. 12, during s_2, s_3, s_5, s_6 only I-stage and III-stage modules are reconfigured. Instead, s_1 and s_4 require a recursive call to the routing algorithm, but all the required operations occur in parallel. In s_1 the reconfiguration of *all* modules internal to c and d occurs in parallel, since c and d are isolated from the network, thus it lasts no more than δ. A similar observation holds for a and b during s_4. Thus each step s_i lasts no more than δ and $T_R \leq 6\delta$. □

As a conclusion, *in large NIR2 networks the configuration time is independent from the size of the switch and from the number of recursion levels to build the actual network*. Comparing Property 2 with Property 4, it is clear that NIR2 networks are superior to NIR1 in terms of reconfiguration time. This is a crucial issue that must be considered when δ is not negligible.

4. NIR networks with port grouping

When a new circuit is added, a given input port must be switched to any output port within a group of ports; this operation is relevant for AMDFs, as discussed previously in Sec. 1.1. Indeed, output grouping allows to simplify the design of multistage networks and reduce the final cost. Such reduction can be easily achieved in classical three stage RNB Clos networks, as shown in [22].

Indeed, consider a $N \times N$ RNB network, built with $n \times n$ modules at the III stage. Now observe that each of them allows the selection of one specific output among n. If a specific input must be connected to any output of a specific

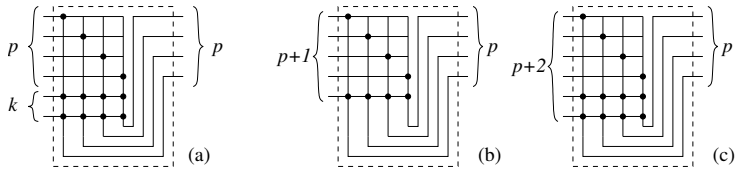

Figure 14: Construction of concentrators of size: (a) $(p + k) \times k$, (b) $(p + 1) \times p$ and (c) $(p + 2) \times p$.

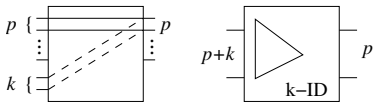

Figure 15: Example of k-restricted divertible module and its symbol.

III-stage module (i.e. within a group of size n), this is achieved independently from the configuration of the III-stage module. Thus, the III-stage modules can be removed and the II-stage modules are connected directly to the outputs, according to some predefined configuration of the III-stage modules; this process is denoted by "pruning". Note that the routing algorithm remains unchanged and runs on the "virtual" network, equivalent to the original Clos network before pruning: when connecting a specific input to any output inside a group, the algorithm connects the input to any idle output within the group, even if the III stage has been pruned. Note that, to fully exploit the cost reduction due to output grouping, the size of the group must be a multiple of n.

Consider now either a SNB or a NIR $N \times N$ Clos network. In such a case, III-stage modules are asymmetric and cannot be pruned as in the case of RNB networks. However, they can be substituted with concentrators [16]. A $(p+k)\times p$ concentrator allows connecting each specific input to *any* output among the p ones that are currently available. Even if there exist different architectures [16] to implement concentrators, NIR networks require concentrators implemented according to the scheme shown in Fig. 14. This is equivalent to starting from a $(p + k) \times p$ crossbar and then removing all active crosspoints. Then, place the active crosspoints: (i) for a $p \times p$ subnetwork, in the intersections between the horizontal line i and the vertical line i, for $1 \leq i \leq p$, (ii) for the remaining $k \times p$ subnetwork, in all the intersections. The final cost in terms of crosspoints is reduced to $p+kp$, instead of p^2+kp of the initial crossbar. Fig. 14 shows two examples of concentrators built according to such scheme, for the specific cases of III-stage modules in a NIR1 network ($k = 1$) and NIR2 network ($k = 1$).

In the case of NIR networks, a generic concentrator does not support the input-divertability required in the III stage to provide NIR operations. We propose a new family of concentrators:

Definition 1. *A network is said to be input* k-restricted divertible *if there exists*

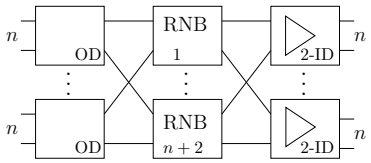

Figure 16: NIR2-(N, n, n) network with output grouping.

a subset of k input ports denoted as backup ports *and it is always possible to add a new path, non-interruptively, from an output of a pre-existing connection to any input backup port.*

Equivalently, in an input *k-restricted divertible* network, a two-cast connection from output j to input i and i' is always permitted, provided that i' belongs to the set of backup inputs, as shown in Fig. 15. An analog definition can be devised for the output k-restricted divertible network.

Property 5. *A $(p + k) \times p$ concentrator adopting the construction in Fig. 14 is input k-restricted divertible.*

Proof. Define the k lower inputs as backup ports. By construction, the corresponding kp lower crosspoints allow to add a new connection from any backup input ports to any active output without any interruption. □

Note that other concentrator architectures, known in the literature [16], do not satisfy Property 5.

We now define the construction of NIR Clos networks supporting output grouping. Let NIR-(N, n, g) be a NIR-(N, n) network supporting output grouping with group size g. For simplicity, we will consider the design of NIR-(N, n, n) networks with $g = n$, even if an analogous methodology can be applied to generic group sizes.

We start with a negative result:

Property 6. *Consider a network built by substituting the III-stage modules of an NIR1-(N, n) with input 1-restricted divertible concentrators. Such a network is not NIR.*

Proof. During step s_1, s_4, s_7 in Conf-NIR1 of Fig. 10, it is necessary to duplicate the path through the backup port c and two other modules a and b. This is incompatible with a 1-restricted divertible concentrator, since it does not support input divertability from any generic input port. □

On the contrary, NIR2 networks support output grouping:

Property 7. *Consider a NIR2-(N, n, n) network shown in Fig. 16, built by substituting the III-stage modules of an NIR2-(N, n) with input 2-restricted divertible concentrators. Such a network is NIR when controlled by routing algorithm Conf-NIR2.*

***** Design-NIR2-**(n^k, n, n^h) *******
use NIR2-(n^k, n, n) // at the first factorization level
for $l = 2$ **to** h // $h-2$ factorization levels with concentrators
  use RNB-(n^{k-l+1}, n, n) at II-stage of RNB-(n^{k-l+2}, n, n)
endfor
for $l = (h+1)$ **to** $(k-1)$ // remaining factorization levels without concentrators
  use RNB-(n^{k-l+1}, n) at II-stage of RNB-(n^{k-l+2}, n)
endfor

Figure 17: Pseudocode to design recursively a large $n^k \times n^k$ NIR2 network with output groups of size n^h, with $1 \leq h < k$.

Proof. In steps s_2 and s_6 in Conf-NIR2 of Fig. 12, always the same two modules c and d are exploited. Now connect these two modules to the backup ports of the 2-restricted divertible concentrators in the III stage. As a consequence, Conf-NIR2 avoids interruptions. □

It is possible to apply the recursive construction to design very large NIR2 networks, using the algorithm shown in Fig. 17. Indeed, we can start with an NIR2 network with group size n and use, internally at the II stage, the recursive RNB construction with concentrators, which support output grouping. At each recursion level, the group size grows by a factor n. After $h-1$ recursion levels, the middle II-stage networks are classical RNB networks, recursively factorized and not supporting output grouping.

4.1. Cost evaluation

We now evaluate the cost in terms of modules and crosspoints for NIR2 networks.

Property 8. *Consider a 3-stage $N \times N$ NIR2 network, supporting groups of size n and built with basic modules of size $n \times n$ and $n \times (n+2)$. Assume N is an integer multiple of n. The cost in terms of crosspoints is:*

$$C_X(NIR2-(N,n,n)) = nN + N^2/n + 5N + 2N^2/n^2$$

Proof. Referring to Fig. 16, by construction, the I stage is built with N/n modules of size $n \times (n+2)$, the II stage with $n+2$ modules of size $n \times n$ and the III stage with N/n concentrators of size $(n+2) \times n$ with output group size n. By summing each contribution:

$$\begin{aligned} C_X(\text{NIR2}-(N,n,n)) &= \frac{N}{n}n(n+2) + (n+2)\left(\frac{N}{n}\right)^2 + \frac{N}{n}(n+2n) \\ &= nN + N^2/n + 5N + 2N^2/n^2 \end{aligned}$$

□

For large networks:

Property 9. *Consider a $N \times N$ NIR2 network with $N = n^k$, supporting groups of size n^h (with $1 \leq h < k$) and built recursively with basic modules of size $n \times n$ and $n \times (n+2)$. The cost in terms of crosspoints is:*

$$C_X(NIR2-(n^k,n,n^h)) = ((2k-h)(n+2)-2n+1)n^k \\ = N((2\log_n N - h)(n+2)-2n+1)$$

Proof. According to the design algorithm in Fig. 17, given $N = n^k$, the total number of stages is $2k-1$ and the number of factorization levels is actually $k-1$. At the first level, the I stage is built with n^{k-1} modules, supporting output divertability, of size $n \times (n+2)$; the III stage with n^{k-1} concentrators of size $(n+2) \times n$; the II stage with $n+2$ RNB networks of size $n^{k-1} \times n^{k-1}$ supporting output groups of size n^{h-1}. Let $C_{NIR2}(n,g)$ and $C_{RNB}(n,g)$ be the cost in terms of crosspoints for, respectively, a NIR2 network and a RNB network of size $n \times n$ and output groups of size n. Hence,

$$C_{NIR2}(n^k,n^h) = n^{k-1}n(n+2) + n^{k-1}(n+2n) + (n+2)C_{RNB}(n^{k-1},n^{h-1}) \\ = n^{k-1}(n^2+5n) + (n+2)C_{RNB}(n^{k-1},n^{h-1}) \quad (2)$$

Now consider the recursive construction of a generic RNB network of size n^k, supporting output grouping with group size n^h:

$$C_{RNB}(n^k,n^h) = n^{k-1}(n^2) + nC_{RNB}(n^{k-1},n^{h-1}) \\ = n^{k+1} + nC_{RNB}(n^{k-1},n^{h-1}) = 2n^{k+1} + n^2C_{RNB}(n^{k-2},n^{h-2})$$

Solving the recurrence equation by the substitution method, it can be shown:

$$C_{RNB}(n^k,n^h) = sn^{k+1} + n^sC_{RNB}(n^{k-s},n^{h-s}) \quad (3)$$

for $0 \leq s \leq h$, i.e. for the first h factorization levels. By setting $s = h$ and substituting k with $k-1$ and h with $h-1$:

$$C_{RNB}(n^{k-1},n^{h-1}) = (h-1)n^k + n^{h-1}C_{RNB}(n^{k-h},1) \quad (4)$$

Now, in the last factorization levels, the RNB network $C_{RNB}(n^{k-h},1)$ is a standard RNB network without grouping, which generalizes the construction of Benes networks for basic modules of size $n \times n$ instead of 2×2. It can be shown that, similarly to Benes networks:

$$C_{RNB}(n^{k-h},1) = (2(k-h)-1)n^{k-h+1} \quad (5)$$

By combining (4) and (5)

$$C_{RNB}(n^{k-1},n^{h-1}) = (h-1)n^k + n^{h-1}(2(k-h)-1)n^{k-h+1} \\ = (2k-h-2)n^k \quad (6)$$

Table 2: Cost in crosspoints for networks supporting output grouping; $N = n^k$ for recursively factorized networks.

Network	C_X
RNB-(N, n, n)	$nN + N^2/n$
NIR2-(N, n, n)	$nN + N^2/n + 5N + 2N^2/n^2$
RNB-(n^k, n, n^h), $1 \leq h < k$	$Nn(2\log_n N - h - 1)$
NIR2-(n^k, n, n^h), $1 \leq h < k$	$N((2\log_n N - h)(n+2) - 2n + 1)$

which can be used in (2) to get

$$C_{NIR2}(n^k, n^h) = n^{k-1}(n^2+5n)+(n+2)n^k(2k-h-2) = n^k((n+2)(2k-h)-n+1)$$

□

Note that Property 9 implies that, for large N, the cost in terms of crosspoints of a NIR2 with grouping grows asymptotically as $(2nN\log_n N)$. Now observe that, thanks to (6), for a RNB network with grouping

$$C_{RNB}(n^k, n^h) = (2(k+1) - (h+1) - 2)n^{k+1} = (2\log_n N - h - 1)nN$$

which tends to $(2nN\log_n N)$ for $N \to \infty$, i.e. the NIR2 network with grouping grows asymptotically as a RNB Clos network with grouping. This implies the cost efficiency of the design construction considered in our paper.

Finally, Table 2 summarizes the costs of NIR2 and RNB networks, supporting output grouping.

When evaluating the *cost reduction* achievable in NIR2 network when supporting output grouping:

Property 10. *The relative cost of NIR2 networks with output grouping is*

$$\frac{C_X(\textit{NIR2-}(n^k, n, n^h))}{C_X(\textit{NIR2-}(n^k, n))} = \frac{(2k-h)\,(n+2) - (n-1)}{(2k-1)\,(n+2)} \tag{7}$$

for any $1 \leq h < k$.

Proof. It can be easily shown that:

$$\begin{aligned} C_X(NIR2(n^k, n)) &= 2n(n+2)n^{k-1} + (n+2)C_X(RNB-(n^{k-1}, n)) \\ &= n^k(n+2)(2k-1) \end{aligned} \tag{8}$$

Now divide the cost evaluated in Property 9 by (8) to get the assertion. □

We now evaluate how (7) asymptotically behaves, when the group size scales linearly with the switch size:

Property 11. *Assume $n^h = \alpha n^k$ for some fixed $\alpha < 1$, and $k - h = -\log(\alpha)$:*

$$\lim_{k\to\infty} \frac{C_X(NIR2\text{-}(n^k, n, n^h))}{C_X(NIR2\text{-}(n^k, n))} = \frac{1}{2}$$

Proof. From (7):

$$\lim_{N\to\infty} \frac{(k - \log(\alpha))(n+2) - n + 1}{(2k-1)(n+2)} = 0.5$$

□

Property 11 implies that the maximum achievable cost reduction in NIR2 networks is 50% thanks to output grouping, which may be a relevant reduction from a practical point of view.

5. Examples of designs for NIR networks

We assume that the overall switch is built using a single basic physical module (denoted as PHY-M). We assume that a single PHY-M modules can implement either a $k \times k$ module with output/input divertability, or a $k \times (k+1)$ and $(k+1) \times k$ module with output/input divertability (needed for NIR1 networks), or a $k \times (k+2)$ and $(k+2) \times k$ module with output/input divertability (needed for NIR2 networks). Finally, we evaluate the design using PHY-M with k between 16 and 128, which are realistic values for the new technologies adopted in our scenario [12]. We can define two topologies:

- *logical topology*, describing the interconnection network in terms of the logical modules (denoted as LOG-M), obtained by the (recursive) construction discussed in Sec. 2 and Sec. 4.
- *physical topology*, mapping each LOG-M to one or more PHY-Ms, according to standard packing techniques, optimized to minimize the number of needed PHY-Ms. For example, a PHY-M can implement many (smaller) LOG-Ms by reserving different group of ports to each LOG-M. As second example, a larger LOG-M can be implemented by interconnecting many PHY-M according to a Clos or crossbar topology.

In the following results, the mapping between logical topology and physical topology has been optimized to minimize the total number of PHY-Ms, hence the final cost.

5.1. NIR networks without output grouping

We present a numerical analysis of the design of NIR1 and NIR2 networks. In particular, we investigate how different design methodologies impact on the final cost and we identify the most convenient conditions to exploit NIR2 rather than NIR1 architectures. We consider $N \times N$ networks, with $N \in [k^2, 2 \cdot 10^5]$, with k being the size of PHY-M.

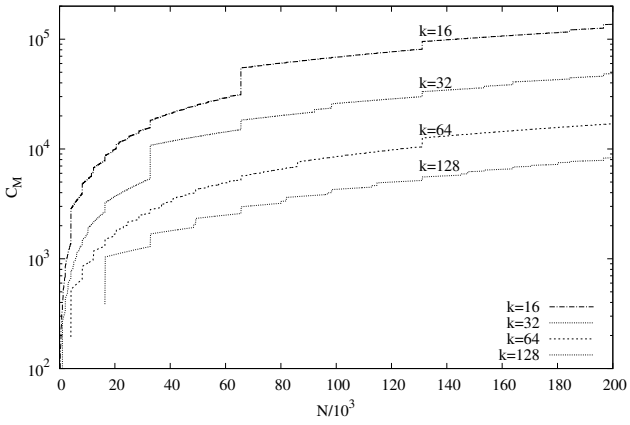

Figure 18: Number of PHY-Ms needed for NIR2-(N, k) networks.

Fig. 18 shows the cost of NIR2 networks, measured in terms of number of PHY-Ms (denoted as C_M), for different values of k and N. All the points in the graph have not been interpolated and correspond to actual values of N. As expected, a larger k implies lower cost in terms of PHY-Ms. However, the difference between different curves is variable, due to the packing strategy to map the logical topology into the physical one; indeed, PHY-Ms can have some unused crosspoints, and this level of "waste" depends on k and N. Hence, the optimal choice of k depends on the actual cost (in monetary terms) of each PHY-M and the specific values of N. Similar results have been observed for NIR1 networks, as shown in Fig. 19.

A comparative analysis between NIR1 and NIR2 networks is reported in Fig. 20, which shows the ratio between the corresponding costs in PHY-Ms. Also in this figure, the points have not been interpolated. Both architectures present similar costs, varying at most by 10%; this behavior depends on the specific values of N and k. Interestingly, for $N \geq 20,000$, NIR2 networks show either lower costs or higher costs less than 1%, with respect to NIR1 networks. This means that all the benefits of NIR2 networks (lower configuration time and output grouping support) are achieved for the network architecture with (almost) minimum cost.

5.2. NIR networks with output grouping

We now investigate the cost benefits due to output grouping. Table 3 shows the relative cost of NIR2 networks with output grouping with respect to NIR2 networks without output grouping. Note that group size equal to one means that no output grouping is supported. Furthermore, cost ratios for $g = 1$ are 1.0 by construction and they are shown here just for the sake of clearness. The first two blocks of lines refer to symmetric networks, and the relative costs have

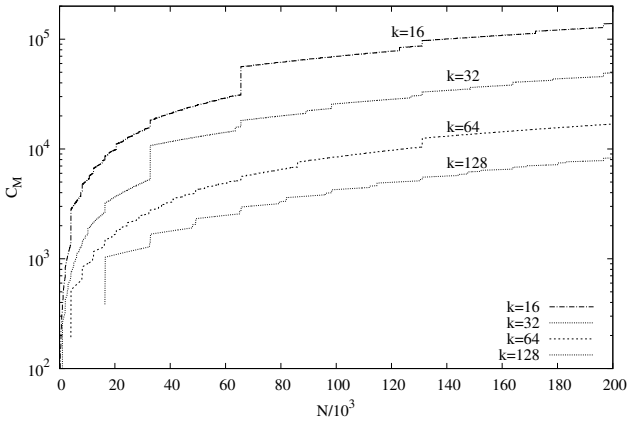

Figure 19: Number of PHY-Ms needed for NIR1-(N, k) networks.

been evaluated according to (7). The last block of lines refers to asymmetric networks, and the relative cost has been obtained by extending the methodologies discussed in the Sec. 4 for these cases. Note that the 4096×6544 size has been chosen to approximate a realistic ratio between user ports and carrier ports in operational MDFs. In all such cases, exploiting output grouping allows to reduce costs relevantly, even if the group size is relatively small. For example, in a 32768×32768 switch, by distributing the outputs in 32 groups of 1024, the final cost is just 82% of the original cost without grouping. Note that this group granularity is more than enough to support 32 different DSLAMs,

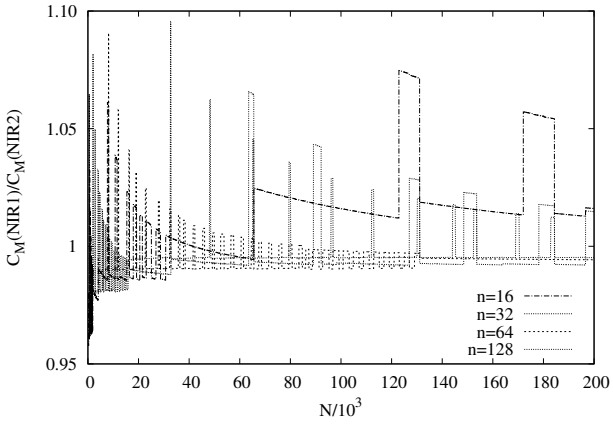

Figure 20: Ratio between the cost of NIR1 and NIR2 networks.

Table 3: Numerical examples showing the cost reduction due to output grouping in NIR2 networks.

Network N	Basic module n	Group size g	$\frac{C_X(\text{NIR2}-(N,n,g)}{C_X(\text{NIR2}-(N,n))}$
1024×1024	32	1	1.00
1024×1024	32	32	0.70
32768×32768	32	1	1.00
32768×32768	32	32	0.82
32768×32768	32	1024	0.62
4096×6544	16	1	1.00
4096×6544	16	16	0.89
4096×6544	16	256	0.60

or the classical 4-6 carriers present in a CO, in which each carrier is assigned a set of groups (with minimum granularity 1024 ports) depending on their actual service offer.

6. Conclusions

We have addressed the design of non-interruptive rearrangeable (NIR) networks in the specific scenario of next-generation large AMDFs located in the access network. We have considered two multi-stage architectures, denoted as NIR1 and NIR2 networks.

Regarding NIR1 networks, we have proposed a routing algorithm, denoted as Conf-NIR1, to minimize the configuration time. We have shown that the configuration time grows logarithmically with the size of the switch, and that NIR1 networks cannot exploit output grouping to minimize final costs. To the contrary, we have shown that NIR2 networks, if controlled by our routing algorithm Conf-NIR2, show a constant configuration time independent from the switch size, and exploit the relevant cost reduction due to output grouping.

Finally, comparing total costs of NIR1 and NIR2 networks, NIR2 costs are lower than NIR1 costs in most of the scenarios. As a conclusion of our investigations, we believe that NIR2 networks will be the reference architectures for the next generation AMDFs, outperforming classical rearrangeable networks at almost no extra cost.

7. Acknowledgements

The authors would like to thank Harry Rudin, Editor-in-Chief of *Computer Network Magazine*, for his thoughtful comments on the paper.

8. References

[1] C. Borna, "Combating customer churn - product information," *The CBS Interactive Business Network*, Mar. 2010.

[2] J. Lu, "Predicting customer churn in the telecommunications industry - an application of survival analysis modeling using SAS," in *SAS Conference Proceedings: SAS User Group International 27*, Apr. 2002.

[3] C. Lange, D. Kosiankowski, C. Gerlach, J.-F. Westphal, and A. Gladisch, "Energy consumption of telecommunication networks," in *European Conference on Optical Communication, ECOC '09*, Sep. 2009.

[4] R. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. Ayre, and W. Sorin, "Evolution of WDM optical IP networks: A cost and energy perspective," *Journal of Lightwave Technology*, vol. 27, no. 3, pp. 243–252, 2009.

[5] E. Goma, M. Canini, A. L. Toledo, N. Laoutaris, D. Kostic, P. Rodriguez, and R. Stanojevic, "Insomnia in the access or how to curb access network related energy consumption," in *ACM SIGCOMM*, Aug. 2011.

[6] A. Finamore, M. Mellia, M. Meo, M. Munafò, and D. Rossi, "Live traffic monitoring with tstat: Capabilities and experiences," in *8th International Conference on Wired/Wireless Internet Communication, WWIC 2010*, 2010, pp. 290–301.

[7] "Automating access to local loop and network facilities," White Paper, Telepath Networks, 2006. [Online]. Available: www.telepathnetworks.com

[8] S. Panattoni, "Telecom Italia view and perspectives," C5 World Forum 2007 - Networks and Technologies. Automated Provisioning of New Services: Automated Distribution Frames, Gruppo Telecom Italia, Tech. Rep., 2007.

[9] A. Nyquist, "ADF in next generation access solutions," White Paper, Network Automation, Aug. 2009. [Online]. Available: www.networkautomation.se

[10] "EZ-MDF architecture," 2011. [Online]. Available: www.simplernetworks.com

[11] F. Kaufhold, "Robotic distribution frame," White paper, UTEL Labs, 2010. [Online]. Available: www.utel.co.uk

[12] S. Braun, J. Oberhammer, and G. Stemme, "MEMS single-chip 5x5 and 20x20 double-switch arrays for telecommunication networks," in *IEEE 20th International Conference on Micro Electro Mechanical Systems*, Jan. 2007, pp. 811–814.

[13] C. Ebeling, F. Reblewski, O. Lepape, and J. Barbier, "Crossbar device constructed with MEMS switches," U.S. Patent US2010/0108479, May 2010.

[14] D. Czaplewski, G. Patrizi, G. Kraus, J. Wendt, C. Nordquist, S. Wolfley, M. Baker, and M. de Boer, "A nanomechanical switch for integration with CMOS logic," *Journal of Micromechanics and Microengineering*, vol. 19, no. 8, p. 085003, Jul. 2009.

[15] Micro magnetic latching switches for automated cross-connect systems. Telepath Networks. [Online]. Available: http://www.telepathnetworks.com/s.nl/sc.5/category.36/.f

[16] J. Hui, *Switching and traffic theory for integrated broadband networks*. Kluwer Acad. Publ., 1990.

[17] A. Johansson, "ADSL Lite - the broadband enabler for the mass market," *Ericsson Review*, no. 4, 1998.

[18] F. Hwang, W.-D. Lin, and V. Lioubimov, "On noninterruptive rearrangeable networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1141–1149, Oct. 2006.

[19] W. Kabacinski, J. Kleban, M. Michalski, M. Zal, A. Pattavina, and G. Maier, "Rearranging algorithms for log2(n, 0, p) switching networks with even number of stages," in *IEEE HPSR*, June 2009, pp. 1–6.

[20] W. Kabacinski and M. Michalski, "The algorithm for rearrangements in the log2(n,0,p) fabrics with an odd number of stages," in *IEEE ICC*, June 2011, pp. 1–5.

[21] D. Cuda, P. Giaccone, and M. Montalto, "Design and control of next generation distribution frames," in *IEEE HPSR*, July 2011, pp. 115–120.

[22] Y. Yang, S. Zheng, and D. Verchere, "Group switching for DWDM optical networks," in *IEEE ICCCN 2004*, Oct. 2004, pp. 193–198.