DESIGN AND EVALUATION OF AN OPTICAL SWITCHING NODE

A Degree's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Maite Inchausti Eceiza

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATIONS SCIENCE AND TECHNOLOGY ENGINEERING

Advisor: Jaume Comellas Colome

Barcelona, June 2014

Abstract

For a good development of elastic optical networks, the design of flexible optical switching nodes is required. This work analyses the previously proposed flexible architectures and, based on the most appropriate, which is the Architecture on Demand (AoD), proposes a specific configuration of the node that includes spatial and spectral switching and the wavelength conversion functionality with a low blocking probability and the minimum amount of modules; the characteristics of the traffic that the designed node is able to cope with are specified in the last chapter. An evaluation of the designed node is also done, and, compared to the other architectures, it is shown that the Architecture on Demand gives better results than others and that it has a higher potential for future developments.

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

telecom BCN

Resum

Per a un bon desenvolupament de les xarxes òptiques elàstiques, es requereix el disseny de nodes de commutació òptiques flexibles. Aquest treball analitza les arquitectures flexibles proposades anteriorment i, basant-se en la més adequada, es a dir, en l'arquitectura sota demanda (AoD), proposa una configuració del node que inclou commutació espacial i espectral i la funcionalitat de conversió de longitud d'ona amb una baixa probabilitat de bloqueig i la quantitat mínima de mòduls; les característiques del tràfic al que el node dissenyat és capaç de fer front s'especifiquen al final. També es fa l’avaluació del node dissenyat, i, en comparació amb les altres arquitectures, es demostra que l'arquitectura sota demanda dóna millors resultats que les altres i que té un major potencial per futurs desenvolupaments.

Resumen

Para un buen desarrollo de las redes ópticas elásticas, se requiere el diseño de nodos de conmutación ópticas flexibles. Este trabajo analiza las arquitecturas flexibles propuestas anteriormente y, basándose en la más adecuada, es decir, en la arquitectura bajo demanda (AoD), propone una configuración del nodo que incluye conmutación espacial y espectral y la funcionalidad de conversión de longitud de onda con una baja probabilidad de bloqueo y la cantidad mínima de módulos; las características del tráfico al que el nodo diseñado es capaz de hacer frente se especifican al final. También se hace la evaluación del nodo diseñado, y, en comparación con las otras arquitecturas, se demuestra que la arquitectura bajo demanda da mejores resultados que los demás y que tiene un mayor potencial para futuros desarrollos.

Laburpena

Sare optiko elastikoen garapen egokia eman dadin, beharrezkoa da kommutazio nodo optiko malguen diseinua. Lan honek aurrez proposatu diren arkitektura malguak aztertzen ditu eta, egokienean oinarrituz, hau da, eskarien menpeko arkitekturan (AoD), leku eta espektro kommutazioa eta uhin luzera bihurketa funtzionalitatea biltzen dituen nodoaren konfigurazio bat proposatzen da, blokeatzeko probabilitate baxuarekin eta gutxieneko modulu kantitatearekin; diseinatutako nodoak aurre egin diezaioken trafikoaren ezaugarriak bukaeran zehazten dira. Diseinatutako nodoaren ebaluazioa ere egiten da, eta, beste arkitekturekin konparatuz, eskarien menpeko arkitekturak emaitza hobeak ematen dituela eta etorkizuneko garapenetarako potentzial handiagoa duela erakusten da.

Hau irakurriko ez dutenei.

Acknowledgements

This work would not be in front of your eyes without the help of my project director, Jaume Comelles, whom I want to thank not only his advices, but also the interesting conversations we kept after every meeting and for always highlighting the importance of the critical spirit.

And I also want to thank every people who had to endure me meanwhile I was carrying out this work; I believe that their support and encouragement has been indispensable for its realization.

Revision history and approval record

Revision	Date	Purpose
0	23/06/2014	Document creation
1	08/07/2014	Document revision
2	10/07/2014	Document revision
3	11/07/2014	Document approval

DOCUMENT DISTRIBUTION LIST

Name	E-mail
Maite Inchausti Eceiza	minchausti006@ikasle.ehu.es
Jaume Comellas Colome	comellas@tsc.upc.edu

Written by:		Reviewed and approved by:	
Date	23/06/2014	Date	dd/mm/yyyy
Name	Maite Inchausti Eceiza	Name	Jaume Comellas Colome
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction[1]

The normalization and globalization of the usage of the Internet in our lives, the variety of new devices that can access to it using mobile broadband, the increment of the usage of mobile phones for browsing the Internet or looking to the e-mail, the evolution of contents from relatively static texts and web pages to high-bandwidth multimedia contents that require low latency (video, VoIP, etc.), and many other factors have influenced the significant growth of the Internet traffic in the last few years. And it seems that this growth will continue in the following years, if no dramatic change happens meanwhile.

Hence, when designing the networks nowadays, it's important to take this into account and make future-proof optical networks in order to be able to deal with the future high-speed super-channels.

On the other hand, the philosophy of the developers of the computer networks has been always to share the elements as much as possible, as this reduces operating costs and allows a better usage of them, avoiding the non-usage of those elements. So in a context where WDM networks are being developed, we can make profit of this philosophy as much as possible, as it has been done in the development of other technologies before. Thus, sharing the elements that compound these networks will be a key factor for reducing its costs and a good performance of them.

These goals can be achieved by means of flexible nodes that enable a good performance, sharing the elements between the incoming connections over time, and architectures that enable an easy extension of the actual node to deal with larger traffic loads. In addition, the elastic nodes will be required to provide additional functionalities such as spectrum defragmentation, wavelength conversion, regeneration, time multiplexing, etc., that will have an important effect in the performance of the whole optical network [1].

1.1. Purpose, requirements and specifications[2]

Thus, in a context where the variable and complex nature of the Internet traffic makes oversizing not efficient and neither economical, flexible nodes are required in order to give a good quality of service in an efficient way as said before.

In order to give answer to these requirements, the aim of this project is to propose an advanced flexible optical switching node that would work in this environment. Hence, a node which

---

[1] This project has been carried out at the ETSETB school placed in the Campus Nord of the UPC and the faculties of Bilbo and Donostia of EHU, using the documentation and computing resources (Internet access and software) that the schools provide.

[2] In order to simplify the work, the losses introduced by the used elements haven't been calculated, although it has been considered by way of using as few elements as possible in the design.

performs spatial and spectral switching[3] adapting to the traffic requirements dynamically, and once it is designed, show which its differential features are.

In order to reduce costs, as has been mentioned previously, the number of blocks and ports used in the design has to be as low as possible; but in the same time, the blocking probability of the incoming connections has to be also as low as possible, guaranteeing an adequate behaviour of the node.

The development of the proposed node will be based on the idea of Architecture on Demand, mainly developed in the referenced documents of this report. This idea can be defined as [2]: "*[…] an optical cross-connect (OXC) that can implement and dynamically adapt its architecture in real time in order to provide the required functionality to fulfil the switching and processing requirements of network traffic.*" Furthermore, this architecture will allow the introduction of additional functionalities: in this work we will introduce the wavelength conversion functionality in order to achieve a better performance.

So as to make an initial approach, in this proposal we will consider 4 input and output ports of the node and 160 wavelengths. Besides, the elastic optical networks work with dynamically partitioned variable-size spectrum slots. So in order to evaluate this, we will consider that the maximum bandwidth of the wavebands is 10 when we evaluate its performance (the connections' bandwidth will have a uniform distribution). We will have to pay attention to this factor, because the variable bandwidth of the incoming connections will have a considerable effect in the performance of the node as we will see later.

1.2. Work Plan

The project has been carried out in four main work packages that can be seen below, as well as the milestone corresponding to each task:

WP1: Define blocks and dimensions

WP2: Simulate different architectures

WP3: Optimization

WP4: Redaction

WP#	Task#	Short title	Milestone
1	1	Analysis of architectures	Understand their performance
	2	Definition of blocks	Have all the blocks defined
2	1	Implementation	Have the script
	2	Simulation and evaluation	Identify the weaknesses of the design
3	1	Change initial design	Achieve an optimum result
	2	Evaluation	Identify the features of the optimized result
4	1	Redaction	Memory
	2	Preparation of presentation	Presentation

Table 1: Tasks and Milestones

[3] Time multiplexing will not be a requirement.

However, the initial planning had to be changed due to some incidences: the implementation of the node in Matlab took more time than expected in the beginning (because I started with a more complex design that further we saw didn't influence so much on the performance, and a simpler design will provide more efficient results), so the finalization of the project has been delayed considerably.

The initial and resulting Gantt diagram are the following, the initial in green and the resulting in blue:

Tasks	MONTH	March		April				May					June				July	
	WEEK	17-23	24-30	31-6	7-13	14-20	21-27	28-4	5-11	12-18	19-25	26-1	2-8	9-15	16-22	23-29	30-6	7-11
Analysis of architectures																		
Definition of elements																		
Implementation																		
Simulation +evaluation																		
Optimization +comparison																		
Redaction																		

Table 2: Gantt diagram

2. State of the art of the technology used or applied in this thesis:

As optical WDM networks are being developed, several works that examine flexible architectures and the possibilities for their design and synthesis have been published recently. The bibliography consists basically in these reports that can give us an idea of what a flexible architecture can be.

In those reports several flexible architectures have been exposed, but having a look to them we can observe that the architecture that has gained importance is the Architecture on Demand (AoD). In the following lines we will see why this has happened and what differences it from the other proposed architectures. After that we will describe the suggested models to design a node basing on the AoD.

2.1. Flexible architectures

In the reports [1] and [3] different flexible architectures are presented, as well as a new method to measure their flexibility, where they link the concepts of maximum entropy (determined by factors such as traffic load, traffic requirements, etc.) with the node's flexibility.

We will not focus on this new method, but in the presented architectures, which can be seen in the following figure [1]:

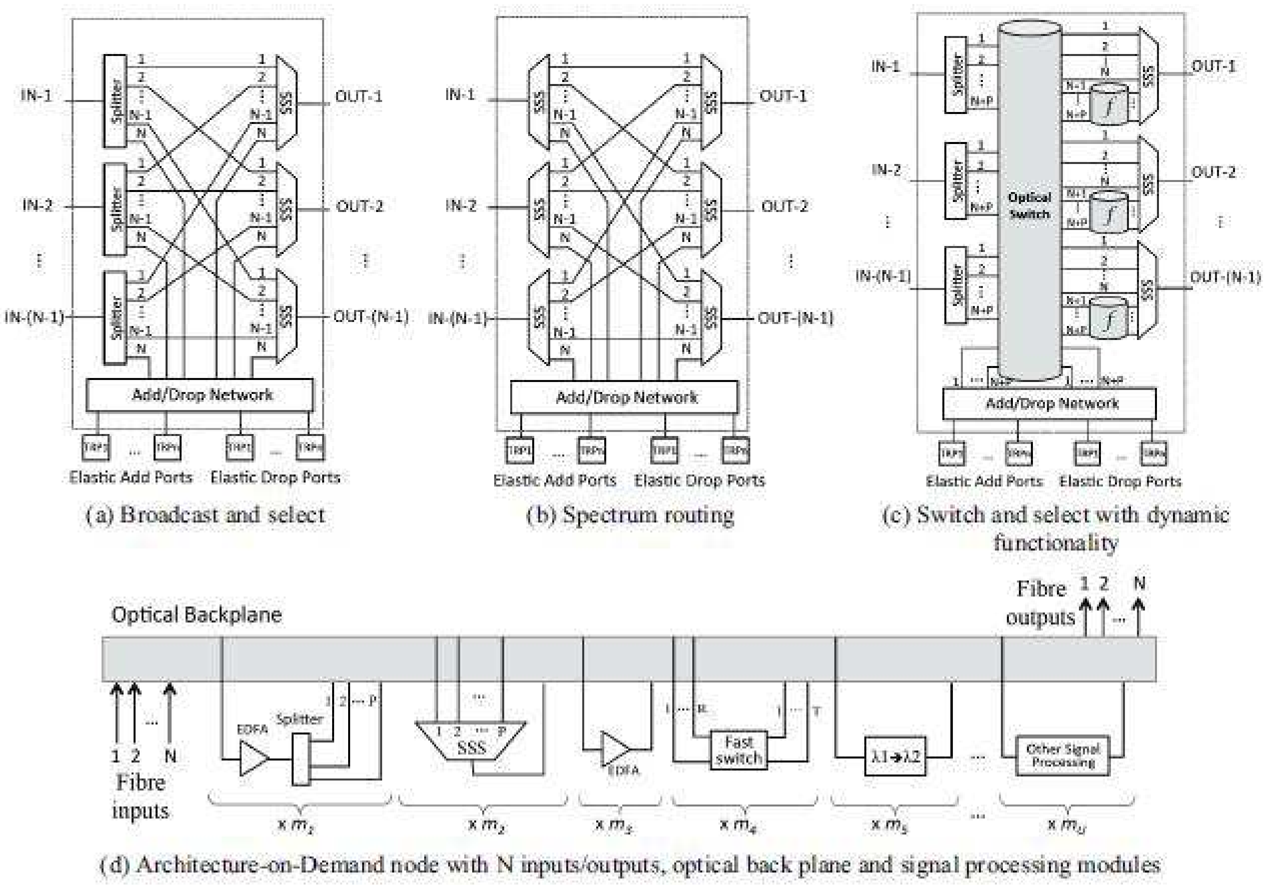

Figure 1: Elastic optical node architectures

They all can be considered elastic because the SSS modules and the optical switches can change their configuration dynamically depending on the incoming traffic's requirements. Even though all of them are flexible, we can preview that not all of them will have the same flexibility, neither the same features. As it can be seen, the Switch and Select architecture and AoD allow placing more modules that will improve the node's behaviour, adding more functionalities to cope with dynamic requirements; something that seems difficult to introduce in the other architectures.

Another differential factor would be the introduced losses, but as said in the introduction in this work we will not focus on them. Just say that introducing an optical backplane will introduce additional losses.

However, with the AoD different arrangements of inputs, modules and outputs can be constructed by setting up appropriate cross-connections in the optical backplane; providing greater flexibility than the other architectures as the components used for optical processing (SSS, power splitters, etc.) can be interconnected together in an arbitrary manner, in a way that it allows sharing the available modules between the connections from any input port. Also, on an AoD node, the required number of SSS ports is not so strictly related to the node degree, as several small port-count SSS may be cascaded if required [1].

On the other hand, if we make a flexibility comparison based on the capability of the architectures to offer additional functionality on demand, we will see that such on-demand functionality is not supported by the BS nor the SR architectures, and that the SS architecture offers on-demand functionality but having the modules that provide it hardwired to the node's outputs.

So, concluding, we will base our flexible node on the AoD architecture, as it is the most flexible architecture and allows sharing the placed elements, reducing the amount of needed elements.

2.2. Synthesis

Once explained why the AoD architecture has been selected to design the optical flexible switching node, we will focus on the synthesis of it. In order to make an idea of how to organize it, we can look to the model exposed in Figure 1 again; but it does not give so many details. However, in a report published afterwards the authors of the previous work specify more the idea and give us a model where we will base the design [2]:

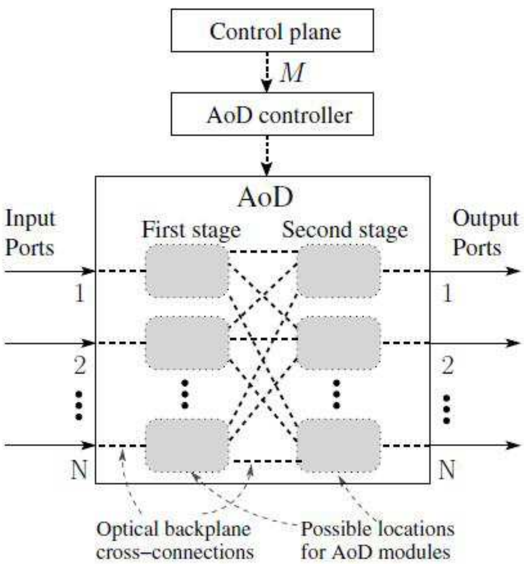

Figure 2: AoD model

In the Figure 2 we can observe that several stages can be placed in order to achieve different functionalities. The optical backplane allows making the cross-connections that requires each incoming connection, and as we said in the previous section the elements are shared between all the incoming ports.

The model focuses on a circuit switching scenario, where wavebands are considered as aggregates of adjacent wavelengths; although we have to take into account that the used modules will be different if we consider that there are wavebands in the system.

We can also see that we need a control plane that will decide which the actual configuration is depending on the incoming traffic. Hence, to build up the architecture, fast and efficient synthesis and selection algorithms that take into account the availability of modules and traffic requests will be necessary [2]. This will be taken into account when writing the script that will simulate the designed node.

In the second part of the mentioned report a specific synthesis technique is proposed. However, it does not use wavelength converters, and we can preview that the usage of wavelength converters in the node will improve its performance. Therefore, although we can take it as an example, the designed node in this report will be different from it.

So, taking in mind the ideas given by other authors about the nodes based on AoD, we can start designing our model.

3. Project development:

In this chapter the chosen structure for the design of the node will be explained, as well as the key parameters for its simulation in the used software, Matlab. After defining the structure of the node that will be used, the parameters that may influence on the behaviour of the node will be defined for the following analysis.

This analysis will provide then the information that will allow deciding which will be the appropriate dimension of the elements for a good performance and which will be the characteristics of the traffic that this node would be able to cope with.

3.1. Structure of the node

As explained in the previous chapters, the design of the node has been based in the AoD architecture. In its realization we have considered several elements: the optical backplane –which is the main differential element of this architecture-, the wavelength converters and the SSS.

The wavelength converters, the SSS and the input and output fibers of the node are all connected to the optical backplane, and the cross-connections are set to connect the different elements dynamically.

As for simplicity in this work the power losses aren't calculated, we will not use amplifiers, and we can consider that in terms of functionality the SSS and the MUX and DEMUX modules are the same (even though in reality the MUX and DEMUX modules will not perform properly with wavebands [2]). Because of that, the modules that are connected to the backplane are compound of single elements; but in practice, if we consider the power losses and if other functionalities have to be added, composed modules can be connected as well.

In the Figure 3 we can see the structure described before:

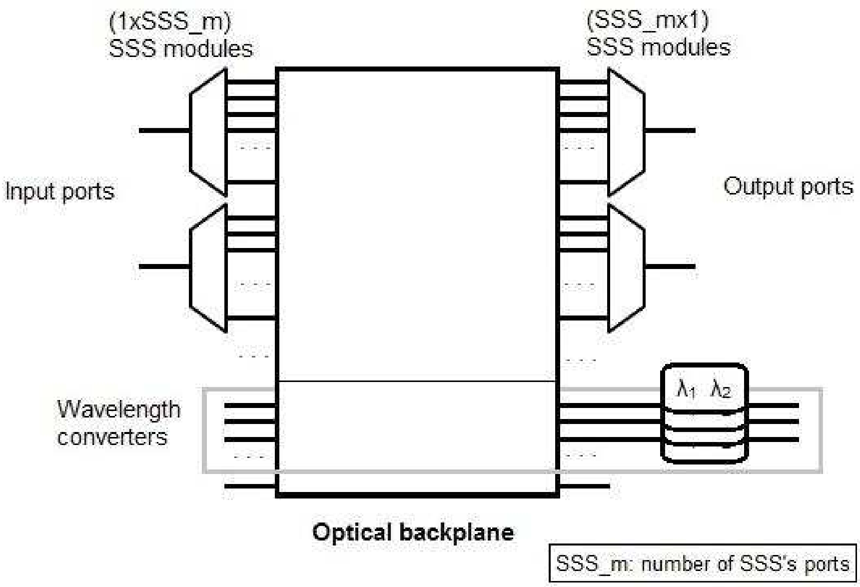

Figure 3: Node's structure

As it can be seen, we can put different modules and the incoming connections can use all of the available ones when the connection request arrives: they share the modules, so the modules are not related to a certain physical input or output port neither to a certain wavelength.

On the other hand, the cross-connections allow changing dynamically the configuration of the node, making it a flexible node as explained before.

3.1.1. The optical backplane

The optical backplane can be considered a big switch, which connects the input and output ports with all the modules by means of cross-connections.

Technologically, for its realization active components based on micro-electromechanical systems (MEMS) are typically used: in [3] a large port count 3D MEMS is mentioned as the optical backplane as an example.

To simulate it on the Matlab tool, we need to formulate it mathematically: we can implement the switch like a matrix (S) where s_{ij} is equal to one for positions we want to switch, being *i* the desired output and *j* the input we want to switch; the rest of the values will be zero. Multiplying the input matrix by the S matrix we will obtain the desired output matrix, with the input rows indicated in the S matrix changed of position.

3.1.2. The wavelength converters

The wavelength converters displace the information of a certain wavelength to another wavelength. In this model we consider that these wavelength converters tune in the central wavelength of the incoming traffic and change it to the desired one; we also consider that they work with wavebands as well.

So the only thing that the function of Matlab that model them does is to displace the incoming traffic to the desired output wavelength.

3.1.3. The Spectrum Selective Switches (SSS)

The SSS modules are the key elements to give flexibility to architectures, thus, as it have been seen before in the chapter 2, all of the exposed architectures work with SSS modules (and most of them use SSS modules for both switching and filtering functionalities). These devices are very proper for flexible nodes, as they feature a fine spectrum granularity that enables the implementation of highly customizable filters with variable bandwidth [1].

They can work in two directions: in one, they work like a MUX as mentioned before (they switch the spectrum slices that contain the signals that require to be passed to a certain output); and in the other direction, they work like a DEMUX (they select the spectrum slices that require to be passed to the common output port from the appropriate input and block the remaining spectrum).

This is achieved by means of control information that indicates the transfer function of the module, which is dynamically programmed by the control plane shown in Figure 2.

In this work, we have considered the model of Wavelength Selective Switches (WSS) to simulate these modules in Matlab. So we have used the model of the WSS that we can see in Figure 5, where the WSS is compounded by three kinds of modules: MUX, switches and DEMUX [4]. It has been made so as to be directional: so the Matlab function detects if it has to work as a 1x*SSS_m* module (compound by 1 DEMUX, *w* switches and *SSS_m* MUX) or as a *SSS_m*x1 module (*SSS_m* DEMUX, *w* switches and 1 MUX).

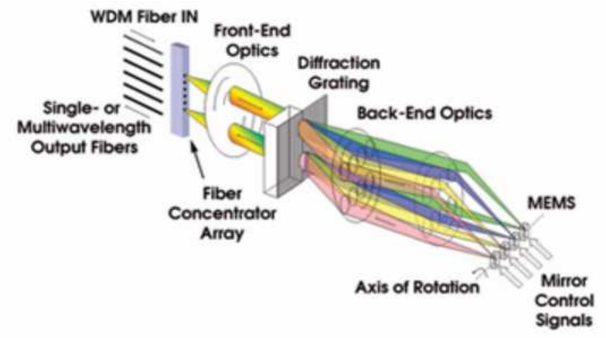

Figure 4: Practical model of a WSS [5]

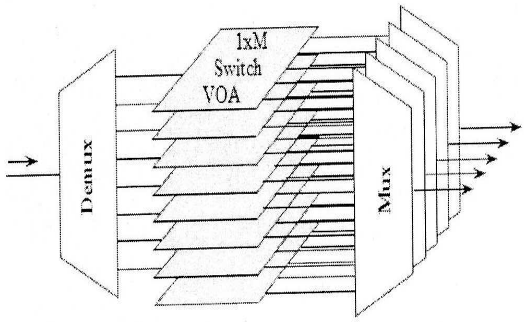

Figure 5: Model of a WSS for simulation

As it can be seen in Figure 5, the three kinds of modules are connected in order to achieve the desired functionality. The DEMUX are modeled by turning the input array into the diagonal of a *w*x*w* matrix, and the MUX as 1x*w* switches, where all the components of the array are equal to 1; we can be sure that there will be no interference as the switches block the input signals that cannot be switched.

The switches block the signals depending on the control information they receive, which they also use to select the desired information that they are going to switch. In the Matlab tool this is made forming the components of the matrix of the switch with the control information received.

On the other hand, we can see that the switch's matrix will be different depending on the direction of the SSS. If it is a 1x*SSS_m* switch, the dimensions of the matrix will be *SSS_m*x1, and vice versa.

3.2. Main parameters

In order to implement this structure shown in Figure 3, two main parameters have been used: *offsets* and *using_ports_B*.

The *offsets* parameter is a 2x2 matrix, and each position indicates the offset port of each kind of module. The number of ports of the backplane –saved in the 2x2 matrix *ports*- that uses each set of modules is shown in the following table:

ports	**Ports used in the input[4]**	**Ports used in the output[5]**
Input/output ports	n	m
Wavelength converters	n_y	n_y

Table 3: Number of used backplane's ports by each kind of module

Knowing these values, we can obtain the offset input and output ports of the backplane for each kind of module (in this case we use just wavelength converters as shown in Figure 3). Thus, we can define the *offsets* parameter –a 2x2 matrix following the structure of the table-:

offsets	**Offsets for the input**	**Offsets for the output**
Input/output ports	1	1
Wavelength converters	ports(1,1)+offsets(1,1)	ports(1,2)+offsets(1,2)

Table 4: Number of the offset backplane's port used by each kind of module

We will use this last parameter to create the *using_ports_B* parameter. This will be a matrix where we indicate which the backplane's ports that each connection uses are. So it let us know which the modules that each connection is using in every moment are, and using it we can set the needed cross-connections in the backplane for a good performance of the node.

In this configuration, each connection can use at most four ports of the backplane: one at the input and another at the output, and another two to connect to a wavelength converter if needed. So the *using_ports_B* parameter will be a 2x2 matrix, indicating in the first row the input of the backplane and in the second, the output. Using this parameter we will be able to set the needed cross-connections of the backplane setting B(*using_ports_B*(2,i,k),*using_ports_B*(1,i,k))=1, being B the matrix that models the backplane, *i* a counter from 1 to 2 (in this case) and *k* the number of connection, calculated in this way:

$$k = first\ wavelength + w * (input\ number - 1)$$

3.3. Parameters to study

The main parameters that will influence on the behaviour of the node will be the amount of elements that compound the node and their dimensions, and the characteristics of the traffic.

As it have been explained in the previous section, the number of SSS modules that will be used will be fixed, as it is the same as the amount of the input ports in the input of the backplane and equal to the amount of output ports in the output of the backplane, as it can be seen in Figure 3. So respect to the SSS modules, the parameter that will be necessary to analyse will be their number of ports.

We can see that the minimum number of ports is equal to the number of input or output ports of the node in each side of the backplane, in order to be able to rout all the connections directly. Then, depending on the number of wavelength converter modules that we have, the needed number of SSS modules' ports will be different. We will have to choose the minimum possible that guarantees a small blocking probability.

[4] With *input* I refer to the left side of the optical backplane.
[5] With *output* I refer to the right side of the optical backplane.

So, as it can be guessed from the previous paragraph, the number of wavelength converter modules will be an important parameter to analyse too. We can appreciate that this parameter is what makes difference between the AoD architecture and the BS, SR and SS architectures, as in the BS there is no option to put wavelength converters because it has no element to separate the different spectrum slices, and in the SR and the SS we would need 4 times more modules (4 is the number of input ports of the node) to achieve the same effect on the behaviour.

However, the dimensions of the modules that will be used are directly related with the optical backplane port count; the more modules we have and the more ports the used modules have, the more ports we will need in the optical backplane to connect all of them. So if we chose a certain port count for the backplane, the number of modules and their dimensions will be limited by this factor. Therefore, the number of ports of the backplane will be a factor to analyse as well.

In the other hand, the features of the incoming traffic will be also crucial parameters that will determine the behaviour of the node. The incoming connections have several statistic parameters: the incoming port and destination port, the first wavelength and the bandwidth are uniform distributions; the inter-arrival time between two connections is a Poisson's distribution and, finally, the duration is an exponential distribution.

Dividing the duration with the inter-arrival time we will obtain the traffic load parameter, which will be a parameter to analyse. Theoretically, if the traffic load is high, the blocking probability will increase. We will see later if this happens with the designed node or not.

In addition, the bandwidth of the connections will be an important parameter too. The following equation, where *w* is the number of wavelengths of the system and *n* is the number of inputs of the node, shows its influence on the number of connections that are alive in the node:

$$E\{number\ of\ connections\} = \frac{w * n}{E\{BW\ of\ connections\}} = \frac{160 * 4}{5.5} \cong 116\ connections$$

The conditions imposed in the specifications involve that, approximately, at any time we will have 116 alive connections in the node, 29 in each incoming port (the incoming port of the new connection is random with a uniform distribution) as it can be seen in the previous expression.

We can appreciate that if we increase the maximum bandwidth of the connections, the number of connections that can keep alive the node will decrease and vice versa. So, as said before, it will be a parameter to study.

3.4. <u>Synthesis Algorithm</u>

When a new connection arrives to the node, the control plane has to decide whether it can switch the new connection or not, and if it can switch it, which configuration it is going to establish in order to switch it.

To allocate modules, cross-connections and to create the control information that is needed for an appropriate behaviour of the SSS modules, the control plane has to launch a synthesis algorithm that takes these decisions.

The algorithm that the designed node follows is relatively simple: we can classify the decisions in two groups, the decisions that have to be taken when new connections require to be switched to a certain output, and, on the other hand, the decisions that have to be taken when connections that were being switched by the node end.

When a new connection arrives, first of all, the control plane has to decide to rout it or not depending on the spectral slices that are available in the output port that the new connection desires to reach. If physically it is possible to switch it, the actual connections that have the same destination have to be analysed in order to see if the connection needs to use a wavelength converter. If it needs it, the control plane will have to look if there are available, and if there are, allocate the resources and update the parameters that indicate which the available modules are. Whether using wavelength converters or not, the SSS modules control information and using ports needs to be configured as well.

In the other case, when it detects that some connections have ended, the control plane has to release the resources that the old connections were using and look whether any wavelength converter that an alive connection is using can be removed or not, in the case that the alive connection needed to change the central wavelength due to any old connection.

Following these steps, the control plane allows sharing the wavelength converters between the different connections over time when needed, using them efficiently.

4. Results

In this chapter we will see the results of the analysis of the different parameters mentioned in the chapter *Parameters to study*, and basing on them we will decide which the most efficient configuration of the node is, concluding its design. After that we will evaluate the designed flexible optical switching node's performance with traffic of different characteristics in the input.

4.1. Design of the node

First of all we will compare the three different architectures presented previously (AoD, BS, SR), using the AoD architecture without wavelength converters, and for a maximum bandwidth of connections of 10 wavelengths. We can see that functionally this configuration is the same as the BS and the SR. The results of the simulation support this initial hypothesis as shown in Figure 6.

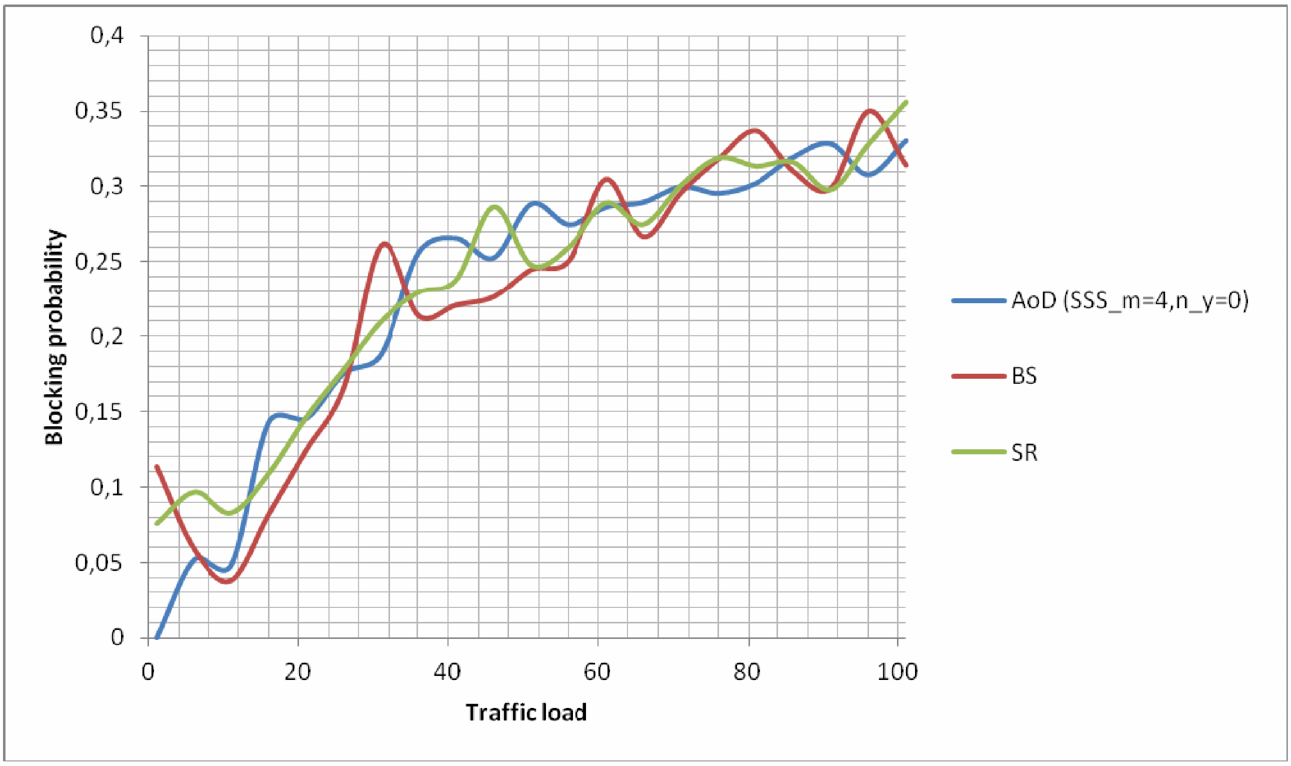

Figure 6: Comparison of the three architectures, with AoD without wavelength converters

We can notice that this result can be improved placing wavelength converters. The question would be how many wavelength converters we need in order to achieve a good performance, which can be considered as a blocking probability around 0.2.

To solve this question we will analyse how the number of ports of the SSS modules and the amount of wavelength converter modules influence on the blocking probability. To do this we will consider a traffic load of 100. After making several simulations this graphics have been obtained:

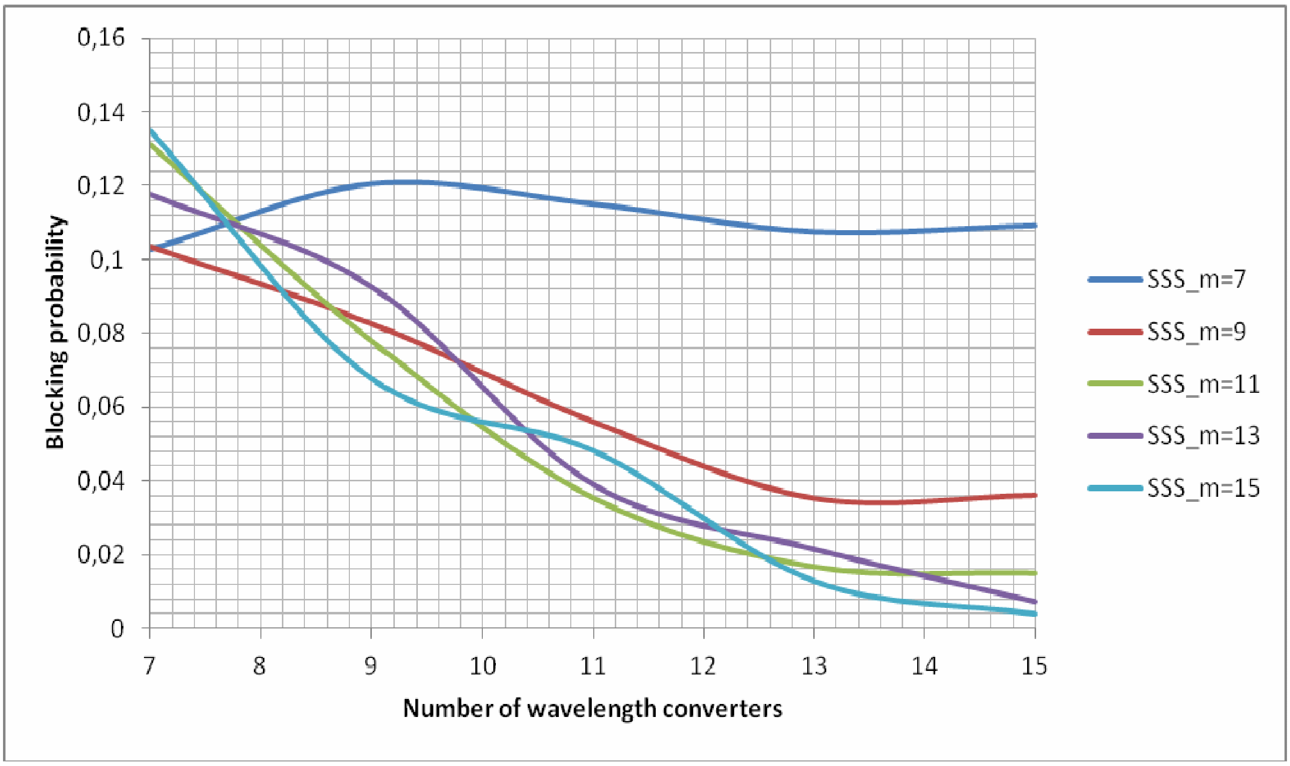

Figure 7: Blocking probability depending on the number of wavelength converters, for different amount of SSS modules' ports

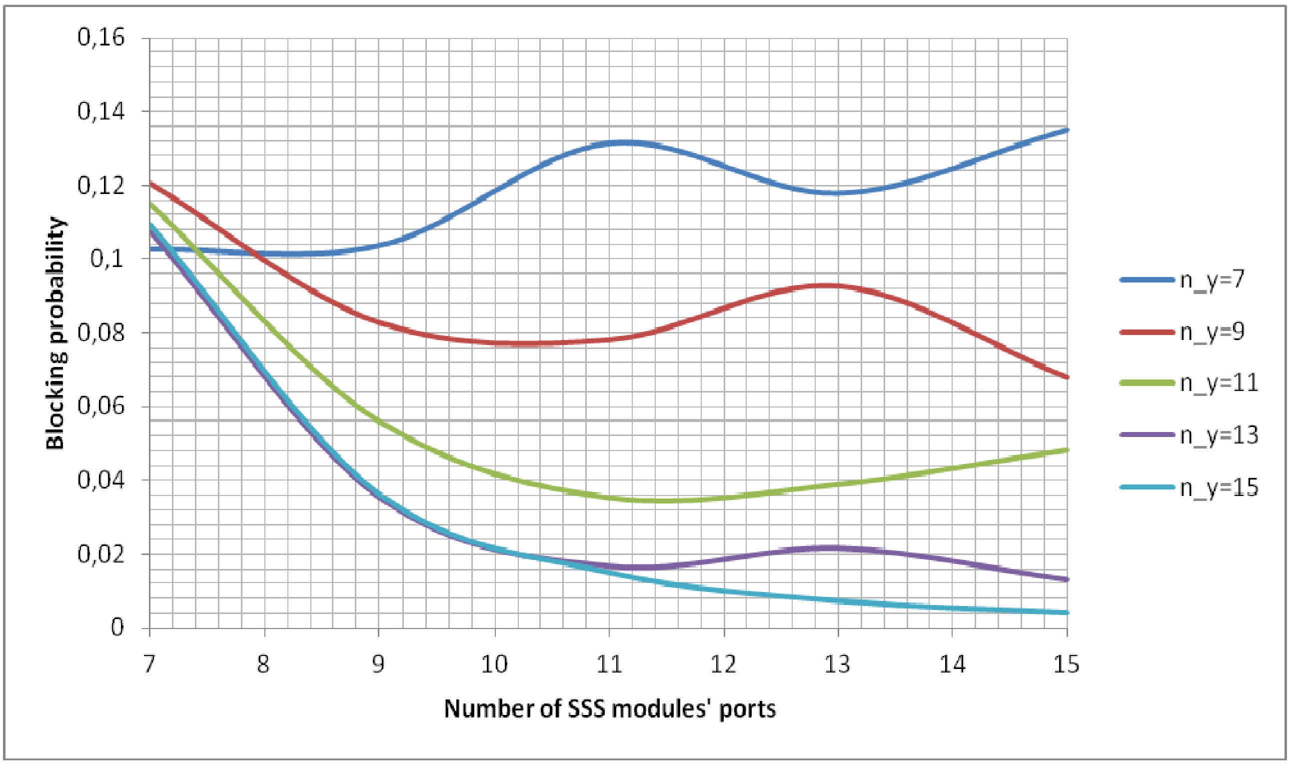

Figure 8: Blocking probability depending on the number of SSS modules' ports, for different amount of wavelength converters

As we can see in Figure 7, the number of SSS modules' ports doesn't influence so much if there are up to 7, and up to 11 ports the resulting blocking probability is more or less the same. This happens because although we have more wavelength converters than SSS modules' ports, not all the wavelength converters are used in the same input; thus, not all the available SSS modules' ports are used. Therefore, **the minimum number of SSS modules' ports for a good behaviour will be 11**.

However, if we look to Figure 8 we can see that the number of wavelength converters used in the node has a considerable influence on the blocking probability until 13 wavelength converters; up to this amount the improvement is not so high, and we can see that 13 wavelength converters are enough to obtain a blocking probability around 0.2. So we conclude that **the minimum number wavelength converter modules for a good behaviour will be 13**.

Once selected the dimensions of the modules that we are going to use, we have to observe which dimensions the optical backplane will need to have in order to connect all the selected modules that will allow achieving an optimal performance of the node. The minimum port count needed will be the summation of all the modules' ports. As we have four SSS modules of 11 ports each and 13 wavelength converters, the optical backplane's minimum port count will be:

Backplane's minimum port count = 4 *SSS modules**11 *ports/each* + 13 *w_converters**1 *port/each* = 57 *ports*

Knowing this parameter we will be able to select the appropriate optical backplane. In the literature, normally port counts of 96 or higher are mentioned; with the calculated minimum port count, we can see that a 96x96 3D MEMS [3] will be able to connect all the desired modules, and using it we have the choice to connect more modules or even more input/output fibers in the future, upgrading the node's nodal degree.

Thus, summarizing, the designed node of nodal degree 4, which 160 wavelengths in each fiber, that performs spatial and spectral switching and with wavelength conversion functionality for an optimum behaviour for traffic load below 100 and maximum bandwidth of 10 wavelengths, will be compounded by connecting this components with the arrangement described in the chapter Structure of the node:

- An **96x96 optical backplane**, which can be a 96x96 3D MEMS
- **8 SSS modules of 11 ports**: 4 in the input of the backplane and another 4 in the output, with the 11 ports connected to the backplane and the port of the other side to the input/output fiber
- **13 wavelength converters**, with the input port connected to the backplane's output and the output port to the backplane's input

Now that we have the optimal configuration, we can repeat the initial comparison between the different flexible architectures, and we will see that the performance of the node based on the AoD architecture has improved substantially, obtaining a much better behaviour.

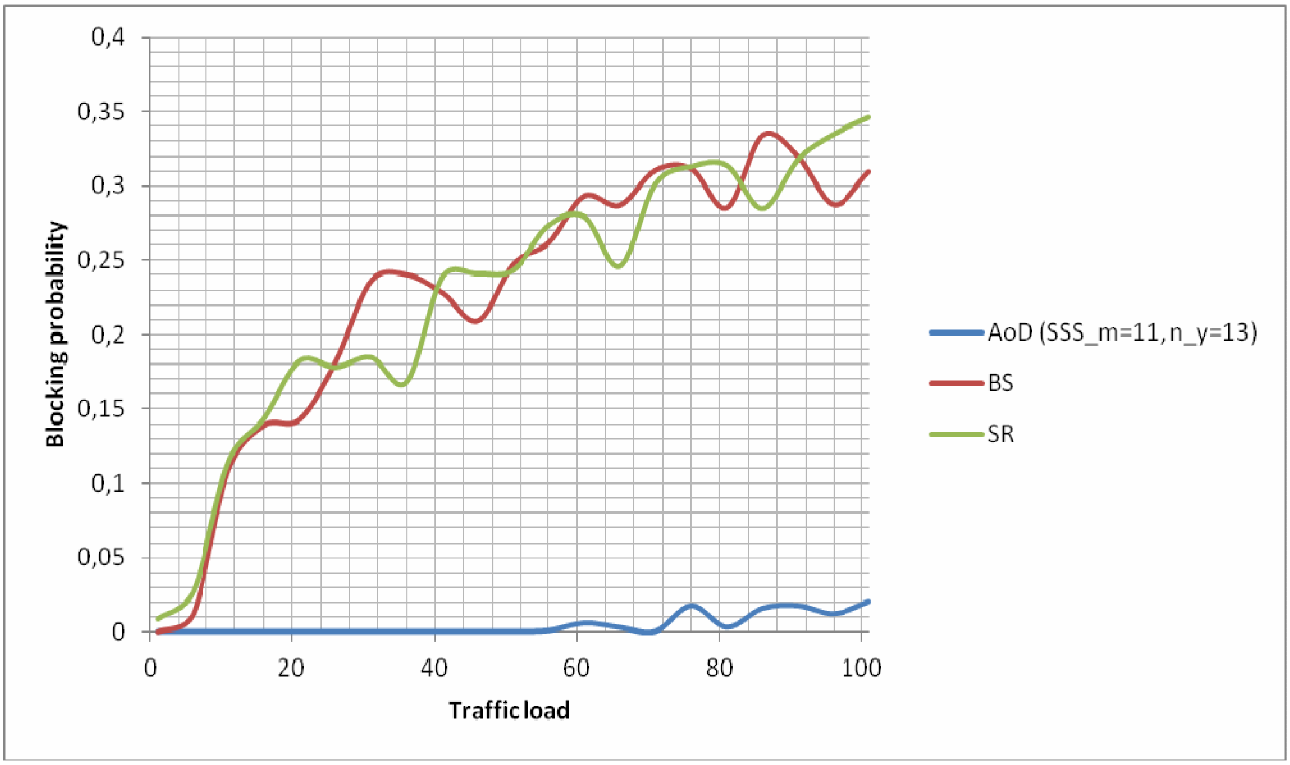

Figure 9: Comparison of the three architectures, with AoD with wavelength converters

4.2. <u>Evaluation with traffic of different nature</u>

Once we have obtained a good configuration of the flexible node that accomplishes the specifications established in the beginning, we will see how this node deals with traffic of different nature. To show this, we will generate traffic with different maximum bandwidth and different traffic loads, and simulate the behaviour of the node when it has to switch them.

Making different simulations with traffic with different characteristics in the input, we can appreciate that having a fixed configuration changes its behaviour depending on the incoming traffic's characteristics as predicted. In the following graphics the behaviour of the designed node (based on AoD architecture and with 13 wavelength converters and 11 SSS modules' ports) is shown depending on the maximum bandwidth of the incoming connections and the traffic load.

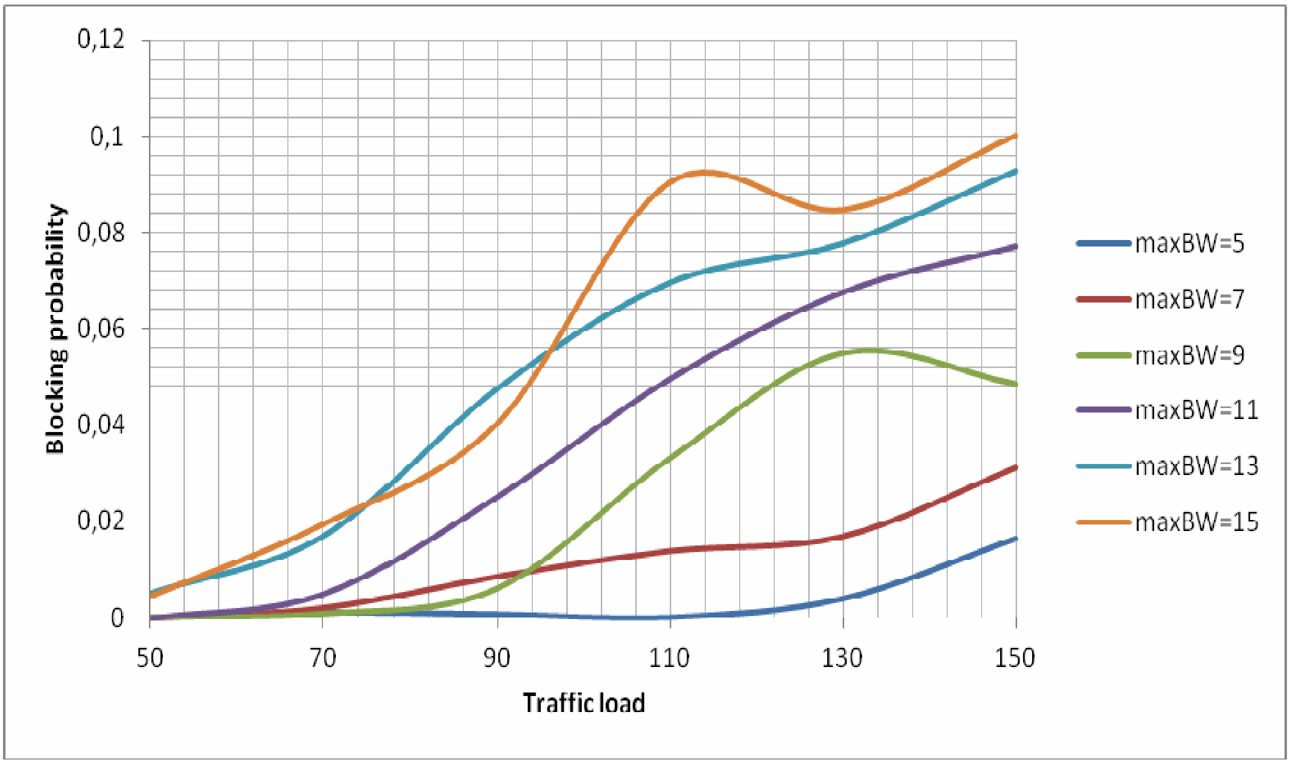

Figure 10: Blocking probability of the designed node depending on the traffic load and maximum bandwidth

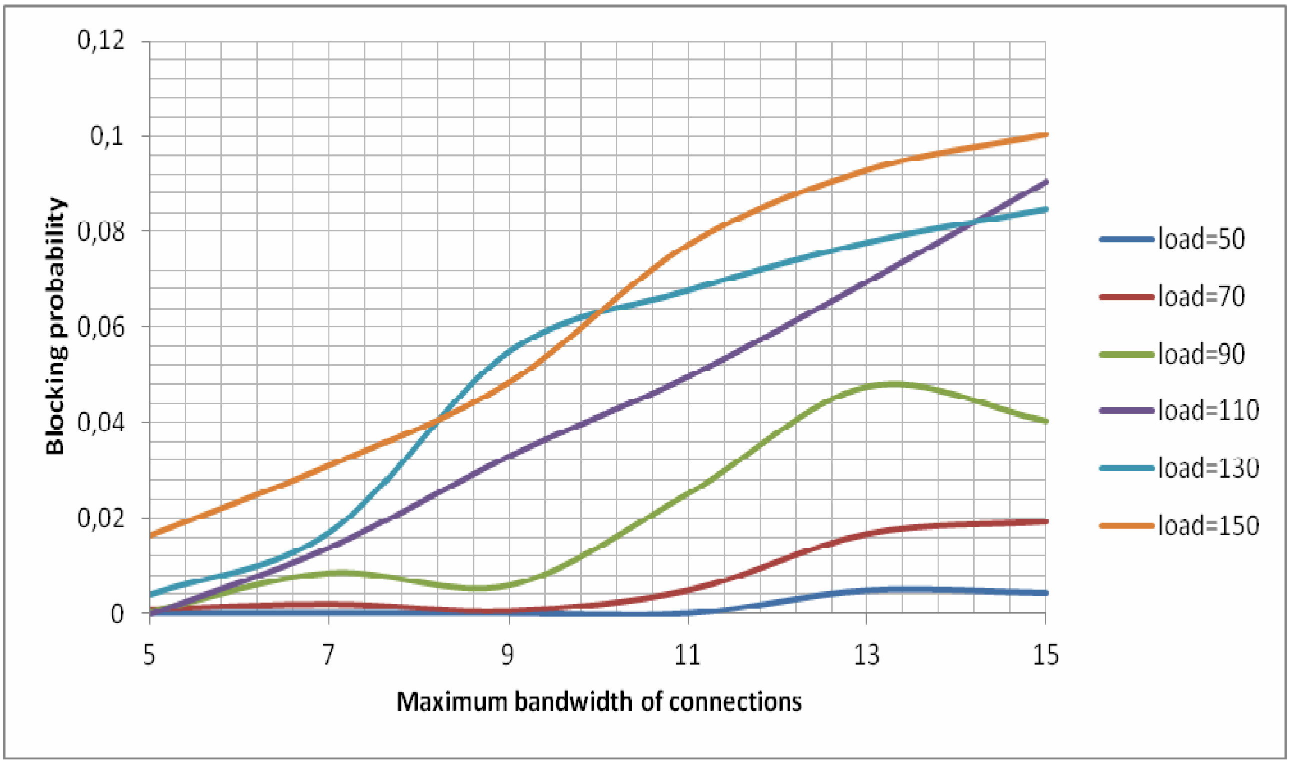

Figure 11: Blocking probability of the designed node depending on the maximum bandwidth and the traffic load

For the chosen configuration, the traffic load 90 can be routed till a maximum bandwidth of 11 more or less, and traffic load 110 till a maximum bandwidth of 8 more or less; it is coherent with the tests that we have done before, which indicated that with a load of 100 and maximum bandwidth of 10 the achieved blocking probability was around 0.2 approximately.

We can also observe that if the traffic load is low, the node can rout all the different maximum bandwidths; and if the maximum bandwidth of the connections is low, the higher traffic load's can be routed as well. The contrary happens when the traffic load or the maximum bandwidth is high: only the traffic with low maximum bandwidth and low traffic load can be switched.

Concluding, we can see that establishing a certain configuration limits the features of the incoming traffic that the node can cope with. As the design has been done for a traffic load of 100 and maximum bandwidth of 10 wavelengths, we can see that the node's behaviour is good below the specified values, but if both parameters are higher than the established, the blocking probability is higher than the acceptable, thus, 0.2.

The increase of the traffic load can be coped by placing more wavelength converters on the designed node. Thus, if the node needs to deal with higher traffic loads than 100, the simulations made in the previous section should be repeated and the new optimal configuration decided, based on the obtained graphics.

However, the increase of the maximum bandwidth influences on the number of alive connections that the system can keep, as seen in the chapter Parameters to study:

$$E\{number\ of\ connections\} = \frac{W * n}{E\{BW\ of\ connections\}}$$

So this factor is physical, and placing more wavelength converters we wouldn't achieve significant improvements as the problem would be the lack of space.

The graphics give much more information, but in general we see that as traffic load increases and the maximum bandwidth of the connections increases, the blocking probability increases as well; something that we could expect.

In conclusion, depending on the features of the incoming traffic we have, we will have to decide if this node is suitable looking to this graphics; and if we see that the node has to cope with higher traffic loads, increase the number of wavelength converter modules in order to improve its performance.

5. Budget

In order to realize this work no physical component has been used. Therefore, to make the economical budget the factors to take into account are two: the cost of the software used, in this case Matlab, and the human work. The final cost of this work can be seen in the following table:

	Weeks	Estimation of the number of hours	Cost per hour (junior engineer)	Cost
Define blocks and dimensions	4	100	8€/hour	800€
Simulate different architectures	8	200	8€/hour	1600€
Optimization	3	75	8€/hour	600€
Redaction	2	40	8€/hour	320€
Software	11	-	-	1000€
TOTAL	-	415	-	**4320€**

Table 5: Economical budget

6. Conclusions and future development:

In this work we have seen that a flexible architecture such as AoD gives very good results, giving the chance to create flexible nodes that change their configuration dynamically depending on the requirements of the incoming traffic. This allows sharing the elements between the different ports, thus, there's no necessity to place elements for every port; the placed elements can be used by any port. This involves costs reduction, as the amount of elements needed for a certain quality of service reduces considerably: by a factor of the number of input ports. It also allows configuring additional on-demand functionalities, improving the features of the node.

However, this project is a simple step that proves the improvements that this kind of nodes can contribute, and it's clear that more improvements can be achieved placing more stages and more kinds of modules, modules compound by a single element as well as modules compound by more than one elements.

Because, on the other side, it must be considered that several factors haven't been analysed in this work; factors that would be interesting to analyse, but which would extend this work and it had to be time limited.

Basically, these factors can be summarized in the possibility and necessity of placing more kinds of elements. Possibility in the one hand because placing more elements will give the chance to add functionalities to our node (for example time multiplexing can be included), and on the other hand necessity, because the factor of losses has not been analysed although placing elements would influence on the power of the connections; thus, it should be considered the possibility of adding optical amplifiers in the node.

In order to make these improvements on the node, a basic script has been made in this work; taking this like a base, making changes will be easier than starting from the scratch. If new modules have to be added in the future works, the amount of backplane's ports that each connection can use will be higher (hence, the parameter *using_ports_B* should be changed), and it would be necessary to add an extra section that corresponds to the new element when allocating the elements to the new connection and when operating the element in the final section of the "node.m" script. In the other hand, if new parameters have to be considered, they would have to be assigned to each connection, as it has been made with the first wavelength of the connection, bandwidth, etc. And the synthesis algorithm will have to be changed as well, creating one that includes the allocation and realise of the new modules.

Thus, basically, we can say that a basic design and script have been set, and modifying it will allow achieving new improvements and new features for the designed flexible optical switching node.

Bibliography:

[1] N. Amaya, G. S. Zervas, D. Simeonidou. "Optical Node Architectures for Elastic Networks: From Static to Architecture on Demand". *In 2012 14th International Conference on Transparent Optical Networks (ICTON)*. 2-5 July 2012, Coventry, UK. pp. 1-4. DOI: 10.1109/ICTON.2012.6254473.

[2] M. Garrich, N. Amaya, G. Zervas, P. Giaccone, D. Simeonidou. "Architecture on Demand: Synthesis and scalability". *In 2012 16th International Conference on Optical Network Design and Modeling (ONDM),* 17-20 April 2012, Colchester, UK. pp. 1-6. DOI: 10.1109/ONDM.2012.6210271.

[3] N. Amaya, G. Zervas, D. Simeonidou. "Introducing Node Architecture Flexibility for Elastic Optical Networks". *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 6, pp. 593-608, June 2013. DOI: 10.1364/JOCN.5.000593.

[4] T. A. Strasser, J. L. Wagener. "Wavelength-Selective Switches for ROADM Applications". *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 5, pp. 1150-1157, September/October 2010. DOI: 10.1109/JSTQE.2010.2049345.

[5] M. Nagy, S. Tibuleac. "Wavelength Selective Switches for Fiber Optic Telecommunications". *Photonics Spectra*, 2006. [Online] Available: http://photonics.com/Article.aspx?AID=27188. [Accessed: 7 April 2014].

[6] M. Garrich, J. R. F. Oliveira, M. Siqueira, N. Amaya, G. Zervas, D. Simeonidou, J. C. R. F. Oliveira. "Flexibility of Programmable Add/Drop Architecture for ROADMs". *In Optical Fiber Communication Conference*, 9-13 March 2014, San Francisco, USA. Pp. 1-3. DOI: 10.1364/OFC.2014.W1C.2.

Glossary

AoD: Architecture on Demand

BS: Broadcast and Select

SR: Spectrum Routing

SS: Switch and Select

MUX: Multiplexer

DEMUX: Demultiplexer

MEMS: Micro-electromechanical Systems

SSS: Spectrum Selective Switches

WSS: Wavelength Selective Switches

SSS_m: Number of SSS modules' ports

w: number of wavelengths used on the system

n: number of input ports of the node

n_y: number of wavelength converters