Simulation of All-Optical Networks Study Case: Minimum-Deflection Routing

Armando Borrero Molina, Andrés Arcia-Moret, Anna Lezama

***Abstract* – This paper presents the design and implementation of an all optical networks simulator. It has been developed for teaching purposes through a minimum-deflection routing algorithm applied on a mesh network. Using a *Poisson* distribution, the packets arrive into the system to simulate data flow.**

The simulator is highly configurable. Each node may serve as an input (with its accessory queue), an output or both. The amount and the location of these nodes can be adjusted, as well as the number of iterations carried out by the simulation, and the packet generation. This allows the user to evaluate and grasp quite a variety of simulation scenarios.

***Key words*— All-optical networks; simulation; algorithm; deflection routing.**

INTRODUCTION

The users' need to obtain more information through the network in less time and a lower cost has increased nowadays. Moreover, thousands of new users wish to access all that information simultaneously. New technologies, such as all-optical networks, have emerged in order to meet this need. In fact, all-optical networks have gained recent relevance due to their high potential for providing vast bandwidth and great speed. Optical transmission strives to use the development and growth of electronic technology to overcome its self-imposed limitations. As it is well known, packets in all-optical networks circulate as optical/luminous signals called photons actually, which are the constituent particles of luminous radiation.

An all-optical network is a very particular type of network. The main difference from other kind of networks is that all-optical communication support is made of optical fiber cables. Also, signals go through the network point-to-point only in the optical domain, without converting to the electrical domain and vice versa. Thus, these networks must face a great inconvenient: there are no efficient optical memories that allow the information to be stored within the nodes. Some methods have been implemented as an alternative to temporary storage, adapting very well to this type of network. That is the case of deflection routing [1].

Deflection solves the contention at nodes level, by temporarily sending some of the packets to an unwanted destination. If several packets compete for an output loop, only the packets within the loop's capacity will take the desired output. The rest of the packets shall be diverted towards another available output. That is, they will be deflected, even when it means sending the packets to a route that does not necessarily approximates them to their final destination [2], [3]. Note that, all the information packets remain in constant movement.

A. All-optical network

The evolution of optical networks can be divided into two generations [3]. The first generation corresponds to the exclusively usage of optical fiber cables for point-to-point transfers. These are called optoelectronic networks, as they carry O/E/O conversion (*Optical-to-Electrical-to-Optical*). The second generation includes all-optical networks, since the information is always dealt with in the optical domain, from origin to destination. In other words, they avoid employing time in conversions, having low latency and a vast bandwidth. Therefore, dealing with the information flow from point to point exclusively in the optical domain will allow the implementation of high speed networks in the near future [4].

B. Deflection routing

Deflection routing was developed by Baran in 1964, and later tested and adapted for all-optical networks, which do not use intermediate storage devices. This routing is a simple way of solving the so called "dispute problem". If a packet does not find a favorable port on a particular node, then the packet will be sent to any available output port. This packet is sent to a destination different from the one originally intended for it, but the loss of packets is thus avoided. This type of routing can be used, as the number of input and output ports is the same. Therefore, each input packet will always find an available output port.

The optical packets go through several optical switches before reaching their appointed final destinations. Hence, it is important to introduce some sort of mechanism to settle the dispute, having an important effect on the network's performance. In electronic networks this problem is solved through storage memories, routing and "*store-and-forward*" shipping. Electronic packets are kept in the RAM (*Random Access Memory*), where they remain until they can be

---

Research Article.

A.B.M., A.A.-M. y A.L. are with the University of Los Andes, located in La Hechicera, Faculty of Engineering. Mérida, Mérida, Venezuela, Contacts: +58-274-2402811/ 2824, Fax: +58-274-2402890, E-mail: borrero@ula.ve, andres.arcia@ula.ve, lezamaanna@ula.ve.

A.B.M. is part of the Telecommunications Investigation Group (GITEL, by its acronym in Spanish).

A.A.-M y A.L. are with the Networks and Distributed Systems Investigation Group (RESIDE, by its acronym in Spanish).

forwarded back to the network. In the case of an optical packet switch, there is not an optical memory and deflection routing has emerged as a natural alternative for this type of networks. An intuitive explanation would be that nodes engage the entire network as a big *buffer*, and the disputing packet gets diverted to the rest of the network. This technique allows multiple packets to reach their destination without any loss.

II. PREVIOUS STUDIES

The minimum-deflection algorithm provides a route assignment strategy for packets circulating in all-optical networks. For other heuristics, Borrero [1] has also performed evaluations; along with examples for various deflection algorithms in this kind of networks. Among all evaluated algorithms, minimum-deflection results the more illustrative of what the problem of routing in a bufferless network is all about. Moreover, Zambrano and Borrero [5] present further evaluation and implementation by simulation in OMNET++ of algorithms described in [1].

As we will see in this section, previous studies have considered the deflection algorithm not only in all-optical networks for cost savings [12] but also as a bufferless strategy in other contexts. However, all of them agree in deflection as a basic building block for routing in bufferless networks. In all-optical networks, related work focus on the evaluation of some heuristics and the results of the deflection algorithms *per se* as a packet routing strategy for various traffic generation functions. On the other hand, Fallin et al. [11] propose an implementation of a deflection algorithm for packet routing for networks on chip (NoC). In this case, NoC share similarities with optical networks on the high cost for storing and forwarding packets inside the network. The authors have shown that by combining deflection with other strategies to deal with an inherent reordering, they show important savings in power consumption by non-significantly degrading the performance within parallel processors.

He and Gary [6], studied the effect of some active queue management schemes, such as *weighted fair queuing* (WFQ) on the input buffer to improve the prioritization of different classes of traffic on a deflection routed all-optical network. The delay introduced by the deflection algorithm has been mitigated by strategies such as the aggregation of packets or assignation of priority to control packets (e.g., TCP-ACKs). Then, authors emphasize that although the deflection technique is a promising one for core network routing, other mechanisms should be employed to mitigate reordering effects.

Kozlovszky et al. [9] show simulations in OMNET++ using fiber links as “buffers”. This technique enables traffic through all-optical networks with measurable and acceptable levels of loss.

The network in our study, just as the one in previous studies, refers to a *core* network (i.e., central network that carries very large amounts of data, with large capacities and high speed). This kind of network represents a common network for tier-1 service providers [5]. In previous revisions of the deflection algorithm, the need to provide quality service in all-optical networks has been mentioned emphasizing that selecting an appropriate routing strategy is the key to optimal results. Thus, our interest on considering modeling the different process for simulation of deflection routing in all-optical networks to gain insight in the routing process. Moreover, to the best of our knowledge, there is no available module for all-optical networks simulation for the major free-licensed simulation tools, namely OMNET++ and ns-2.

III. SIMDROP: THE ALL-OPTICAL NETWORK SIMULATOR FOR TEACHING PURPOSES

The topology chosen for the simulation model is a 2D net, as shown in Fig. 1, mainly due to its simple implementation and its approximation to a real topology [5]. This topology consists of nodes and links in synchronous disposition and it does not have any storage or processing device in the intermediate nodes. Only the input nodes have a queue with all the packets that want to enter the net but cannot do so immediately. Moreover, a queue does not have a limited fixed size.

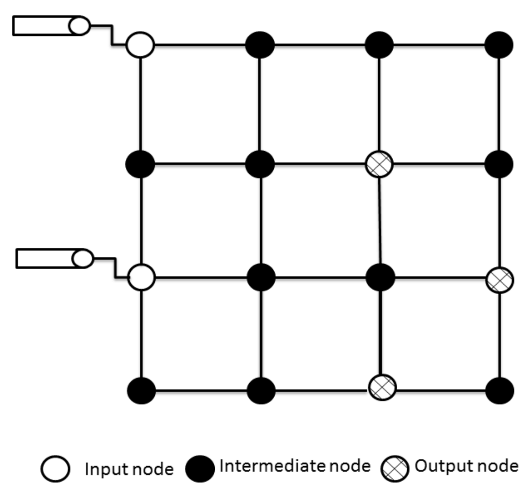

Fig. 1 2D network topology used for the simulation model

As can be seen in Fig. 1, in the model network the nodes are arranged as a grid (for any n there is n x n nodes). This disposition is based considering it is a *core* network in which performance is the key to its functioning with bandwidths hovering around hundreds of megabits per second.

Each node is in charge of passing the packet to the following node according to the *Minimum-Deflection* algorithm (see Fig. 3). This algorithm will behave depending on the number of pre-configured iterations and the packet-generation rate.

The minimum unit in the system is the packet, which transits from one node to the other within the net through an optical link between the nodes (Fig. 1). The packet generation answers to a Poisson distribution with random seed. This guaranties an amount of traffic that allows the observation of the deflection algorithms.

The simulator was developed with the general-purpose programming language, C++, the GUI with the Qt library [7] and the network what modeled using graphs defined in the *Aleph* library [8]. For the particular case of this simulation, and because they are all-optical networks, it is assumed that there are no packet losses due to corruption within the net and eventually each packet is capable of reaching the output node. However, the delay and the reordering inherent to deflection may affect a packet.

The class diagram that supports the simulator is described in Fig. 2.

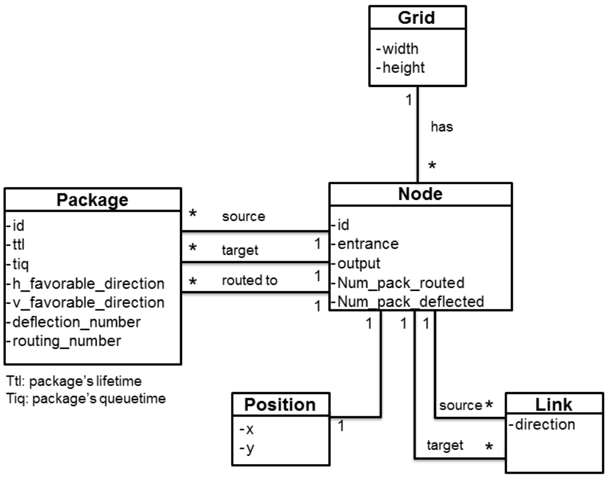

Fig 2.

Class diagram for the simulator

A. The simulation process

Below we find the processes with the higher impact in the simulator.

a) Execution of the minimum-deflection algorithm: Through this process, the routes for each packet within the 2D net are assigned. This means that a packet has to be moved from the output node (in each stage) to the corresponding link.

b) Moving Packets: It consists on the mobilization of packets from the link they have been assigned by the deflection algorithm to the destination node.

c) Packet generation: It consists on going through all the input nodes and generating an amount of packets according to the Poisson distribution, average chosen by the user.

d) Stop condition: The simulation process ends when one of the following conditions is met. Either the counter of the number or routing iteration reaches its end, or all the packets placed at the input have exited the network.

The minimum-deflection algorithm showed in Fig. 3 is used to route the packets from one node to the other [1], [5]. It tries to send a packet in the most convenient direction depending on the packet's destination. If the route is occupied it is sent on another direction, this being what is known as 'deflection' (described in *I.B.*). Below is an algorithm adaptation applied to the simulator that has been developed.

This algorithm is applied in each of the nodes and in each stage of the simulation.

B. Considerations about routing

Some properties, inherent to the net, were taken into consideration when modeling the routing process, as follows:

- Each node possesses four directions at most, which are: North, South, East and West.
- A packet may have one (1) or two (2) convenient directions, being two (2) the maximum.
- A packet cannot have as convenient direction two physically opposite directions. That is, a packet cannot have as convenient directions North and South or East and West at the same time. The directions a packet may take, have to be near and adjacent. For example: North and West or South and West.
- Packets have only one convenient direction if their current node is in the same row or column that the destination node. Therefore, in all the other cases there will be two convenient directions.
- For a *p* packet, if its convenient direction or directions have already been taken by another *q* packet coexistent in the current node, then the *p* packet will be sent to another node that does not form part of the shorter road to its destination. In consequence, *p* will be deflected.
- The number of coexisting packets within the node (taken from the queue) depends on both the capacity of the link (one packet per link in our case) and the number of available links. That is, for a central node (with four output links) we could potentially accommodate four different routes for four different packets.

C. Minimum-Deflection Algorithm

A distributed algorithm that deflects the packet or packets passed onto the node is presented in Fig. 3. It aims towards the minimum number of deflections during its stay in the net. As can be observed on lines 5, 8 and 16 of Fig. 3, each node tries to send the packets to their optimal routes or the shortest path to their destinations. This path is potentially the one with the least number of deflections.

```
1  ∀ node (n) ∈ grid
2  While n has packets
3   p is the current packet
4   i is the number of favorable directions of p
5   if (i=0) then
6      Deflect p to the first available link
7      Return While
8   if (i=1) then
9      D is the only favorable direction of p
10     if link to D is unavailable then
11        i ← 0
12        Return While
13     Routed p to D
14     Return While
15  if (i=2) then
16     DH is the horizontal favorable direction
17     if DH is available then
18        Route p to DH
19        Return While
20     DV is the vertical favorable direction
21     if DV is available then
22        Route p to DV
23     else
24        i ← 0
25 End While
```

Fig. 3. Minimum-Deflection Algorithm with minimum capacity

Fig. 4 presents the detail of a routing in a 2D net. Links *a* and *b* conveniently approach the packet about to be routed to its destination node, that is, to the North or to the East. In other words, for inconvenient directions West or South (links *c* or *d* in Fig. 4), the packet is being deflected.

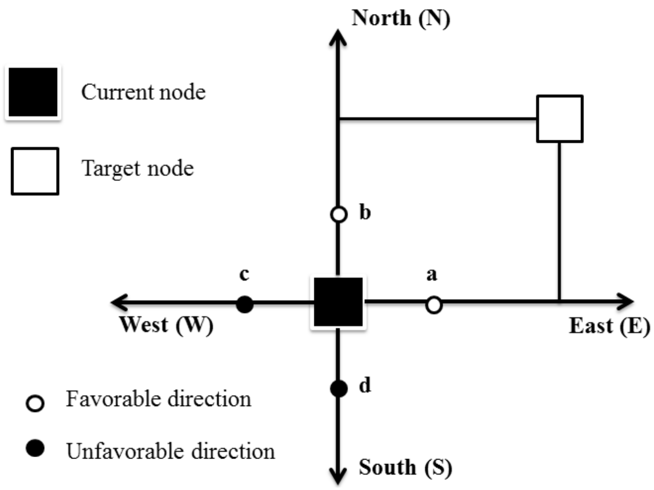

Fig. 4 Routing on a 2D net

D. Simulation Parameters and Performances Measurements

In this section is dedicated to defining the simulation parameters and the measurements related to monitoring the performance of the simulations within the optical network.

Parameters:

1. *Number of Iterations:* At the beginning of the simulation the user must define the number of iterations correspondent to the expected amount of routing operations.

2. *Number of Packets Generated:* Total amount of packets that will be generated in all the inputs during the whole simulation. Packet generation responds to the Poisson distribution.

Performance Measurements:

1. *Number of Queued Packets:* Amount of packets that could not enter the net and were left on queue at the end of the simulation limited by the number of iterations.
2. *Number of Delivered Packets:* Amount of packets that arrived to their destination after the simulation.
3. *Number of Circulating Packets:* These are the packets left circulating the net at the end of the simulation limited by the number of iterations.
4. *Average time for a queued packet:* Average number of iterations that a packet stays in queue.
5. *Average time for a circulating packet:* Average number of routings applied to circulating packets. In other words, it refers to the number of simulation stages that a packet has remained in the net, or the time it takes a packet to get from its origin to its destination. It is also called packet "age".
6. *Deflection Frequency:* On a table, it displays the amount of deflections that have occurred during the simulation. This measure does not only stipulate the maximal amount of deflections occurred but also counts the total of packets that incurred by each frequency for every integer number of deflections (in a range from 0 to *MaxDeflections*).

IV. SIMULATION TESTS

The simulator has a net modeled with a data structure correspondent to a Euclidian graph with matrix coordinates, in which there are nodes and links.

The *nodes* have counters for the packets that have transited through them and a list of the packets in them at any given iteration. Specially, the input nodes also have a queue which storages the packets that are about to enter a net.

The *links* have the source node and the destination node. With it, it is possible to know the way of the direction within the net and thus determine the link's availability.

The *packets* are the minimum data unit to be considered by the simulator. They have information of the destination node, which does not change from when it was generated and of the previous node which changes with each routing iteration. The packet also has its own counters for routing and deflection used for the measurements presented in Section III.D.

There are only two movements for any given packet in the net: *routing* and *deflection.* Routing involves passing a packet from one node to another available and convenient one, while deflection is the passage of a packet from one node to another available one but in an inconvenient direction as shown in Fig 5.

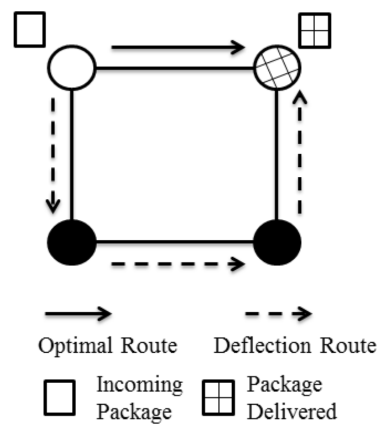

Figura 5: Routing Behavior

The simulator allows the user to set the height and width of the net, as well as the average of packet arrival according to the *Poisson*. Before starting the simulation, the input and output nodes are set.

A. Step by Step Simulation:

For illustration purposes, a 2x2 net with one input node and one output node behaves as follows:

As shown in Fig. 6, the first iteration occurs at the arrival of three packets labeled A, B and C according to the arrival rate set by the user. Packet A is *routed* east, this being its only convenient direction. Then, even though packet B is headed in the same direction as A, it is *deflected* south since its convenient direction is currently occupied by packet A (which has priority). Packet C is queued at the input node until it finds an available link in the next iterations.

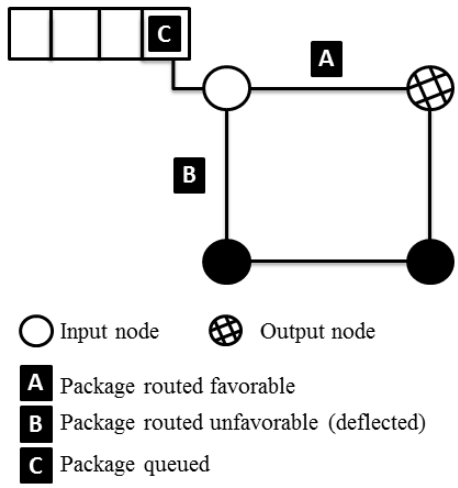

Fig. 6 First Iteration of the Simulator

On the next iteration, displayed in Fig. 7, packet A is delivered, packet B follows the deflection route and C goes from being queued to being *routed*. The arrival of other packets labeled D and E causes D to be inconveniently deflected while E is queued.

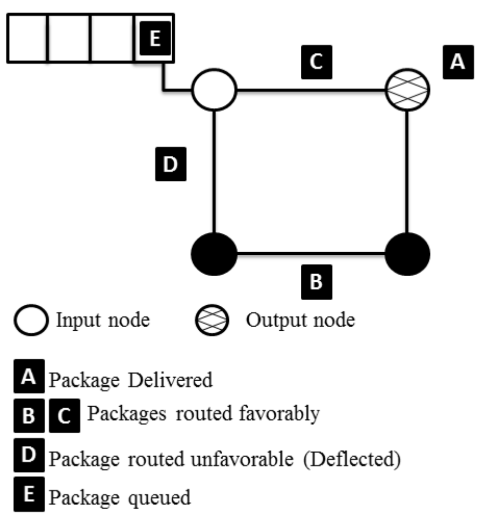

Fig. 7 Second Iteration of the Simulator

Packet C is delivered on the third iteration. Packet B is routed to the output node. Packet D is still on the deflection route and packet E, which was queued, is now finally *deflected*. This behavior continues in the same way on a net with those characteristics.

To determine which packet will be deflected next, it has been established that priority goes to the packet already circulating within the net over those which are just arriving. This way, a packet wanting to enter the net must wait for its input node to route or deflect one or more packets (internally) and finally allow entrance to the new packet.

B. Simulation Evaluation.

This section presents some thoughts after having carried out several simulations with different random seed maintaining the same simulation parameters.

As the net size reduces, the queues get larger. However, the number of available outputs in a node (according to its location on the net) seems to be a determining factor in this phenomenon. Therefore, an input located with four possible routing links (at the center of the net) shows queue sizes similar to the one located at the edges (with 2 and 3 outputs).

Having input queues proved to be beneficial. However, setting them to create unnecessary waiting must be avoided. On the other hand, we also know the low-capacity queues can generate packet loss. Thus, a well-chosen queue size can cooperate with the deflection algorithm in optimizing the routing within an all-optical network.

One of the possible policies for input queue design corresponds to the calculus of its size proportional to the optical network.

The net size was varied during different experiments. When there were between one and three input nodes and the same number of output node, it was observed that packets were never queued but actually flowed quite fast. It was observed that when the net size is reduced maintaining the same number of input and output nodes, the number of queued packets increases.

Fig 8. presents the decrease of queued packets related to the total number of packets on the simulation as the net size increases.

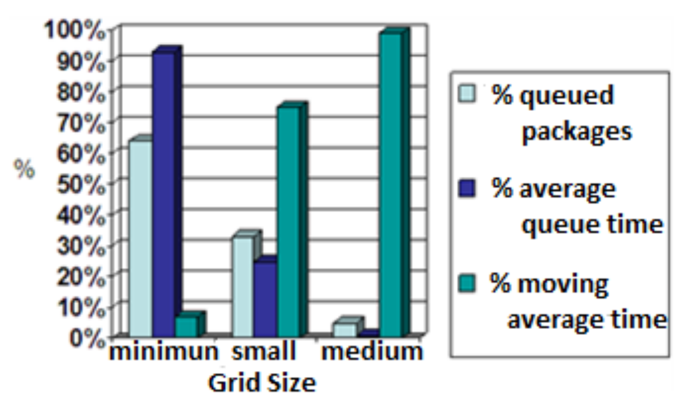

Fig. 8. Percentages of time and number of queued packets related to matrix size

Minimum nets are those which dimensions are between dos and four; small matrix have dimensions between four and six, whereas medium matrix would have dimensions over seven. Let it be noted that the classification of simulated nets is for test and teaching purposes. In practice then, a small all-optical network corresponds to an approximate of fifty nodes.

Finally, during the different simulations it was observed that traffic on the edges turned out to be more fluent and less prone to deflections that traffic to the central nodes. That is, central nodes on the net experienced more congestion.

V. Conclusions

This article presented a simulation model for all-optical networks implemented in the C++ language. Also, the deflection algorithm it was implemented and explained in detail through a graphs model.

As for the functioning of the simulator we have observed that: a) As the size of the nets increases, there is less occupation of the input queues, due partly to the net itself acting as a memory and downloading the packets quickly from the input; b) the packets located at the center of the net experience more deflection and, in consequence, more congestion.

The architecture of the simulator that was developed allows easy incorporation of different routing algorithms.

The capacity of defining irregular topologies can be provided on future studies. We also consider worthy of study other phenomena within optical topologies, providing the simulator with larger capacity links and asynchronous processing capacity.

Even for other topologies such as bus topology where extensive advances have been conducted on simulation for Ethernet [10]

Acknowledgements

On this section we wish to express our gratitude to the Computing Postgraduate Program at the Universidad de Los Andes and its members, where several discussions where held with the students in the Computer Network class.

References

[1] A. Borrero Molina, "**Simulation of all-optical networks with deflection routing**" Consejo de Publicaciones de la Universidad de los Andes. Serie Akademia, Julio 2004.

[2] Kozlovszky, Miklos, Berceli, Tibor and Kozlovszky, Viktor, "**NSOSS: the non-synchronized optical switch simulator"** ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools: 1—6. 2007

[3] Rajiv Ramaswami, Kumar N. Sivarajan Galen H. Sasaki. "**Optical Networks. A Practical Perspective**" Tercera Edición. Morgan Kaufmann, Elsevier. 2010.

[4] Barth D., Bethomé P., y Cohen J. "**The Eulerian Stretch of a Digraph and the Ending Guarantee of a Convergence Routing**". Reporte Técnico. LRI. *Université* Paris-Sud, Orsay. France. 2000.

[5] F. Zambrano, A Borrero Molina, "**Selección de Heurísticas para asignación de rutas a paquetes de información en una red completamente óptica usando enrutamiento por deflexión**". Universidad de los Andes, Facultad de Ingeniería, Escuela de Sistemas, Diciembre 2006.

[6] Jingyi He and S.-H. Gary Chan, "**TCP and UDP performance for Internet over optical packet-switched networks," Department of Electrical and Electronic Engineering**, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Abril 2004.

[7] Blanchette D. y Summerfield M. "**C++ GUI Programming with Qt 4**". Prentice Hall. 2008

[8] León, L. "**Tejiendo Algoritmos**". Universidad de los Andes. Consejo de Publicaciones. Mérida. 2012.

[9] Kozlovszky, M., Berceli, T., Kovacs, G. and Kozlovszky, V., "**Measurements on Optical Buffering Models Made from Fiber Delay Lines**", MMS 2007: Mediterranean Microwave Symposium and 12th Microcoll Conference. 2007

[10] Steinbach, Till, Kenfack, Hermand Dieumo, Korf, Franz and Schmidt, Thomas C. "**An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy**", SIMUTools 2011 -- 4th International OMNeT++ Workshop. 2011

[11] Fallin, C. Craik, C., Mutlu, O. "**CHIPPER: A low-complexity bufferless deflection router**" High Performance Computer Architecture (HPCA), IEEE 17th International Symposium. 2011

[12] T. Chich, P. Fraigniaud, and J. Cohen. Unslotted deflection routing: a practical and efficient protocol for multihop optical networks. IEEE/ACM Transactions on Networking, 2001.