

Selección de heurísticas para asignación de rutas a paquetes de información en una red completamente óptica usando enrutamiento por deflexión

Selection of heuristics for assigning routes to information packages in an all-optical network using deflection routing

Borrero, Armando* y Zambrano, Frank

Departamento de Ciencias Aplicadas y Humanísticas, y Departamento de Computación
Facultad de Ingeniería, ULA.
Mérida 5101, Venezuela
*borrero@ula.ve

Recibido: 17-06-2007

Revisado: 11-03-2008

Resumen

En este trabajo se propone el desarrollo de un simulador usando la herramienta OMNeT++, con el fin de evaluar distintas estrategias para el enrutamiento por deflexión de paquetes de Información, para las emergentes generaciones de redes (Redes Completamente Ópticas). Se presenta el Algoritmo de Mínima Deflexión (AMD) con la aplicación de heurísticas apropiadas, y la evaluación del desempeño de esta estrategia de enrutamiento, en una red síncrona 2D (tipo malla) con funciones de tráfico uniforme y no uniforme. En este sentido, se calcularon algunas estadísticas que permiten evaluar mejor el rendimiento de este tipo de redes, con la utilización del algoritmo de enrutamiento y las heurísticas usadas para el diagnóstico de errores. Se obtuvo un mejor rendimiento respecto al tiempo de retardo de los paquetes cuando se empleó el AMD con una función de tráfico uniforme sin la aplicación de heurísticas.

Palabras clave: Redes completamente ópticas, simulación, enrutamiento por deflexión, heurística y evaluación del desempeño.

Abstract

This paper proposes the development of a simulator using the OMNeT ++ tool, with the purpose of evaluating different strategies for deflection routing in information packages for next generation networks (All-Optical Networks). A Minimal Deflection Algorithm (MDA) is proposed, with the application of some heuristics and the evaluation of performance of this routing strategy in a 2D mesh with uniform and non uniform traffic functions. In this sense, some statistics were calculated in order to evaluate the performance of this type of networks, using the routing algorithm and the heuristics for error detection. There was a better performance in the packages delay, when using the MDA with a uniform traffic functions without the application of heuristics.

Key words: All-Optical networks, simulation, deflection routing, heuristics, analysis of performances.

1 Introducción

En la actualidad existe una necesidad continua y persistente de aumentar capacidades y bajar costos en las redes de telecomunicaciones. El desarrollo de la Internet, con todas sus aplicaciones y servicios, es una de las principales razones del crecimiento de esta demanda, (Tomsu, Schmutzer, 2002), combinada con una gran necesidad de comuni-

cación, debido al incremento en el número de computadoras y usuarios, junto con la exigencia de compartir el acceso a toda esa información al mismo tiempo. Es por esta razón que algunos autores han denominado esta época, “Era de la Información” (Ramaswami y Sivarajan, 2002).

La obligación de crear nuevos servicios ha generado el uso extensivo de más redes; de ahí la necesidad de aumentar el ancho de banda. El incremento de la demanda de ca-

pacidades de transmisión exige cada vez más ancho de banda. Este comportamiento no presenta signos de detenerse, incluso aumenta cada día de manera exponencial. Uno de los principales problemas que han tenido que afrontar las redes es cómo responder eficientemente a este crecimiento del tráfico. Esta es precisamente una de las razones por las cuales las tecnologías ópticas han experimentado un desarrollo importante. Las principales razones de su éxito han sido la gran capacidad y alta velocidad que solo la óptica puede ofrecer.

Algunos autores presentan la evolución de las redes ópticas dividida en varias generaciones (Ramaswami y Sivarajan, 2002). La primera generación utiliza los cables de fibra óptica básicamente para el traslado punto a punto de la información.

En la segunda generación, en las redes completamente ópticas, se realiza el tratamiento de la información, desde el origen hasta el destino, siempre en el dominio óptico. Esto ha motivado la creación de nuevos paradigmas de asignación de rutas para los paquetes de información que transitan por la red, puesto que los antiguos protocolos que aún cuando podrían trabajar bien en las redes de primera generación, no necesariamente trabajan bien en este tipo de redes.

En las redes completamente ópticas, las memorias son excesivamente caras y poco eficientes, lo cual sumado a las altas velocidades de transmisión involucradas, obliga a que las estrategias de asignación de rutas deben ser sencillas, limitar el acceso a las memorias y sobre todo deben ser rápidas, para que puedan adaptarse a las altas velocidades de la óptica (Barth y col, 2001, Ramaswami y Sivarajan, 2002). Por eso el enrutamiento por deflexión es el método más usado (Borrero, 2004). Se usa la deflexión como una alternativa para el almacenamiento temporal. (Baran, 1964).

2 Formulación del problema

En un nodo particular de una red que usa enrutamiento por deflexión, si varios paquetes compiten por un lazo de salida, solo tomarán la salida deseada, tantos paquetes como lo permita la capacidad del lazo. El resto de los paquetes serán desviados hacia otra salida disponible. Serán “deflectados”; aún cuando esto signifique enviar los paquetes hacia una ruta que no necesariamente los aproxime a su destino final (Barth y Berthomé, 2000; Brassil y Cruz, 1995). En esta estrategia, todos los paquetes de información permanecen siempre en movimiento.

En esta estrategia debe ponerse especial atención en la selección de los paquetes a deflectar, pues una mala gestión en su tratamiento conduce a que los paquetes permanezcan errando en la red sin alcanzar su destino y por tanto la red puede saturarse más rápidamente.

Se ha hecho necesario encontrar un algoritmo de enrutamiento que permita obtener mejoras en el rendimiento, respecto al tiempo de retardo de los paquetes que atraviesan una red completamente óptica. Intuitivamente, si los paque-

tes no son desviados de su ruta, llegarán más rápidamente a su destino y por tanto saldrán más rápido de la red. En (Borrero, 2002), el autor ha demostrado el “*Algoritmo de Deflexión Mínima*”, que es un algoritmo distribuido que deflecta el menor número de paquetes en cada nodo de la red.

Con la finalidad de obtener mejoras en el rendimiento de una red completamente óptica, en el presente trabajo se establecerán algunas heurísticas aplicadas al mencionado algoritmo, que permitirán evaluar el desempeño de la red, respecto a la distribución de la variable antes mencionada.

En el trabajo se plantea la hipótesis de que es posible disminuir el tiempo que demoran los paquetes de información en atravesar una red completamente óptica, con topología de malla 2D, si se hace una correcta elección de los paquetes a deflectar. De acuerdo a esto, se asignan rutas a los paquetes, basándose en diversas heurísticas:

- Asignar primero la ruta, a los paquetes que hayan sufrido mayor cantidad de deflexiones durante su trayectoria origen – destino.
- Asignar antes la ruta, a los paquetes que tengan mayor tiempo en la red, es decir, mayor edad o permanencia.
- Asignar mayor prioridad a los paquetes que estén más cerca de su destino final.

3 Algoritmo de enrutamiento

Tomando en cuenta que no existen “memorias ópticas” eficientes, es necesario explotar todos los recursos de la red para ofrecer una mejor calidad de servicio (*Quality of Service, QoS*) con el fin de garantizar que se transmitirá cierta cantidad de datos en un tiempo dado, de acuerdo con los requerimientos actuales.

3.1 Características y conceptos relacionados

La topología escogida para nuestro modelo es una malla 2D, principalmente por su simplicidad de implementación y su aproximación a una topología real, siendo considerada por algunos autores como la más general, al ser compatible con una natural dispersión de las manzanas de una ciudad. Según (Barth y Berthomé, 2000; Brassil y Cruz, 1995) [4,5] la malla 2D es la mejor topología para la estrategia de encaminamiento por deflexión.

El tamaño de la red modelada es de 7×7 . Se ha considerado este tamaño tomando en cuenta que es una Red Óptica de Paquetes (*Optical Packet Network*) del tipo “*Core Network*”, que son redes de gran capacidad y alta velocidad. Por otra parte, es conveniente recordar aquí que un paquete es una unidad de datos enviados a través de una red y el mensaje es la información llevada por el paquete. El formato del paquete usado en la red modelada es de duración fija. La razón es que en el dominio óptico es el que mejor se adapta por la ausencia de memoria óptica. Cuando hablamos de “*Optical Packet Networks*”, nos referimos a redes cuyos paquetes son transportados en la forma de impulsos luminosos modulados y no usa dispositivos de almacena-

miento intermedio. Los paquetes son encaminados individualmente a través de los nodos (*routers*), sin establecer previamente un camino de comunicación. Los paquetes son encaminados a su destino a través del camino más conveniente. Determinado por algún algoritmo para la asignación de la ruta óptima. Esta estrategia es usada para optimizar el uso del ancho de banda disponible en una red y para minimizar la latencia. Por esta razón, se ha vuelto necesario concebir una función de encaminamiento que pueda permitir a cualquier paquete, en cualquier situación alcanzar su destino en el tiempo mas corto posible. Debido a la ausencia de buffers de nivel óptico para controlar el tráfico, se emplea el *enrutamiento por deflexión*, que es la estrategia de encaminamiento usada en nuestro simulador.

El *enrutamiento por deflexión* consiste en resolver la contención en el nivel de un nodo enviando algunos paquetes hacia un destino no deseado.

Los algoritmos de enrutamiento por deflexión no explican como elegir los paquetes a ser desviados, o incluso el número de tales paquetes. La selección y la decisión de encaminamiento son elecciones locales sin conocimiento global y debe ser realizado en cada nodo y en cada instante de tiempo. Por este motivo se han desarrollado varios algoritmos y heurísticas para hacer la asignación de rutas a paquetes de información. No obstante, este proceso de selección es de gran importancia dentro de nuestro análisis, siendo características fundamentales a estudiar por ejemplo, el número de deflexiones promedio que sufren los paquetes durante una cierta cantidad de etapas de simulación. Allí podremos monitorear el desempeño de las heurísticas aplicadas al algoritmo de Mínima Deflexión dentro de nuestra investigación. Sin embargo, estos algoritmos son bastante complejos de implementar por el control de los *conmutadores* ópticos. Tradicionalmente los algoritmos de deflexión son vistos más como una manera para evitar el *almacenamiento intermedio* de mensajes y es por esto que son muy populares en las redes completamente ópticas.

Se han estudiado y creado algunos algoritmos para elegir la secuencia de encaminadores que deben tomar los paquetes en cada paso a lo largo de la red (Ramswami y Sivaraman, 2002; Barth y Berthomé, 2000), desde su origen a su destino. Estos son métodos que usan encaminamiento por deflexión. Y tienen la propiedad de reducir el número de deflexiones que sufren los paquetes en su camino.

Como se ha dicho, en nuestro simulador las redes usan conmutación de paquetes, pero como están en el dominio óptico, el protocolo es llamado conmutación óptica de paquetes (*Optical Packet Switching*, OPS) y en las redes completamente ópticas también es llamado conmutación de paquetes fotónicos (*Photonic Packet Switching*, PPS). En las redes ópticas los mensajes están hechos de luz y no pueden ser almacenados en ningún medio. La velocidad de transporte de la información se ve afectada cuando se producen las conversiones al modo electrónico. Por esta razón se han creado nuevos paradigmas de encaminamiento y nuevas estrategias de enrutamiento en redes de comunicación

completamente ópticas. El encaminamiento debe reaccionar muy rápido tomando en cuenta las velocidades de la óptica. Por consiguiente la información que necesita un nodo para calcular la ruta debe ser mínima.

La característica más importante de este tipo de encaminamiento es que en cada nodo siempre se debe tomar la mejor decisión, que significa enviar el paquete hacia la ruta que esté más cerca de su destino, mientras sea posible. Esta estrategia de encaminamiento, intenta encontrar la ruta óptima sin embargo algunos paquetes serán deflectados usando este algoritmo para encontrar un enrutamiento óptimo.

3.2 Encaminamiento en una malla 2D

Cuando se habla de ruta óptima, significa el camino mas corto, con el mínimo numero de deflexiones. Entonces, se define el camino mas corto como la mínima trayectoria que toma un paquete desde que entra a la red hasta que alcanza su nodo destino.

Se ha proporcionado un algoritmo distribuido que deflecta el menor número de clientes en cada nodo de la red. Cada nodo intenta enviar los paquetes hacia sus rutas óptimas, o los caminos mas cortos hacia sus destinos, el camino con el menor numero de deflexiones.

En la Fig.1 se observa el enrutamiento en una malla 2D. Allí se aprecia que las posiciones *a* y *b*, aproximan los paquetes hacia su nodo destino, es decir, hacia el Norte y el Este, en otras palabras donde los paquetes no son deflectados. Y hay otras dos direcciones *c* y *d* que representan las posiciones en el caso de deflexión, que son hacia el Sur y el Oeste.

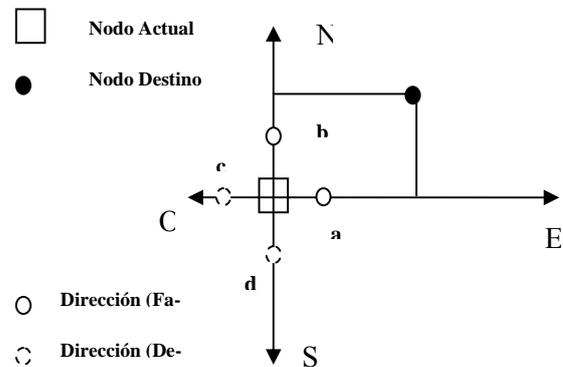


Fig. 1. Encaminamiento en una malla 2D

3.3 Algoritmo de mínima deflexión

En el presente trabajo, se ha desarrollado un simulador basado en el *Algoritmo de Mínima Deflexión* (Borrero, 2002), cuyo principio se fundamenta en minimizar las deflexiones en cada nodo de la red, y por tanto, reducir el número de pasos que los paquetes realizan para alcanzar su destino final. Por otra parte, es importante recordar que en este modelo el *índice de deflexión* (el número de pasos añ-

didos al camino del paquete, desde su origen a su destino, por cada deflexión sufrida), es de dos (2) pasos, debido a las propiedades de la topología de la malla.

Una importante propiedad en este algoritmo es que puede manipular grandes valores de capacidad de los nodos, es decir, gran número de longitudes de onda. Para efectos del estudio desarrollado en este trabajo, de acuerdo a las características de la red y para facilitar su implementación, la capacidad de los nodos que se utilizó es la mínima.

Presentaremos a continuación, la nomenclatura usada en este algoritmo; llamaremos:

Cientes a los paquetes que se encuentran en la red. *Nodo* a cada vértice de la malla, *Nodo actual* de un paquete en un momento dado, al nodo donde se encuentra el paquete al inicio del correspondiente *instante* de tiempo, *Dirección favorable* para un cliente, a la dirección de enrutamiento que coloca al cliente más cerca de su destino. *Dirección disponible*, a una dirección de enrutamiento que no ha sido asignada todavía. *Direcciones adyacentes*, a las direcciones que se encuentran localizadas una al lado de la otra, tales como: Norte y Este ó Sur y Oeste. Se refiere a las direcciones que no son opuestas en el sentido físico, como lo son: el Norte respecto al Sur y el Este con el Oeste. (Ver fig. 2).

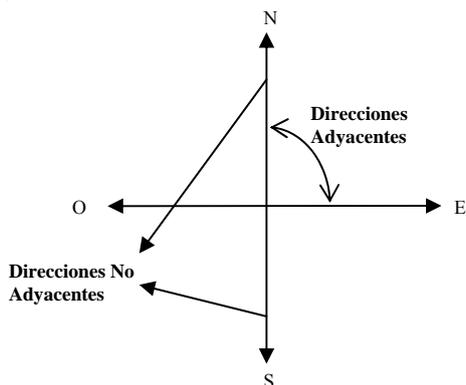


Fig. 2. Direcciones adyacentes y no adyacentes.

Es importante señalar que para el análisis y diseño del algoritmo (Borrero, 2002), se tomaron en cuenta algunas propiedades inherentes a la malla, que mencionaremos a continuación:

- Cada nodo posee a lo sumo cuatro direcciones, que son: Norte, Este, Sur, y Oeste.
- Un cliente puede tener una (1) ó dos (2) direcciones favorables, siendo dos (2) el máximo.
- Un cliente no puede tener como direcciones favorables, dos (2) direcciones opuestas en el sentido físico. Es decir, un cliente no puede tener como direcciones favorables, Norte y Sur o Este y Oeste al mismo tiempo; a lo sumo una de las dos en cada caso. Las direcciones que un cliente puede tomar deben ser siempre próximas, adyacentes, por ejemplo: Norte y Oeste o Sur y Este. Lo que significa,

que debe estar una al lado de la otra y nunca en sentidos opuestos.

Cada cliente tiene un nodo origen y un nodo destino, los cuales siempre son diferentes. Los clientes tienen una sola dirección favorable, si su nodo actual está en la misma fila o en la misma columna que su nodo destino. Tendremos dos direcciones favorables en todos los demás casos. Para un cliente, si su dirección o direcciones favorables, ya fueron tomadas por otros clientes del nodo actual, entonces ese cliente será enviado a otro nodo que no es parte del camino más corto hacia su destino, es decir, será deflectado. Solo algunos paquetes podrán transitar por el lazo que ellos solicitan, los otros tendrán que ser deflectados. El número de clientes depende de la capacidad del lazo.

3.3.1 Algoritmo de mínima deflexión usando mínima capacidad.

Con el propósito de entender mejor como funciona este algoritmo, veremos su estructura cuando la capacidad del lazo de unión entre los nodos es igual a uno, lo que significa, a lo sumo un cliente por cada dirección.

Lema 3.1 Este algoritmo opera para D direcciones y C clientes, donde $C \leq D$.

El algoritmo es el siguiente:

- Cada cliente es marcado como **no encaminado**.
- **Mientras** exista algún cliente marcado como **no encaminado**
- **Si** hay un cliente que no ha sido encaminado todavía, con 0 direcciones favorables disponibles:
- **Entonces** lo marcamos como siendo enrutado hacia **X** (Este cliente será deflectado. Fase 3 del algoritmo).
- **De lo contrario:**
- **Si** hay un cliente que no ha sido enrutado todavía con 1 dirección favorable disponible:
- **Entonces** marcarlo como siendo encaminado hacia esa dirección, y marcar esa dirección como no disponible.
- **De lo contrario:**
- **Si** hay un cliente con 2 direcciones favorables disponibles: Enviar hacia la dirección menos solicitada y marcar el paquete como siendo enrutado hacia esa dirección, y marcarla como no disponible.

Fin Mientras.

- Cada uno de los clientes marcados con X (a ser deflectados), son encaminados a hacia una dirección disponible.

• Ejemplo:

Usaremos cuatro clientes para este ejemplo, *a*, *b*, *c* y *d*. El cliente *d* posee sólo una dirección favorable. Se presenta una disputa entre los clientes *a* y *c*, resultando deflectado el

cliente *c* en la fase 2a del algoritmo, para ser luego enviado a la dirección Oeste en la fase 3 de algoritmo.

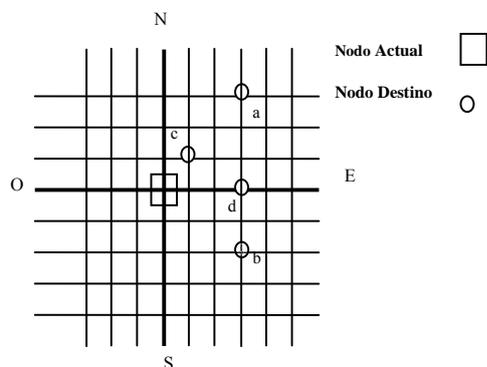


Fig. 3. Asignación de rutas.

Observamos en la fig. 3 cada uno de los destinos para los clientes *a*, *b*, *c* y *d* ubicados en su respectivo nodo actual denotado por el cuadrado de la figura.

Tabla 1. Asignación de rutas

Cliente	Fase Alg.	Ruta Asignada	N	E	S	O
d	1		2	4	1	0
d	2b	d → E	-	*	-	-
a	2b	a → N	*	*	-	-
c	2a	c → X	*	*	-	-
b	2b	b → S	-	*	*	-
c	3	c → O				

Se aprecia en la Tabla 1, el acoplamiento de cada uno de los clientes a sus respectivos destinos de acuerdo al algoritmo de encaminamiento de mínima deflexión.

4 Selección de heurísticas

La principal pregunta que se debe responder es, *¿que clientes deben ser deflectados?*, es por esta razón que se han elegido algunas heurísticas para observar el tiempo de retardo de los paquetes de información que atraviesan una red completamente óptica, con topología de malla 2D, y el comportamiento del algoritmo de Mínima Deflexión si se asignan prioridades a los paquetes, de acuerdo a diversas heurísticas. La primera heurística utilizada está basada en *asignar mayor prioridad a los paquetes que hayan sufrido mayor cantidad de deflexiones* durante su trayectoria origen – destino. La segunda heurística da *mayor prioridad a los paquetes que tengan mayor tiempo en la red*, es decir, “*mayor edad*” o permanencia. Y la tercera y última heurística *asigna prioridades a los paquetes que estén más cerca de su destino final*.

Es conveniente resaltar, que los paquetes pueden conservar en su cabecera, además de sus respectivos nodos ori-

gen y destino, la información referente al número de deflexiones sufridas durante su recorrido o el número de etapas de permanencia en la red.

5 Implementación del simulador

Se ha empleado la herramienta de simulación de redes de telecomunicaciones OMNeT++ (*Objective Modular Network Testbed in C++*), ya que es una herramienta eficiente, enfocada al área académica y desarrollada para simular eventos discretos en redes de telecomunicaciones. Básicamente, este simulador de redes recrea dichos eventos discretos por medio de módulos orientados a objetos; los cuales pueden ser utilizados para modelar el tráfico de información sobre las redes, los protocolos de red, multiprocesadores y otros sistemas de hardware distribuido. Además es empleado para evaluar arquitecturas de hardware y rendimiento de sistemas complejos (Varga, 2002). Fue escogida esta herramienta de simulación debido a: Porque es una herramienta “*open source*” (de gratis acceso para propósitos académicos). Corre en sistemas operativos como Windows y Unix. Posee una arquitectura totalmente orientada a objetos. Permite la reutilización de códigos fuentes, gracias a su arquitectura formada por módulos simples y compuestos desarrollados en lenguaje C++. La topología es definida a través de archivos de texto (excelente para la generación automática de topologías de redes)

5.1 El sistema programado

Es importante mencionar, que la metodología que se utilizó para la implementación del simulador, es el *Modelo de Desarrollo Evolutivo Incremental*. El cual tiene por filosofía, aplicar secuencias lineales de forma escalonada, mientras progresa el tiempo en el calendario. Es decir, el desarrollo del software es hecho en partes pequeñas pero utilizables, llamadas incrementos. En general cada incremento se construye sobre aquel que ya ha sido entregado (Pressman, 2002).

La implementación de nuestro simulador se basó en varias etapas, las cuales describiremos a continuación:

Este simulador está formado principalmente por un módulo compuesto, el cual contiene a su vez submódulos que hacen posible el funcionamiento adecuado de la red. Un módulo compuesto consta de tres submódulos, los cuales describiremos brevemente a continuación:

Se define la estructura de la red en lenguaje NED

Durante esta etapa de desarrollo especificamos la disposición de la red, la cual se hizo mediante el lenguaje de NED, en forma de código. La definición de la red en este lenguaje, permite una descripción modular de la red (Varga, 2002). Esto significa que la red consta de un conjunto de componentes descrito por medio de módulos simples y compuestos, los cuales pueden ser reutilizados en otras con-

figuraciones.

- **Submódulo Gen:** módulo encargado de generar los paquetes que van a entrar a la red.
- **Submódulo Sink:** su función es eliminar el paquete de la red; una vez que alcanza su destino final, además es en este módulo, justo antes de que cada paquete salga de la red, cuando se toman las estadísticas respectivas.
- **Submódulo Rte:** este es el módulo que se encarga de encastrar los paquetes de acuerdo al algoritmo de enrutamiento desarrollado en este trabajo.

A continuación se observa de manera gráfica en la Fig. 4, como está constituido el módulo compuesto (Nodo), formado por los tres submódulos (*gen*, *sink* y *rte*).

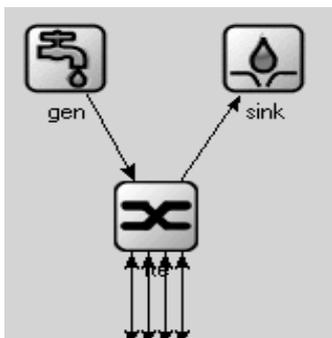


Fig. 4. Disposición de los submódulos *gen*, *sink* y *rte* dentro del simulador.

Se configura el comportamiento con lenguaje C++

Configuramos los parámetros y características de la red mediante el lenguaje de programación C++, para cada uno de los submódulos, anteriormente mencionados, que componen el simulador. Es necesario definir el comportamiento de los módulos (inicialización, finalización y tratamiento de mensajes). Cada módulo simple se programa como una clase en C++. Los módulos simples contienen algoritmos como funciones en C++, así la gran flexibilidad y el poder de programación de este lenguaje puede ser usado, respaldado por las librerías de clases del simulador OMNeT++ (librería API). Los objetos de simulación, como: mensajes, módulos, compuertas, etc., son representados por clases en C++. Ellos son designados para trabajar juntos, creando una poderosa herramienta de simulación. Entre los principales objetos con los que se trabajó en nuestra simulación están: módulos, mensajes, compuertas o conexiones y nodos.

Se configura la simulación con el archivo *omnetpp.ini*

Es en este archivo donde se configura la simulación, de acuerdo a los distintos escenarios que se analizarán. Dentro de este archivo se puede modificar, por ejemplo, la tasa lambda de la distribución de poisson para la generación de paquetes. También se pueden fijar los distintos tiempos de simulación, o incluso cambiar la semilla para la generación

de paquetes dentro de la simulación. La principal ventaja que tiene el trabajar con este archivo, es que se pueden hacer modificaciones para los distintos escenarios con los cuales se va a emplear la simulación, sin necesidad de estar compilando el programa entre corrida y corrida.

6 Escenarios de simulación y evaluación del desempeño

Se procedió a desarrollar las simulaciones de acuerdo a las diferentes funciones de tráfico evaluadas, como son: uniforme, no uniforme hacia el centro y no uniforme hacia la parte superior de la red.

El número de simulaciones realizadas en total, fue de 144, aproximadamente, establecidas de la siguiente manera:

En primer se ejecutaron las simulaciones de la red utilizando el Algoritmo de Mínima Deflexión con las tres (3) funciones de tráfico mencionadas, para al menos seis (6) valores distintos de la tasa de entrada de paquetes (λ), con cuatro (4) valores para el número de etapas de simulación. Lo que da un total aproximado de 72 simulaciones realizadas.

Posteriormente se aplicaron al algoritmo, las tres (3) heurísticas explicadas en capítulos anteriores, con igual cantidad de valores para la tasa de llegadas de paquetes y el mismo número de valores para las etapas de simulación, lo que produce un total de otras 72 simulaciones.

En la Fig. 5 se puede observar el porcentaje acumulado de clientes que sufren deflexiones, utilizando una función de tráfico uniforme y para diferentes valores de lambda. Es conveniente aclarar que cuando indicamos que el valor de lambda es igual a 0.01, estamos expresando en realidad que el número de paquetes generados por unidad de tiempo es 100. Así, un valor de lambda de 0.05, corresponde a una tasa de llegadas de 20 paquetes por unidad de tiempo. De la misma manera, 0.1 corresponde a 10 paquetes, y así 0.5, 1.5 y 4, corresponden respectivamente a 2, 0.67 y 0.25 paquetes por unidad de tiempo. También se puede observar en la Fig. 5 que con una función de tráfico uniforme y para una tasa de 0.01 paquetes, el porcentaje de clientes que sufre entre 0, 1 y 2 deflexiones es del 70%. De manera muy similar ocurre para los valores de lambda 0.05, 0.1, 0.5 y 1.5 donde la cantidad de clientes entre 0, 1 y 2 deflexiones se mantiene entre el rango del 70 al 75% de los clientes. Sin embargo cuando el valor de lambda es 4, la cantidad de

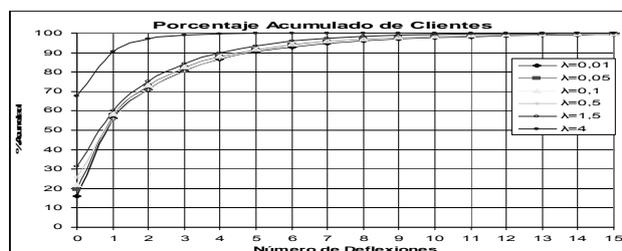


Fig. 5. Porcentaje acumulado de clientes por número de deflexiones para diferentes valores de lambda. Tiempo de simulación 15E04.

clientes que no sufren deflexiones es de casi el 70% de los clientes y el 97% de los clientes solo sufren 0, 1 ó 2 deflexiones.

De acuerdo a la Fig. 6, es posible apreciar que para la curva que representa a la función de tráfico uniforme, la mayor frecuencia corresponde a los paquetes que han sufrido una (1) deflexión. A partir de ese valor, la curva es siempre decreciente. En lo que respecta a la curva de la función no uniforme hacia el centro de la red, la mayor cantidad de paquetes solo sufre una deflexión, aún cuando la curva presenta un máximo relativo en el valor 4, lo que la hace más irregular. Mientras que en la función de tráfico no uniforme hacia la parte superior de la red, la mayor frecuencia corresponde a paquetes que no han sido deflectados. La curva es siempre decreciente

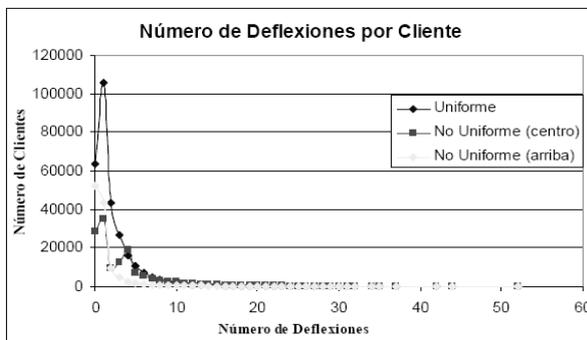


Fig 6. Comparación de funciones de tráfico Uniforme y No Uniforme. $\lambda = 0.1$. Tiempo de simulación 15E04

La Fig. 7 muestra el comportamiento de cada una de las heurísticas, comparadas con el *Algoritmo de Mínima Deflexión de Mínima Capacidad* para las diferentes funciones de tráfico. Se observa que la dispersión de los datos es mayor cuando se aplican las heurísticas, comparada con la estrategia de encaminamiento de mínima deflexión con función de tráfico uniforme. Aunque si se estudia con más detalle, la proporción de clientes con más de 12 unidades de tiempo en la red, es menor cuando es aplicada la heurística que le da prioridad a los paquetes que están mas cerca de su destino final, (*HeHopFTU*), lo que significa que para esta medida particular, la heurística implementada se comporta mejor que el algoritmo de mínima deflexión original. Considerando que cuanto mayor es la cantidad de paquetes que entran a la red, mejor se comportan dichas heurísticas. Se ha establecido este patrón de comparación, debido a las características propias de la red que se está simulando. Ya que para una topología de red malla 2D, la cantidad máxima de pasos entre dos nodos, es de $2N-2$, (para $N=N$, igual número de filas que de columnas), es decir, en nuestro caso, 7×7 , entonces la cantidad máxima de pasos que un paquete puede dar, sin sufrir deflexiones, entre su nodo origen y su nodo destino es de 12. Simplemente es el camino mas largo que existe entre el nodo origen y el nodo destino dentro de una malla 2D.

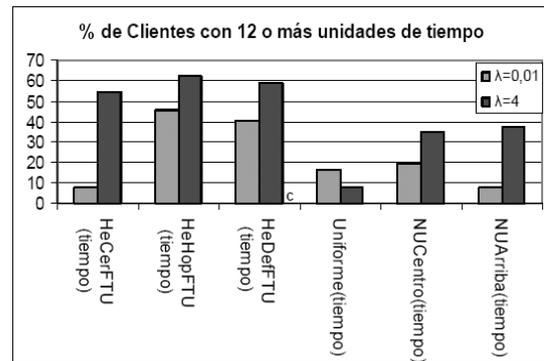


Fig.7. Promedio de clientes con 12 o más unidades de tiempo (edad), antes de alcanzar su destino final. $\lambda=0,01$ y $\lambda=4$ Tiempo de simulación 15E04

7 Conclusiones

El producto principal de este proyecto es un simulador de redes completamente ópticas implementado en el paquete de simulación de redes de telecomunicaciones OM-NeT++, empleando un algoritmo distribuido de asignación de rutas a paquetes de información, conocido como *Algoritmo de Mínima Deflexión*. Luego de realizar la investigación se llegó a las siguientes conclusiones:

Se ha evaluado la estrategia del encaminamiento por deflexión en una red de tamaño 7×7 , considerando que la red simulada es de tipo "Core Network"; que por su gran capacidad y alta velocidad, la hemos considerado muy apropiada para una red de paquetes fotónicos. Se implementó la estrategia de encaminamiento por deflexión conocida como *Algoritmo de Mínima Deflexión*, en una malla 2D síncrona, por su sencillez, facilidad de implementación y su semejanza a una red real. El uso de la herramienta de simulación de eventos discretos OMNeT++, demostró ser de gran ayuda y funcionalidad para el desarrollo del simulador, teniendo características muy útiles, tales como: implantación de distintas funciones de tráfico, tasas de llegadas de paquetes e incluso una elección de manera dinámica del tamaño de la red a simular, igualmente de la versatilidad de la herramienta para la implantación de funciones a través de cada uno de los módulos que componen el simulador empleando la librería API, que contiene gran cantidad de funciones ya implementadas. Además de ser una herramienta de libre distribución para fines académicos, lo que permite la reutilización de códigos fuentes.

Utilizando las ventajas y beneficios que nos proporciona el paquete OMNET++, se logró diseñar un simulador para redes completamente ópticas, que proporciona una ágil interacción con el usuario, una fácil operatividad, un entorno gráfico amigable y poderoso, que permite observar el funcionamiento de la red en tiempo real. Se evaluó el desempeño del simulador de redes completamente ópticas, empleando el *Algoritmo de Mínima Deflexión* para funciones de tráfico uniformes y no uniformes, observándose que dicho algoritmo es sensible a la manera como se disponen

los destinos de los paquetes dentro de la red. Como era de esperarse, se observó el mejor desempeño en el tiempo de retardo de los paquetes cuando se utilizó una función de tráfico uniforme. Mientras que para las funciones de tráfico no uniformes se puede concluir que la función orientada hacia la parte superior de la red funciona un poco mejor que la orientada hacia el centro de la red. Además se evaluó el desempeño para diferentes escenarios de simulación utilizando distintas tasas de llegada de paquetes según la función de distribución de *Poisson* y se observó que para tasas de llegada de paquetes muy altas, existen retrasos más considerables, sin embargo el *Algoritmo de Mínima Deflexión* se acopla muy bien para el encaminamiento, cuando se emplean grandes volúmenes de entrada de paquetes por unidad de tiempo. Para tratar de disminuir el promedio de retardo de los paquetes dentro de la red se implementaron heurísticas aplicadas al *Algoritmo de Mínima Deflexión*, que le asignaron prioridades a los paquetes de acuerdo a diversos criterios, obteniéndose como resultado de la aplicación de las mismas, un rendimiento inferior al esperado para disminuir dichos tiempos de retardo de los paquetes. El hecho de asignar prioridades para la asignación de rutas, a los paquetes que están más cerca de su destino, no incide favorablemente en la disminución del tiempo de retardo de los paquetes que transitan la red. Sin embargo es importante señalar que cuando se evaluó la proporción de clientes que permanecieron más de 12 unidades de tiempo dentro de la red, se observó que para esta medida particular, la heurística implementada se comporta mejor que el *Algoritmo de Mínima Deflexión* original. Por otra parte, en cuanto a las heurísticas para asignación de prioridades a los paquetes que han sufrido mayor cantidad de deflexiones y con mayor tiempo de permanencia en la red, podemos afirmar que no se demostró una incidencia favorable sobre la distribución del tiempo de retardo de los paquetes.

Es aconsejable el progreso en el estudio e implementación de nuevas heurísticas para el encaminamiento a fin de tener una mejor selección de los paquetes a ser deflectados, para reducir el tiempo de retardo de los paquetes en el camino desde su nodo origen a su nodo destino. Esto permitiría profundizar la investigación en esta área de extensas posibilidades de desarrollo.

Se ha demostrado que *Algoritmo de Mínima Deflexión* trabaja muy bien con una topología de red malla 2D. Recomendamos adaptarlo a otras topologías, tales como: hiper-cubo o torus y así evaluar su desempeño y adaptabilidad.

Se ha implementado un simulador que usa un algoritmo de encaminamiento con cuatro posibles direcciones, podría considerarse adaptar este algoritmo para trabajar con mas direcciones en función de topologías más irregulares.

Podría evaluarse el algoritmo para funciones de tráfico dirigidas a otros sectores de la red. Además de utilizar otra estrategia para la generación de clientes, por ejemplo ráfagas.

En base a la gran cantidad de datos analizados, es posible hacer un estudio estadístico exhaustivo, a fin de evaluar las posibles correlaciones existentes entre cada una de las variables, para obtener un estudio analítico más profundo, que permita ahondar el estudio de los comportamientos para los diferentes escenarios evaluados en este proyecto.

Podría hacerse una implementación del simulador de redes completamente ópticas, utilizando otros paquetes de simulación a fin de establecer comparaciones y evaluar el rendimiento de cada simulador de acuerdo al paquete de simulación y el lenguaje de programación empleado.

8 Agradecimientos

Este trabajo ha sido financiado por el Consejo de Desarrollo Científico, Humanístico y Tecnológico (CDCHT) de la Universidad de Los Andes bajo el proyecto N° I-986-06-02-F.

Referencias

- Baran P, 1964, On distributed communication networks, IEEE, Transactions on Communication Systems, CS-12,,
- Barth D, Berthom P, Borrero A, Fourneau JM, Laforest C, Quesette F y Vial S, 2001, Performance comparisons of eulerian routing and deflection routing in a 2D-mesh all optical network, In ESM.
- Barth D, Berthomé P y Cohen J, 2000, The eulerian stretch of a digraph and the ending guarantee of a convergence routing, Technical report, LRI, Université Paris-Sud, Orsay, France.
- Borrero M, 2004, Simulation of all – optical networks with deflection routing, Tesis Doctoral.
- Brassil J y Cruz R, 1995, Bounds on maximum delay in networks with deflection routing, IEEE Transactions on Parallel and Distributed Systems, Vol. 6, No. 7, pp.724-732.
- Pressman RS, 2002, Ingeniería del software, un enfoque practico, McGrawHill, Quinta edición.
- Ramaswami R y Sivarajan K, 2002, Optical networks, A practical perspective, Editorial Morgan Kaufman.
- Tomsu P y Schmutzer C, 2002, Optical networks, Prentice Hall.
- Varga, 2001, The OMNeT++ discrete event simulation system, proceedings of the european simulation, Multiconference (ESM'2001), Praga.