

# Diseño e Implementación de un Simulador para Redes Completamente Ópticas. Caso de Estudio: Enrutamiento por Deflexión Mínima.

Armando Borrero Molina, Andrés Arcia-Moret, Anna Lezama, Alejandro Mujica, Ana Rodríguez

**Resumen**— En este trabajo presentamos el diseño e implementación de un simulador para redes completamente ópticas, elaborado con fines didácticos. Se utiliza el algoritmo de enrutamiento por deflexión mínima aplicado sobre una red de tipo malla. Para simular el tráfico de datos, realizamos el ingreso de paquetes al sistema mediante una distribución de Poisson.

El simulador es altamente configurable. Cada nodo puede servir de entrada (con su respectiva cola) y/o salida. Además, de poder variar a voluntad la cantidad y ubicación estos nodos, el número de iteraciones de la simulación y la generación de paquetes; permitiéndole así al usuario, evaluar y entender un gran número de escenarios de simulación.

**Palabras claves**— Redes completamente ópticas; simulación; algoritmo, enrutamiento por deflexión.

## I. INTRODUCCIÓN

Hoy en día ha aumentado la necesidad de los usuarios por obtener mayores cantidades de información a través de la red, en tiempos cada vez menores y a más bajos costos. Además, miles de usuarios desean acceder a toda esa información al mismo tiempo. Con la finalidad de cubrir dichas necesidades, han surgido nuevas tecnologías, tales como las redes completamente ópticas. Estas han tomado gran relevancia recientemente debido a su elevado potencial para proporcionar un enorme ancho de banda y una gran velocidad. La transmisión óptica busca aprovechar el desarrollo y madurez de la tecnología electrónica para superar las limitaciones impuestas por esta última. Como es sabido, en las redes completamente ópticas, los paquetes circulan en forma de señales ópticas/luminosas, fotones en última instancia, que son las partículas que conforman la radiación luminosa.

Es conveniente recordar que las redes completamente ópticas son un tipo muy especial de redes, cuya principal diferencia con el resto de las redes es que su soporte de

comunicación son los cables de fibra óptica, y además, que las señales atraviesan la red de lado a lado, únicamente en el dominio óptico; sin hacer conversiones al dominio eléctrico y viceversa.

Un inconveniente grande que deben enfrentar estas redes, es que no existen memorias ópticas eficientes que permitan almacenar la información en los nodos. Se han implementado algunos métodos, como el enrutamiento por deflexión [1], que se usan como una alternativa para el almacenamiento temporal y que se adapta muy bien a este tipo de redes.

La deflexión resuelve la contención al nivel de los nodos, enviando algunos paquetes hacia un destino no deseado. 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, es decir, serán deflectados. Aun cuando esto signifique enviar los paquetes hacia una ruta que no necesariamente los aproxime a su destino final [2,3]. Todos los paquetes de información permanecen siempre en movimiento.

### A. Redes completamente ópticas

La evolución de las redes ópticas se puede dividir en generaciones [4]. Se denomina primera generación, a aquella que utiliza los cables de fibra óptica básicamente para el traslado punto a punto de la información. Se les denomina redes optoelectrónicas, pues realizan conversión O/E/O, Óptica-Electrónica-Óptica, (*Optical-to-Electrical-to-Optical*). La segunda generación, se les denomina redes completamente ópticas, pues se realiza el tratamiento de la información siempre en el dominio óptico, desde el origen hasta el destino. Es decir, hacen posible evitar los tiempos de conversión, poseen además baja latencia y gran ancho de banda. Por tanto, manejar el tráfico de fin a fin únicamente en el dominio óptico y evitar el procesamiento eléctrico intermedio, permitirá implementar en el futuro redes de altas velocidades.

### B. Enrutamiento por deflexión

El enrutamiento por deflexión fue inventado por Baran en 1964, y posteriormente probado para adaptarlo a las redes completamente ópticas, (*All-optical networks*), que no utilizan dispositivos de almacenamiento intermediario.

Artículo recibido el 3 de Febrero de 2012.

A.B.M., A.A.-M., A.L., A.M. y A.R. están con la Universidad de Los Andes, Sector La Hechicera, Facultad de Ingeniería. Mérida, Estado Mérida, Venezuela, Tlf. +58-274-2402811/ 2824, Fax: +58-274-2402890, E-mail: borrero@ula.ve, andres.arcia@ula.ve, lezamaanna@ula.ve, alejandromujica@ula.ve, ana.cegarra@gmail.com

A.B. pertenece al grupo de investigación en telecomunicaciones (GITEL). A.A.-M., A.L. y A.M. pertenecen al grupo de investigación en Redes y Sistemas Distribuidos (RESIDE).

Este enrutamiento es una manera simple de resolver el denominado “problema de la disputa”. Si en un nodo, un paquete no encuentra desocupado un puerto de salida, este será enviado a cualquier puerto de salida que se encuentre disponible. Este paquete es enviado a un destino que no es el deseado, pero por medio de esta estrategia se evita la pérdida de paquetes. Este tipo de enrutamiento puede ser usado gracias a que el número de puertos de entrada y de salida son iguales. Cada paquete de entrada encontrará siempre disponible un puerto de salida.

Los paquetes ópticos pasan a través de varios conmutadores ópticos antes de llegar a sus respectivos destinos finales. Es importante entonces introducir algún mecanismo para dar un acuerdo en la disputa, esto tiene un impacto importante en el desempeño de la red. En redes electrónicas este problema es resuelto por medio de memorias de almacenamiento de paquetes enrutamiento y envío “*store-and-forward*”. Los paquetes electrónicos son guardados en la memoria RAM (*Random Access Memory*) y permanecen allí hasta que pueden ser enviados nuevamente a la red (*Forwarded*). En el caso de un conmutador óptico de paquetes, no se tiene memoria óptica y el enrutamiento por deflexión ha aparecido como una alternativa natural para este tipo de redes. Una explicación intuitiva es que los nodos usan toda la red como un gran *buffer* y encaminan el paquete en disputa al resto de la red. Esta técnica permite que múltiples paquetes alcancen su destino sin pérdida alguna

## II. TRABAJOS PREVIOS

El algoritmo de mínima deflexión fue propuesto con la finalidad de proporcionar una herramienta para la asignación de rutas a los paquetes que circulan en redes totalmente ópticas incluyendo además la revisión de múltiples estrategias de enrutamiento [1]. También se ha utilizado con aplicación de heurísticas y realizado la respectiva evaluación de desempeño, [5]. Se muestran además ejemplos de aplicaciones del algoritmo de deflexión para este tipo de redes.

En general, los trabajos anteriores han propuesto y estudiado el algoritmo de deflexión en simulaciones de redes completamente ópticas en donde se evaluaron algunas heurísticas y los resultados del algoritmo en sí como estrategia de enrutamiento de paquetes, para diversas funciones de tráfico.

También se ha estado revisando el efecto de algunos esquemas como *weighted fair queueing* (WFQ) en el buffer de ingreso como servicios importantes a ser diseñados en la red y a ser considerados al momento de mejorar el rendimiento obtenido por el algoritmo [6].

La red de nuestro estudio, al igual que en los trabajos previos, se refiere a una red de tipo CORE, que transmite grandes cantidades de datos y que por sus características, gran capacidad y alta velocidad, resultan en unas de las más apropiadas para la transferencia óptica. [5] Es de hacer notar que el rendimiento es la clave de su funcionamiento.

En las anteriores revisiones del algoritmo de deflexión, se ha mencionado la necesidad de proporcionar calidad de servicio en las redes completamente ópticas y se ha hecho

énfasis en que la selección de una acertada estrategia de enrutamiento define que los resultados sean óptimos.

Esto es crítico, considerando la importancia de las redes que se están simulando y la función de estas redes CORE en la Internet.

El retardo ocasionado por la deflexión de paquetes se ha intentado aliviar con estrategias como la asignación de prioridades a los paquetes con variados esquemas [5], sin embargo estas estrategias no han proporcionado un aumento considerable del rendimiento de la red. Actualmente nuestro estudio se enfoca más en presentar las estadísticas de tiempos de los paquetes en cola y en tránsito por la red. Gracias a esto se podrá visualizar la relación entre el número de deflexiones y los tiempos de permanencia en la red de un paquete.

Otro de los problemas visualizados en los estudios anteriores [6], consiste en el reordenamiento de los paquetes ó mejor dicho la recepción desordenada de los paquetes, la cual se dejará en manos de la capa transporte y se continuará realizando la simulación sin considerar este elemento ya que nuestro alcance se limita hasta la capa de red.

El ingreso de la cola como una estrategia para mejorar el rendimiento y el retardo introducido por el algoritmo de deflexión [6], ha sido utilizado combinado con estrategias como la agregación de paquetes ó la asignación de prioridades a paquetes de control como los ACK, sin embargo en nuestro enfoque se estará utilizando de manera exclusiva para observar el comportamiento y la llegada de los paquetes a la red simulada.

## III. SIMDROP: EL SIMULADOR DIDÁCTICO DE REDES ÓPTICAS

La topología escogida para el modelo de simulación es una malla 2D como se muestra en la Fig. 1, principalmente por su simplicidad de implementación y su aproximación a una topología real [6]. Esta topología está conformada por nodos y enlaces, de orientación síncrona, y no dispone de ningún tipo de dispositivo de almacenamiento o procesamiento en los nodos intermedios. Solo los nodos de entrada cuentan con una cola, en la cual se colocan los paquetes que desean ingresar a la malla y que no puedan hacerlo inmediatamente. Esta cola no tiene un tamaño fijo limitativo.

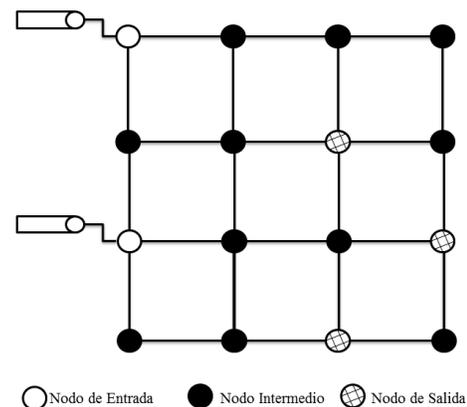


Fig. 1 Topología de malla 2D utilizada para el modelo de simulación.

Como se puede apreciar en la Fig. 1 en la red modelada lo nodos se disponen de forma cuadriculada (para cualquier  $n$ , se tienen  $n \times n$  nodos). Esta disposición está fundamentada tomando en cuenta que se trata de una red de tipo *Core Network*, donde el rendimiento es la clave de su funcionamiento, con anchos de banda que rondan los cientos de megabits por segundo.

Cada nodo está encargado de hacer pasar al paquete al nodo siguiente, de acuerdo con el algoritmo de *deflexión mínima* (ver Fig. 3). Este algoritmo se comportará de acuerdo al número de iteraciones pre-configuradas y a la tasa de generación de paquetes.

La unidad mínima del sistema es el paquete, el cual transita de un nodo a otro dentro de la malla a través del enlace óptico que hay entre los nodos (Fig. 1). La generación de paquetes responde a una distribución de Poisson con semilla aleatoria. Esto garantiza que se pueda tener una cantidad de tráfico que permita ver el comportamiento de los algoritmos de deflexión.

El simulador fue desarrollado con el lenguaje de programación de propósito general, C++, la interfaz gráfica con la biblioteca Qt [7] y la red fue modelada utilizando grafos, definida en la biblioteca *Aleph* [8]. Para el caso particular de este simulador, y por tratarse de redes ópticas, se asume que no existen pérdidas de paquetes por corrupción dentro de la malla y que eventualmente, cada paquete es capaz de alcanzar el nodo de salida. Sin embargo, un paquete puede verse afectado por el retardo y el reordenamiento inherente a la deflexión.

En la Fig. 2 se describe el diagrama de clases que le da soporte al simulador.

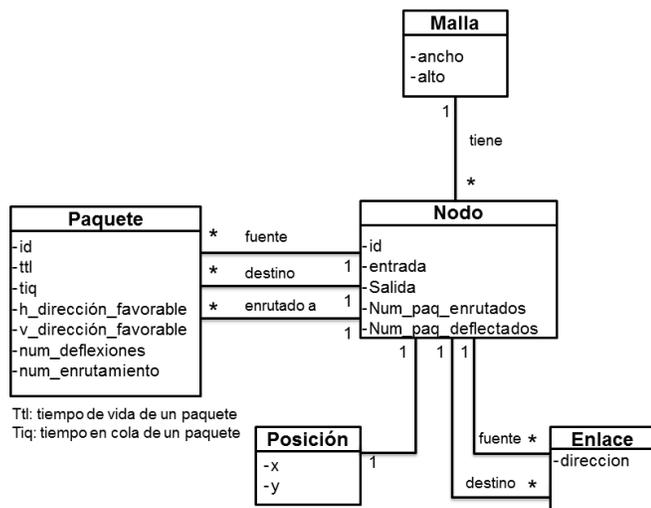


Fig 2. Diagrama de clases del simulador

### A. El proceso de simulación

A continuación se presentan los procesos de mayor impacto dentro del simulador.

a) *Ejecución del algoritmo de mínima deflexión*: A lo largo de este proceso, se asigna la ruta que toma cada paquete dentro de la malla 2D. Esto implica que un paquete ha de ser movido desde el nodo saliente (en cada etapa) hacia el enlace que le corresponde.

b) *Traslado de Paquetes*: Consiste en la movilización de los paquetes desde el enlace que se les ha asignado por el algoritmo de deflexión, hasta el nodo destino.

c) *Generación de Paquetes*: Consiste en recorrer todos los nodos de entrada y generar una cantidad de paquetes según la distribución de Poisson con media elegida por el usuario.

d) *Condición de Parada*: El proceso de simulación culmina cuando se cumplen alguna de las siguientes dos condiciones: 1. El contador de número de iteraciones de enrutamiento llega a su fin. 2. Todos los paquete puestos en las entradas han salido de la red.

Para enrutar los paquetes de un nodo a otro se utiliza el algoritmo mostrado en la Fig. 3, de deflexión mínima [1][5] que trata de enviar un paquete en la dirección que más le favorezca, según su destino. Si ésta ruta se encuentra ocupada, es enviado a otra dirección, siendo esto lo que se conoce como “deflexión” (descrita en *I.B*). A continuación se presenta una adaptación algorítmica aplicada al simulador desarrollado.

Este algoritmo se ejecuta en cada uno de los nodos y en cada etapa de la simulación.

### B. Consideraciones sobre el enrutamiento:

Para el modelado del proceso de enrutamiento, se tomaron en cuenta algunas propiedades inherentes a la malla, que se mencionan a continuación:

- Cada nodo posee a lo sumo cuatro direcciones, que son: Norte, Sur, Este y Oeste.
- Un paquete puede tener una (1) ó dos (2) direcciones favorables, siendo dos (2) el máximo.
- Un paquete no puede tener como direcciones favorables, dos (2) direcciones opuestas en el sentido físico. Es decir, un paquete no puede tener como direcciones favorables Norte y Sur o, Este y Oeste, al mismo tiempo. Las direcciones que un paquete puede tomar deben ser siempre próximas y adyacentes, por ejemplo: Norte y Oeste o Sur y Oeste.
- Los paquetes tienen una sola dirección favorable, si su nodo actual está en la misma fila o columna de su nodo destino. Tendremos, entonces, dos direcciones favorables en todos los demás casos.
- Para un paquete  $p$ , si su dirección o direcciones favorables ya fueron tomadas por otro paquete  $q$  coexistente en el nodo actual, entonces el paquete  $p$  será enviado a otro nodo que no es parte del camino más corto hacia su destino. En consecuencia,  $p$  será deflectado.
- El número de paquetes coexistentes dentro nodo (tomados de la cola) dependen tanto de la capacidad del enlace (1 paquete por enlace en nuestro caso) y del número de enlaces disponibles. Es decir, que para un nodo central (con cuatro enlaces de salida) podríamos potencialmente alojar 4 rutas diferentes para 4 paquetes diferentes.

### C. Algoritmo de Deflexión Mínima

En la Fig. 3, se presenta un algoritmo distribuido que deflecta el o los paquetes pasados al nodo, apuntando a tener

el mínimo número de deflexiones durante su estadía en la malla. Como puede observarse en las líneas 5, 8 y 15 de la Fig. 3, cada nodo intenta enviar los paquetes hacia sus rutas óptimas, o los caminos más cortos hacia su destino. Este camino, es potencialmente, el de menor número de deflexiones.

```

1  V nodo (n) ∈ red
2  Repita Mientras existan paquetes en n
3  Sea p el paquete actual
4  Sea i el número de direcciones favorables de p
5  Si (i=0) entonces
6      Desviar p al primer enlace disponible
7      Retornar Repita Mientras
8  Si (i=1) entonces
9      Sea D la única dirección favorable de p
10     Si enlace hacia D ocupado entonces
11         i ← 0
12         Retornar Repita Mientras
13     Enrutar p hacia D
14     Retornar Repita Mientras
15 Si (i=2) entonces
16     Sea DH la dirección horizontal favorable
17     Si DH está disponible entonces
18         Enrutar p hacia DH
19         Retornar Repita Mientras
20     Sea DV la dirección vertical favorable
21     Si DV está disponible entonces
22         Enrutar p hacia DV
23     De lo contrario
24         i ← 0
25 Fin Repita Mientras
26
27
    
```

Fig. 3 Algoritmo de mínima deflexión con mínima capacidad

En la Fig. 4 presentamos el detalle del enrutamiento en una malla 2D. Allí, se aprecia que los enlaces a y b aproximan al paquete a punto de ser enrutado favorablemente a su nodo destino, es decir, hacia el Norte o hacia el Este. De otra manera, para las direcciones desfavorables Oeste o Sur (enlace c o d en la Fig. 4), el paquete estaría siendo deflectado.

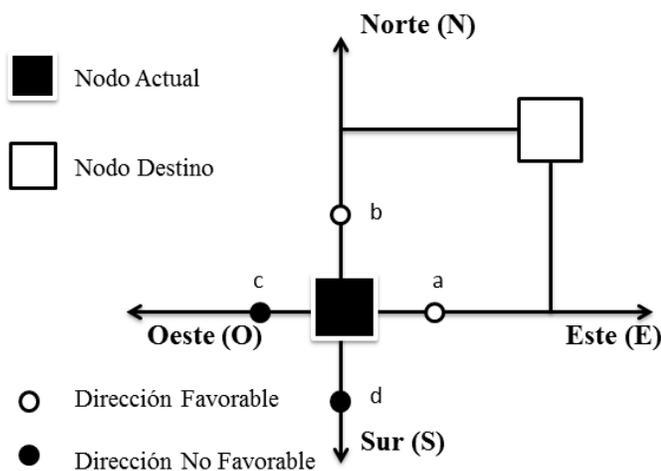


Fig. 4 Enrutamiento en una malla 2D

D. Parámetros de Simulación y Medidas de rendimiento

En esta sección se definen los parámetros de la simulación y las métricas consideradas para la medición del rendimiento de las simulaciones dentro de la red óptica.

Parámetros:

1. *Número de Iteraciones:* Al inicio de la simulación el usuario debe definir la cantidad de iteraciones correspondientes a la cantidad de operaciones de enrutamiento esperadas.

2. *Número de Paquetes Generados:* Cantidad total de paquetes que serán generados en todas las entradas, durante la ejecución completa de la simulación. La generación de paquetes se generan según la distribución de Poisson.

Medidas de Rendimiento:

1. *Número de Paquetes en Cola:* Cantidad de paquetes que no pudieron ingresar a la red, y que quedaron en cola de espera, al finalizar la simulación limitada por el número de iteraciones.

2. *Número de Paquetes Entregados:* Cantidad de paquetes que llegaron a su destino durante la simulación.

3. *Número de Paquetes en Movimiento:* Son los paquetes que quedan circulando en la malla al finalizar la simulación, limitada por el número de iteraciones.

4. *Tiempo promedio de un paquete en la cola:* Cantidad de iteraciones promedio que permanece un paquete en cola.

5. *Tiempo promedio de un paquete en movimiento:* Cantidad de enrutamientos promedio que se ha realizado a los paquetes en movimiento. Dicho de otra forma, se refiere al número de etapas de simulación que ha permanecido un paquete en la red, o tiempo que le toma a un paquete en llegar desde su origen hasta su destino. También se le denomina “Edad” de un paquete.

6. *Frecuencia de deflexiones:* Muestra tabuladamente, la cantidad de deflexiones que han ocurrido durante la simulación. Esta métrica no solo estipula la cantidad de deflexiones máximas ocurridas, sino que por cada número entero de deflexiones (en un rango de 0 a *MaxDeflexiones*), se cuentan el total de paquetes que incurrieron por cada frecuencia.

IV. PRUEBAS DE SIMULACION

El simulador cuenta con una malla que está modelada con una estructura de datos correspondiente a un grafo euclidiano con coordenadas matriciales, en donde existen nodos y enlaces.

Los *nodos* poseen contadores de paquetes que han transitados por ellos y una lista de paquetes que se encuentran en el en una iteración determinada. En especial, los nodos de entrada, además cuentan con una cola, que almacenan los paquetes que están por entrar a una red.

Los *enlaces* tienen el nodo fuente y el nodo destino. Con ello es posible conocer el sentido de la dirección dentro de la malla y así poder determinar la disponibilidad del enlace.

Los *paquetes* son la unidad mínima de datos a movilizar en el simulador. Ellos cuentan con la información de nodo destino, que no cambia desde su generación, y nodo previo que cambia con cada iteración de enrutamiento. El paquete también tiene sus contadores para enrutamiento y deflexión, utilizados para las métricas expuestas en la Sección III.D.

Existen solamente dos movimientos dentro de la malla para un paquete cualquiera: *enrutamiento* y *deflexión*, el enrutamiento consiste en el paso de un paquete de un nodo a otro disponible y favorable, en tanto que la deflexión consiste en el paso de un paquete de un nodo a otro disponible pero con dirección no favorable como se muestra en la Fig. 5.

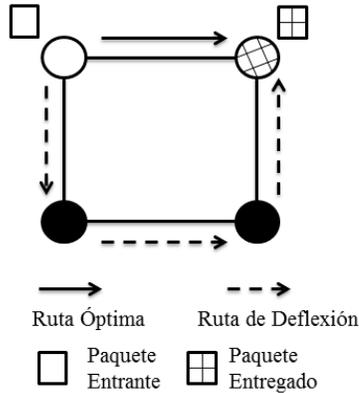


Figura 5: Comportamiento del enrutamiento

El simulador hace configurable el ancho y alto de la malla, así como la media de llegada de los paquetes según la distribución de Poisson. Antes de iniciar la simulación se definen los nodos de entrada y los de salida.

A. Simulación paso a paso:

A modo ilustrativo, el comportamiento en una malla de dimensiones 2x2 con un nodo de entrada y otro de salida es el siguiente:

Como se muestra en la Fig. 6, en la primera iteración ocurre la llegada de tres paquetes denominados A, B y C según la tasa de llegada definida por el usuario. El paquete A es *enrutado* al Este, siendo ésta, su única dirección favorable. Seguidamente, aunque el paquete B se dirige a la misma dirección que A, es *deflectado* hacia el Sur, ya que la dirección favorable que posee se encuentra ocupada, prioritariamente por el paquete A. El paquete C se encola en el nodo de entrada hasta que encuentre algún enlace disponible en próximas iteraciones.

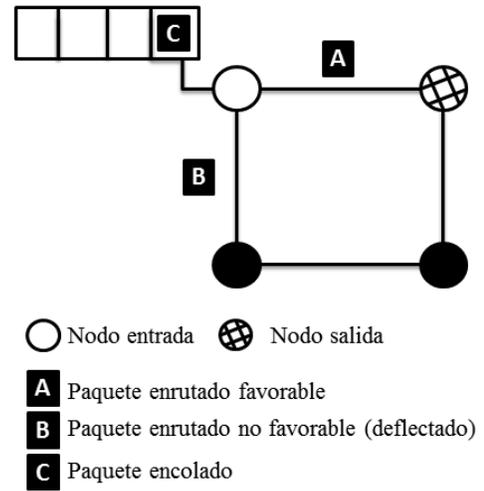


Fig. 6 Primera iteración del simulador

En la iteración siguiente, ilustrada en la Fig. 7, el paquete A es entregado, el paquete B sigue la ruta de deflexión y C que estaba en la cola, pasa a ser *enrutado*. La llegada de otros paquetes denominados D y E causan que D sea enrutado desfavorablemente y que E permanezca en cola.

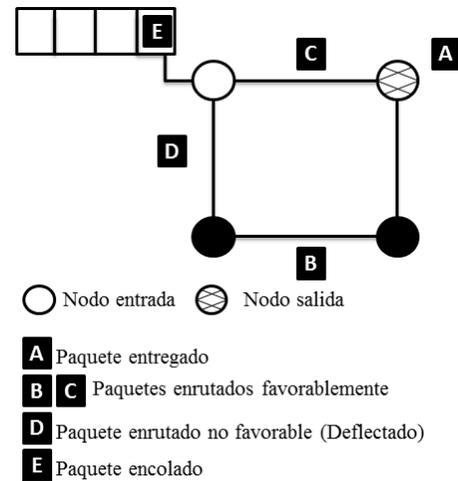


Fig. 7 Segunda iteración del Simulador

El paquete C, es entregado en la tercera iteración. El paquete B es enrutado al nodo de salida. El paquete D sigue enrutado en la ruta de deflexión y el paquete E que se encontraba en cola, es finalmente *deflectado*. Este comportamiento continúa así sucesivamente en una malla con esas características.

Para determinar al paquete que será próximamente deflectado, se estableció que la prioridad corresponde al paquete que ya se encuentra en movimiento dentro de la malla, por encima de los que vienen llegando. De esta manera, un paquete que desea entrar en la malla debe esperar que su nodo de entrada pueda enrutar o deflectar uno o más paquetes (internamente) para finalmente darle entrada.

### B. Evaluación de la simulación

En esta sección mostramos algunas reflexiones luego de haber realizado varias simulaciones con distintas semillas aleatorias, dejando los mismos parámetros de simulación.

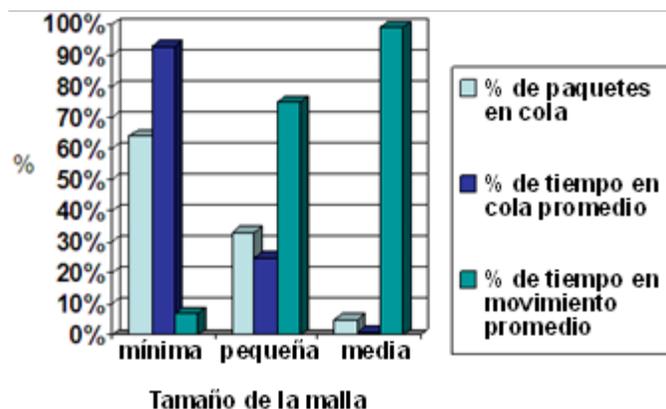
Las colas crecen más conforme disminuye el tamaño de la malla. Sin embargo, ni la cantidad de salidas disponibles en un nodo (de acuerdo a su ubicación en la malla), parecen ser el factor determinante respecto a este fenómeno. Pues, una entrada ubicada con cuatro posibles enlaces de enrutamiento (en el centro de la malla), muestra tamaños de cola similares a aquella que ubicadas en los extremos (con 2 y 3 salidas).

El hecho de tener colas de entrada demostró ser beneficioso. Sin embargo, hay que cuidar de no configurarlas para que creen una espera indeseada. Por otro lado, sabemos también que colas de muy poca capacidad, puede generar pérdidas de paquetes. Así podemos observar, que un tamaño de cola bien escogido puede cooperar con el algoritmo de deflexión, para la optimización del enrutamiento dentro de la red óptica.

Una de las posibles políticas para el diseño de las colas de entrada corresponde al cálculo de su tamaño proporcional a la malla óptica.

Durante diferentes experimentos, se varió el tamaño de la malla. Cuando se tenían entre uno y tres nodos de entradas y la misma cantidad para nodos de salida. Se observó que los paquetes nunca llegaron a encolarse, sino que estuvieron fluyendo con bastante rapidez. Al decrementar la dimensión de la malla para la misma, y manteniendo el mismo número de nodos de entrada y salida, se observó un aumento de la cantidad de paquetes en cola.

En la Fig. 8, se puede observar la disminución del número de paquetes en cola, respecto al número de paquetes totales de la simulación, a medida que aumenta el tamaño de la malla.



**Fig. 8. Porcentajes de tiempos y números de paquetes en cola respecto al tamaño de las matrices**

Las mallas mínimas son aquellas cuyas dimensiones están entre dos y cuatro, las matrices pequeñas tienen dimensiones entre cuatro y seis y las matrices medias estarían ubicadas con dimensiones por encima de siete. Nótese que la clasificación de las mallas simuladas es con propósitos de prueba y con fines didácticos. Pues, en la práctica, una red óptica pequeña corresponde a con un aproximado de cincuenta nodos.

Finalmente, durante las distintas simulaciones se observó que el tráfico en los extremos resultó ser más fluido y menos sujeto a deflexiones que el tráfico hacia los nodos centrales. Es decir, que los nodos centrales de la malla experimentaron una mayor congestión.

### V. CONCLUSIONES

En este artículo se presentó un modelo de simulación para redes completamente ópticas implementado en el Lenguaje C++. Además, se implementó y explicó a detalle el algoritmo de deflexión mínima a través de un modelo de grafos.

Respecto al funcionamiento del simulador hemos observado que: a) A medida que se aumenta el tamaño de las mallas, se tienen menor ocupación de las colas de entrada, esto se debe en parte a que la misma malla hace el rol de memoria y puede descargar rápidamente los paquetes de la entrada. b) Los paquetes ubicados en el centro de la malla experimentan mas deflexiones y consecuencia, mayor congestión.

La arquitectura del simulador desarrollada permite fácil incorporación de distintos algoritmos de enrutamiento.

Para trabajos futuros, pensamos en proveer la capacidad de definir topologías irregulares, así como también las de forma de anillo. Igualmente consideramos interesante para estudiar otros fenómenos dentro de las topologías ópticas, proveer al simulador con enlaces de mayor capacidad y con capacidad de procesamiento asíncrono.

### AGRADECIMIENTOS

Quisieramos expresar nuestra gratitud al Postgrado en Computación de la Facultad de Ingeniería, de la Universidad de Los Andes y a sus miembros. Allí se llevaron a cabo distintas discusiones con los estudiantes de la materia Redes de Computadores.

### REFERENCIAS BIBLIOGRÁFICAS

- [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] Kozlovsky, Miklos, Bercei, Tibor and Kozlovsky, Viktor, "NSOSS: the non-synchronized optical switch simulator" ValueTools '07: Proceedings of the 2nd international conference on Performance evaluation methodologies and tools: 1-6.
- [3] Rajiv Ramaswami, Kumar N. Sivarajan Galen H. Sasaki. "Optical Networks. A Practical Perspective" Third Edition. 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". Technical Report. 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.