

# Dispersión de Tráfico Fractal Mediante Algoritmos Inspirados en Colonias de Hormigas

Sandra M. Suárez, *Student Member, IEEE* y Marco A. Alzate, *Member, IEEE*

**Resumen**—La ingeniería de tráfico permite balancear la carga entre diferentes rutas, para lo cual tiene en cuenta tanto la topología de la red como la congestión en los enlaces. En este trabajo consideramos un parámetro adicional basado en la variabilidad del tráfico. Proponemos dos algoritmos distintos para realizar el balance, ambos basados en las propiedades emergentes de las colonias de hormigas tales como la inteligencia de enjambre, la adaptabilidad al medio y la auto-organización. Consideramos dos topologías, una simple con dos enrutadores unidos por dos enlaces y otra multisalto con siete nodos, diez enlaces y seis posibles rutas. En todos los casos encontramos que las hormigas no sólo consiguen dispersar óptimamente el tráfico balanceando las cargas sobre los enlaces sino que, si se diseñan adecuadamente, logran adaptarse a los cambios multiescala del tráfico con gran flexibilidad. Consideramos que este resultado favorece el nuevo paradigma de la complejidad en el estudio de redes de comunicaciones, especialmente a través de novedosas aplicaciones de la inteligencia computacional.

**Palabras Clave**—Optimización por Colonia de Hormigas, Tráfico Fractal, Enrutamiento mediante hormigas.

## I. INTRODUCCION

La ingeniería de tráfico permite balancear la carga entre los diferentes enlaces de una red, de manera que ninguno de ellos permanezca sobrecargado ni subutilizado. MPLS, por ejemplo, permite dirigir un flujo de paquetes IP por una ruta determinada, o LSP, para lo cual se debe compartir dinámicamente información sobre la topología de la red y el estado de congestión de los enlaces, usando algún algoritmo de señalización adecuado (RSVP o LDP). Esto permite hacer un cálculo "en línea" de las mejores rutas, de acuerdo con la condiciones dinámicas de la red en cada instante [1]. Esta capacidad de MPLS para hacer una juiciosa implementación de ingeniería de tráfico se ve limitada por la falta de un conocimiento preciso de las características del tráfico que circula por la red. Un conocimiento más detallado de dichas características y de la correspondiente variación dinámica del estado de la red permitiría tomar mejores y más oportunas decisiones de ingeniería de tráfico [2]. Sin embargo, en redes modernas de comunicaciones es de esperarse que la variabilidad del tráfico obedezca a complejas estructuras de correlación, incluyendo fractalidad [3], lo cual imposibilita el uso de técnicas tradicionales de optimización [4].

S. M. Suárez estudia en la Universidad Distrital Francisco José de Caldas., Bogotá, Colombia (e-mail: sandramilena@iee.org).

M. A. Alzate, es profesor de la Universidad Distrital Francisco José de Caldas., Bogotá, Colombia (e-mail: malzate@udistrital.edu.co).

En efecto, en muchos campos distintos surgen problemas de optimización cuya complejidad no les permite ser resueltos dentro de restricciones razonables de tiempo [5]. En estos casos se recurre a heurísticas que guíen una búsqueda dentro del espacio de soluciones. Existen algunas heurísticas de aplicabilidad general tales como los algoritmos genéticos [6], la búsqueda Taboo [7] o las redes neurales [8]. La optimización mediante colonia de hormigas (ACO - Ant Colony Optimization-) es una metaheurística propuesta recientemente [10], la cual genera repetitivamente nuevas soluciones utilizando decisiones aleatorias pero influenciadas por la experiencia obtenida de soluciones anteriores. Los algoritmos ACO hacen parte de los algoritmos de "inteligencia de enjambre" (SI -Swarm Intelligence-) [10], la cual emerge de la acción autónoma de numerosos agentes simples. Los algoritmos SI se inspiran en las sorprendentes capacidades colectivas de los insectos sociales. Entre ellos, los algoritmos ACO se basan en el comportamiento de ciertas especies de hormigas cuando buscan comida para traer al hormiguero, el cual se puede describir mediante el concepto de auto-organización (los patrones macroscópicos que se originan en interacciones microscópicas). En efecto, aunque cada hormiga actúa autónomamente sin que nadie coordine la labor colectiva, el efecto es una ordenada organización de la colonia en su conjunto, en la que el comportamiento colectivo complejo surge de la interacción entre hormigas simples. Esta auto-organización le permite a la colonia adaptarse a los cambios en el ambiente con una gran robustez ante las fallas individuales de cada hormiga.

En el contexto de una colonia de hormigas, la auto-organización se da mediante interacciones simples como la realimentación positiva (estigmergia: las hormigas depositan rastros de feromonas cuando regresan de una fuente de comida al hormiguero para que otras hormigas los huelan y tiendan a seguir el mismo rastro) y el comportamiento aleatorio (cada hormiga parece seguir una caminata aleatoria, aunque probabilísticamente influenciada por los rastros de feromonas de otras hormigas). La evaporación de feromonas es otro mecanismo importante de interacción. Estos mecanismos son los que se intentan emular en un algoritmo ACO para resolver problemas algorítmicamente complejos, tales como aquellos asociados con manipulación de grafos. El enrutamiento en redes de comunicaciones es un buen ejemplo de este tipo de problemas, en los que el nuevo paradigma de los agentes de software móviles promete soluciones apropiadas [11].

Los algoritmos de enrutamiento deben transferir los paquetes de datos desde la fuente hasta el destino en el menor tiempo posible, mientras minimizan la pérdida de los mismos. Con este

propósito, la red se suele modelar como un grafo en el que los nodos son los vértices y los enlaces son los arcos. El costo del uso de los enlaces (retardo, velocidad, tasa de errores, confiabilidad, etc.) se usa como entrada para determinar cuál es la mejor ruta entre dos nodos. Desafortunadamente, como vimos, en las redes modernas de comunicaciones los costos de los enlaces varían dinámicamente en un amplio rango de escalas de tiempo con complejas estructuras de correlación, de manera que la naturaleza estática y determinística de los algoritmos clásicos de enrutamiento no les permiten adaptarse suficientemente rápido a dichas variaciones.

Los primeros algoritmos de enrutamiento inspirados en colonias de hormigas se desarrollaron hacia 1990 como una aplicación de ACO (Ant Colony Optimization) [12]. AS (Ant System) [13] fue una de las primeras propuestas, de la que surgió ACS (Ant Colony System) [14], el cual incorporó mejoras importantes: Las hormigas se mueven sobre las rutas más probables, sólo las hormigas con resultados óptimos actualizan las trazas de feromonas, existe la realimentación negativa mediante evaporación y se utiliza una lista de información para guiar las decisiones de enrutamiento. ARS (Ant Routing System) [15] es otro algoritmo ACO para enrutamiento de redes que mezcla juiciosamente la exploración aleatoria y la estigmergia para converger a buenas soluciones. En [16] se investigó si ARS podía obtener simultáneamente un aumento de la eficiencia en el uso de los recursos y una reducción de la pérdida de paquetes, cuando se utiliza en una red sometida a una alta carga por parte de los usuarios. Se concluyó que, al combinar realimentaciones negativa y positiva con heurísticas locales, se puede asegurar un bajo retardo y una baja pérdida de paquetes simultáneamente.

Los autores de este artículo consideramos que la adaptabilidad propia de los algoritmos de optimización mediante hormigas los hacen adecuados para encontrar rutas que varíen dinámicamente de acuerdo con los cambios temporales del tráfico a diferentes escalas de tiempo. Como ejemplo, diseñamos dos algoritmos de dispersión basados en ARS y presentamos distintos resultados experimentales con los que se verifica que los rastros de feromonas depositados en los enlaces responden a las variaciones del tráfico en múltiples escalas de tiempo, alcanzando un balance óptimo de la carga en cada instante.

En la sección II presentamos algunos experimentos en los que se pretende distribuir el tráfico entre dos enlaces paralelos, encontrando las características de adaptabilidad y optimalidad deseadas. En la sección III exploramos una topología más compleja, encontrando los mismos resultados con algoritmos ligeramente diferentes, lo que sugiere la generalidad de estas dos propiedades para el enrutamiento mediante hormigas. La sección IV concluye el artículo.

## II. DISPERSIÓN OPTIMA SOBRE DOS ENLACES

Inicialmente nos limitamos a una topología sencilla compuesta por dos enlaces entre un par de nodos, sobre la cual determinamos el efecto de los diferentes parámetros en la calidad de la ruta seleccionada. La red está compuesta por dos enrutadores, unidos por medio de dos enlaces con capacidades  $C_1$  y  $C_2$ , como muestra la figura 1. El objetivo es distribuir el tráfico de entrada  $\lambda$  en dos flujos  $\lambda_1$  y  $\lambda_2$  de manera que se

maximice el caudal y se minimicen los retardos y las pérdidas.

El algoritmo de [16] se basa en el uso adecuado de la información disponible, combinando las características del algoritmo Bellman-Ford [17] con las de ARS. Cada paquete que llega al primer enrutador mira cuál es su mejor opción de acuerdo con la experiencia de los paquetes que lo han precedido.

Cada cierto intervalo de tiempo se evalúa cuántos bytes se han recibido exitosamente, cuántos se han perdido y cuántos hay en la cola de espera de cada enlace. Esta información se normaliza dentro del intervalo  $[0, 1]$  para calcular así la probabilidad de tomar el primer enlace,  $P$ , de acuerdo con la ecuación (1), tomada de [16]. Esa información se obtiene del último intervalo evaluado, desechando los valores de los anteriores intervalos, lo cual asegura que la información corresponda al estado inmediatamente anterior.

$$P = \frac{\frac{(\text{recibidos\_en\_1})^\alpha}{(\text{perdidos\_en\_1})^\beta \cdot (\text{en\_cola\_1})^\delta}}{(\text{recibidos\_en\_1})^\alpha \frac{(\text{recibidos\_en\_2})^\alpha}{(\text{perdidos\_en\_1})^\beta \cdot (\text{en\_cola\_1})^\delta} + (\text{perdidos\_en\_2})^\beta \cdot (\text{en\_cola\_2})^\delta} \quad (1)$$

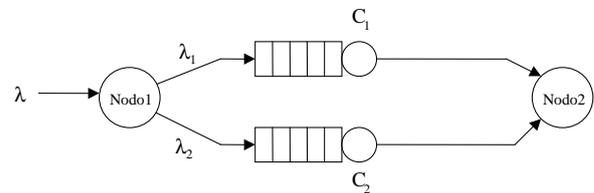


Fig 1. Topología de dos enlaces

Lo primero que queremos saber es cuáles son los valores que deben tomar alfa, beta y delta para conseguir que el balance de la carga obedezca las variaciones del tráfico en un amplio rango de escalas, esto es, queremos ponderar óptimamente la realimentación positiva, la realimentación negativa y las reglas heurísticas locales. Este punto se debería resolver planteando un problema de optimización donde se minimice el costo medido por las pérdidas, los retardos y el caudal. Sin embargo, dada la dependencia de dichos parámetros en las características estadísticas del tráfico y las dinámicas particulares de las colas, preferimos utilizar un procedimiento de muestreo aleatorio que, con un número adecuado de repeticiones independientes, nos asegure unos intervalos y niveles de confianza estadísticamente significativos.

La combinación donde alfa es igual a 0.1, beta igual a 0.7 y delta igual a 0.7 es, comparativamente, la que ofrece los mejores resultados en cuanto que se maximiza el caudal y se minimiza el retardo, bajo la restricción de una cota máxima en las pérdidas. Además, se observó que estos valores también minimizan las varianzas de las cantidades mencionadas.

De la misma manera encontramos que el intervalo de tiempo en que se actualiza la probabilidad de cada enlace puede ajustarse para alcanzar un compromiso entre la adaptabilidad del algoritmo y la sobrecarga computacional impuesta a los nodos. Efectivamente, a medida que disminuimos el tiempo de actualización, las pérdidas y los retardos disminuyen, pero aumenta el tiempo de convergencia a la solución óptima. Las simulaciones con tráfico tipo Poisson mostraron que siempre es posible obtener un buen compromiso entre los dos criterios en un amplio rango de intensidades de tráfico y distribuciones de la capacidad total entre los dos enlaces.

En las pruebas anteriores, para mirar la convergencia del algoritmo a la solución óptima, ejecutábamos el algoritmo para tráfico Poisson con una tasa fija y observábamos el tiempo necesario para alcanzar la convergencia. En el siguiente conjunto de pruebas evaluamos el comportamiento del algoritmo con un tráfico tipo Poisson no estacionario en el que la tasa de bits varía senoidalmente entre cero y la suma de las capacidades de los dos enlaces, con un período variable desde 6 s hasta 300 ms. Se encontró que, en condiciones de bajo tráfico, el algoritmo utiliza solamente el enlace más rápido y que, en condiciones de alto tráfico, se utilizan ambos enlaces en proporción a sus capacidades, lo cual concuerda con el resultado esperado [18]. No alcanzamos a observar una diferencia estadísticamente significativa en las soluciones para diferentes frecuencias en la variación de las tasas de llegada, lo cual indica que, en el rango de escalas de tiempo que se contempló, las hormigas se adaptan adecuadamente. En vez de seguir explorando frecuencias más altas, esta generalidad nos motivó a explorar directamente la posible adaptabilidad multiescala. En efecto, hasta ahora las pruebas nos permitieron observar las características de adaptabilidad del algoritmo frente a un tráfico variable, en una escala fija de tiempo. Sin embargo, como hemos dicho, en las redes modernas de comunicaciones el tráfico presenta variaciones significativas en múltiples escalas de tiempo, lo cual se modela mediante fractalidad en las estadísticas del tráfico.

Para ver el comportamiento del algoritmo cuando el tráfico en la fuente tiene propiedades fractales, usamos las trazas VBR codificadas por Fitzek usando H.263 [19]. Aunque las pruebas incluyeron “Mr Bean”, “From Dusk Hill Dawn”, “First Contact”, “Jurassic Park”, “Silence of the Lambs”, “Star Wars IV”, y, “The Firm”, sólo presentamos los resultados correspondientes a Jurassic Park.

La traza contiene una hora de video codificado a un promedio de 756 kbps, con picos de hasta 2.7 Mbps. Los paquetes son de longitud variable y se generan a una tasa constante de 25 paquetes por segundo. Utilizamos un enlace de 600 kbps y otro enlace de 400 kbps, para los cuales actualizamos la probabilidad de selección cada 480 ms (cada 12 tramas de video). La figura 2(a) muestra la tasa de llegadas (en Mbps) del tráfico a la entrada, la figura 2(b) muestra la tasa de llegadas al enlace 1, y la figura 2(c) muestra la probabilidad de escoger el enlace 1.

Como se puede observar, el enlace 1 casi siempre recibe más del 60% del tráfico de entrada y, en ocasiones de baja carga, puede recibir hasta el 100%, como era de esperarse. Pero, más interesante aún, la probabilidad de seleccionar el primer enlace forma una señal autosemejante que conserva el mismo parámetro de Hurst de la traza original de tráfico (0.8), lo que demuestra la adaptabilidad multiescala del algoritmo.

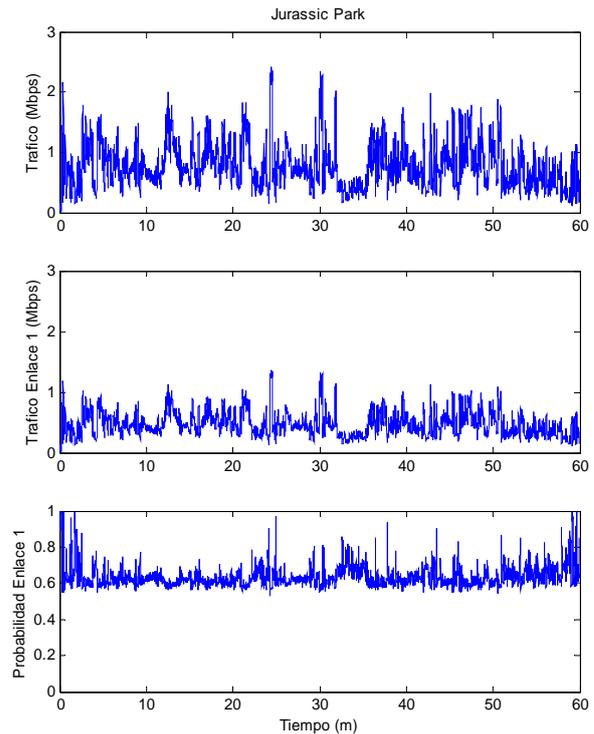


Figura 2. Resultados con tráfico fractal en la topología de dos enlaces

### III. EXTENSIÓN A RUTAS MULTISALTO

Ahora supondremos que disponemos de una red  $\{N, E\}$ , donde  $N$  es un conjunto de nodos y  $E$  es un conjunto de enlaces entre algunos de ellos. Queremos encontrar la mejor ruta desde un nodo fuente  $f \in N$  hasta un nodo destino  $d \in N$ .

Cada enlace  $e \in E$  tiene asociado un costo dado por una función  $c: E \rightarrow \mathbb{R}^+$ , de manera que el criterio de selección es el mínimo costo total de la ruta, dado por alguna función de los costos de los enlaces que la componen. Para esto se generan hormigas en el nodo  $f$ , donde cada una de ellas realiza un procedimiento semejante al descrito en el numeral anterior, pero modificado de la siguiente manera para poder manejar múltiples saltos:

Procedimiento Hormiga

$i=1$

$Ruta_i=f$

Mientras  $Ruta_i \neq d$

$A=Posibles\_Saltos(Ruta_i)$

$P=Probabilidades\_de\_Transición(A)$

$i=i+1$

$Ruta_i=Siguiente\_Nodo(P)$

Fin

Deposite\_Feromonas( $Ruta$ )

Fin

La función *Posibles\_Saltos(nodo)* retorna el conjunto de los nodos vecinos de  $nodo \in N$ , y la función *Probabilidades\_de\_Transición* calcula la probabilidad de cada uno de esos vecinos como siguiente salto en la ruta desde  $f$  hasta  $d$ , de acuerdo con la cantidad de feromona depositada en el enlace correspondiente. La función *Siguiente\_Nodo* retorna uno de esos nodos, seleccionado aleatoriamente de acuerdo con la distribución  $P$ . Por último, la función *Deposite\_Feromonas* añade feromonas a cada uno de los enlaces de la ruta seleccionada de acuerdo con la experiencia de la respectiva hormiga, para lo cual usa un paquete de reconocimiento de alta prioridad.

De esta manera, la hormiga construye una ruta ( $Ruta_1=f, Ruta_2, \dots, Ruta_{n-1}, Ruta_n=d$ ) siguiendo una caminata aleatoria en la que las probabilidades de transición están influenciadas por el rastro de feromonas depositado en cada enlace, y al llegar a su destino, determina cuánta feromona depositar de acuerdo con la calidad de la ruta encontrada. Como se pudo ver respecto a la topología sencilla de dos enlaces, es posible definir muchos algoritmos de enrutamiento especificando distintas funciones *Probabilidades\_de\_Transición* y/o *Deposite\_Feromonas*. Para determinar la generalidad de la capacidad de adaptación multiescala de los algoritmos basados en hormigas, en el siguiente algoritmo sólo vamos a considerar el retardo y las pérdidas como criterios de deposición de feromonas, a diferencia del algoritmo usado en la topología de dos enlaces.

Todos los paquetes en la red transportan hormigas y, por consiguiente, deben mantener una lista en su encabezado registrando la ruta recorrida para actualizar el rastro de feromonas (la hormiga vuelve al hormiguero en un paquete de reconocimiento de alta prioridad).

En el instante  $t$  el enlace  $e_{ij}=(n_i, n_j) \in E \subset N \times N$  tiene un rastro de feromonas  $\tau(t, n_i, n_j, d)$  -en cuanto a su uso para llegar al nodo destino  $d$ - . En consecuencia, si un nodo  $n_i \in N$  tiene  $m$  vecinos, su tabla de enrutamiento tendrá  $m$  rastros de feromona por cada uno de los  $|N| - 1$  posibles destinos (donde  $|A|$  es la cardinalidad del conjunto  $A$ ). Inicialmente, para un destino dado  $d$ , los enlaces de salida de cada  $n \in N$  tienen una mínima cantidad de feromonas,  $b$ .

Para estimular la exploración haremos que, con probabilidad  $q$ , un paquete que se encuentre en el nodo  $n_i$  escoja un enlace de salida aleatoriamente, sin tener en cuenta el rastro de feromonas. Con probabilidad  $1-q$ , el enlace de salida se escoge entre los vecinos no visitados de  $n_i$ ,  $\{n_k\}$ , de acuerdo con la siguiente distribución

$$P(t, n_i, n_j) = \frac{\tau(t, n_i, n_j)}{\sum_{n_k} \tau(t, n_i, n_k)} \quad \forall n_j \in \{n_k\} \quad (2)$$

Si el tiempo que tomó el paquete-hormiga en llegar de  $f$  a  $d$  en la ruta  $\{Ruta\}$  es  $T_R$ , la cantidad de feromonas que la hormiga deposita en cada enlace de  $\{Ruta\}$  será  $D_{min}/T_R$ , donde el factor  $D_{min}$  es un parámetro igual al mínimo tiempo de transmisión entre todos los enlaces. Cada  $t_{act}$  segundos el rastro de feromonas de cada uno de los enlaces se reduce a un  $evap \cdot 100\%$  de su valor actual, cuidando que nunca llegue a ser menor que  $b$ .

Un paquete perdido en el nodo  $n_i$  al intentar utilizar el enlace  $(n_i, n_j)$  genera una realimentación negativa inmediata reduciendo en un factor  $\gamma \leq 1$  la cantidad de feromonas en el enlace que lo condujo al nodo  $i$ .

La razón de ser de los parámetros  $q, b, \gamma$  y  $evap$  es la de evitar la convergencia prematura de la solución. Mientras los parámetros  $q, b$  y  $evap$  permiten explorar nuevas soluciones, el parámetro  $\gamma$  se encarga de la realimentación negativa ante pérdidas de paquetes.

Para probar las características de este nuevo algoritmo utilizamos la topología de red de la figura 3. Cada enlace está marcado con su capacidad (en paquetes por ms). Todas las colas de los enlaces tienen un cupo máximo de 10 paquetes. El nodo 0 quiere transmitir un flujo de paquetes al nodo 6, los cuales se generan según un proceso Poisson o según una traza muestral de tráfico fractal.

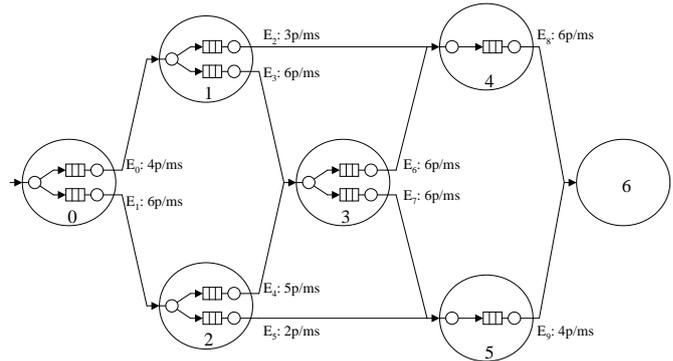


Figura 3. Topología de la Red Multisalto

Para el caso de tráfico Poisson, variamos la tasa promedio de llegadas de 0 a 30 paquetes por milisegundo y, para cada tasa de llegadas, medimos el correspondiente caudal (o tasa efectiva de paquetes que llegan a su destino), para cuatro algoritmos de enrutamiento diferentes: En uno de ellos, se escoge estáticamente la ruta más rápida, teniendo en cuenta solamente la capacidad de los enlaces. En otro algoritmo se consideran las dos rutas más rápidas y el tráfico se dispersa entre ellas estáticamente de acuerdo con sus capacidades. El tercer algoritmo es uno ideal en el que cada paquete, cuando nace, conoce completamente el estado de la red en ese instante y escoge la mejor ruta para ese momento. Finalmente, consideramos también el algoritmo de paquetes-hormiga descrito anteriormente.

En la figura 4 se compara el desempeño de cada algoritmo en términos del caudal de paquetes que efectivamente llegan al nodo 6, en función de la demanda ofrecida a la red desde el nodo 1. Claramente se observa cómo el desempeño del algoritmo de hormigas se acerca al del algoritmo ideal.

Para el tráfico fractal usamos una traza sintética de 80 s obtenida mediante un modelo wavelet multifractal [20] con un promedio de 8 paquetes por milisegundo y parámetro de Hurst igual a 0.8. La figura 5 muestra la variabilidad del tráfico a múltiples escalas así como la variabilidad de las trazas de feromonas en cada uno de los enlaces. Las columnas muestras, de izquierda a derecha, la tasa de llegadas y las trazas de feromonas en los enlaces 0 y 1, 2 y 3, 4 y 5, y 6 y 7. Las filas muestran los promedios a distintas escalas de tiempo que van desde 1 ms hasta 625 ms.

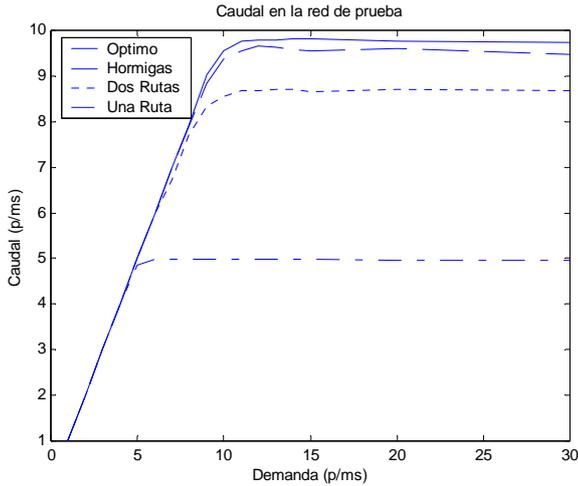


Figura 4. Caudal de los algoritmos de enrutamiento bajo tráfico de Poisson

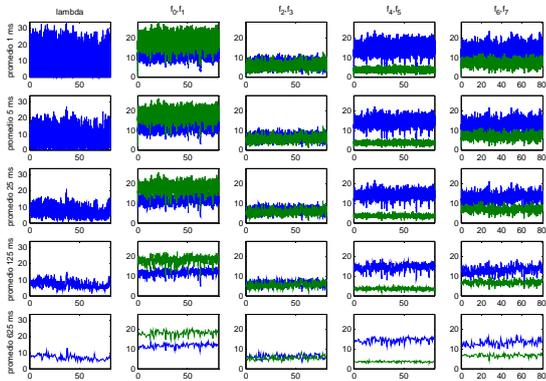


Figura 5. Resultados con tráfico fractal en la red multisalto

Se puede observar cómo la variabilidad de las trazas de feromonas en cada enlace siguen el comportamiento multiescala del tráfico de entrada, con la particularidad de que, en escalas de tiempo suficientemente grandes (del orden de 1 s), el promedio de las trazas de feromonas logran dispersar el tráfico de una manera óptima sobre todos los enlaces de la red. Nuevamente, las hormigas demuestran adaptabilidad y optimalidad cuando se trata de dispersar tráfico fractal sobre múltiples rutas alternas, a pesar de las diferencias topológicas y algorítmicas con los experimentos del capítulo anterior.

IV. CONCLUSIONES

Hemos verificado la adaptabilidad multiescala y la optimalidad de las soluciones basadas en las propiedades emergentes de las colonias de hormigas para balancear la carga entre los enlaces de una red. Diferentes tipos de algoritmos en diferentes tipos de red muestran las mismas propiedades de adaptabilidad y optimalidad, si se diseñan adecuadamente, lo cual sugiere que éstas son propiedades inherentes a la inteligencia de enjambre aplicada en este tipo de problemas complejos.

REFERENCIAS

- [1] D.Awduche, J.Malcolm, J.Agogbua, M.O'Dell and J.McManus. Requirements for Traffic Engineering Over MPLS. Request for Comments: 2702, September 1999
- [2] A.Elwalid, C.Jin, S.Low and I.Widjaja "MATE: MPLS Adaptive Traffic Engineering" IEEE Infocom 2001.
- [3] W. Leland, M. Taqu, W. Willinger and D. Wilson. "On the self-similar nature of Ethernet Traffic". IEEE/ACM Trans. Networking, 2:1-15, 1994.
- [4] W. Willinger, and V. Paxson, "Where Mathematics meets the Internet," Notices of the American Mathematical Society, vol. 45, no. 8, Aug. 1998, pp. 961-970.
- [5] J.F.Traub and A.G.Werschulz. "Complexity and Information", Cambridge University Press, Cambridge, UK, 1999.
- [6] M.Michelle "An introduction to genetic algorithms" MIT Press, Cambridge, MA, 1998
- [7] F.W.Glover and M.Laguna. "Tabu Search" Kluwer Academic, 1998.
- [8] J.M.Zurada. "Introduction to Artificial Neural Networks", West Publishing Company, St. Paul, MN, 1992.
- [10] E.Bonabeau, M.Dorigo and G.Theraulaz. "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, 1999.
- [11] S. Appleby and S. Steward. "Mobile software agents for control in telecommunications networks". BT Technology Journal, vol. 12, No. 2, april 1994.
- [12] M.Dorigo. "Optimization, Learning and Natural Algorithms". Ph.D.Thesis, Politecnico di Milano, Italia, 1992
- [13] M.Dorigo, V.Maniezzo y A.Colonni. "The Ant System: Optimization by a Colony of Cooperating Agents". IEEE Transactions on Systems, Man and Cybernetics, Vol 26, No 1, pp.1-26, enero de 1996.
- [14] G.DiCaro and M.Dorigo. "Ant colonies for adaptive routing in packet-switched communications networks". Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature, pages 673--682. Springer-Verlag, 1998.
- [15] Bundgaard M., Damgaard T. y cols: Ant Routing System. University of Copenhagen – Internet Technology. 2002.
- [16] Suárez S. y Alzate M.: Enrutamiento de Paquetes en una Red Simulada por Medio de Algoritmos Inspirados en Hormigas. Third LACCEI International Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2005) "Advances in Engineering and Technology: A Global Perspective". Junio 2005; Cartagena de Indias, COLOMBIA. www.laccei.org
- [17] G. Malkin. "RIP Version 2", RFC 2453, 1998
- [18] D. Bertsekas y R. Gallager: Data Networks, 2nd edition, Prentice Hall, 1992.
- [19] Fitzek F. y Reisslein M.: MPEG-4 and H.263 Video Traces for Network Performance Evaluation. IEEE Network Magazine. Noviembre 2001; Vol. 15, n°6, pp.40-54. .
- [20] R. Riedi, et. al. "A Multifractal Wavelet Model with Application to Network Traffic". IEEE Trans. Inf. Theory, 45(3):992-1018, 1999.

**Sandra M. Suárez** estudiante de Ingeniería Electrónica de la Universidad Distrital Francisco José de Caldas. Ha sido miembro IEEE durante 3 años consecutivos. Fue presidenta del Capítulo Estudiantil de Procesamiento de Señales IEEE y Presidenta de la Rama Estudiantil IEEE de la Universidad Distrital, Bogotá, Colombia (2005).

**Marco A. Alzate** es ingeniero electrónico de la Universidad Distrital, Bogotá, Colombia y Máster en Ingeniería Eléctrica de la Universidad de los Andes en Bogotá. Actualmente adelanta su doctorado en la Universidad de los Andes, en colaboración con la Universidad de Maryland y la Universidad del Sur de la Florida. Es profesor titular de la Facultad de Ingeniería de la Universidad Distrital.