UNIVERSIDAD Privada DR. RAFAEL BELLOSO CHACÍN

TÉLÉMATIQUE

APLICABILIDAD DE LA METAHEURÍSTICA DE COLONIAS DE HORMIGAS AL ENRUTAMIENTO DE DATOS CON LOS PROTOCOLOS RIP & EIGRP

(Applicability of the colony of ant's metaheuristic data routing protocols RIP & EIGRP)

Recibido: 11/11/2014 **Aceptado:** 05/06/2015

Graterol, Verónica
Universidad del Zulia, Venezuela
vgraterol@gmail.com

Perozo, Ricardo
Universidad del Zulia, Venezuela
ricardoperozo@gmail.com

Pirela, Gerardo
Universidad del Zulia, Venezuela
gepirela@fec.luz.edu.ve

Jakymec, Juan
Universidad del Zulia, Venezuela
juan.jakymec@gmail.com

RESUMEN

En una arquitectura de red de datos, el algoritmo de ruteo se encarga de llevar los paquetes de datos desde el origen hasta el destino a través de rutas y con el uso de estructuras de datos, asegurando la eficacia y eficiencia de los envíos. RIP y EIGRP son dos ejemplos de tales algoritmos, o protocolos de enrutamiento. La metaheurística de colonias de hormigas (CH) surge como alternativa para los algoritmos clásicos de búsqueda de caminos óptimos. El presente artículo describe la aplicabilidad de la CH a los protocolos RIP y EIGRP. Se muestran los algoritmos RIP-ACO y EIGRP-ACO junto con un ambiente de simulación y pruebas que permitió variar la topología de red y los parámetros con los que se probaron los algoritmos para la medición de la efectividad y eficiencia de los mismos. Las pruebas arrojaron que el algoritmo RIP-ACO mostró mejor eficiencia que el RIP clásico, en cuanto a rapidez, la primera vez que se corre; sin embargo, RIP-Clásico mostró ser más eficiente cuando hay cambios en la topología de la red. Además, para topologías de baja densidad (mallas parciales, incompletas o estrellas de brazos largos), RIP-ACO no halla todas las distancias con los parámetros con los que fue implementado. Respecto el protocolo EIGRP, se apreció que el 80% del tiempo, la versión clásica encontró rutas óptimas, mientras que el 20% del tiempo, EIGRP-ACO fue capaz de obtener dichas rutas óptimas (las mismas halladas por la versión clásica). No obstante, debido a la naturaleza estocástica de EIGRP-ACO, sus resultados varían de ejecución a ejecución. Es necesario un extenso análisis teórico y empírico de los algoritmos para ajustar sus parámetros y evaluar mejor su rendimiento frente a las respectivas versiones clásicas.

Depósito Legal: PPPX200002ZU2142 / ISSN: 1856-4194. Volumen 14 Edición No 2 Julio – Diciembre 2015

Palabras clave: protocolo de enrutamiento, RIP, EIGRP, metaheurística de colonias de hormigas.

ABSTRACT

In data network architecture, the routing algorithm takes data packages from source to destination via paths and using data structures, ensuring delivery efficacy and efficiency. RIP and EIGRP are towed such routing algorithms or protocols. Ant-colony metaheuristics (AC) emerges as an alternative to the classic optimal-path algorithms. This paper describes AC applicability to the RIP and EIGRP protocols. We show the RIP-ACO and EIGRP-ACO algorithms along with a simulation and testing environment which allows changing network topology and algorithm's parameters to measure affectiveness and efficiency of both algorithms. Tests show that RIP-ACO turns out to be more efficient tan classic RIP with regards to running time the first time it is run; however, classic RIP behaves more efficiently when there are changes to the network topology. Furthermore, for low-density topologies (partial mesh, incomplete mesh, or long-arm star formations) RIP-ACO did not find all distances with the parameters with which it was implemented. Regarding EIGRP, 80% of the time the classic version found all shortest paths and distances while 20% of the time EIGRP-ACO was able to find all shortest paths and distances (the same ones that were found by the classic version). Nevertheless, due to the stochastic nature of EIGRP-ACO, these results vary among runs. Extensive theoretical and empiric analyses are necessary to both algorithms to adjust the parameters and better assess their performance compared to the respective classic versions.

Keywords: routing protocol, RIP, EIGRP, ant colony metaheuristics.

INTRODUCCIÓN

Teniendo en cuenta las necesidades y los avances producidos en la sociedad de la información y del conocimiento resulta de gran importancia la transmisión de la información, así como la necesidad de que ésta llegue a un destino en el momento preciso mediante el uso de las redes. Los principales cambios estructurales de la sociedad se producen ahora entorno al tratamiento y la transmisión de la información.

Por su parte, Hedrick (1988) explica que el RIP (Protocolo de enrutamiento de información) es un protocolo de enrutamiento basado en el algoritmo vector-distancia de Bellman-Ford y busca caminos óptimos mediante el conteo de saltos, considerando que cada router atravesado para llegar a su destino es un salto. Este algoritmo se ha utilizado para el enrutamiento de paquetes en redes de ordenadores desde los primeros días de la Arpanet.

El Algoritmo de Bellman-Ford se basa en generar caminos eficientes (desde su punto de vista) desde un nodo origen de grafo ponderado (el peso de algunas de las aristas pueden ser negativos) al resto de los nodos del mismo, solucionando el problema de la ruta más corta o camino mínimo desde un nodo origen. Una variante del algoritmo de Bellman-Ford se usa en protocolos de enrutamiento basados en vector-distancia, esto es el protocolo de enrutamiento de información (RIP). Sin embargo, Hedrick (1988) explica

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

tres limitantes que presenta el protocolo de enrutamiento de información (RIP): (a) no permite más de quince saltos; (b) el protocolo utiliza métricas fijas para comparar rutas alternativas; (c) cambios de topología en la red.

La limitante respecto a los saltos significa que dos enrutadores más alejados de la red no pueden distar más de 15 saltos, tomando al salto 16 como infinito y marcándolo inalcanzable en la tabla de enrutamiento. Este problema puede surgir en situaciones atípicas en las cuales se puedan producir bucles, ya que estos bucles pueden producir retardos e incluso congestión en redes en las cuales el ancho de banda sea limitado.

La limitante respecto al uso de métricas fijas se refiere que al comparar rutas alternativas el protocolo no está adecuado para escoger rutas que dependan de parámetros a tiempo reales como por ejemplo retardos, carga del enlace y ancho de banda del mismo. Finalmente, la limitante respecto a los cambios de topología de la red, es que no se reflejan rápidamente ya que las actualizaciones se distribuyen nodo por nodo. Además, RIP es un protocolo que genera muchísimo tráfico al enviar toda la tabla de ruteo en cada actualización, con la carga de tráfico que ello conlleva.

Asimismo, Cisco (2004) desarrolla en 1992 el protocolo de enrutamiento EIGRP (Enhanced Interior Gateway Routing Protocol), una versión mejorada de IGRP. El propósito principal de este desarrollo fue lanzar un protocolo de enrutamiento sin clase y con soporte de VLSM. Una de las mejoras más significativas es la implementación del algoritmo DUAL; evita los loops de enrutamiento rastreando todas las rutas y por medio de la métrica selecciona rutas eficientes y sin bucles, de esta forma termina seleccionando la ruta de menor costo. DUAL también garantiza una rápida convergencia en la actualización de los routers.

Para mejorar su mecanismo de trabajo el protocolo de enrutamiento EIGRP utiliza tres tablas de información; la tabla de vecinos guarda las rutas hacia los routers vecinos (directamente conectados); la tabla de topología contiene la información sobre las rutas EIGRP recibidas en actualizaciones y sobre rutas originadas localmente; la tabla de enrutamiento recibe la información de la tabla de topología para seleccionar la mejor ruta hacia cada destino.

Actualmente, existen técnicas alternativas de búsquedas llamadas metaheurísticas que han suplantado técnicas o algoritmos clásicos exitosamente, en particular existen la metaheurísticas de colonias de hormigas basadas en el comportamiento natural de las hormigas donde son capaces de seguir rutas más corta en su camino de ida y vuelta entre la colonia y una fuente de abastecimiento. Esto es debido a que las hormigas pueden transmitirse información entre ellas gracias a que cada una de ellas, al desplazarse, va dejando un rastro de una sustancia llamada feromona a lo largo del camino seguido.

Además, el algoritmo resultó ser eficaz sobre grafos pesados positivamente, con una complejidad computacional aproximadamente polinómica, dependiente de los parámetros básicos (cantidad de hormigas, ratas de aumento y evaporación de feromonas) además del típico tamaño del grafo. Los parámetros del algoritmo resultante pueden ser ajustados

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

para alcanzar una complejidad mejor que la del algoritmo de Bellman, pero no mejor que la del algoritmo de Dijkstra

Se plantea entonces estudiar la aplicabilidad, efectividad y eficiencia de estas técnicas de optimización emergente basadas en colonias de hormigas, para implementar el algoritmo de enrutamiento RIP y el algoritmo del protocolo EIGRP; en pos de evaluar si éstas técnicas son capaces de mejorar el desempeño computacional del algoritmo.

METODOLOGÍA

La metodología seleccionada para la aplicabilidad de la metaheurística de colonias de hormigas al enrutamiento de información con el protocolo RIP y al enrutamiento con el protocolo EIGRP fue la desarrollada por Kendall y Kendall (2005). Ellos plantean tres estrategias para el desarrollo de sistemas a saber: (a) el método clásico del ciclo de vida de desarrollo de sistemas; (b) el método de desarrollo por análisis estructurado y (c) el método de construcción de prototipos de sistemas.

De allí que la metodología a utilizar para el desarrollo del proyecto fue una adaptación del método clásico del ciclo de vida de desarrollo de sistemas, el cual a su vez consta de seis (6) fases.

A saber, la primera es la identificación del problema, oportunidades y objetivos; la segunda es la determinación de los requerimientos de la información; la tercera es el diseño del sistema recomendado; la cuarta es el desarrollo del sistema; la quinta es la prueba de la simulación de los algoritmos; la sexta es la comparación de los resultados de los dos algoritmos desarrollados.

Fase 1: identificación del problema, oportunidades y objetivos:

El desarrollo de toda aplicación de software se plantea por la manifestación de una idea principal potencialmente beneficiosa para el proceso del acceso a la información. La idea principal constituye qué es lo que se quiere crear y el cómo, identificando las necesidades y reconociendo el problema, definiendo el propósito de la aplicación del software.

De acuerdo con lo planteado, el problema del algoritmo del protocolo de enrutamiento RIP es que no permite más de quince saltos, igualmente, utiliza métricas fijas para comparar rutas alternativas y los cambios en la topología de red no se reflejan rápidamente (Kendall y Kendall, 2005). Mientras que el algoritmo del protocolo EIGRP es la implementación del algoritmo DUAL; el cual evita los loops de enrutamiento rastreando todas las rutas y por medio de la métrica selecciona rutas eficientes y sin bucles, de esta forma termina seleccionando la ruta de menor costo.

Fase 2: determinación de los requerimientos de la información:

En esta fase se establecieron las condiciones de utilización en la simulación del funcionamiento del protocolo de enrutamiento RIP, protocolo de EIGRP y la Metaheurística de colonias de hormigas. En este sentido, el sistema operativo fue de

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Microsoft Windows XP Profesional con las últimas actualizaciones instaladas, versión 2002, con Service Pack 2003, el hardware utilizado fue una Pentium (R) Dual-core CPU 2.496Hz con 992MB de RAM. Se seleccionó el lenguaje de programación java, específicamente en el Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) NetBeans en su versión 6.8.

De igual forma, se instaló el Java Development Kit (JDK), la versión del JDK que se utilizó fue la 1.6.0_18. También, se necesitó instalar previamente el Kit de Desarrollo de Java J2SE (Java 2 Platform Standard Edition) porque ofrece un completo entorno para escribir applets y para el desarrollo de aplicaciones en servidores, con el lenguaje de programación java.

Además, estas herramientas están diseñadas para utilizarse desde la línea de comandos, sin ofrecer un interfaz gráfico de usuario. Este kit incluye una JVM (Java Virtual Machine, Máquina Virtual de Java), una ApplicationProgramming Interfaces (API), Interface de Programación de Aplicaciones. Una vez que se instaló el kit de desarrollo, la máquina virtual de Java (JVM) permitió el funcionamiento del programa sobre cualquier sistema.

Fase 3: diseño del sistema recomendado:

En esta investigación, la idea principal es tener una herramienta de simulación de tráfico en redes que contenga las características topológicas de nodos (routers) en redes de datos (grafos) incorporando el algoritmo del protocolo RIP basado en Bellman-Ford,el algoritmo del protocolo EIGRP basado en su funcionamiento convencional, el algoritmo de RIP y EIGRP basado en colonias de hormigas.

Fase 4: desarrollo del sistema:

Según Kendall y Kendall (2005), el analista trabaja con los programadores para desarrollar cualquier software original que se necesite. Algunas de las técnicas estructuradas para el diseño del software incluyen diagramas de flujo, y pseudo-código realizados en ésta investigación. Los programadores tienen un papel principal en esta fase conforme diseñan, codifican y eliminan errores de sintaxis de los programas de computadora para asegurar la calidad, un programador puede realizar ya sea un diseño o un ensayo de código, explicando las partes complejas del programa a un equipo de otros programadores.

De modo similar, se codificó en lenguaje java la simulación del tráfico de red, desarrollando el algoritmo del protocolo RIP y del protocolo de enrutamiento EIGRP en su forma convencional y después realizando el algoritmo de colonias de hormigas para realizar una comparación entre ambas simulaciones de algoritmos.

Fase 5: prueba de la simulación de los algoritmos:

Antes de colocar el sistema en funcionamiento es necesario realizarle pruebas iniciales que permitan analizar las posibles modificaciones o correcciones de errores, tanto en interfaz como en los otros aspectos considerados en su elaboración.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

En este estudio, se creó un escenario de prueba seleccionando la topología de red tipo estrella con una cantidad de nodos y enlaces para comprobar el funcionamiento del algoritmo ACO con trescientas (300) hormigas y luego se transformó ese grafo de topología tipo estrella colocándole Ip de clase C a cada uno de los nodos (router) para probarlos en el algoritmo de RIP-Bellman en su forma convencional. Además, se utilizaron como tipo de pruebas, grafos con topología tipo estrella, grafos densos y grafos dispersos; cuyos parámetros de prueba fueron: tiempo, complejidad computacional, tablas de enrutamiento, la topología (configuración de nodos y las conexiones), la cantidad de hormigas.

Mientras que para el algoritmo EIGRP se creó un escenario de prueba seleccionando la topología de red tipo estrella con una cantidad de nodos y enlaces para comprobar el funcionamiento del algoritmo ACO con cien (100) hormigas y luego se transformó ese grafo de topología tipo estrella colocándole Ip de clase C a cada uno de los nodos (router) para probarlos en el algoritmo de EIGRP en su forma convencional.

Fase 6: comparación de los resultados de los dos algoritmos desarrollados:

En esta fase, se compararon los resultados de enrutamiento utilizando el algoritmo del protocolo RIP, protocolo EIGRP y la versión basada en colonias de hormigas; y luego se procedió a tomar los resultados obtenidos en la etapa anterior utilizando varias métricas de efectividad y eficiencia algorítmica, en consecuencia, se compararon los resultados de enrutamiento empleando el algoritmo del protocolo RIP, protocolo EIGRP y la versión basada en colonias de hormigas.

SIMULADOR DE PROTOCOLO RIP

RIP Clásico:

A continuación, se presenta la descripción del funcionamiento de cada una de las clases que conforman el simulador RIP, la cual lleva por nombre SimulRIP, dando una descripción del funcionamiento, estructuras, métodos y algoritmos que forman parte de ellas. De este modo se explicarán solamente las clases utilizadas:

Clase Grafo: la clase Grafo de RIP implementa Runnable y representa la red de nodos en su totalidad la cual maneja las estructuras que se necesita para simular los nodos, para ellos se manejan unas estructuras (nodos, hilo, detener, flag, eventos, ips).

Clase Nodo: la clase Nodo de RIP es aquella clase que representa los routers y pcs que se manejan en la clase grafo. Esta contiene un objeto tipo vector tabla enrutamiento, los seriales que manejan los routers y pcs, el tiempo, un objeto tipo vector eventos y un objeto valor.

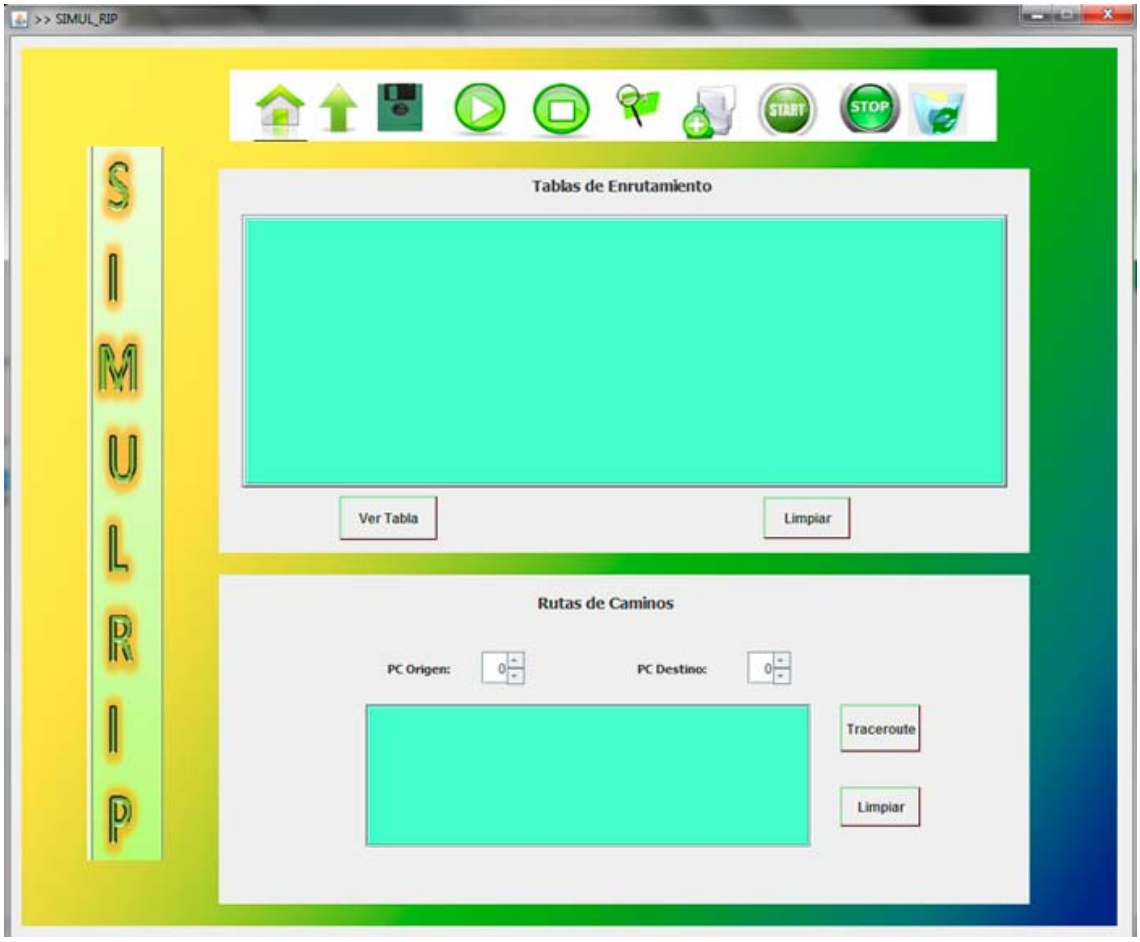
Clase Tabla de Enrutamiento: la clase Tabla de Enrutamiento de RIP es aquella estructura que usa el objeto nodo para modelar la tabla de enrutamiento que utiliza el protocolo de enrutamiento (RIP). Están también los métodos para modificar las mismas.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Clase Protocolo_RIP: la clase Protocolo_RIP es la clase principal del simulador, se encarga de leer un grafo genérico, que comparten caracteres en común del formato con que trabajan las tablas de enrutamiento del Protocolo RIP.

Clase RoutingInformationProtocol: la clase RoutingInformationProtocol es la clase que maneja toda la simulación del protocolo RIP que se hace visible al ejecutar inicialmente el programa, y presenta las diversas opciones para el funcionamiento del simulador.

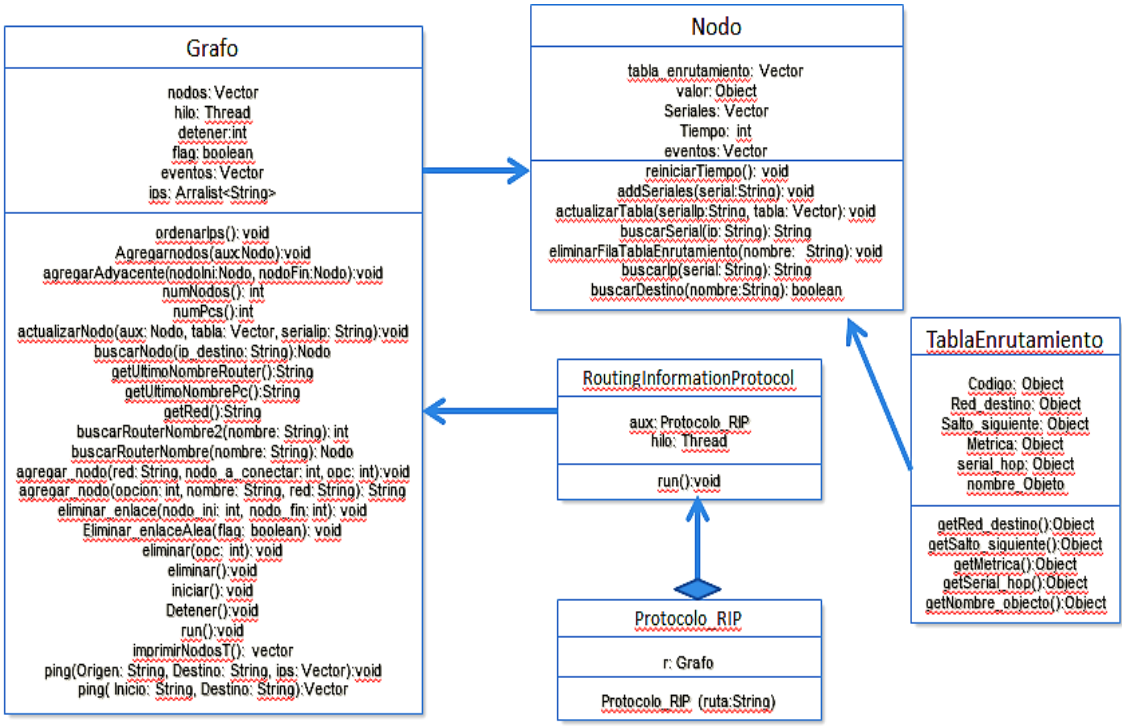
Figura 1. Diagrama de Clases de RIP



Fuente: elaboración propia.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Figura 2. Diseño de la interfaz de la clase RoutingInformationProtocol



Fuente: elaboración propia.

IMPLEMENTACIÓN DEL ALGORITMO DE HORMIGA

A continuación, se presentan las clases que conforman el algoritmo de hormiga, la cual lleva por nombre ACO dando una descripción del funcionamiento, estructuras, métodos y algoritmos que forman parte de ellas:

Clase Grafo: la clase Grafo de ACO se basa en las teorías de grafos y éstas se implementan para su modelado, dentro de este método se manejan estructuras tales como Vector nodos; intnodoNido.

Clase Nodo: la clase Nodo de ACO es aquella clase que representa los nodos que se manejan en la clase grafo, la feromona que representa los arcos hacia los adyacentes

Clase Adyacente: la clase Adyacente de ACO implementa Compare es aquella clase que la utiliza el nodo_siguiente (Grafo g) para ordenar nodos con máximo valor de feromona.

Clase Hormiga: la clase Hormiga de ACO es aquella clase que representa a la hormiga que maneja estructuras de almacenamiento (camino, acumulado, activa, nod_actual, nodo_anterior, escojecomida, comida, nuevas_comidas, comidas_posibles).

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Clase Colonia: la clase Colonia de ACO es aquella clase que representa a la Colonia de Hormiga y maneja estructuras de almacenamiento (hormigas [ ], comidas posibles, frecuencias de comidas posibles) y los métodos para modificar las mismas. De igual manera, presenta métodos para poder almacenar los distintos estados que presenta la hormiga como el obtengo comida, actualizar comidas posibles y ACO. En el constructor de Colonia se pasa por parámetro cantidad de hormigas y Grafo.

Clase Ruta: la clase Ruta de ACO es aquella estructura que usa variables privadas tipos enteras para modelar la tabla de enrutamiento, tiene como atributos (nod_destino, salto siguiente y la métrica). Están también los métodos para modificar las mismas

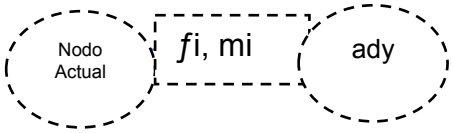
Clase Main: la clase main de ACO es aquella clase que define los grafos en listas de adyacencia que utilizan para la ejecución del algoritmo de hormigas. Se instancia grafo y se le pasa por parámetro matriz_peso definitiva en el main, el nodoNido de donde va a estar corriendo ACO y el contador igual a 5. Después, se instancia colonia pasándole por parámetro la cantidad de hormiga y g llamando al método ACO de la Clase Colonia.

FUSIÓN DE LOS ALGORITMOS RIP-ACO

Para el proceso de selección del nodo siguiente durante la ejecución del algoritmo, se tomó como muestra el algoritmo desarrollado por los autores Saeedi y Mahdavi (2007), y se adaptó a esta investigación de la siguiente manera:

Dado el nodo actual y sus adyacentes, ady; tomando los niveles de feromona fi y métrica ni de las aristas, como se muestra a continuación:

Figura 3. Proceso de Selección de Nodos



Fuente: elaboración propia.

Se realizó la adaptación de los dos algoritmos para su unión.

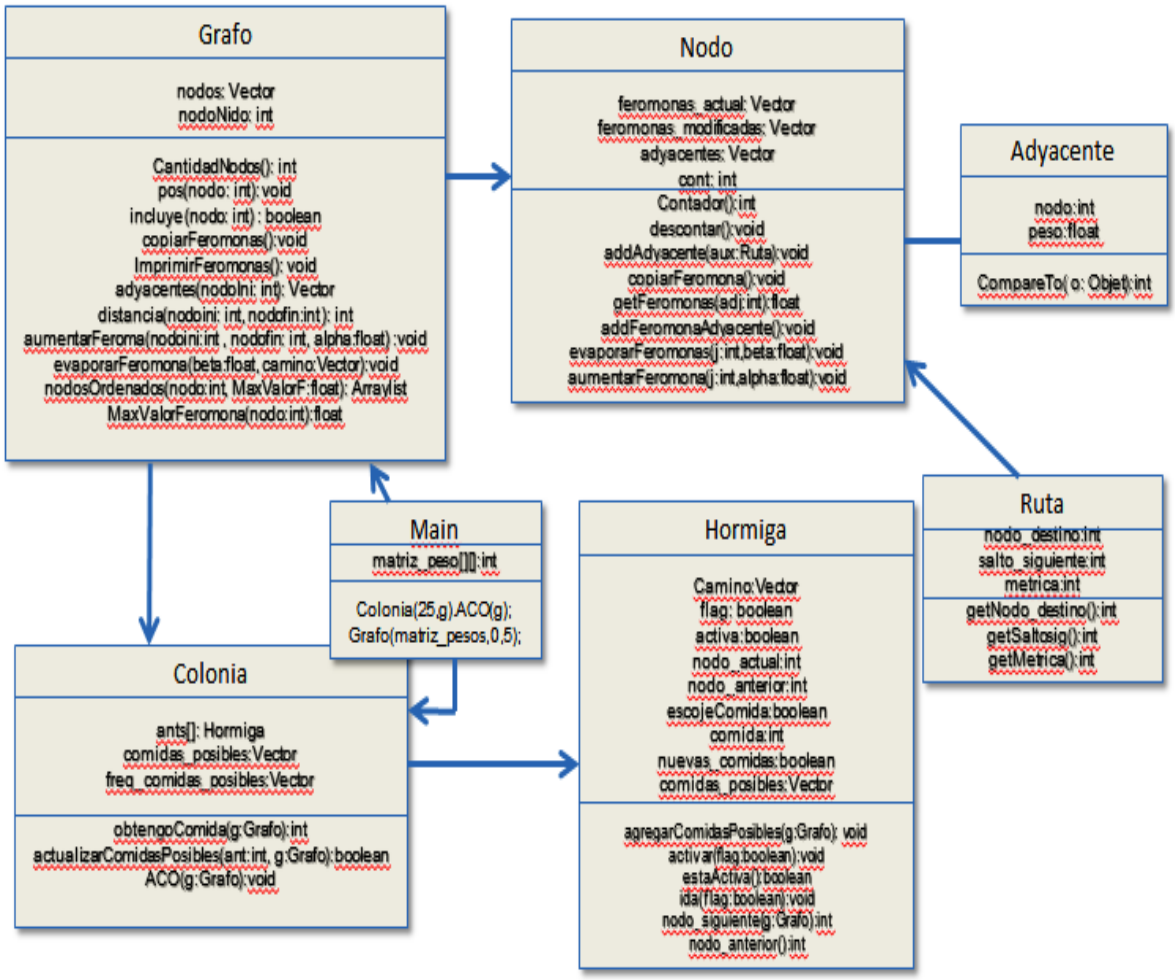
Al empezar la simulación, por el algoritmo de ACO, ACO va a correr desde cada nodo con hilo, cuando se termina de correr ACO el resultado de ACO va a hacer las tablas de enrutamiento de cada nodo para luego empezar a simular el resto de RIP.

Ahora, si en plena simulación un nodo se cae o entra un nuevo nodo a la red, se va a correr de nuevo el algoritmo de hormiga dentro de la simulación. Si entra un nodo nuevo, este tiene que mandar a correr su hormiga para tener su tabla. Si cae un nodo, todos los nodos tienen que mandar a correr hormiga para actualizar su tabla, ACO se va a mandar a correr en todos los nodos antes que el RIP.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

De esta manera, se ejecutará el algoritmo de ACO sólo para lo que en RIP se corría Bellman, es decir, el relax del algoritmo de Bellman se convirtió en ACO. En vez de ejecutar el relax se va a mandar a correr ACO hacia todos los nodos de la red.

Figura 4. Diagrama de Clases de RIP-ACO



Fuente: elaboración propia.

PRUEBAS DE EFICACIA Y EFICIENCIA RIP-ACO

A continuación, se muestra el funcionamiento del simulador con diversos ejemplos de grafos cargados por archivo. Así mismo, se muestran resultados del algoritmo de ACO usados para entonar el parámetro referente a cantidad de hormigas con el que se incorporará el mencionado algoritmo ACO al simulador de RIP.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

En estas pruebas, se construyeron varios grafos con distintas topologías, grados de conectividad y densidad, y se corrió ACO variando el número de hormigas para determinar la cantidad suficiente de estas para el cálculo efectivo de las distancias.

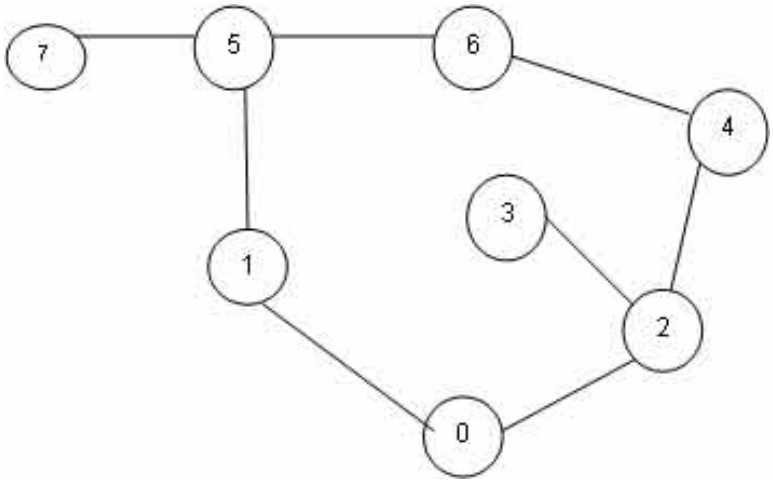
Los valores de los parámetros α y β se mantuvieron constantes, según recomendaciones de los autores Dorigo y Stültze (2004): α= 1 y β= 0.1. Las fórmulas para el aumento y evaporación de feromonas se usaron tal y como fueron descritas en el apartado (a) ruta de seguimiento y actualización de feromona, así como (b) evaporación del camino de feromona), respectivamente:

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta_{\tau}^{k} \qquad \text{(a)}$$

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \in A, \qquad \text{(b)}$$

A continuación, se muestran tabulados los resultados de estas pruebas para cada grafo considerado.

Figura 5. Grafo para Calcular la Distancia con 100 Hormigas



Fuente: elaboración propia.

En esta figura se observa un grafo entrelazado con una cantidad de 8 nodos y de 8 aristas, con una densidad de grafo de 0<0,28<1 cercano a cero (grafo disperso), y una topología de malla parcial; el algoritmo de ACO se está corriendo desde el nodo 0, es decir, nodo fuente (nido) de la colonia de donde saldrá una cantidad de 100 hormigas en busca de su comida en los nodos del grafo que son seleccionados aleatoriamente como comidas posibles, y al regresar al nido reportan su mejor distancia para garantizar el cálculo efectivo de las distancias.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

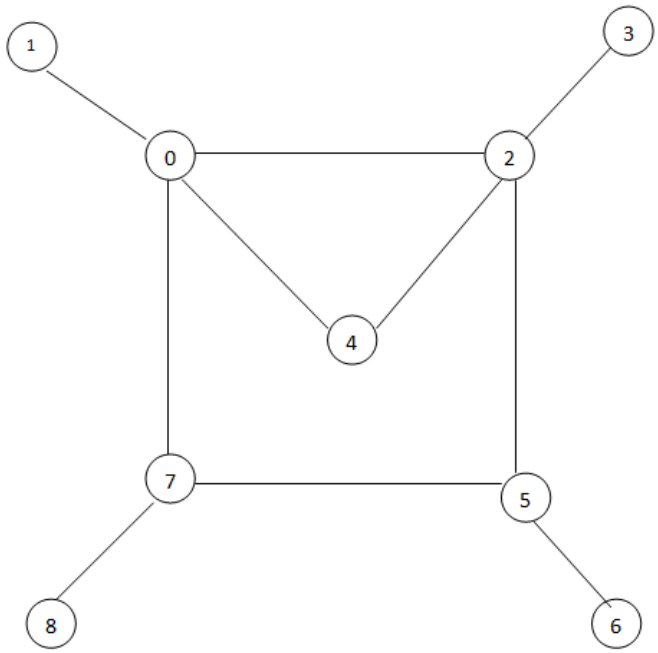
Tabla 1. Cálculo de la Distancia con 100 Hormigas

#Hormigas/N	Nodos	0	1	2	3	4	5	6	7
25	Deltas	0	1	1	2	4	2	3	3
	NSig	-1	-2	-2	2	1	1	1	1
50	Deltas	0	1	1	2	2	4	3	5
	NSig	-1	-2	-2	2	2	2	2	2
75	Deltas	0	1	1	2	2	4	3	5
	NSig	-1	-2	-2	2	2	2	2	2
100	Deltas	0	1	1	2	2	2	3	3
	NSig	-1	-2	-2	2	2	1	1	1

Fuente: elaboración propia.

Se observa en la Tabla 1 que fue necesaria una cantidad de 100 hormigas para garantizar el cálculo efectivo de todas las distancias; mientras tanto, con 25, 50 y 75 hormigas no son óptimas para el cálculo de las distancias.

Figura 6. Grafo para calcular la distancia con máximo 100 Hormigas



Fuente: elaboración propia.

En esta figura se observa un grafo entrelazado con 9 nodos y 10 aristas, con una densidad de 0,27 (grafo moderadamente disperso) y una topología de malla parcial; el

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

algoritmo de ACO se corrió desde el nodo 0, es decir, nodo fuente (nido de la colonia) de donde salieron desde 25 y hasta 100 hormigas en busca de su comida en los demás nodos del grafo, seleccionados aleatoriamente como comidas posibles.

Tabla 2. Cálculo de la distancia con máximo 100 hormigas

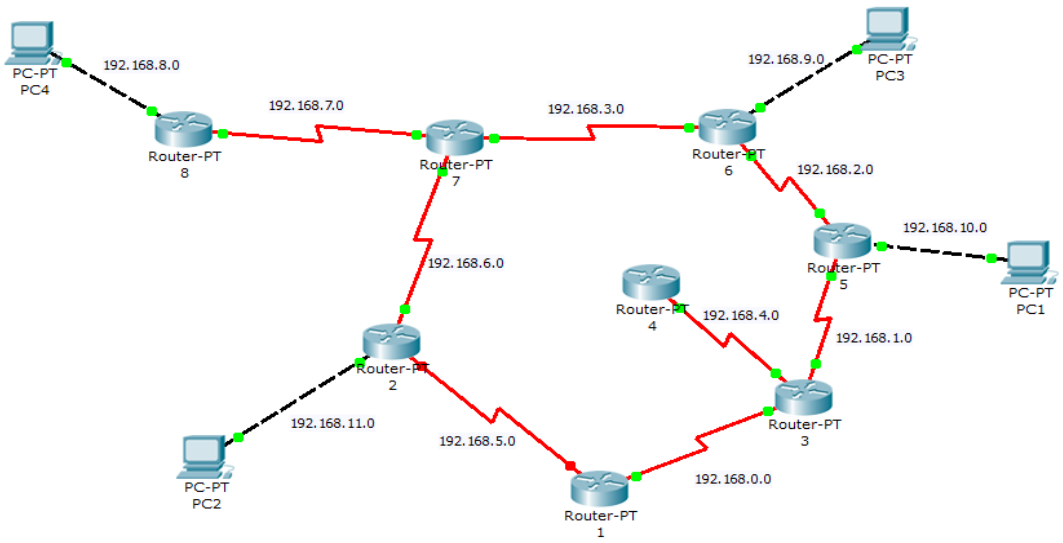
#Hormigas/N	Nodos:	0	1	2	3	4	5	6	7	8
25	Deltas	0	1	1	2	1	2	3	1	4
	NSig	-1	-2	-2	2	-2	2	2	-2	2
50	Deltas	0	1	1	3	1	3	4	1	5
	NSig	-1	-2	-2	4	-2	4	4	-2	4
75	Deltas	0	1	1	4	1	2	3	1	2
	NSig	-1	-2	-2	7	-2	7	7	-2	7
100	Deltas	0	1	1	2	1	2	3	1	2
	NSig	-1	-2	-2	2	-2	2	2	-2	7

Fuente: elaboración propia.

Se observa de nuevo en la Tabla 2 que fue necesaria una cantidad máxima de 100 hormigas para garantizar el cálculo efectivo de todas las distancias, mientras tanto, 25, 50 y 75 hormigas no son suficientes para el cálculo de las distancias.

Se puede evidenciar que la repetición de tales resultados se deba a la similitud de topologías de las redes de pruebas: ambas de malla parcial, moderadamente dispersas (densidad= 0,275=0,005).

Figura 7. Topología de Malla Parcial



Fuente: elaboración propia.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

En esta figura se observa una red con una topología de malla parcial con una cantidad de 8 routers, 12 interfaz y 4 PCs; se utiliza Ips de clase C cuya red es 192.168.0.0 con máscara /24.

Tabla 3. Tablas de Enrutamientos de RIP-Clásico

Tabla de Router 1: R Red destino: 192.168.9.1 Métrica: 4 Salto siguiente: 192.168.5.1Serial:3Router Destino:PC3
Tabla de Router 2: R Red destino: 192.168.10.1 Métrica: 4 Salto siguiente: 192.168.6.2Serial:6Router Destino:PC1
Tabla router 3: R Red destino: 192.168.11.1 Métrica: 3 Salto siguiente: 192.168.0.1Serial:2Router Destino:PC2
Tabla router 4: R Red destino: 192.168.6.2 Métrica: 4 Salto siguiente: 192.168.4.1Serial:3Router Destino:R7
Tabla de router 5: R Red destino: 192.168.11.1 Métrica: 4 Salto siguiente: 192.168.1.2Serial:2Router Destino:PC2
Tabla de router 6: R Red destino: 192.168.11.1 Métrica: 3 Salto siguiente: 192.168.3.1Serial:3Router Destino:PC2
Tabla de router 7: R Red destino: 192.168.4.2 Métrica: 4 Salto siguiente: 192.168.3.2Serial:2Router Destino:R4
Tabla de router 8: Red destino: 192.168.10.1 Métrica: 4 Salto siguiente: 192.168.7.2Serial:2Router Destino:PC1

Fuente: elaboración propia.

Se puede observar que algunas de las tablas de enrutamiento de RIP-Clásico se revela que en los router 1, router 2 router 4, router 5, router 7, router 8 alcanza todas las distancias, ya que hay varios caminos con la misma longitud. Sin embargo, en router 3 y router 6 en algunas de las actualización de sus tablas alcanzó distancia con métrica 3 y también existen varios caminos con la misma longitud.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Tabla 4. Tablas de Enrutamientos de RIP-ACO

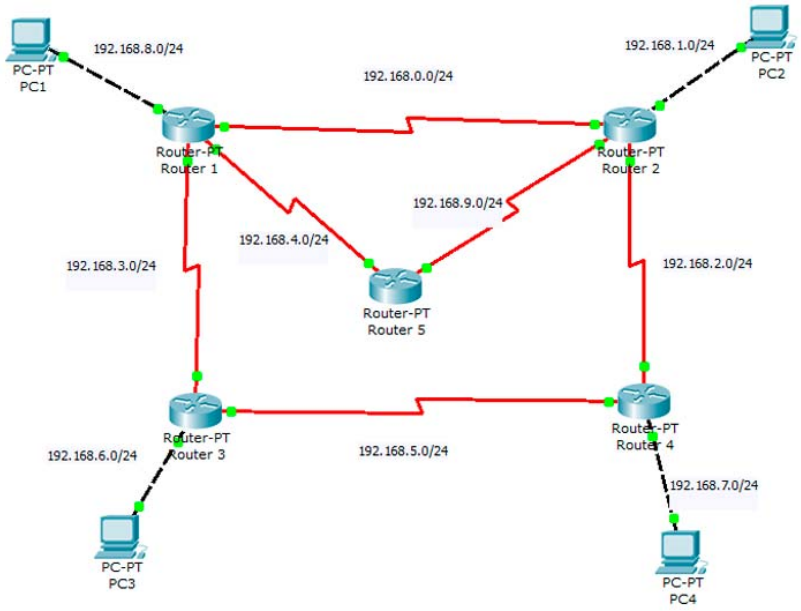
Tabla de Router 1: R Red destino: 192.168.9.1 Métrica: 4 Salto siguiente: 192.168.5.1Serial:3Router Destino:PC3
Tabla de Router 2: R Red destino: 192.168.10.1 Métrica: 4 Salto siguiente: 192.168.6.2Serial:2Router Destino:PC1
Tabla de Router 3: R Red destino: 192.168.11.1 Métrica: 3 Salto siguiente: 192.168.0.1Serial:6Router Destino:PC2
Tabla de Router 4: R Red destino: 192.168.2.1 Métrica: 3 Salto siguiente: 192.168.4.1Serial:2Router Destino:R6
Tabla de Router 5: R Red destino: 192.168.3.1 Métrica: 2 Salto siguiente: 192.168.2.1Serial:2Router Destino:R7
Tabla de Router 6: R Red destino: 192.168.8.1 Métrica: 3 Salto siguiente: 192.168.3.1Serial:2Router Destino:PC4
Tabla de Router 7: R Red destino: 192.168.4.2 Métrica: 4 Salto siguiente: 192.168.6.1Serial:6Router Destino:R4
Tabla de Router 8: R Red destino: 192.168.5.2 Métrica: 3 Salto siguiente: 192.168.7.2Serial:2Router Destino:R1

Fuente: elaboración propia.

Se puede observar que en algunas de las tablas de enrutamiento de RIP-ACO se revela que en los router 1, router 2, router 7, se alcanzan todas las distancias, ya que hay varios caminos con la misma longitud. Sin embargo, en router 3, router 4, router 6 y router 8 en algunas de las actualización de sus tablas se alcanzó distancia con métrica 3 y también existen varios camino con la misma longitud. Asimismo, router 5 alcanzó distancia con métrica 2.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Figura 8. Topología de Malla Parcial 2



Fuente: elaboración propia.

En esta figura se representa una implementación de la red equivalente al grafo mostrado en la Figura 5.

La figura 7 muestra una captura de pantalla de Packet Tracer referente a la implementación de la red, donde se observa una red con una topología de malla parcial con una cantidad de 5 routers, 10 interfaz y 4 PCs; se utiliza Ips de clase C cuya red es 192.168.0.0 con máscara /24. Para esta red se corrió el RIP-CLÁSICO y el RIP-ACO para comparar los resultados de ambos algoritmos.

Tabla 5. Tabla de Enrutamiento del Router 1 corriendo RIP-Clásico

Tablas de Enrutamiento R1
C Red destino: 192.168.8.2 Métrica: 1 Salto siguiente: *Serial:0Router Destino:PC1
C Red destino: 192.168.0.2 Métrica: 1 Salto siguiente: *Serial:2Router Destino:R2
C Red destino: 192.168.3.2 Métrica: 1 Salto siguiente: *Serial:3Router Destino:R3
C Red destino: 192.168.4.2 Métrica: 1 Salto siguiente: *Serial:6Router Destino:R5
R Red destino: 192.168.1.1 Métrica: 2 Salto siguiente: 192.168.0.2Serial:3Router Destino:PC2
R Red destino: 192.168.2.2 Métrica: 2 Salto siguiente: 192.168.0.2Serial:3Router Destino:R4
R Red destino: 192.168.6.2 Métrica: 2 Salto siguiente: 192.168.3.2Serial:2Router Destino:PC3
R Red destino: 192.168.7.2 Métrica: 3 Salto siguiente: 192.168.0.2Serial:3Router Destino:PC4

Fuente: elaboración propia.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Tabla 6. Tabla de Enrutamiento del Router 1 corriendo RIP-ACO

Tabla de Enrutamiento R1
C Red destino: 192.168.8.2 Métrica: 1 Salto siguiente: *Serial:0Router Destino:PC1
C Red destino: 192.168.0.2 Métrica: 1 Salto siguiente: *Serial:2Router Destino:R2
C Red destino: 192.168.3.2 Métrica: 1 Salto siguiente: *Serial:3Router Destino:R3
C Red destino: 192.168.4.2 Métrica: 1 Salto siguiente: *Serial:6Router Destino:R5
R Red destino: 192.168.2.2 Métrica: 2 Salto siguiente: 192.168.0.2Serial:2Router Destino:R4
R Red destino: 192.168.1.1 Métrica: 2 Salto siguiente: 192.168.0.2Serial:2Router Destino:PC2
R Red destino: 192.168.7.2 Métrica: 3 Salto siguiente: 192.168.3.2Serial:3Router Destino:PC4
R Red destino: 192.168.6.2 Métrica: 2 Salto siguiente: 192.168.3.2Serial:3Router Destino:PC3

Fuente: elaboración propia.

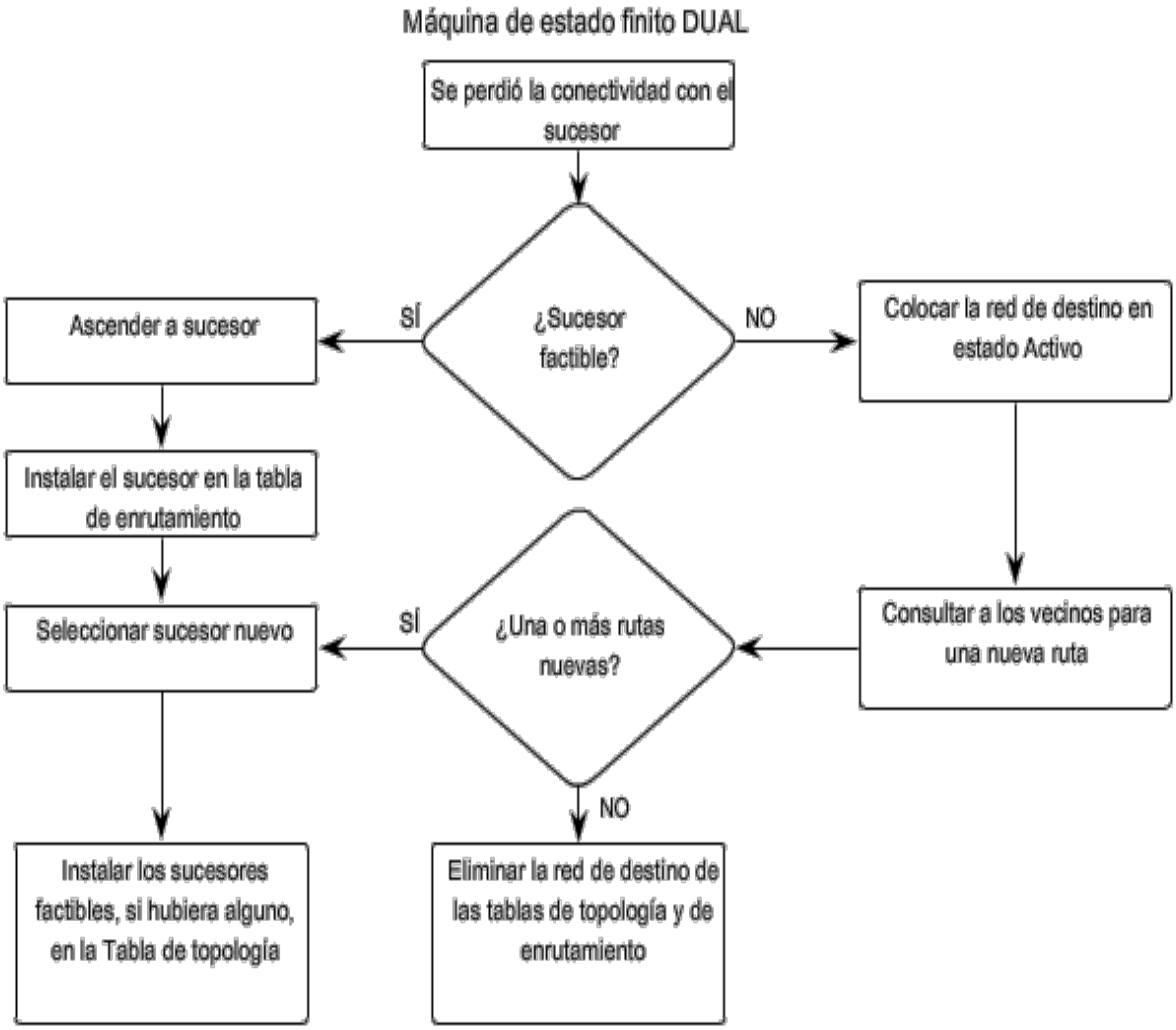
Se puede observar en la Tabla 5, la tabla de enrutamiento del Router 1 de RIP-Clásico muestra las métricas alcanzadas partiendo del router 1 hacia los demás nodos de la red, actualizando la mejor métrica a medida que se intercambia la información.

Por su parte, en la Tabla 6 se observa la tabla de enrutamiento del Router 1 de RIP-ACO, donde revela las métricas alcanzada hacia los nodos de la red, partiendo de router 1. Finalmente, se observó que ambos algoritmos obtuvieron las mismas métricas evidenciando la efectividad de RIP-ACO con respecto al grafo descrito anteriormente.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

MÁQUINA DE ESTADOS FINITO DUAL- EIGRP CLÁSICO

Figura 9. Funcionamiento de Dual-EIGRP Clásico



Fuente: elaboración propia.

Según Cisco (2004), la máquina de estados finito dual es la encargada de garantizar una operación sin bucles durante todo el cálculo de rutas, lo que permite la sincronización simultánea de todos los routers involucrados en el cambio de topología.

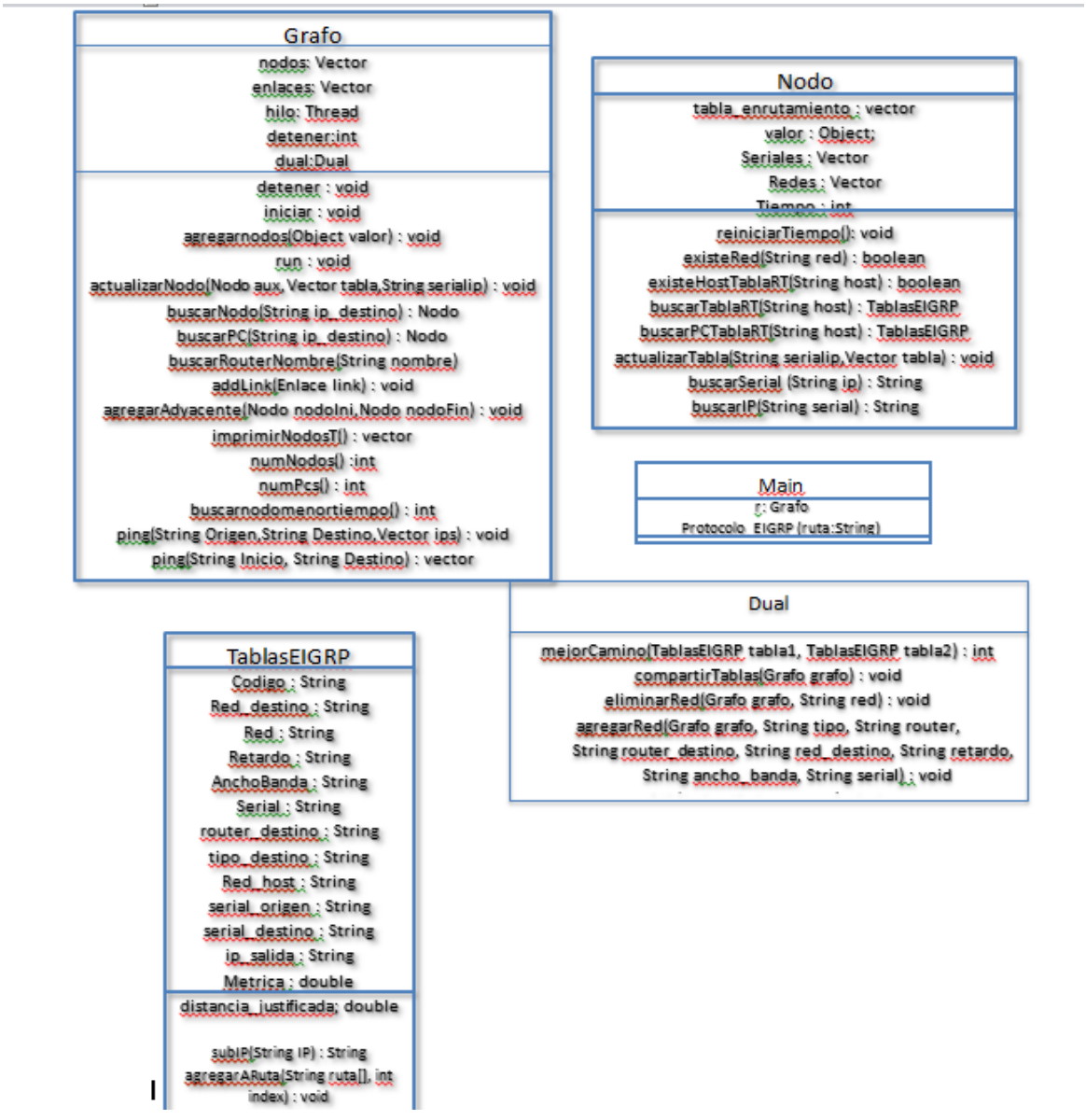
Los routers que no se ven afectados por los cambios de topología no se encuentran involucrados en el recálculo. Este método proporciona a EIGRP mayor tiempo de convergencia que a otros protocolos de enrutamiento vector distancia. La máquina de estados finitos DUAL está a cargo de la decisión para todos los cálculos de ruta

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

SIMULADOR DE PROTOCOLO EIGRP

A continuación, se presenta la implementación de cada una de las clases que conforman el simulador EIGRP-ACO:

Figura 10. Diagrama de Clases de EIGRP-Clásico



Fuente: elaboración propia.

Clase Dual: realiza las actualizaciones de las tablas, también se encarga de eliminar y agregar redes. Es la clase fundamental de la simulación ya que en ella se lleva a cabo el

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

proceso dinámico para gestionar todos los eventos que involucran las tablas de enrutamiento.

Clase Grafo: representa la red de nodos en su totalidad la cual maneja las estructuras que se necesita para simular los nodos.

Clase Nodo: representa los routers y pcs que se manejan en la clase grafo

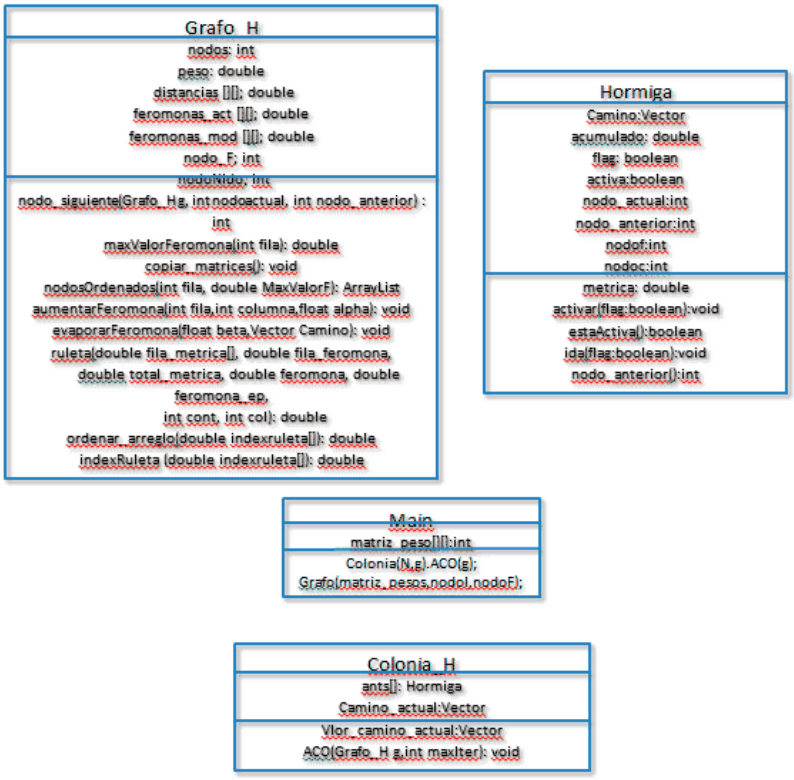
Clase TablasEIGRP: obtiene la dirección de destino, de la red y los seriales tanto de salida como de entrada del destino.

Main: maneja la clase Protocolo EIGRP que es la principal de la simulación. También se encarga de levantar el archivo de entrada.

Clase Principal: maneja toda la simulación del protocolo EIGRP que se hace visible al ejecutar inicialmente el programa, y presenta las diversas opciones que posee el programa para el funcionamiento del simulador.

EIGRP-ACO

Figura 11. Diagrama de Clases de EIGRP-ACO



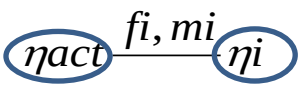
Fuente: elaboración propia.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Para el proceso de selección del nodo siguiente durante la ejecución del algoritmo, se tomó como muestra el algoritmo desarrollado por Saeedi y Mahdavi (2007) y se adaptó a esta investigación de la siguiente manera:

Dado el nodo actual, nact, y sus adycentes, ni; tomando los niveles de feromona fi y métrica mi de las aristas, como se muestra a continuación:

Se selecciona el nodo siguiente a través del proceso estocástico descrito a continuación:



$$Sea\ \eta\text{act}, \text{G}(\eta\text{act}) = \eta \rightarrow \#\ \text{adyacentes}$$
$$\text{sum} = 0$$
$$\forall\, \eta i \in \eta act.Adyacente(\ )$$
$$\rho i = \frac{fi + \varepsilon}{\sum (fi + \varepsilon)} * \frac{1 - \frac{mi}{\sum mi}}{n - 1}$$
$$sum += pi$$
$$\forall\, \eta i \in \eta act.Adyacente(\ )$$
$$ri = pi / sum;$$

Donde pi obtiene el resultado de multiplicar la frecuencia relativa por la conjugada de la frecuencia de la métrica, esto se hace debido a que se desea obtener el camino con menor métrica.

Donde cada valor ri representara la probabilidad que tendrá cada nodo adyacente (potencial nsig) de ser seleccionado utilizando un proceso pseudo-aleatorio que emula una ruleta.

Clase Colonia_H: busca el mejor camino desde el nido hasta la comida, enviando cierta cantidad de hormigas en un número determinado de iteraciones.

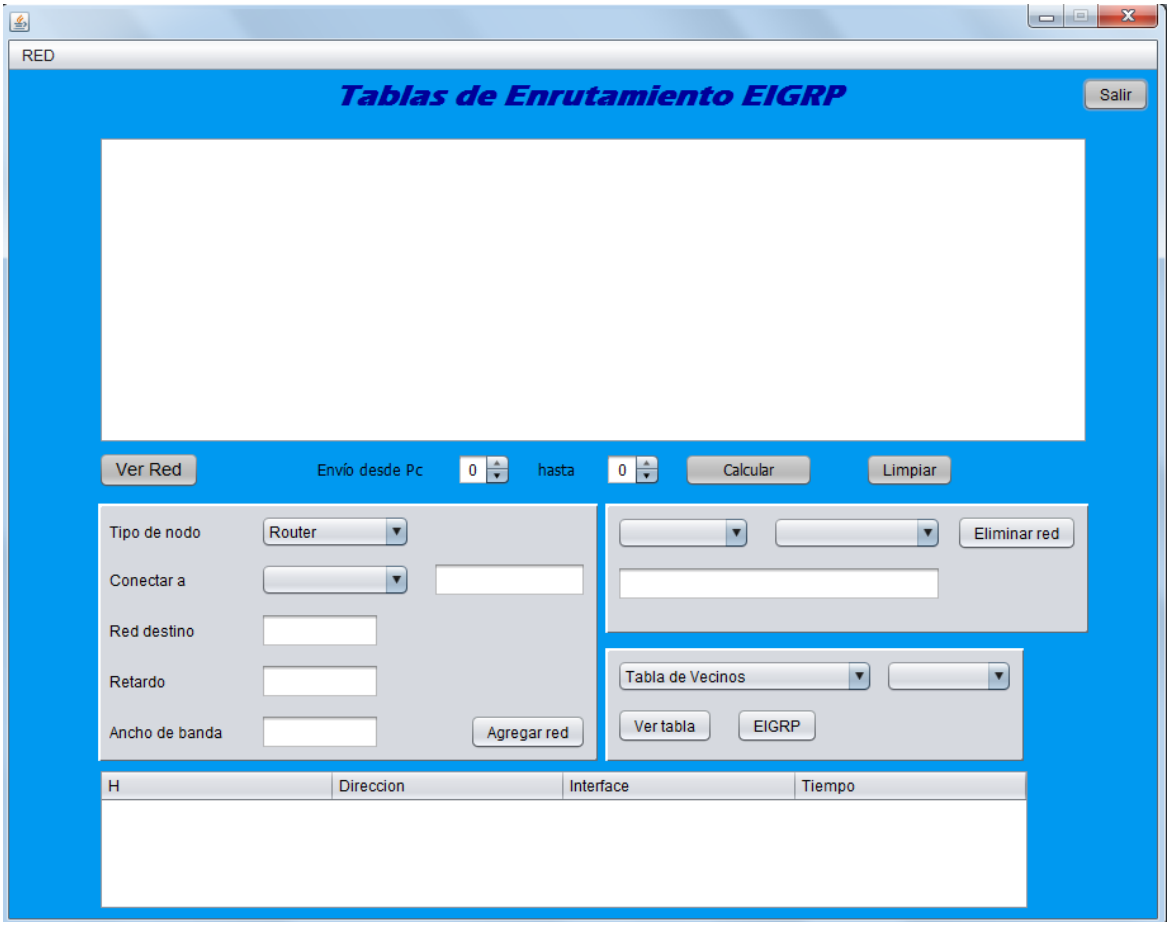
Clase Main: define los grafos para la ejecución del algoritmo de hormigas.

public class Hormiga: crear los objetos hormigas que serán los encargados de realizar el recorrido por el grafo para de esta manera encontrar caminos óptimos desde el nido a la comida.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

public class Grafo_H: realiza los cálculos correspondientes para encontrar los mejores caminos entre el nodo nido y el nodo comida. feromonas_act[ ] almacena la feromona al comienzo de la simulación que es 0; feromonas_mod[ ] [ ], almacena la feromona final para cada arco luego del recorrido de las hormigas.

Figura 12. Pantalla Inicial del Sistema



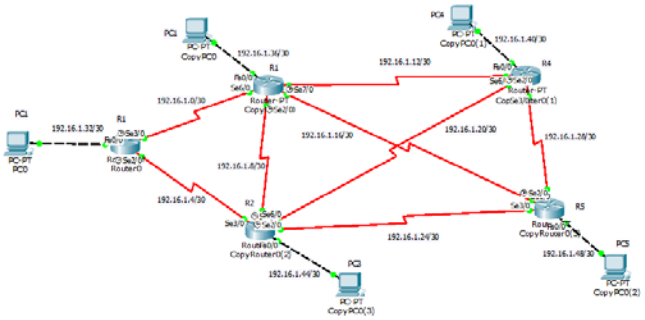
Fuente: elaboración propia.

PRUEBAS DE EFICACIA Y EFICIENCIA EIGRP-ACO

En estas pruebas se construyeron varios grafos con distintas topologías, grados de conectividad y densidad, y se corrió ACO con una cantidad de 100 hormigas y 25 iteraciones para determinar el cálculo efectivo de las distancias.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Figura 13. Representación del grafo. Red número 1



Fuente: elaboración propia.

La Figura 12 representa el grafo 1 o la red #1 construida. Está compuesta por 10 nodos, 5 para representar los routers y 5 para representar las PC's. La topología es de tipo malla parcial.

Tabla 7. Envío desde PC1 a todas las PC en la red número 1

		EIGRP	**EIGRP-ACO**
PC2	Caminos	R1-R2-PC2	R1-R2-PC2
	Métrica	14067316	14067316
PC3	Caminos	R1-R3-PC3	R1-R3-PC3
	Métrica	15059259	15059259
PC4	Caminos	R1-R3-R2-R4-PC4	R1-R3-PC3
	Métrica	10669227	431565597
PC5	Caminos	R1-R3-R2-R5-PC5	R1-R3-R4-R5-PC5
	Métrica	20480819	431565597

Fuente: elaboración propia.

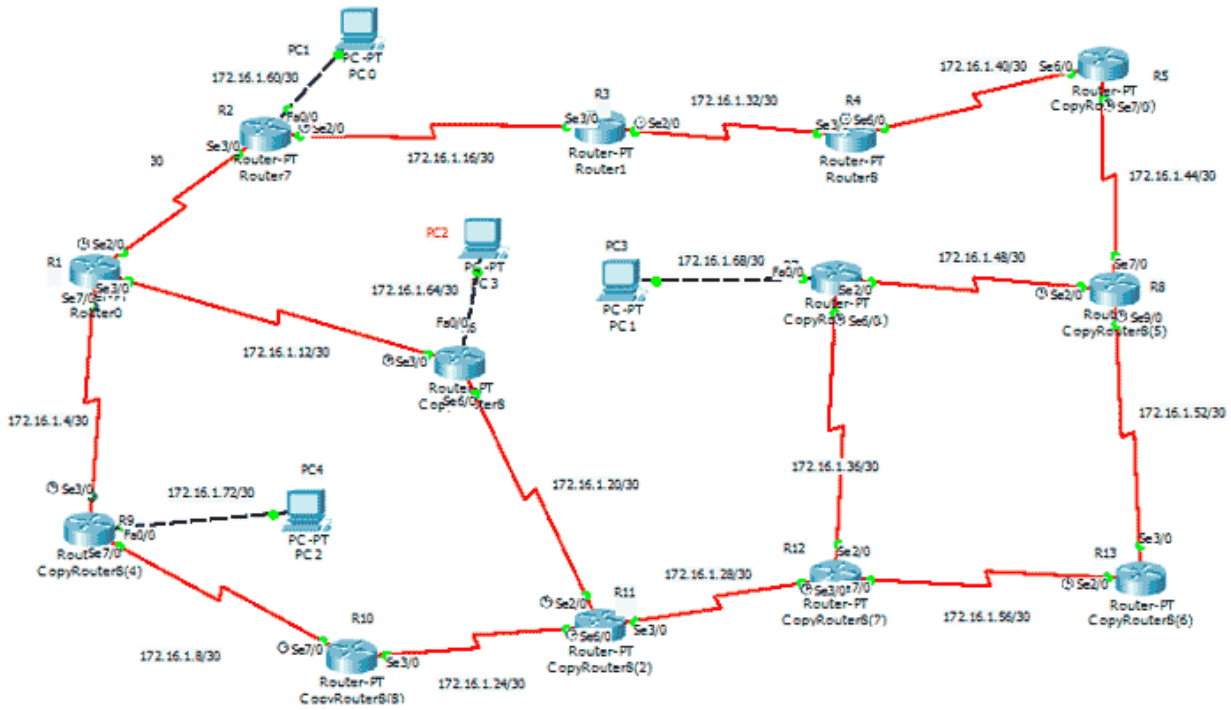
Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

La Tabla 7 muestra los caminos óptimos calculados por cada uno de los algoritmos, tomando a PC1 como el nodo origen y el resto de las PC como destinos, de manera sistemática para las pruebas realizadas.

Como puede notarse en la tabla anterior, el mejor camino encontrado para las PC2 y PC3 en ambos algoritmos coinciden, recorriendo la misma ruta y acumulando la misma métrica.

Con respecto a los caminos hacia PC4 y PC5 difieren entre sí. La ruta encontrada por EIGRP hacia PC4 debe recorrer una mayor cantidad de routers (nodos) y su métrica reportada es más baja que la métrica acumulada por EIGRP-ACO, a pesar que este recorre menos routers. La ruta hacia PC5 ambos presentan la misma cantidad de routers, tomando la ruta anunciada por EIGRP ya que acumula la mejor métrica.

Figura 14. Representación del grafo. Red número 2



Fuente: elaboración propia.

En la Figura 14 representa el grafo 2 o la red número 2 construida. Está compuesta por 17 nodos, 13 para representar los routers y 4 para representar las PC's. La topología es tipo anillo.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

Tabla 8. Envío desde PC1 a todas las PC en la red número 2

		EIGRP	EIGRP-ACO
PC2	Caminos	R2-R1-R6-PC2	R2-R1-R9-R10-R11-R6-PC2
	Métrica	2594538	14631117
PC3	Caminos	R2-R1-R6-R11-R12-R7-PC3	R2-R3-R4-R5-R8-R7-PC3
	Métrica	3259506	18543956
PC4	Caminos	R2-R1-R9-PC4	R2-R1-R6-R11-R10-R9-PC4
	Métrica	3778320	18357120

Fuente: elaboración propia.

La Tabla 8 muestra los caminos óptimos calculados por cada uno de los algoritmos, tomando a PC1 como el nodo origen y el resto de las PC como destinos representados en la red #2, de manera sistemática para las pruebas realizadas.

La Tabla 8 muestra cada uno de los caminos arrojados por cada uno de los algoritmos en cada caso. En el destino hacia PC2 la ruta óptima de EIGRP prevalece tanto en el camino como en la métrica acumulada. Hacia PC3 la ruta se equipara en cuanto a routers recorridos, destacándose la métrica acumulada por el algoritmo EIGRP. Y por último, la ruta óptima encontrada hacia PC4 es la reportada

CONCLUSIONES

El ambiente de prueba construido permite la corrida del algoritmo de enrutamiento de redes de datos, RIP-Clásico y la versión diseñada por los autores RIP basado en hormigas donde admite el análisis de los resultados para lograr una comparación entre ambos algoritmos, el algoritmo de RIP con hormiga mostró mejor eficiencia en cuanto rapidez la primera vez que se corre pero el RIP-Clásico mostró ser más eficiente cuando hay cambios en la topología de la red después de haber estado en funcionamiento.

Para el objetivo comparar la efectividad y la eficiencia de ambas versiones del protocolo RIP incorporadas al simulador. El algoritmo de RIP-ACO mostró mejor eficiencia en cuanto a rapidez la primera vez que se corre pero el RIP-Clásico mostró ser más eficiente cuando hay cambios en la topología de la red después de haber estado en funcionamiento. Con los parámetros implementados para ciertas topologías de densidad baja, como por ejemplos: mallas incompletas, estrella de brazos largos y buses; se determinó tras varias pruebas que el algoritmo RIP-ACO no logra hallar todas las distancias.

Por lo tanto, ninguna cantidad de hormiga fue suficiente para el cálculo de absolutamente todos los nodos, pero con 300 hormigas se determinó que era suficiente

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015

para calcular la cantidad máxima de hormiga para el algoritmo.

Debido a la naturaleza estocástica del algoritmo se notó que esas 300 hormigas para las topologías antes descritas a las que el algoritmo es susceptible, no siempre se calcula la misma cantidad de nodos, pero, se determinó 300 hormigas porque es la cantidad de hormigas para la cual, la gran mayoría de la veces el algoritmo calcula cierta cantidad más o menos aceptable de nodos. No obstante, para mallas parciales y mallas completas 100 hormigas resultaron suficientes para que el algoritmo RIP-ACO halle toda la distancia necesaria.

La efectividad y la eficiencia de la versión EIGRP-ACO incorporadas al simulador se midieron a través del diseño de varias redes que incorporaran características típicas de topología y densidad de interconexión, para luego correr simultáneamente dicho algoritmo y la versión EIGRP-Clásica. En todos los casos probados, la versión clásica mostró mejor o igual eficiencia en cuanto al camino encontrado, hallando siempre la métrica más baja que se obtiene de todos los caminos posibles. En el 20% de las pruebas realizadas, el algoritmo EIGRP-ACO equiparó los resultados con EIGRP-Clásico, con una ralentización apenas marginal de los tiempos de respuesta del mismo.

REFERENCIAS BIBLIOGRÁFICAS

Cisco System (2004). CCNA Exploration - Conceptos Y Protocolos De Enrutamiento. México. Editorial Pearson. Cisco.

Dorigo, M. y Stützle, T. (2004). Ant Colony Optimization. Estados Unidos. Massachusetts Institute of Technology: Brand Ford Book.

Hedrick, C. (1988). RFC 1058 Routing Information Protocol. Network Working Group, Documento en línea. Disponible en: http://www.rfc-editor.org/rfc/rfc1058.txt. Consulta: 02/06/2011.

Kendall, K. y Kendall, J. (2005). Análisis y Diseño de Sistemas. México. Pearson Educación.

Saeedi, S. y Mahdavi, I. (2007). Using ant colony optimization for shortest path problem. Documento en línea. Disponible en: http://dr-saeedi.ir/download/download/Papers/SPT_BAI2007.pdf. Consulta: 15/07/2011.

Depósito Legal: PPX200102ZU2313 / ISSN: 1856-6189. Volumen 14 Edición No 2 Julio – Diciembre 2015