UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

**PROGRAMA DE INGENIERÍA DE TELECOMUNICACIONES Y
REDES**

**“Desarrollo de una aplicación de software orientada a
la mejora de la gestión del encaminamiento de la
información en redes de computadora utilizando el
algoritmo de colonia de hormigas”**

TESIS

Para optar el título profesional de: Ingeniero de Telecomunicaciones y
Redes

AUTOR

Romero Quiroz, Percy Omar (0000-0002-5928-7590)
Chacayán Ventura, Nick Nelson (0000-0002-2158-3335)

ASESOR DE TESIS

Kemper Vásquez, Guillermo Leopoldo (0000-0002-7800-7769)

Lima, 24 de febrero de 2017

Nick Chacayán: A los que se fueron (mi abuelito Maximo, Ghommy y Sookie) y a los que siempre están (Andy). Gracias por la ayuda.

Percy Romero: A Dios, a quienes me cuidan desde el cielo (Mi hermano Yuri, mi mamá Coco y mi tío Lizardo), mi madre, mi padre y toda mi familia; también a Paolita, profesores y todas las personas que hicieron realidad este proyecto.

RESUMEN

En el Perú, la infraestructura de telecomunicaciones es muy dinámica ya que presentan muchos cambios en su topología de red y esto se debe a que conforme avanzan los años la infraestructura de telecomunicaciones crece y no necesariamente proporcional al crecimiento de usuarios. Este dinamismo se debe tomar en cuenta en los algoritmos de enrutamiento, ya que un algoritmo de enrutamiento es óptimo cuando este es más adaptable a los cambios en una topología de red y a los patrones de tráfico que existe en un determinado momento.

Este trabajo propone, desarrollar e implementar una aplicación de software para mejorar la gestión del encaminamiento de información en redes de computadoras basada en la optimización por colonia de hormigas, el cual tiene como base el modelamiento del comportamiento utilizado por las hormigas para la resolución del problema de obtención del camino mínimo entre su colonia y su fuente de alimentación. Por ello, el trabajo tiene como objetivo el uso de “hormigas artificiales” que simulen dicho comportamiento para buscar la ruta más rápida para poder transmitir información. Dicha ruta se actualiza en el tiempo según la cantidad de servidores y usuarios tenga dicha red. Con eso se busca cubrir en parte los problemas ocasionados por la desproporción entre la infraestructura de la red de telecomunicaciones, la cantidad de usuarios y por ende la alta carga de información transmitida. Además, de mejorar la eficiencia de la obtención de rutas mínimas cuando hay cambios en la topología de red de datos como incremento de servidores o caídas de dichos servidores.

Palabras clave: enrutamiento / algoritmo / optimización por colonia de hormigas

ABSTRACT

In Perú, the telecommunications infrastructure is very dynamic because it has many changes in their network topology and this is because as the years pass the telecommunications infrastructure is growing. This dynamism must be taken into account in the routing algorithms, since a routing algorithm is more optimal when it is adaptable to the changes in a network topology and to the traffic patterns that exist at a certain moment.

This work proposes, develops and implements a software application to improve the management of information routing in computer networks based on Ant Colony Optimization (ACO), which is based on the modeling of the behavior used by ants to solve the problem of obtaining the minimum path between their colony and their food source. Therefore, the work aims to use "artificial ants" that simulate this behavior to find the fastest route to be able to transmit information. This path is updated over time according to the number of servers and users that network has. This is intended to cover in part the problems caused by the disproportion between the infrastructure of the telecommunications network, the number of users and the high burden of information transmitted. In addition, to improve the efficiency of obtaining minimum paths when there are changes in the topology of data network as increase of servers or falls of such servers.

Keywords: routing / algorithm / ant colony optimization (ACO)

Tabla de Contenido

Tabla de Contenido.....	5
Índice de tablas	7
CAPÍTULO 1: ASPECTOS INTRODUCTORIOS	10
1.1.- Situación Problemática y Definición del Problema.....	10
1.1.1.- Situación Problemática.....	10
1.1.2.- Definición del Problema	16
1.2.- Estado del Arte	17
1.2.1.- Productos y soluciones existentes.	17
1.2.2.- Publicaciones Científicas/Ingenieriles	21
1.3.- Justificación.....	22
1.3.1.- Justificación Ingenieril:	23
1.3.2.- Justificación Económica:.....	23
1.3.3.- Justificación Social:	23
1.4.- Objetivos	24
1.4.1.- Árbol de objetivos	24
1.4.2.- Objetivo General	25
1.4.3.- Objetivos Específicos de implementación	25
1.5.- Breve Descripción de la Solución Propuesta	25
1.5.1.- Diagrama de bloques pictórico.....	25
1.5.2.- Funcionamiento.....	26
1.5.3.- Limitaciones de la solución.....	29
1.5.4.- Resultados esperados	29
1.6.- Aplicaciones y usuarios potenciales del producto.....	30
1.7.- Viabilidad.....	31
1.7.1.- Viabilidad técnica.....	31
1.7.2.- Viabilidad económica.....	31
1.7.3.- Viabilidad operativa	33
1.7.4.- Alternativas	33
CAPÍTULO 2: MARCO TEÓRICO	34
2.1.- Aspectos Básicos de Red.....	34

2.1.1.- Tipo de Redes: Red LAN y WAN	35
2.1.2.- Modelo de Red y Protocolo de Red	36
2.1.3.- Conmutación de Paquetes	38
2.1.4.- Retardos y pérdidas de paquetes en Redes de Computadoras.....	39
2.1.5.- Medición de Performance de Enlaces	42
2.2.- Conceptos matemáticos previos al ACO.....	43
2.2.1.- Introducción a la teoría de grafos	43
2.2.2.- Optimización Combinatoria	45
2.2.3.- Problema de optimización combinatoria.....	45
2.2.4.- Heurística	46
2.2.5.- Estructura de vecindad	47
2.2.6.- Óptimo local con respecto a una estructura de vecindad	47
2.2.7.- Metaheurísticas	47
2.3.- Optimización basada en Colonias de Hormigas.....	48
2.3.1.- De las colonias de hormigas a las hormigas artificiales.....	48
2.4.- La metaheurística de Optimización basada en Colonias de Hormigas	54
CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA REQUERIDO Y ALGORITMO PROPUESTO	58
3.1.- Sistema de red requerido	58
3.2.- Algoritmo propuesto.....	59
3.3.- Interfaz propuesta	62
CAPÍTULO 4: PRUEBAS, RESULTADOS Y VALIDACIÓN	68
4.1.- Prueba 1	68
4.2.- Prueba 2	74
4.3.- Prueba 3	80
4.4.- Resultados Comparativos	83
4.5.- Costo de la Implementación	85
Conclusiones y comentarios finales	86
Recomendaciones para trabajos futuros	90
Referencias bibliográficas	91
Anexos.....	94

Índice de tablas

Tabla 1.1: Principales indicadores del sector comunicaciones, 2001 -2012	11
Tabla 1.2: Estaciones de Teleservicio, por tipo, 2001 -2012	12
Tabla 1.3: OSPF vs AntNet: <i>Throughput</i>	15
Tabla 1.4: Costo bruto del proyecto	32
Tabla 1.5: Costo de inversión neto para el proyecto	33
Tabla 4.1: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 1.....	74
Tabla 4.2: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 2.....	78
Tabla 4.3: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 3.....	82
Tabla 4.4: Costos Reales	85

Índice de Figuras

Figura 1.1: Comparación de Algoritmos de enrutamiento	15
Figura 1.2: Árbol del Problema	16
Figura 1.3: Árbol de Objetivos	24
Figura 1.4: Diagrama pictórico del producto.	26
Figura 1.5: Topología lógica de la simulación	28
Figura 2.1: Convergencia de Redes	35
Figura 2.2: Redes WAN y LAN	36
Figura 2.3: Modelo OSI	37
Figura 2.4: Principio de Conmutación de Paquetes	38
Figura 2.5: Retardos y pérdidas en una transmisión IP	39
Figura 2.6: Retardo Nodal en el <i>router</i> A	40
Figura 2.7: Retardo medio de Cola VS Intensidad de Tráfico (La/R)	41
Figura 2.8: Ejemplos de un grafo simple (a), de un pseudografo (b), de un multígrafo (c) y de un grafo dirigido (d)	44
Figura 2.9: experimento real llevado a cabo por Deneubourg et al.	50
Figura 2.10: Imagen del Experimento real llevado a cabo por Goss et al.	51
Figura 2.11: Algoritmo 1	55
Figura 3.1: Sistema de red requerido	59
Figura 3.2: Diagrama de Bloques del Algoritmo propuesto	61
Figura 3.3: Diagrama de Bloques de la Interface Propuesta	62
Figura 3.4: Interfaz Gráfica del Algoritmo propuesto	63
Figura 3.5: Interface Monitoreo	64
Figura 3.6: Diagrama de Bloques del Algoritmo de Monitoreo	65
Figura 3.7: Grafica de la Topología de Red	66
Figura 3.8: Reporte Estadístico del Algoritmo Propuesto	67
Figura 4.1: Topología de Red 1	69
Figura 4.2: Tiempos desde la IP del Host de monitoreo hacia los demás nodos de red	70
Figura 4.3: IP del host de monitoreo	70
Figura 4.4: Lista de nodos vecinos de las IPs 10.11.128.190 y 10.11.128.211	70

Figura 4.5: Estadísticas de tiempos 10.11.128.190(izquierda) y 10.11.128.211(derecha)	72
Figura 4.6: Estadísticas de los <i>throughputs</i> : nodos 10.11.128.190 y 10.11.128.211	72
Figura 4.7: Estadísticas del generador de hormigas artificiales en los nodos de red 10.11.128.190 y 10.11.128.211	73
Figura 4.8: Resultados de la Prueba 2	75
Figura 4.9: Topología de Red 2	75
Figura 4.10: Tiempos desde la IP del Host de monitoreo hacia los demás nodos de red	76
Figura 4.11: Nodos vecinos para las IPs 10.11.11.80 y 10.11.11.81	76
Figura 4.12: Estadísticas de tiempos 10.11.11.80 (izquierda) y 10.11.11.81 (derecha)	77
Figura 4.13: Estadísticas de los <i>throughputs</i> de los nodos 10.11.11.80 y 10.11.11.81 .	77
Figura 4.14: Estadísticas del generador de hormigas en los nodos 10.11.11.80 y 10.11.11.81	77
Figura 4.15: Topología de Red 2 con el nodo 10.11.11.81 desconectado.....	79
Figura 4.16: Actualización de los nodos de red frente a un cambio en la topología de Red	79
Figura 4.17: <i>Throughput</i> de la IP 10.11.11.80	80
Figura 4.18: Topología de Red 3	80
Figura 4.19: Estadísticas de tiempos 10.11.11.80 (izquierda) y 10.11.11.81 (derecha)	81
Figura 4.20: Nodos vecinos para las IPs 10.11.11.80 y 10.11.11.81	81
Figura 4.21: Estadísticas de los <i>throughputs</i> de los nodos 10.11.11.80 y 10.11.11.81 .	82
Figura 4.22: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red para la topología de Red 2 y Red 3	83
Figura 4.23: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red con el algoritmo basado en ACO para la topología de Red 1, Red 2 y Red 3.....	84
Figura 4.24: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red sin el algoritmo basado en ACO para la topología de Red 1, Red 2 y Red 3.....	85

CAPÍTULO 1: ASPECTOS INTRODUCTORIOS

1.1.- Situación Problemática y Definición del Problema

1.1.1.- Situación Problemática

Con el avance tecnológico han ido apareciendo diversos equipos electrónicos que han mejorado la calidad de vida. Incluso muchos de ellos se han vuelto indispensables para las personas como por ejemplo lo son los teléfonos inteligentes y las laptops. Gracias a equipos electrónicos como estos se puede realizar llamadas, video conferencias o hasta acciones tan simples como navegar en la Internet y enviar/recibir un correo electrónico. Todas estas funcionalidades se pueden realizar gracias a que estos equipos están interconectados entre sí mediante una red de datos. Esta red está conformada por diversos equipos que reciben y envían paquetes de datos de tal forma que se pueda garantizar el envío y recepción de la información. Según el avance de los años la cantidad de usuarios de una red de datos va incrementándose vertiginosamente. Esto se puede corroborar en la **Tabla 1.1** que fue elaborado por el Instituto Nacional de Estadísticas e Informáticas (INEI) en donde se puede observar que para el año 2000 había aproximadamente 1.3 millones de líneas de servicios móviles hasta llegar en el 2012 a casi 30 millones de líneas de servicios móviles. Según este cuadro, esta tendencia de crecimiento se observa en todos los demás servicios de telecomunicaciones¹.

¹ Cfr. INEI 2016

Año	Líneas en servicio			Servicios de Telecomunicaciones			
	Telefonía fija	Telefonía pública	Servicios móviles	Radiodifusión		Radiocomunicación	
				Estación radiodifusión Sonora	Estación de radiodifusión por televisión	Estación de teleservicio privado	Radioaficionados
1997	1 537 341	40 129	435 706	1 479	496	17 940	3 285
1998	1 553 874	49 399	736 294	1 394	569	21 085	3 330
1999	1 609 884	63 508	1 045 710	1 425	596	21 314	3 371
2000	1 617 582	84 087	1 339 667	1 491	672	28 838	2 268
2001	1 570 956	95 415	1 793 284	2 065	818	17 160	2 467
2002	1 656 624	113 190	2 306 943	2 359	933	20 241	2 453
2003	1 839 165	129 366	2 930 343	2 434	936	24 763	2 456
2004	2 049 915	143 777	4 092 558	2 450	903	23 451	2 465
2005	2 250 991	151 686	5 583 356	2 492	891	22 702	2 469
2006	2 400 603	158 314	8 772 479	2 596	933	29 586	2 440
2007	2 677 847	171 083	15 417 368	2 236	1 040	31 517	2 458
2008	2 875 385	196 659	20 951 834	2 146	1 043	30 174	2 376
2009	2 965 283	192 765	24 702 060	2 371	1 084	32 050	2 255
2010	2 941 794	200 233	29 002 791	2 781	1 155	33 669	1 997
2011	2 951 144	207 651	32 305 455	3 256	1 260	35 422	1 370
2012	3 083 650	225 764	29 451 584	3 424	1 272	35 681	...

Fuente: Instituto Nacional de Estadística e Informática

Tabla 1.1: Principales indicadores del sector comunicaciones, 2001 -2012

El incremento de la cantidad de usuarios debería traer un crecimiento y mejora de la infraestructura de la red de telecomunicaciones para así poder soportar la nueva cantidad de usuarios. Esto se puede observar en la **Tabla 1.2** elaborado por INEI. En dicho cuadro es muy notorio el crecimiento de las estaciones de teleservicio entre el año 2001 y 2012. Por ejemplo, de 11 106 estaciones móviles terrestres en el año 2001 paso a 30 631 estaciones para el año 2012. Es decir, hubo un incremento de casi 276% de la cantidad de estaciones móviles en tan sólo 11 años. De igual forma en los demás tipos de estaciones se tiene un incremento, pero en menor proporción².

² Cfr. INEI 2016

Año	Total	Tipo de estación			
		Base terrestre	Fija terrestre	Móvil terrestre	Otros
2001	17 160	1 854	4 200	11 106	-
2002	20 241	2 178	4 632	13 431	-
2003	24 763	2 558	5 028	17 177	-
2004	23 451	1 876	4 690	16 885	-
2005	22 702	1 541	4 440	16 721	-
2006	29 586	2 518	4 959	22 109	-
2007	31 517	2 505	4 993	24 019	-
2008	30 174	2 159	6 155	21 860	-
2009	32 050	2 229	6 099	23 722	-
2010	33 669	2 740	4 405	24 653	1 871
2011	35 422	2 938	4 417	26 037	2 030
2012	39 062	3 268	4 547	30 631	616

Fuente: Instituto Nacional de Estadística e Informática

Tabla 1.2: Estaciones de Teleservicio, por tipo, 2001 -2012

Para que pueda haber una comunicación óptima y segura debe haber un crecimiento proporcional entre la cantidad de usuarios y la infraestructura que tiene que soportar la comunicación de los nuevos usuarios. Sin embargo, según los datos proporcionados por el INEI, el crecimiento de líneas de servicio no ha sido en la misma proporción que el crecimiento de estaciones de teleservicio. Existen diversos motivos el cual ocasiona esta falencia en la cantidad de estaciones contra la cantidad de usuarios. Por ejemplo, los problemas que existen durante el despliegue de la infraestructura de telecomunicaciones. Según el Ministerio de Transportes y Comunicaciones (MTC), los principales problemas son³:

- Demoras en tramitación de permisos municipales.
- Prohibición de instalación de postes.
- Prohibición de tendido de cableado aéreo. Barrera de acceso en canalizados.
- Informalidad (piratería) en el sector.

³ Cfr. MTC 2016

- Prohibición de instalación de antenas (temores infundados ante “daños” a la salud)

Esta desproporción del crecimiento de la cantidad de líneas en servicios de telecomunicaciones con su infraestructura trae problemas en las comunicaciones entre las líneas tales como lentitud, cobertura, interrupciones y/o corte, entre otros. Agregado a ello, es común que en ocasiones una de las estaciones se averíe incrementando los problemas de comunicaciones ya existentes.

Para que las redes de telecomunicaciones estén interconectadas entre ellas, indistintamente la tecnología o medio de transmisión que se use, requieren de un algoritmo de enrutamiento. Gracias al enrutamiento existente entre las redes de datos se puede generar diversas estrategias de encaminamiento de información. De esta forma lograr obtener las rutas óptimas de transmisión de datos desde un nodo de red origen hacia un nodo de red destino⁴. Sin embargo, Arturo Imas Rodríguez en su tesis de maestría en Inteligencia Artificial en la Universidad Politécnica de Madrid menciona que:

« Los algoritmos clásicos propuestos hasta la fecha suelen ser algoritmos centralizados que tratan de gestionar una arquitectura claramente distribuida, que en escenarios estacionarios pueden mantener un buen rendimiento, pero que **no funcionan bien en escenarios donde se dan continuos cambios en la topología de red o en los patrones de tráfico** »

(Imas 2013).

Según lo mencionado en párrafos anteriores, la infraestructura de telecomunicaciones en Perú es muy dinámica ya que presentan muchos cambios en su topología de red. Esto se debe a que conforme avanzan los años la infraestructura de telecomunicaciones va creciendo. Sin embargo, este crecimiento no lo logra hacer en la misma proporción al crecimiento de usuarios de dicha infraestructura. Ese dinamismo se debe tomar en cuenta en los algoritmos de enrutamiento, ya que un algoritmo de enrutamiento es óptimo cuando este es más adaptable a los cambios en una topología de red y a los patrones de tráfico que existe en un determinado momento (datos transmitidos). Se dice que cuando un algoritmo tiene una menor

⁴ Cfr. Imas 2013: 1-2

cantidad de secuencias de interacciones para obtener la ruta más corta tendrá una mayor capacidad de convergencia.

Por ello, la velocidad de propagación de la información de enrutamiento y el cálculo de los caminos óptimos son considerados las propiedades de convergencia más importantes en un protocolo de enrutamiento. Para medir el rendimiento de este tipo de algoritmos las medidas habituales que se usan son el *throughput* (volumen de información neto que fluye a través de un sistema como, por ejemplo, una red de computadoras) y el tiempo de entrega entre paquetes de datos (*end-to-end delay*)⁵.

Los protocolos de enrutamiento clásicos que tienen mayor velocidad de convergencia son el protocolo OSPF (*Open Shortest Path First*) y EIGRP (*Interior Gateway Routing Protocol*)⁶. Sin embargo, estos protocolos tienen algoritmos que no presentan el comportamiento deseado en escenarios con muchos cambios de red y alta transmisión de información⁷. Vicent Verstraete realizó estudios del comportamiento del protocolo OSPF generando transmisiones de datos superiores a las soportadas por el enlace entre dispositivos de prueba.

De esta forma, logra obtener sus valores de *throughput* en este tipo de ambientes demostrando que el protocolo OSPF pierde el 49.9% de los paquetes enviados (*packet loss*) en una carga de transmisión del doble de la soportada por el enlace (ver **Tabla 1.3**)⁸.

Por su parte, Gianni Di Caro y Marco Dorigo también realizaron estudios del comportamiento del protocolo OSPF. Para la cual hicieron simulaciones en donde disminuyen los tiempos de interarribo (entre 2.4 y 2) para obtener los valores de información del volumen que fluye en el sistema (*throughput*) y el retardo que presentan. Esto se refleja en la **Figura 1.1** donde se puede observar que el protocolo OSPF tiene hasta 2.5 segundos de retardo con un interarribo de 2⁹.

⁵ Cfr. Imas 2013: 9 - 34

⁶ Cfr. Cisco 2012

⁷ Cfr. Imas 2013: 1-2

⁸ Cfr. Verstraete y otros 2006: 3

⁹ Cfr. Di Caro y Dorigo 1998: 342-348

Load	Packet loss OSPF	Packet loss AntNet
110 %	8.95 %	2.19 %
150 %	33.2 %	12.86 %
200 %	49.9 %	26.65 %

Fuente: Verstraete y otros (2006)

Tabla 1.3: OSPF vs AntNet: *Throughput*

Fuente: Di Caro y Dorigo (1998)

Figura 1.1: Comparación de Algoritmos de enrutamiento

En este proyecto de tesis se propone un software con un algoritmo robusto y estable que sea más adaptable a los cambios. Este algoritmo está basado en algoritmos de optimización por colonia de hormigas (ACO, del inglés, *Ant Colony Optimization*) el cual toma como base el modelamiento del comportamiento utilizado por las hormigas para la resolución del problema de obtención del camino mínimo entre su colonia y su fuente de alimentación.

Por ello, el proyecto tiene como objetivo el uso de “hormigas artificiales” que simulen dicho comportamiento para buscar la ruta más rápida para poder transmitir información. Dicha ruta se actualiza en el tiempo según la cantidad de servidores y usuarios tenga dicha red. Con eso se busca cubrir en parte los problemas ocasionados por la desproporción entre la infraestructura de la red de telecomunicaciones, la cantidad de usuarios y por ende la alta carga de información

trasmitida. Además, de mejorar la eficiencia de la obtención de rutas mínimas cuando hay cambios en la topología de red de datos como incremento de servidores o caídas de dichos servidores.

1.1.2.- Definición del Problema

1.1.2.1.- Árbol del Problema

El problema principal junto a sus causas primarias, causas secundarias y consecuencias secundarias y consecuencia principal se puede observar en árbol del problema en el **Figura 1.2**.

Fuente: Elaboración propia

Figura 1.2: Árbol del Problema

1.1.2.2.- Problema General

Limitada gestión de encaminamiento de la información en redes de computadoras.

1.1.2.3.- Problema Ingenieril

¿Qué consideraciones tecnológicas e ingenieriles de diseño de telecomunicaciones y software se tienen que tener en cuenta para llevar a cabo la implementación de una aplicación de software basada en la optimización por colonia de hormigas con el objetivo de mejorar el encaminamiento de datos en redes de computadoras para satisfacer los requerimientos y optimizar la reconstrucción de su ruta de transmisión de información y restablecimiento del servicio?

1.2.- Estado del Arte

A continuación, se presentan los estudios anteriormente publicados y los productos existentes que se ofrecen en el mercado, los cuales se pueden considerar como una alternativa de solución a la situación problemática planteada.

1.2.1.- Productos y soluciones existentes.

Existen diversas soluciones planteadas basadas en el Algoritmo de Optimización por Colonias de Hormigas. Estas están siendo tomadas en cuenta como ayuda para solucionar el problema planteado en este proyecto de tesis, ya que la solución planteada también se basa en el algoritmo mencionado. Estos proyectos son:

1.2.1.1.- AntNet: ACO routing algorithm in practice¹⁰

AntNet es un protocolo de enrutamiento adaptativo basado en el algoritmo de optimización por colonia de hormigas creado por M. Domingo y Di Caro. AntNet fue creado como alternativa al ya conocido protocolo de enrutamiento OSPF (*Open Shortest Path First*). A diferencia del OSPF que busca el camino más corto mediante encaminamientos jerárquicos, AntNet también busca el encaminamiento más corto a base de agentes móviles usando estigmergia (colaboración entre distintos agentes

¹⁰ Cfr. Verstraete

independientes, en una forma de organización donde la colaboración juega un rol clave).

Ventajas:

- En comparación con el protocolo OSPF, AntNet pierde menos paquetes en ambientes con alta carga de transmisión. Por ejemplo, con una carga de transmisión del 200% OSPF logra perder el 49.9% de los paquetes y AntNet el 26.65% (ver **Tabla 1.3**).
- AntNet distribuye la transmisión de datos por toda la red. Siendo esto una manera más eficiente en transmisiones de gran cantidad de datos

Desventajas:

- Este algoritmo fue solamente simulado y no implementado en una red física.
- AntNet fue simulado en una red pequeña con solo cinco routers y dos hosts
- Con respecto a la performance en los enlaces de falla, el protocolo OSPF es más robusto que AntNet

1.2.1.2.- Assigning cells to switches in mobile networks using an ant colony optimization heuristic¹¹

En comunicaciones móviles, una determinada área geográfica es cubierta por una celda y a esta se le es asignada una estación base de telefonía móvil (BTS, del inglés, *Base Transceiver Station*). Sin embargo, es muy común que cuando una persona haga una llamada cuando se está trasladando de un punto a otro; es decir, cabe la posibilidad que se traslade de un área geográfica de una celda hacia el área geográfica de otra celda. Para que este tipo de comunicación no se corte se debe hacer una conmutación de celdas, pero con diferentes frecuencias de transmisión, ya que no se puede reutilizar la misma frecuencia en celdas adyacentes. Este producto busca optimizar dicha comunicación móvil y para ello usa un algoritmo

¹¹ Cfr. Fournier 2013

basado en algoritmos metaheurísticos como los algoritmos de optimización por colonias de hormigas (ACO) y algoritmos de optimización k-opt.

Ventajas:

- En comparación con estudios anteriores, los métodos utilizados en este proyecto son más eficientes en cuestión de calidad y tiempo de ejecución de los algoritmos
- Según los autores de este algoritmo, es posible aumentar el tamaño de los casos de problemas sin aumentar el tiempo de ejecución con soluciones de calidad equivalente.

Desventajas:

- El algoritmo carece de cierta robustez con respecto a los valores de algunos parámetros, ya que ciertos valores aumentan significativamente en el tiempo sin necesariamente mejorar los resultados.
- Es aproximadamente 1% más costoso que las soluciones planteadas en estudios anteriores

1.2.1.3.- Una versión de ACO para Problemas con Grafos de muy Gran Extensión¹²

Los Algoritmos basados en las colonias de hormigas (ACOs) presentan limitaciones como tener la cantidad de nodos o grafos conocidos y esta cantidad debe ser un número pequeño para así poder almacenar en memoria los datos de cada grafo. Otra limitación de los ACOs es de tener un número de grafos fijo o en su defecto esta cantidad debe encontrarse acotado por un valor conocido. Este proyecto plantea una nueva versión llamada ACOhg el cual no posee las limitaciones mencionadas, ya que se puede aplicar para problemas con grafos de tamaño desconocido y/o demasiados grandes como para ser almacenados en memoria.

Ventajas:

- En los resultados de las pruebas de ACOhg muestran que la tasa de éxito aumenta con la longitud máxima y longitudes medias.

¹² Cfr. Alba y Chicaco 2013

- Este algoritmo disminuye la tasa de error y tiempo para longitudes alta o medias.

Desventajas:

- El modelo ACOhg aún se encuentra en estudio; por ello, no se encuentra claramente la desventaja de este algoritmo.

1.2.1.4.- Algoritmos bioinspirados para el encaminamiento de datos en redes inalámbricas de sensores¹³

Las redes inalámbricas de sensores (RIS) presentan limitantes las cuales les impide el uso de protocolos de redes tradicionales. Por ello en las redes inalámbricas de sensores se debe adaptar sus protocolos al uso específico que se debe utilizar. Por ello, este proyecto de tesis tiene como objetivo diseñar un algoritmo de encaminamiento de redes basado en el algoritmo optimización de colonia de hormigas. Para eso el autor propone un algoritmo que toma las partes principales de los algoritmos ACLR (Algoritmo de encaminamiento consciente de la ubicación basado en ACO) y EEABR (Algoritmo de encaminamiento basado en hormigas con eficiencia de energía). A dicho algoritmo lo nombra IACAR (Algoritmo de encaminamiento propuesto para mejorar el tiempo de vida de la RIS). IACAR para poder obtener la ruta más corta utiliza la probabilidad de salto al siguiente nodo de ACLR y toma el método de control de movimiento de hormigas a través de los nodos de la red.

Ventajas:

- El algoritmo propuesto, IACAR, logra optimizar el consumo de energía.
- IACAR logra encontrar la ruta más corta que ACLR y EEABR.

Desventajas:

- ACLR es más efectivo para obtener la ruta más corta.
- IACAR tiene ligeramente mayor latencia en transmisión de información por a RIS que ACLR.

¹³ Cfr. Dominguez 2011

1.2.2.- Publicaciones Científicas/Ingenieriles

Existen diversos estudios que han enfocado en el algoritmo de optimización de colonias de hormigas. Siendo los estudios que se basó este proyecto los siguientes:

- **Enrique Alba y Francisco Chicano en la publicación en un proyecto parcialmente financiado por el Ministerio de Educación y Ciencia y FEDER de España con título “Una nueva Versión de ACO para Problemas con Grafos de muy Gran Extensión”** mencionan que ACOs se han aplicado con éxito a problemas de optimización combinatoria que pueden transformarse en una búsqueda dentro de un grafo. También, en sus resultados finales muestra que la memoria y tiempo de latencia disminuyen al permitir que el algoritmo construya caminos más largos y la tasa de éxito aumenta con la longitud máxima. Adicionalmente, los autores comparan su nueva versión de ACO con técnicas clásicas como búsqueda en profundidad (DFS) y búsqueda en amplitud (BFS) consiguiendo mejores resultados en latencia y tiempo.
- **Daniel Aparicio Guirao en su tesis de final de la carrera Ingeniería Industrial de la Universidad Politécnica de Cataluña con título: “Aplicación de los algoritmos de hormigas para la resolución de un problema de equilibrado de líneas de montaje con robotizados”** diseña un algoritmo por ACO llamado RALBP-MMAS para solucionar el problema de equilibrado de líneas de montaje de diseño productivos que consisten en la asignación de las tareas para el montaje del producto a las estaciones de trabajo de una línea de producción. Este software lo compara con un software comercial obteniendo mejores resultados.
- **Arturo Imas Rodriguez en su Tesis de Master en Inteligencia Artificial de la Universidad Politécnica de Madrid con título “Algoritmos Inspirados en *Swarm Intelligence* para el Enrutamiento en Redes de Telecomunicaciones”** dice « El enrutamiento determina las estrategias a utilizar por los nodos de una red para encontrar las rutas óptimas entre un origen y un destino en el envío de información (...) Los algoritmos clásicos propuestos hasta la fecha suelen ser algoritmos centralizados que tratan de gestionar una arquitectura claramente distribuida, que en escenarios estacionarios pueden mantener un buen rendimiento, pero que no funcionan bien en escenarios donde se dan continuos cambios en la topología de red

o en los patrones de tráfico ». Por ello, él propone un algoritmo de optimización de colonias hormigas llamado Abira el cual muestra un resultado eficiente.

- **Pedro Isasi Viñuela y David Quintana Montero en su proyecto de fin de carrera de Ingeniería Informática de la Escuela Politécnica Superior** realiza un estudio del algoritmo de optimización de colonias de hormigas para poder obtener la distancia mínima entre puntos dentro de una red. Para ello se basan en estudios realizados a problemas combinatoriales en la búsqueda del camino más corto entre dos puntos. Ellos logran comprobar la viabilidad del uso de algoritmo en cuestión para poder obtener resultados óptimos y eficientes.
- **Matthew Guidry en su publicación “*Artificial Intelligence in Networking: Ant Colony Optimization*”** menciona que el Algoritmo por optimización de colonias hormigas está capacitado de forma eficiente para hacer frente cuando un *router* es apagado, ya que este algoritmo presenta una gran capacidad de adaptación; en otras palabras, el ACO presenta técnicas eficientes para resolver problemas cuando el camino de la “hormiga” ha sido bloqueada por alguna razón. También, menciona que una ventaja muy importante del ACO es que cada “hormiga” tiene un tamaño muy pequeño y gracias a ello se logra agregar una carga mínima adicional a la información transmitida. Sin embargo, Matthew también menciona que es muy costoso actualizar los *routers* para que se agreguen nuevas técnicas de enrutamiento (entre ellas el ACO), ya que estas serían muy costosas.

1.3.- Justificación

El proyecto que se implementará tiene como fin mejorar la gestión de transmisión de información entre redes de computadoras. Las limitadas opciones para la gestión en la transmisión de información producen la necesidad de implementar un aplicativo en capa de usuario a fin de satisfacer esta necesidad. Por otro lado, la gran mayoría de aplicativos de software propietario para gestionar la transmisión de información no brindan una personalización detallada en las redes de

computadoras del usuario, sin embargo, el aplicativo a implementar en este trabajo se basa en los mínimos detalles de la red en donde este aplicativo se ejecutará.

1.3.1.- Justificación Ingenieril:

El proyecto se justifica a nivel ingenieril debido a que implica implementar y resolver diferentes problemas y situaciones aplicando fundamentos y técnicas de ingeniería como, por ejemplo:

- Diseño de algoritmo pseudocódigo del ACO.
- Implementación del aplicativo ACO en Visual Studio C#.
- Configuración de Sistemas Operativos CentOS.
- Diseño de red de computadoras para la simulación del ACO.

1.3.2.- Justificación Económica:

El proyecto se justifica a nivel económico debido a que reducirá costos al cliente ya que la instalación de los aplicativos para la gestión de la transmisión de información no tendrá un costo adicional. Asimismo, actualmente el software propietario, Microsoft Visual Studio se comercializa de manera gratuita para pequeñas empresas o de uso particular, por lo que cualquier modificación adicional en el código implementado no aumentará los costos de operación. Por otro lado, el sistema operativo en el cual el aplicativo funciona es Windows sin embargo las modificaciones se realizan en el software libre, CentOS como también los servicios de comunicación que se utilizarán para sustentar el proyecto.

1.3.3.- Justificación Social:

El proyecto se justifica a nivel social debido a que en poblaciones rurales y/o alejadas de las grandes urbes se puede implementar el ACO sin utilizar muchos

recursos físicos con la finalidad de obtener una red óptima y eficiente sin la necesidad de gastar en una gran cantidad de hardware.

1.4.- Objetivos

1.4.1.- Árbol de objetivos

El objetivo general junto a los logros secundarios, logros principales, objetivo específico y objetivo secundario se pueden observar en el árbol de objetivos en la **Figura 1.3.**

Fuente: Elaboración propia

Figura 1.3: Árbol de Objetivos

1.4.2.- Objetivo General

Diseñar e implementar un algoritmo de software basada en la optimización por colonia de hormigas para mejorar la gestión del encaminamiento de información en redes de computadoras.

1.4.3.- Objetivos Específicos de implementación

- Implementar una red en la cual se pueda simular y validar el uso del algoritmo de optimización por colonias de hormigas
- Implementar un algoritmo dinámico frente a cambios constantes de la topología de red.
- Medir en cada nodo de red la cantidad de datos transmitidos correctamente en una unidad de tiempo (*throughput*).
- Proporcionar un marco conceptual comprensivo para la implementación del método de optimización por colonia de hormigas.
- Medir el desempeño del aplicativo del encaminamiento de datos en redes de computadoras con el cálculo de porcentaje de mejora.

1.5.- Breve Descripción de la Solución Propuesta

1.5.1.- Diagrama de bloques pictórico

A continuación, se muestra en la **Figura 1.4** el diagrama del proyecto mostrando las fases que son:

1. Comienza transmisión de datos.
2. Agente Monitor inicia algoritmo ACO, reconoce los Agentes Servidores (líneas punteadas rojas).
3. Agente Monitor selecciona ruta óptima inicial para la transmisión, la cual es enviada a los Agentes Servidores e ingresada al sistema de transmisión. (línea punteada verde).
4. Agentes Servidores inician la transmisión de datos.

5. Se corta la transmisión de datos en forma inesperada, Agente Monitor inicia algoritmo ACO y genera una nueva ruta.
6. Agente Monitor envía la nueva ruta a los Agentes Servidores, la cual es actualizada en el sistema de transmisión.
7. Se restablece la transmisión de datos.

Fuente: Elaboración propia

Figura 1.4: Diagrama pictórico del producto.

1.5.2.- Funcionamiento

La solución final consiste en la implementación de una aplicación de software, la cual cuando el usuario empiece una transmisión de datos, el agente monitor inicie el algoritmo de colonia de hormigas (ACO) basado en los parámetros de transmisión como pesos, seleccione la ruta óptima inicial para la transmisión de datos, la que es enviada a los agentes servidores para dar inicio a la conexión entre servidores y la transmisión datos. Asimismo, este software aplicativo también permite identificar la caída de un servidor, en este caso el agente monitor procede a inicializar el algoritmo de hormigas (ACO) y genere una nueva ruta óptima que será enviada a los agentes servidores para actualizar los servidores y continuar con la transmisión de

datos de modo transparente. Las tareas de la aplicación se encuentran ligadas a la parte de núcleo y transporte del sistema de comunicación de información que se desee optimizar.

Esta aplicación ha sido elaborada en dos módulos, el **primer módulo** consiste en el monitoreo de envío de mensajes de alerta del estado del enlace por parte del agente servidor hacia el agente monitor y el **segundo módulo**, consiste en la selección de la ruta optima de transmisión inicial, las actualizaciones de la ruta ante una pérdida de conexión y las modificaciones de los archivos de configuración en el servidor CentOS por el cual se lleva a cabo la transmisión de datos.

En el primer módulo, el software aplicativo nos indicará si un servidor se encuentra fuera de servicio o no disponible, es decir, el agente servidor enviará la información al agente monitor sobre el estado del enlace entre ambos nodos. El agente monitor al recibir el aviso de que un servidor se encuentra fuera de servicio, inicia el algoritmo de hormigas (ACO) con el fin de generar una nueva ruta para la transmisión de información que será enviada al agente servidor de donde se inició la transmisión de datos. El agente servidor se encargará de informar al agente monitor el estado del servidor del cual proviene. Asimismo, se encargará de actualizar el nuevo camino de transmisión entre los servidores CentOS y seguir con la transmisión de información. De acuerdo a las premisas del desarrollo del software aplicativo, el primer módulo del software aplicativo se realizará con el lenguaje de programación C#, es decir, el agente monitor será programado utilizando el software Microsoft Visual Studio 2013 y los agentes servidores serán programados en consola en el sistema operativo CentOS en cada uno de los servidores CentOS, la comunicación entre estos dos aplicativos será vía *sockets*.

En el segundo módulo, el software aplicativo, en este caso el agente monitor procederá a iniciar el algoritmo de hormigas (ACO) y generará la nueva ruta óptima para la transmisión de datos, esta nueva ruta será enviada al agente servidor por el cual se inició la transmisión de datos a fin de actualizar la nueva ruta de transmisión.

El monitoreo es realizado vía *sockets* UDP ya que estos no son orientados a conexión. Solo se abre un *socket* y se empieza a escribir mensajes en él o leer, sin necesidad de esperar que otro servidor se conecte en el otro extremo del *socket*. El monitoreo es

llevado a cabo cada segundo desde que se inicia el agente monitor previa inicialización de los agentes servidores.

Si en algún momento el agente monitor deja de recibir mensajes por algún agente servidor dispara el aviso de nodo fuera de servicio o no disponibilidad de conexión del nodo al algoritmo ACO. El objetivo del agente monitor es comprobar la conectividad entres servidores y su disponibilidad. Asimismo, el agente monitor debe tener en su base de datos qué servidores se encuentra en servicio y disponibles en su topología física y según esta, el algoritmo de hormigas (ACO) seleccionara la ruta óptima para la transmisión de información.

Fuente: Elaboración propia

Figura 1.5: Topología lógica de la simulación

En la **Figura 1.5** la línea de color verde indica el camino predeterminado para la transmisión de datos entre el Agente Servidor A y el Agente Servidor C, la línea de color amarillo indica el camino alternativo para la transmisión de datos de los agentes servidores mencionados entregado por el agente monitor y las líneas de color rojo representan el monitoreo del Agente Monitor.

1.5.3.- Limitaciones de la solución

El aplicativo que se implementará presenta las siguientes limitaciones:

- El aplicativo elaborado en software propietario se ejecuta solo en sistemas operativos de tipo Windows.
- El aplicativo no robustece el enlace de transmisión, ni lo hace invulnerable a las caídas de servicio, solo mejora la gestión del encaminamiento de información.
- El aplicativo solo reconoce transmisión de datos en redes LAN.
- Se enviará un archivo para validar el caso específico cuando se utiliza un medio de comunicación que utiliza encaminamiento de datos se encuentre no disponible o fuera de servicio, no se tiene a disposición un mecanismo automático o aplicación de software en capa de aplicación el cual reconstruya la ruta de encaminamiento y restablezca el servicio.
- Para la sustentación del proyecto el aplicativo enviará un archivo entre los Servidor A y Servidor H.
- Se utilizará máquinas virtuales de CentOS como servidores, a fin de enviar los archivos vía consola y calcular el tiempo de envío entre servidores.

1.5.4.- Resultados esperados

Se espera que la solución cumpla con los siguientes puntos para satisfacer la necesidad planteada antes mencionada:

- Desarrollar un pseudocódigo para el encaminamiento de datos en redes de computadoras como guía para futuras aplicaciones de los alumnos de la facultad de electrónica de la Universidad Peruana de Ciencias Aplicadas.
- Presentar un cuadro comparativo estadístico de la tasa de transferencia inicial y la tasa de transferencia al utilizar el ACO. Así como también, un cuadro de los parámetros usados en el aplicativo, la variación del tiempo de respuesta en el algoritmo de

hormigas y las otras etapas que conforman el aplicativo de encaminamiento de información.

- Obtener una reducción en la pérdida de paquetes en la transmisión de información y una mejora en el encaminamiento de información.
- Obtener el índice de mejora de transmisión de información que es el resultado de la fracción de la diferencia de tiempos sin ACO y con ACO entre el tiempo sin ACO.
- Obtener los parámetros para la configuración de la red y los sistemas operativos donde se simulará y validará el algoritmo de optimización.
- Presentar la documentación del marco conceptual para la implementación de un algoritmo basado en colonia de hormigas para mejorar la gestión del encaminamiento de información en una red de computadoras.

1.6.- Aplicaciones y usuarios potenciales del producto

La aplicación de software basada en el algoritmo de optimización mediante colonia de hormigas tiene como aplicación directa la optimización del tráfico de ruteadores en la capa de aplicación de una red de telefonía VoIP. El algoritmo propuesto tiene como aplicaciones indirectas el control vehicular de semáforos y optimizar la ruta de viajes en el transporte de material en una empresa.

Asimismo, Rita Peñabaena, presenta un diseño y optimización de un modelo matemático para la transición entre el cambio de planes de los tiempos de reparto de los semáforos, el cual propone una reducción en el coste social que incluye reducir la demora, las emisiones de gases y el consumo de combustible. La gran parte de las metrópolis no cuenta con un plan de semaforización ni de una propuesta de transición, lo cual hace que este problema perdure a lo largo del tiempo. Este diseño beneficiaría a la población que depende de un vehículo para movilizarse.¹⁴

Por otro lado, Juan Arias, propone una aplicación de un modelo de optimización en la planeación de rutas de los buses escolares del colegio Liceo de Cervantes Norte ubicado en Bogotá, Colombia y la cantidad de usuarios potenciales abarca a 466

¹⁴ Cfr. Peñabaena 2015:

alumnos. Un proyecto de similares características aplicado a nuestros actuales sistemas de transporte como el metropolitano, línea amarilla y el corredor de Javier Prado podría beneficiar a miles de personas que viven en Lima.¹⁵

1.7.- Viabilidad

1.7.1.- Viabilidad técnica

Este proyecto de tesis tiene viabilidad técnica debido a que:

- El proyecto es viable en cualquier tipo de red LAN y se puede instalar un servidor en la misma red sin ninguna configuración adicional.
- El equipamiento para el desarrollo del proyecto se encuentra disponible en cualquier proveedor de equipos de computación.
- Se tiene conocimientos de programación en Microsoft Visual C#, sistema operativo CentOS como servidores y optimización por algoritmo de hormigas.
- Se dispone de un laboratorio para llevar a cabo una simulación de una red con varios servidores.

1.7.2.- Viabilidad económica

El proyecto propuesto tiene viabilidad económica debido a que el costo de inversión neto es mínimo ya que los materiales disponibles propios como los dispuestos por la Universidad Peruana de Ciencias Aplicadas no tienen costo y los materiales que se compraran no superan los S/. 50 soles.

A continuación, se presenta la **Tabla 1.4** en donde se detalla los costos de los artículos necesarios para la implementación del proyecto propuesto.

¹⁵ Cfr. Aria 2010:

Costo Bruto				
Item	Imagen	Cant.	Costo Unitario	Costo Total
Computadora Dell Precision T1700		8	4099.00	32,796.00
Patch Panel AMP Category 6		1	150.00	150.00
HP Envy 15-q002la		1	3799.00	3799.00
Toshiba Satellite C45-ASP4310FL		1	1999.00	1999.00
CentOS v7.0		1	0.00	0.00
Visual Studio 2015 - Estudiantil		1	0.00	0.00
Cableado estructurado para PCs		8	6.25	50.00
Switch D-Link 8 puertos		1	60.00	60.00
			Total	38,854.00

Fuente: Elaboración Propia

Tabla 1.4: Costo bruto del proyecto

En la **Tabla 1.4** se muestra el costo de inversión neto del proyecto propuesto en donde se detalla los materiales disponibles dispuestos por la Universidad Peruana de Ciencias Aplicadas y los materiales propios de los alumnos.

Item	Cant.	Costo	Costo Total	Dispuesto UPC	Dispuesto Alumno	Inversion
Computadora Dell Precision T1700	8	4,099.00	32,796.00	-32,796.00		0.00
Patch Panel AMP Category 6	1	150.00	150.00	-150.00		0.00
HP Envy 15-q002la	1	3,799.00	3,799.00		-3,799.00	0.00
Toshiba Satellite C45-ASP4310FL	1	1,999.00	1,999.00		-1,999.00	0.00
CentOS v7.0	1	0.00	0.00			0.00
Visual Studio 2015 - Estudiantil	1	0.00	0.00			0.00
Cableado estructurado para PCs	8	6.25	50.00			50.00
Switch D-Link 8 puertos	1	60.00	60.00	-60.00		0.00
					Total	S/.50.00

Fuente: Elaboración Propia

Tabla 1.5: Costo de inversión neto para el proyecto

1.7.3.- Viabilidad operativa

Este proyecto tiene viabilidad operativa porque se puede operar sin problemas en una red sin requerimientos adicionales en hardware. Asimismo, solo se necesita ejecutarse el aplicativo en una red LAN con ordenadores que utilicen Windows. Los servidores se ejecutan en máquinas virtuales por lo que no se requiere alguna configuración adicional en los ordenadores.

1.7.4.- Alternativas

El diseño e implementación del algoritmo de hormigas para el encaminamiento de datos en una red de computadoras tiene como objetivo reducir el tiempo entre la comunicación de un servidor en el caso de que uno de ellos falle y/o se encuentre saturado. Para poder medir la efectividad de este problema se compara los tiempos con los protocolos de conexión actuales versus con la propuesta que planteamos.

CAPÍTULO 2: MARCO TEÓRICO

En este capítulo se presentan los principales fundamentos teóricos para comprender el diseño e implementación del algoritmo propuesto. El desarrollo se centra en fundamentos teóricos de redes IP y encaminamiento de datos. Por ello, se iniciará describiendo conceptos elementales sobre teoría de redes en las cuales se mencionará los protocolos, topologías y los parámetros de medición de calidad de servicio. Después, se hará una descripción de algoritmos en los cuales se basa el ACO como también la teoría de grafos (algoritmo heurístico) y algoritmos metaheurísticos para finalmente describir detalladamente el fundamento teórico del Algoritmo de Optimización por Colonias de Hormiga.

2.1.- Aspectos Básicos de Red

Todas las personas buscan enviar y recibir mensajes de voz, texto y/o video a través de diversas aplicaciones ya sea por computadoras, teléfonos inteligentes u otros dispositivos (hosts). Para esto, los hosts utilizados tienen aplicaciones las cuales están conectados a una misma red convergente que está gobernada por reglas y protocolos que se utilizan para comunicarse entre sí (En la **Figura 2.1** se puede observar como una red móvil, doméstica, empresarial, entre otras redes convergen). Para ello, estos protocolos han tenido que ser aprobados por la industria de *Networking* y ratificados por una organización de estándares, ya que de esta forma se puede asegurar que los productos de diferentes fabricantes puedan funcionar conjuntamente y así lograr una comunicación eficiente. Actualmente, el estándar utilizado es un conjunto de protocolos denominado TCP/IP (Protocolo de control de transmisión/Protocolo de Internet) siendo este el protocolo primario en la Internet el cual garantiza que los mensajes sean entregados al destinatario final¹⁶.

¹⁶ Cfr. Cisco 2012

Fuente: Kurose y Ross (2010)

Figura 2.1: Convergencia de Redes¹⁷

2.1.1.- Tipo de Redes: Red LAN y WAN

Las redes por la cual se envía la información se dividen según su tamaño en redes LAN (Red de Área Local de las siglas en inglés *Local Area Network*) y WAN (Red de Área Ampla de sus siglas en ingles *Wide Area Network*). Las redes LAN por lo general es administrada por una organización única o una red local la cual se conectan varios equipos con un alcance limitado por una conexión por cables o inalámbrica. En este tipo de redes se tiene por ejemplo una red de un edificio, campus o la red que abastece un hogar. Las redes WAN son redes que conectan las redes locales en ubicaciones geográficas separadas; en otras palabras, una red WAN es una red de computadoras que une varias redes LAN a pesar de que estas no se encuentren en una misma ubicación física¹⁸. En la **Figura 2.2** se muestra una imagen

¹⁷ Cfr. Kurose y Ross 2010: 10-11

¹⁸ Cfr. Cisco 2012

donde se puede observar que la conexión de varias redes de un hogar (redes LAN) llega a formar una red WAN.

Fuente: Cámaras de Seguridad Panamá 2016

Figura 2.2: Redes WAN y LAN¹⁹

2.1.2.- Modelo de Red y Protocolo de Red

2.1.2.1.- Modelo OSI

El modelo OSI, de las siglas en inglés *Open Systems Interconnect*, fue desarrollado por la Organización Internacional de Estándares (ISO) en 1983 y está documentada en el estándar 7498. Básicamente, este modelo define métodos y protocolos que se utilizan para conectar diversos *hosts* dentro de una red de datos. El modelo OSI se divide en 7 capas en las cual cada capa superior depende de los servicios que ofrece la capa de nivel inferior. En la **Figura 2.3** se puede observar las capas que presenta este modelo siendo la primera la Capa Física, seguida consecutivamente por las Capas de Enlace de Datos, Red, Transporte, Sesión, Presentación y, finalmente, la Capa de Aplicación.²⁰

¹⁹ Cfr. Cámaras de Seguridad Panamá 2016

²⁰ Cfr. Halberg 2010: 28 -30

Fuente: Halberg (2010)

Figura 2.3: Modelo OSI

2.1.2.2.- Protocolo de Red

Gracias a la estructura de capas puede analizar una parte específica y bien definida de un sistema grande y complejo. Es por eso que los diseñadores de redes para proporcionar una estructura al diseño del protocolo de red lo organizan en capas²¹. Los protocolos de enrutamiento se encuentran dentro de la capa de red y los más clásicos son los protocolos OSPF y SPF

- **OSPF:** Viene de las siglas en inglés *Open Shortest Path First*. OSPF es un protocolo de enrutamiento de estado de enlace sin clase que utiliza el concepto de áreas para realizar la escalabilidad. Define las métricas como un valor arbitrario llamado costo, siendo el de menor costo la de distancia mínima. También, hay que tener en cuenta que este protocolo es de convergencia rápida.²²
- **SPF:** Viene de las siglas *Shortest Path First*. SPF es un algoritmo adaptativo que utiliza métricas dinámicas para evaluar los costos de cada enlace. Estos costos se

²¹ Cfr. Kurose y Ross 2010: 46-50

²² Cfr. Cisco 2012

calculan mediante el promedio entre los menores y mayores valores reales que reflejen la demora en intervalos fijos.²³

2.1.3.- Conmutación de Paquetes

Existen dos métodos o modelos fundamentales que permiten transportar datos a través de una red. Estos métodos son conmutación de circuitos y conmutación de paquetes. Siendo el de mayor rendimiento el modelo de conmutación de paquetes²⁴. Por eso, este proyecto se basa en dicho modelo. La tecnología de conmutación de paquetes consiste en que un mensaje es dividido en bloques de mensajes más pequeños (paquetes). Cada paquete contiene una cabecera con información de control y un área de datos (*payload*) la cual tiene parte de la información del mensaje y una cola (*trailer*) que incluye algún mecanismo de control de errores. La información de control contiene básicamente información de direccionamiento, origen y destino final (también puede tener información de prioridad). Con esta información se puede enviar estos bloques de mensajes (paquetes) por diversas rutas de una red y rearmar el mensaje original en el destino final²⁵. La división del mensaje original en paquetes y transmisión de estos en una red se puede observar en la **Figura 2.4** que se muestra a continuación.

Fuente: Facultad de Ciencias Exactas y Naturales y Agrimensura de la Universidad Nacional del Nordeste - UNNE (2016)

Figura 2.4: Principio de Conmutación de Paquetes²⁶

²³ Cfr. Di Caro y Dorigo 1988: 334 -335

²⁴ Cfr. Kurose y Ross 2010: 23 - 31

²⁵ Cfr. Imas 2013: 5-8

²⁶ Cfr. UNNE 2016

2.1.4.- Retardos y pérdidas de paquetes en Redes de Computadoras

El caso ideal sería que mediante la Internet se pueda transferir los datos que quisiéramos en la velocidad que uno desee. Sin embargo, esto es imposible, ya que existen factores físicos que limitan la transmisión de datos y a su vez introduce retardos y pérdidas de paquetes de datos. Es por eso, que para medir la eficiencia de una red de computadoras es necesario cuantificar estos valores. Por ello, en esta sección se tratará los retardos y pérdidas de paquetes que se genera en una conmutación de paquetes y en un encolamiento. En la **Figura 2.5** se muestra referencialmente las perdidas y retardos en una transmisión de datos en una red IP.

Fuente: Kurose y Ross (2010)

Figura 2.5: Retardos y pérdidas en una transmisión IP

2.1.4.1.- Retardos en Redes de conmutación de paquetes

Una comunicación en una red por computadoras es básicamente una transferencia de datos entre un host origen y un host destino. Los datos transmitidos sufren diversos retardos en los nodos que se encuentran en la ruta de encaminamiento. Siendo los retardos que más se toman en cuenta los de procesamiento nodal, cola, transmisión y propagación los cuales en su conjunto forman el llamado retardo nodal total.²⁷

²⁷ Cfr. Kurose y Ross 2010: 34 - 35

En la **Figura 2.6**, se representa el retardo nodal que existe en el *router* A. En dicha imagen se muestra un enlace de salida entre el *router* A hacia el *router* B y para la transferencia de datos entre ellos primero se presenta un retardo por procesamiento nodal seguido por los retardos de cola y transmisión finalizado con los retardos de transmisión y propagación. El retardo de transmisión básicamente indica el tiempo que se toma en examinar la cabecera del paquete de datos para poder determinar, entre otras cosas, a donde será enviado el paquete. El retardo por cola es tiempo que se pierde esperando que se realice la transmisión de dicho paquete. En cambio, el retardo de transmisión es la cantidad de tiempo que se toma para que el *router* saque fuera todo el paquete; por ello, este retardo depende de la longitud del paquete y de la velocidad de transmisión. Finalmente, se observa el retardo por propagación el cual es el tiempo necesario para enviar un bit del paquete de datos desde el inicio del enlace hasta finalizar el enlace entre dos hosts (en este caso del *router* A hacia el *router* B). Por ello, a diferencia del retardo de transmisión, el retardo de propagación depende de la distancia del enlace y no de la longitud del paquete de datos²⁸

Fuente: Kurose y Ross (2010)

Figura 2.6: Retardo Nodal en el *router* A

2.1.4.2.- Retardos de cola y pérdidas de paquetes

En una transmisión de paquetes de datos, los retardos ocasionados por una cola de datos pueden variar dependiendo del paquete. En otras palabras, el paquete que está más cercano al inicio de la cola demorará menos que el que se encuentra al final de la cola; incluso, la demora también será de acuerdo al tamaño del paquete de

²⁸ Cfr. Kurose y Ross 2010: 34 - 38

datos. Por ello, se puede decir que los retardos de cola dependen, entre las más importantes, de la velocidad del tráfico de cola y transmisión de enlace.²⁹

Debido a las variantes de medición para cada paquete de datos en una cola, se suele medir los retardos de cola mediante valores estadísticos como los retardos medios, varianza, etc. En esta sección, se va a tomar en cuenta como parámetro de medición a la intensidad de tráfico (λ/μ) donde λ es la velocidad media que llegan los bits a la cola y μ es la velocidad de transmisión en la que los bits salen de la (tanto λ como μ se miden en bits/segundo). En la **Figura 2.7**, se muestra la dependencia cualitativa del retardo de cola con respecto a la intensidad de tráfico. En ella se puede observar que mientras la intensidad de tráfico se acerca cada vez más a 1 el retardo medio de cola aumenta rápidamente.³⁰

Fuente: Kurose y Ross 2010

Figura 2.7: Retardo medio de Cola VS Intensidad de Tráfico (λ/μ)

Lo que se observa en **Figura 2.7** no es tan real, ya que según dicha figura el retardo medio de cola se aproxima al infinito cada vez que la intensidad de tráfico se aproxima a 1. Sin embargo, hay que tener en cuenta también las limitantes de hardware, ya que no los equipos en la realidad tienen una capacidad máxima de almacenaje de información generando esto un tamaño máximo de cola. Por ello, cuando se ha llegado a la máxima capacidad de cola y llega un paquete este es

²⁹ Cfr. Kurose y Ross 2010: 38 - 39

³⁰ Cfr. Kurose y Ross 2010: 38 - 40

eliminado; en otras palabras, el paquete se pierde. De lo mencionado se puede deducir que la probabilidad de que los paquetes perdidos aumenten si la intensidad de tráfico aumenta.³¹

2.1.5.- Medición de Performance de Enlaces

Para poder saber el desempeño de una red de computadoras es necesario tener valores cuantitativos en los cuales se pueda basar para dar un diagnóstico. Para este proyecto nos basaremos en valores de métricas como el *Throughput* y el tiempo de demora de un punto a otro.

2.1.5.1.- Tiempo de punto a punto

Es el tiempo necesario para transmitir un paquete de datos desde un host inicial a un host destino. También, se debe tener en cuenta que los retardos y pérdidas de paquetes mencionados en el punto 2.1.4 influyen en el tiempo que le toma a un paquete de datos llegar a su destino.

2.1.5.2.- *Throughput*

Según la Real Academia de Ingeniería (RAI), es la cantidad de caudal o tasa de información que es transmitida por una red³². Agregado a ello Di Caro y Dorigo mencionan que *throughput* es la cantidad de bits correctamente transmitidos por unidad de tiempo (bits/segundo) en una red de datos³³

³¹ Cfr. Kurose y Ross 2010: 38 - 41

³² Cfr. RAI 2016

³³ Cfr. Di Caro y Dorigo 1988: 321

2.2.- Conceptos matemáticos previos al ACO

En este punto se introduce conceptos y definiciones que son utilizados en el resto del proyecto propuesto, los cuales son necesarios para una mejor comprensión de los temas a tratar.

2.2.1.- Introducción a la teoría de grafos

La teoría de grafos es un campo de las matemáticas discretas cuya implementación ha estado motivada por sus diversas aplicaciones. Esta teoría puede ser utilizada para analizar cualquier situación en la que intervenga un conjunto de elementos en el que varios pares de ellos estén relacionados según una misma propiedad, como puede ser un circuito eléctrico, una red de carreteras, una red de computadoras, etc.³⁴

2.2.1.1.- Definición y representación de grafos

Gutiérrez y Lanchares (2010: 98) explican que “un grafo $G = (V, E)$ consiste en un conjunto finito de V , cuyos elementos reciben el nombre de vértices, y un conjunto E de pares de elementos de V , cuyos elementos se conocen como aristas. Si $\{u, v\}$ es una arista de G , se dice que los vértices u y v son adyacentes y llamaremos a u y v extremos de arista”.

En la **Figura 2.8** se muestra algunas de las variaciones de la idea de grafo, como son las siguientes:

- Multígrafos: grafos con varias aristas entre dos vértices.
- Pseudografos: grafos en donde se permiten aristas cuyos extremos coinciden.
- Digrafos o grafos dirigidos: grafos en los que se asigna un orden a los extremos de cada arista. En este caso, las aristas dirigidas se denotan de la forma (a, b) y se ha de entender que el orden es importante ($(a, b) \neq (b, a)$). En el caso de una arista dirigida

³⁴ Cfr. Gutiérrez y Lanchares 2010: 97

(a, b), el vértice a se le llama origen o fuente de la arista y al vértice b se le llama termino o vértice terminal de la arista.³⁵

Gutiérrez y Lanchares (2010: 98) indican que se hace referencia a un grafo simple cuando se trate de un grafo no dirigido que además no es multígrafo ni pseudografo.

Fuente: Gutiérrez y Lanchares, (2010).

Figura 2.8: Ejemplos de un grafo simple (a), de un pseudografo (b), de un multígrafo (c) y de un grafo dirigido (d).

Gutiérrez y Lanchares (2010: 98 - 99) indican que el grado de un vértice v , que se denotara por $\delta(v)$, de un grafo $G = (V, E)$ es el número de aristas de G que confluyen en v . En el caso de los pseudografos, cada lazo contribuye dos veces al grado del vértice correspondiente. Además, se dirá que un grafo $G = (V, E)$, con $|V| = n$, es regular (de grado r) si $\delta(x) = \delta(y) = r < n$ para todo $x, y \in V$. Los grafos admiten una presentación gráfica. Los vértices se representan por puntos y las aristas por líneas que conectan pares de vértices. Esta representación es de gran ayuda para estudiar propiedades de grafos no demasiado grandes. Además, el carácter intuitivo de estas representaciones sirve para formular y entender argumentos abstractos. Así, por ejemplo, se utilizan para estudiar redes de ordenadores, para saber si un circuito se puede implementar sobre un tablero plano, para distinguir compuestos químicos

³⁵ Cfr. Gutiérrez y Lanchares 2010: 98

con la misma fórmula molecular, para encontrar el camino más corto en una red de transporte, etc.

2.2.2.- Optimización Combinatoria

La optimización es el proceso de tratar de conseguir la mejor solución posible para un problema en específico. En el caso de un problema de optimización existen diferentes soluciones, un criterio para diferenciarlas y el objetivo de encontrar la mejor. La mayoría de los problemas de optimización de importancia teórica y/o práctica consisten en la búsqueda de una mejor configuración de un conjunto de variables para lograr ciertos objetivos. Estos problemas se dividen naturalmente en dos categorías: aquellos en los que las soluciones están codificadas con un valor real de las variables, y aquellos en los que las están codificadas con variables discretas. Entre estos últimos, se encuentra una clase de problemas llamados Problemas de Optimización Combinatoria (POC). En los POC, se busca un objeto de un conjunto finito, es decir, un número entero, un subconjunto de ellos, una permutación, o una estructura de grafo.³⁶

Para explicar los siguientes conceptos (desde el punto 2.2.3 al punto 2.4) citaremos textualmente lo dicho por Franco Luis Alejandro Arito en su trabajo final para obtener el grado de Licenciado en Ciencias de la Computación en la Universidad Nacional de San Luis con título “Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados”.

2.2.3.- Problema de optimización combinatoria

« Un problema de optimización combinatoria $P = (S, \Omega, f)$ puede ser definido por:

- Un conjunto de variables $X = \{x_1, \dots, x_n\}$;
- Dominios de variables $\{D_1, \dots, D_n\}$;
- Un conjunto de restricciones Ω entre las variables;

³⁶ Cfr. Alejandro 2010: 1

- Una función objetivo $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}_0^+$ a ser minimizada (o maximizada).

El conjunto de todas las posibles asignaciones es:

$$S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} | v_i \in D_i, \text{ y } s \text{ satisface } \Omega\}$$

Al conjunto S se lo llama espacio de búsqueda, ya que cada elemento del conjunto puede ser visto como una solución candidata. Para resolver un POC se tiene que encontrar una solución $s^* \in S$ que tenga un valor de función objetivo mínimo, esto es $f(s^*) \leq f(s), \forall s \in S$. A una solución s^* se la llama óptimo global de P , y al conjunto $S^* \subseteq S$ se lo llama el conjunto de todas las soluciones que son óptimos globales.

(...) En la actualidad, siguen apareciendo nuevos problemas de este tipo, lo que ha dado lugar a muchas propuestas de algoritmos para tratar de resolverlos. Las técnicas existentes se pueden clasificar básicamente en dos, algoritmos exactos y algoritmos aproximados. Los algoritmos exactos intentan encontrar una solución óptima y demostrar que la solución obtenida es de hecho la óptima global; estos algoritmos incluyen técnicas como, por ejemplo: procesos de vuelta atrás (*back-tracking*), Ramificación y Acotación (*branch and bound*), programación dinámica, etc. Debido a que los algoritmos exactos muestran un rendimiento pobre para muchos problemas se han desarrollado múltiples tipos de algoritmos aproximados que proporcionan soluciones de alta calidad (soluciones con valores de función objetivo relativamente bajos, aunque no necesariamente sean óptimas) para estos problemas combinatorios en un tiempo computacional razonable. »

(Alejandro 2010: 1 - 2)

2.2.4.- Heurística

« Los algoritmos aproximados, se pueden clasificar a su vez en dos tipos principales: los algoritmos constructivos y los algoritmos de búsqueda local. Los primeros se basan en generar soluciones desde cero añadiendo componentes a cada solución de paso a paso. Un ejemplo bien conocido son las heurísticas de construcción voraz o heurísticas voraces. Su gran ventaja es la velocidad: normalmente son muy rápidas y, además, a menudo devuelven soluciones razonablemente buenas. Sin embargo, no puede garantizarse que dichas soluciones sean óptimas con respecto a pequeños cambios a nivel local. En consecuencia, una mejora típica es refinar la solución obtenida por la heurística voraz utilizando búsqueda local. Los algoritmos de búsqueda local intentan repetidamente mejorar la solución actual con movimientos a soluciones vecinas (con la esperanza de que sean mejores). El caso más simple son los algoritmos de mejora

iterativa: si en el vecindario de la solución actual s se encuentra una solución mejor s' , ésta, reemplaza a la solución actual y se continúa la búsqueda a partir de s' ; si no se encuentra una solución mejor en el vecindario, el algoritmo termina en un óptimo local (...). »

(Alejandro 2010: 2-3).

2.2.5.- Estructura de vecindad

« Una estructura de vecindad es una función $N: S \rightarrow 2^S$ que asigna a cada solución $s \in S$ un conjunto de vecinos $N(s) \subseteq S$. Se dice que $N(s)$ es la vecindad de la solución s .

La introducción de la estructura de vecindad permite definir el concepto de soluciones que son óptimos (mínimos) locales. Las soluciones que son óptimos locales, pueden ser vistas como las “mejores” soluciones dentro de una región acotada del espacio de búsqueda. »

(Alejandro 2010: 3)

2.2.6.- Óptimo local con respecto a una estructura de vecindad

« Una solución s^+ es un óptimo local con respecto a una estructura de vecindad N si $\forall s \in N(s^+)$ se cumple $f(s^+) \leq f(s)$. Uno de los inconvenientes de los algoritmos de mejora iterativa es que puede estancarse en soluciones de baja calidad (óptimos locales muy lejanos al óptimo global). Para permitir una mejora adicional en la calidad de las soluciones, la investigación en este campo en las últimas dos décadas ha centrado su atención en el diseño de técnicas de propósito general para guiar a la construcción de soluciones o a la búsqueda local en las distintas heurísticas. Estas técnicas son conocidas como metaheurísticas (...). »

(Alejandro 2010: 3)

2.2.7.- Metaheurísticas

« El término metaheurística fue introducido por Glover, y deriva de la composición de dos palabras griegas. El sufijo meta significa “más allá de, en un nivel superior”, y heurística deriva del verbo heuriskein (...) que significa “encontrar, descubrir”. Existen muchas definiciones de metaheurística, pero en este trabajo se adopta formula en:

“Una metaheurística se define formalmente como un proceso de generación iterativo el cual guía a una heurística subordinada combinando inteligentemente diferentes conceptos para explorar y explotar el espacio de búsqueda, son utilizadas

estrategias de aprendizaje para estructurar la información con el objetivo de encontrar eficientemente soluciones óptimas.”

En resumen, se puede decir que las metaheurísticas son estrategias de alto nivel para explorar espacios de búsqueda usando diferentes métodos. Además, es de gran importancia que exista un balance dinámico entre diversificación e intensificación. El termino diversificación generalmente se refiere a la exploración del espacio de búsqueda (promover al proceso de búsqueda a examinar regiones no visitadas del espacio de búsqueda, para generar soluciones que difieran de manera significativa de las soluciones ya vistas), mientras que el termino intensificación se refiere a la explotación de la experiencia de búsqueda acumulada (enfocar la búsqueda en examinar soluciones que pertenezcan a la vecindad de las mejores soluciones encontradas).

Las metaheurísticas incorporan conceptos de muchos y diversos campos como la genética, la biología, la inteligencia artificial, las matemáticas, la física y la neurología entre otras (...). »

(Alejandro 2010: 3 - 4)

2.3.- Optimización basada en Colonias de Hormigas

« (...) ACO fue propuesto (...) como un método para resolver problemas de optimización combinatorios (POCs) duros. Los algoritmos de optimización basados en colonias de hormigas son parte de la rama inteligencia colectiva, este es, el campo de investigación que estudia algoritmos inspirados en la observación del comportamiento de enjambres (swarms). Los algoritmos de inteligencia colectiva están compuestos de individuos simples que cooperan a través de la auto-organización, es decir sin ninguna forma de control central sobre los miembros del enjambre. »

(Alejandro 2010: 18)

2.3.1.- De las colonias de hormigas a las hormigas artificiales

« La metaheurística ACO está inspirada en la observación del comportamiento de hormigas reales. En esta sección, se presenta observaciones hechas en experimentos con hormigas, y luego se muestra como estos experimentos inspiraron el diseño de la metaheurística. »

(Alejandro 2010: 18)

2.3.1.1.- Las colonias de hormigas

« Las hormigas son insectos sociales, los cuales viven en colonias y cuyo comportamiento está dirigido más hacia la supervivencia de la colonia como un todo que al de una simple componente individual. Los insectos sociales han capturado la atención de muchos científicos por el gran nivel de estructuración que alcanzan sus colonias, especialmente cuando se lo compara con la simplicidad relativa de los componentes de la colonia.

Uno de los primeros científicos en investigar el comportamiento social de insectos, fue el entomólogo francés Pierre-Paul Grassé. Entre los años 40 y 50, se dedicó a observar el comportamiento de termitas – en particular, las especies *Bellicositermes natalensis* y *Cubitermes*. Descubrió (...) que estos insectos son capaces de reaccionar a lo que llamó “estímulos significantes”, los cuales son señales que activan una reacción codificada genéticamente. Observó (...) que los efectos de estas reacciones pueden actuar como nuevos estímulos significantes, tanto para el insecto que las produjo como para los otros insectos en la colonia. Grassé usó el término estigmergía³⁷ (...) para describir este particular tipo de comunicación indirecta en la cual “los miembros de la colonia son estimulados por el desempeño que han logrado”.

Ejemplos de estigmergía, también pueden ser observados en colonias de hormigas. Muchas especies de hormigas, mientras caminan desde el nido hacia una fuente de comida y viceversa, depositan en el suelo una sustancia llamada feromona. Otras hormigas pueden oler esta sustancia, y su presencia influencia la elección de su camino³⁸, esto es, tienden a seguir concentraciones de feromona fuertes. La feromona depositada en el suelo forma un rastro de feromona, el cual permite a las hormigas encontrar buenas fuentes de comida que han sido previamente identificadas por otras hormigas.

Algunos científicos investigaron experimentalmente este comportamiento de depositar feromona y seguir rastros de feromona para entenderlo mejor y poder cuantificarlo. Deneubourg et al, diseñaron un experimento llamado “experimento del puente binario”. Ellos usaron hormigas *Linepithema humile* (hormigas originarias de Argentina). El nido de las hormigas fue conectado a una fuente de comida por dos puentes de igual longitud, ver [Figura 2.9] (...). Las hormigas pueden elegir libremente que brazo del puente usar cuando buscaban comida y la traían de vuelta al nido. Su comportamiento fue observado por un período de tiempo.

³⁷ La palabra estigmergía deriva de las palabras griegas stigma (marca, señal) y ergon (trabajo, acción).

³⁸ Cabe recordar que muchas especies de hormigas son ciegas.

Fuente: Alejandro, (2010)

Figura 2.9: experimento real llevado a cabo por Deneubourg et al.

En este experimento, inicialmente no hay feromona en ninguno de los dos puentes. Las hormigas empiezan explorando los alrededores del nido y eventualmente cruzan uno de los puentes, alcanzando la fuente de comida. Al caminar hacia la fuente de comida y de vuelta, las hormigas depositan feromona en el puente que usaron. Inicialmente, cada hormiga elige aleatoriamente uno de los puentes. Sin embargo, debido a cambios aleatorios, después de algún tiempo habrá más feromona depositada en uno de los puentes que en el otro. Debido a que las hormigas tienden a preferir (en probabilidad) seguir un rastro de feromona más fuerte, el puente que tiene más feromona atraerá más hormigas. Esto a su vez hace que el rastro de feromona crezca más, hasta que finalmente la colonia de hormigas converge hacia el uso del mismo puente³⁹, [tal como se muestra en la **Figura 2.10.**]

³⁹ Deneubourg et al, realizaron varios experimentos, y los resultados mostraron que cada uno de los dos puentes era usado en aproximadamente el 50% de las cosas.

Fuente: Alejandro, (2010)

Figura 2.10: Imagen del Experimento real llevado a cabo por Goss et al.

Este comportamiento a nivel de la colonia, basada en autocatálisis (retroalimentación positiva), puede ser explotado por las hormigas para encontrar el camino más corto entre una fuente de comida y su nido. Esto se demostró en otro experimento dirigido por Goss et al, en el que los dos puentes no eran de la misma longitud: uno era significativamente más largo que el otro, ver [Figura 2.10] (...). En este caso, las fluctuaciones aleatorias sobre la opción inicial de un puente eran más reducidas: debido a que aquellas hormigas que elegían por casualidad el puente más corto también eran las primeras en alcanzar el nido, y al volver al nido, escogían el puente más corto con mayor probabilidad cuando tenía un rastro de feromona más fuerte. Por consiguiente, las hormigas – gracias al mecanismo de seguir y depositar feromona – convergían rápidamente al uso del puente más corto. »

(Alejandro 2010: 18 - 20)

2.3.1.2.- Hormigas artificiales

« Estimulado por los interesantes resultados, descritos en la sección anterior, Goss et al, desarrollaron un modelo para explicar el comportamiento observado en el experimento del puente de dos brazos. Asumiendo que después de t unidades de tiempo desde el comienzo del experimento, m_1 hormigas han usado el primer brazo del puente y m_2 el segundo, la probabilidad p_1 para la $(m+1)$ -ésima hormiga de elegir el primer brazo del puente puede estar dada por (...) [la Ecuación 1:]

$$p_{1(m+1)} = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h} \quad (1)$$

Donde los parámetros k y h son necesarios para ajustar el modelo a los datos experimentales. La probabilidad que la misma $(m+1)$ -ésima hormiga elija el segundo brazo es $p_{2(m+1)} = 1 - p_{1(m+1)}$. Simulaciones de Monte Carlo, ejecutadas para probar si el modelo corresponde a los datos reales, mostraron muy buen ajuste para valores de $k \approx 20$ y $h \approx 2$.

Este modelo básico, el cual explica el comportamiento de las hormigas, puede ser usado como una inspiración para diseñar hormigas artificiales que resuelvan problemas de optimización definidos de una manera similar. En el ejemplo descrito anteriormente del comportamiento forrajero de las hormigas, la comunicación por estigmergia ocurre a través de la feromona que las hormigas depositan en el suelo. Análogamente, las hormigas artificiales pueden simular la acción de depositar feromona modificando apropiadamente variables de feromona asociadas con estados del problema que visitan mientras construyen soluciones al problema de optimización. También, de acuerdo al modelo de comunicación por estigmergia, las hormigas artificiales sólo tendrían acceso en forma local a estas variables de feromona.

Por lo tanto, las características principales de estigmergia mencionadas en la sección anterior pueden ser extendidas a agentes artificiales:

- asociando variables de estado con diferentes estados de problema; y
- dándole a los agentes sólo acceso local a estas variables.

Otro aspecto importante de la conducta forrajera de las hormigas reales que puede ser explotado por hormigas artificiales es la relación entre el mecanismo autocatalítico⁴⁰ y la evaluación implícita de soluciones. Por evaluación implícita de soluciones, se entiende al hecho de que caminos más cortos (los cuales corresponden a soluciones de menor costo en el caso de las hormigas artificiales) son completados antes que los más largos, y por lo tanto reciben refuerzos de feromona más rápido. La evaluación implícita de soluciones junto con el autocatálisis puede ser muy efectivo: mientras más corto es el camino, más pronto se deposita la feromona, y más hormigas usan el camino más corto. Si se usa apropiadamente, puede ser un mecanismo poderoso en algoritmos de optimización basados en población (por ejemplo, en los algoritmos evolutivos, el autocatálisis es implementado a través del mecanismo de selección/reproducción).

⁴⁰ El concepto de autocatálisis define al tipo de procesos propios de sistemas que son capaces de producir algunos de los elementos que son necesarios para su recurrencia continuada en el tiempo.

La estigmergia, junto con la evaluación implícita de soluciones y el comportamiento autocatalítico, ha dado lugar a los algoritmos ACO. La idea básica de los algoritmos ACO sigue muy estrechamente la inspiración biológica. Por lo tanto, hay muchas similitudes, entre las hormigas reales y las hormigas artificiales. Ambas colonias de hormigas (reales y artificiales) están compuestas de una población de individuos que trabajan juntos para lograr un cierto objetivo. Una colonia es una población de agentes simples, independientes y asíncronos que cooperan para encontrar una buena solución al problema a resolver. En el caso de las hormigas reales, el problema es encontrar la comida; mientras que en el caso de las hormigas artificiales, es el de encontrar una buena solución a un problema de optimización dado. Una sola hormiga (real o artificial) es capaz de encontrar una solución a su problema, pero sólo la cooperación entre muchos individuos a través de estigmergia les permite encontrar buenas soluciones.

En el caso de las hormigas reales, ellas depositan y reaccionan a una sustancia química llamada feromona. Las hormigas reales simplemente la depositan en la tierra mientras caminan. Las hormigas artificiales viven en un mundo virtual, de aquí que ellas sólo modifican valores numéricos (llamados por analogía, feromona artificial) asociados con diferentes estados de problema. Una secuencia de valores de feromona asociados con estados del problema se llama rastro de feromona artificial. En los algoritmos ACO, los rastros de feromona artificial son los únicos medios de comunicación entre las hormigas. Un mecanismo análogo a la evaporación física de feromona en colonias de hormigas reales permite a las hormigas artificiales olvidar la historia y enfocarse en nuevas direcciones de búsqueda prometedoras.

Así como las hormigas reales, las hormigas artificiales crean sus soluciones secuencialmente moviéndose de un estado del problema a otro. Las hormigas reales simplemente caminan, escogiendo una dirección basada en las concentraciones de feromona local y en una política de decisión aleatoria. Las hormigas artificiales también crean soluciones paso a paso, moviéndose a través de estados del problema disponibles y tomando decisiones aleatorias a cada paso.

Sin embargo, hay algunas diferencias importantes entre las hormigas reales y artificiales:

- Las hormigas artificiales viven en un mundo discreto se mueven secuencialmente a través de un conjunto finito de estados del problema.
- La actualización de feromona (es decir, el depósito y evaporación de feromona) no se realiza exactamente de la misma manera por las hormigas artificiales como por las reales. A veces la actualización de feromona es hecha sólo por algunas de las hormigas artificiales, y a menudo sólo después de que una solución se ha construido.

- Algunas implementaciones de hormigas artificiales usan mecanismos adicionales que no existen en el caso de las hormigas reales. Los ejemplos incluyen mirar hacia adelante, búsqueda local y vuelta atrás, entre otros. »

(Alejandro 2010: 20 - 22)

2.4.- La metaheurística de Optimización basada en Colonias de Hormigas

« Los algoritmos de Optimización basados en Colonias de Hormigas han sido formalizados en una metaheurística de optimización de combinatoria por Dorigo et al, y desde entonces han sido usados para resolver muchos Problemas de Optimización Combinatoria (POC). Dado un POC, el primer paso para la aplicación de un algoritmo ACO, para resolverlo consiste en definir un modelo adecuado. Luego este modelo es usado para definir el componente central de los algoritmos ACO: el modelo de feromona.

El modelo de un POC es usado para derivar el modelo de feromona utilizado por los algoritmos ACO. Primero, una variable de decisión instanciada $X_i = v_i^j$ (es decir, una variable X_i con un valor v_i^j asignado de su dominio $\mathbf{D}_i$) se llama componente de una solución y se denota por c_{ij} . El conjunto de todas las posibles componentes de soluciones es denotado por $\mathbf{C}$. Un parámetro de rastro de feromona T_{ij} es asociado con cada componente c_{ij} . El conjunto de todos los parámetros de rastros de feromona es denotado por $\mathbf{T}$. El valor de un parámetro de rastro de feromona T_{ij} es denotado por τ_{ij} (y es llamado valor de feromona). Este valor de feromona es usado y actualizado por el algoritmo ACO durante la búsqueda, y permite modelar la distribución de probabilidad de diferentes componentes de una solución.

En los algoritmos ACO, las hormigas artificiales construyen una solución para un POC travesando un grafo llamado grafo de construcción, $G_C(\mathbf{V}, \mathbf{E})$. El grafo de construcción (totalmente conectado) consiste de un conjunto de vértices $\mathbf{V}$ y un conjunto de arcos $\mathbf{E}$. El conjunto de componentes $\mathbf{C}$ puede ser asociado con el conjunto de vértices $\mathbf{V}$, o con el conjunto de arcos $\mathbf{E}$. Las hormigas se mueven de un vértice a otro vértice a lo largo de los arcos del grafo, construyendo incrementalmente una solución parcial. Además, las hormigas depositan una cierta cantidad de feromona sobre las componentes, es decir, en los vértices o en los arcos que atraviesan. La cantidad de feromona $\Delta\tau$ depositada puede depender de la calidad de la solución encontrada. Las hormigas siguientes utilizan la información de la feromona como una guía hacia regiones más prometedoras del espacio de búsqueda.

La metaheurística ACO se muestra en el Algoritmo 1. El mismo consiste en una fase de inicialización y una iteración sobre tres componentes. Esta iteración consiste de la construcción de soluciones por todas las hormigas, la mejora de la solución (fase optativa) con el uso de un algoritmo de búsqueda local, y la actualización de feromona. A continuación, se explican estos tres componentes en detalle.

Algoritmo 1 Metaheurística ACO

Establece parámetros, inicializar rastros de feromona
While (no se cumpla condición de terminación) **do**
 ConstruirSolucionesporHormigas
 AplicarBúsquedaLocal { Opcional }
 ActualizarFeromona
end while

Fuente: Elaboración propia

Figura 2.11: Algoritmo 1

ConstruirSolucionesporHormigas Un conjunto de m hormigas artificiales construyen soluciones a partir de elementos de un conjunto finito de componentes de soluciones disponibles $C = \{c_{ij}\}, i = 1, \dots, n, j = 1, \dots, |D_i|$. La construcción de una solución empieza con una solución parcial vacía $s^p = \emptyset$. Luego, en cada paso de la construcción, la solución parcial actual s^p es extendida agregando una componente de solución factible del conjunto de vecinos factibles $N(s^p) \subseteq C$. El proceso de construcción de soluciones puede ser considerado como un camino en el grafo de construcción $G_C(V, E)$. Los caminos permitidos en G_C están definidos implícitamente por el mecanismo de construcción de soluciones que define el conjunto $N(s^p)$ con respecto a una solución parcial s^p .

La elección de una componente de solución de $N(s^p)$ se hace probabilísticamente en cada paso de la construcción. Las reglas exactas para la elección probabilística de componentes de soluciones varían entre diferentes variantes de algoritmos ACO. La regla más conocida es de Ant System [se muestra en la **Ecuación 2**]:

$$p(c_{ij}|s^p) = \frac{\tau_{ij}^\alpha \cdot \eta(c_{ij})^\beta}{\sum_{c_{ij} \in N(s^p)} \tau_{ij}^\alpha \cdot \eta(c_{ij})^\beta}, \forall c_{ij} \in N(s^p) \quad (2)$$

Donde τ_{ij} es el valor de feromona asociado con la componente c_{ij} , y $\eta(\cdot)$ es una función que asigna a cada paso de la construcción un valor heurístico para cada componente de solución factible $c_{ij} \in N(s^p)$. Los valores que son retornados por esta función son llamados normalmente información heurística. Además, α y β son parámetros positivos cuyos valores determinan la importancia relativa de la feromona y de la información heurística respectivamente. La ecuación 2 es una generalización de la

ecuación 1. Como se puede apreciar, la formalización de los algoritmos ACO sigue estrechamente la inspiración biológica.

AplicarBúsquedaLocal Una vez que las soluciones han sido construidas, y antes de actualizar la feromona, a veces pueden ser necesarias algunas acciones opcionales. Estas acciones son llamadas acciones auxiliares⁴¹, y pueden ser usadas para implementar acciones centralizadas y/o específicas del problema, las cuales no pueden ser realizadas por simples hormigas. La acción auxiliar más usada consiste en la aplicación de búsqueda local a la solución construida: las soluciones localmente óptimas son usadas para decidir cómo actualizar feromona.

ActualizarFeromona El objetivo de la actualización de feromona es incrementar los valores de feromona asociados con soluciones buenas o prometedoras, y decrementar aquéllos valores que están asociados con malas soluciones. Normalmente, esto se logra:

- Decrementando todos los valores de feromona a través de la evaporación de feromona.
- Aumentando los niveles de feromona asociados con un conjunto de buenas soluciones S_{upd} elegido

$$\tau_{ij} \leftarrow (1 - \rho) \cdot \tau_{ij} + \rho \cdot \sum_{s \in S_{iter} | C_{ij} \in s} F(s) \quad (3)$$

Donde S_{upd} es el conjunto de soluciones que son usadas para la actualización, $\rho \in (0,1]$ es un parámetro llamada factor de evaporación, y $F: S \rightarrow \mathbb{R}_0^+$ es una función tal que $f(s) < f(s') \Rightarrow F(s) \geq F(s'), \forall s \neq s' \in S$. La función $F(\cdot)$ es llamada función de fitness.

La evaporación de feromona es necesaria para evitar una convergencia demasiado rápida del algoritmo. Implementa una forma útil de olvidarse y favorecer la exploración de nuevas áreas en el espacio de búsqueda. Diferentes algoritmos ACO, por ejemplo, MAX-MIN Ant System o Ant Colony System difieren en la forma en que actualizan feromona.

Instanciaciones de la regla de actualización presentada en la ecuación 3 son obtenidas por diferentes especificaciones de S_{upd} , la cual en muchos casos es un subconjunto de $S_{iter} \cup \{s_{bs}\}$ donde S_{iter} es el conjunto de soluciones que fueron construidas en la iteración actual, y s_{bs} es la solución best-so-far (mejor hasta ahora), esto es, la mejor solución

⁴¹ El término original en inglés es daemon actions, concepto que caracteriza a procesos que se ejecutan cuando se cumplen determinadas condiciones.

encontrada desde la primera iteración del algoritmo. Un ejemplo muy conocido es la regla de actualización de Ant System, donde:

$$S_{upd} \leftarrow S_{iter} \quad (4)$$

Un ejemplo de regla de actualización de feromona que es muy utilizada en la práctica es la regla de actualización IB (donde IB significa la mejor de la iteración).

$$S_{upd} \leftarrow \arg \max_{s \in S_{iter}} F(s) \quad (5)$$

La regla de actualización IB introduce un sesgo mucho más fuerte hacia las soluciones buenas encontradas respecto de la regla de Ant System. Aunque esto aumenta la velocidad con que se encuentran buenas soluciones, también aumenta la probabilidad de convergencia prematura. Un sesgo aún más fuerte es introducido por la regla de actualización BS, donde BS se refiere al uso de la solución mejor-hasta-ahora s_{bs} . En este caso, S_{upd} toma el valor de $\{s_{bs}\}$. En la práctica, los algoritmos ACO que usan variantes de las reglas de actualización IB o BS y que adicionalmente incluyen mecanismos para evitar convergencia prematura obtienen mejores resultados que aquellos que usan la regla de actualización de Ant System. »

(Alejandro 2010: 23 - 25)

CAPÍTULO 3: DESCRIPCIÓN DEL SISTEMA REQUERIDO Y ALGORITMO PROPUESTO

En este capítulo se describirá el sistema de red requerido con el cual se implementó el algoritmo basado en ACO; a su vez se desarrollará el esquema de dicho algoritmo. También, se describirá la interface final del proyecto y su integración con el sistema de red y el algoritmo propuesto. Para ello se hará uso de sus diagramas de bloques.

3.1.- Sistema de red requerido

El algoritmo basado en ACO tiene como objetivo poder generar rutas de encaminamiento para poder transmitir datos de una forma eficiente. Para fines de demostración didáctica del desempeño del algoritmo propuesto se limitó su implementación en una red LAN con ocho servidores las cuales están dentro de un mismo segmento de red. Con estos servidores se implementó tres topologías diferentes en las cuales se analizará el desempeño del algoritmo basado en ACO (esto se detallará más en el capítulo siguiente). Para ello, se requirió de un sistema red como el que se muestra en la **Figura 3.1** en la cual se está resaltando la ubicación de la PC en donde se encontrará el aplicativo propuesto (IP 10.11.11.80)

Fuente: Elaboración propia

Figura 3.1: Sistema de red requerido

3.2.- Algoritmo propuesto

El algoritmo propuesto se encuentra basado en ACO y para su implementación se utilizó el lenguaje de programación C# en el compilador Visual Studio 2013. El algoritmo en mención inicia con la creación de hormigas artificiales por medio de generadores, valga la redundancia, de hormigas artificiales. Estos generadores se encargan de crear hormigas artificiales a las cuales se les denominada Hormigas Forwards quienes tienen como objetivo encontrar la ruta más corta hacia un nodo de red destino. Estas hormigas son creadas en cada nodo de red y de forma asíncrona; para ello, se crean hilos para así lograr que el proceso de cada hormiga artificial no afecte a las hormigas generadas en los demás nodos de red; es decir, los generadores y las hormigas creadas por ellos trabajan en forma paralela y no en forma secuencial.

Las Hormigas Forward son enviadas de forma aleatoria en busca de un nodo destino. Sin embargo, la elección de la ruta a seguir por dicha hormiga no es de forma aleatoria, sino que su elección se basa en la probabilidad descrita en la Ecuación (1) la cual fue detallada en el Capítulo 2. Agregado a ello, a cada Hormiga Forward se le asigna un tiempo de vida máximo. Si pasado ese tiempo de vida no ha llegado a un destino, la hormiga artificial es eliminada en conjunto con el hilo en la cual se ha estado ejecutando el proceso de dicha

hormiga. En el caso que la Hormiga Forward si logre llegar a un nodo de red destino, se crea una nueva Hormiga artificial llamada Backward. En ese momento la Hormiga Forward transfiere la información que recopiló en su camino hacia su nodo destino (ruta seguida, nodos visitados y sus costos). Una vez transferida esta información a la Hormiga Backward, la Hormiga Forward es eliminada. Inmediatamente la hormiga artificial Backward crea una feromona artificial la cual es depositada en su camino de retorno al nodo origen siguiendo la ruta obtenida de información brindada por la hormiga Forward. Esta feromona va disminuyendo su intensidad según transcurre el tiempo. Es decir, mientras más tiempo le tome llegar a la hormiga Backward al nodo origen la feromona tendrá menos intensidad (intensidad de la feromona es inversamente proporcional al tiempo transcurrido). Una vez que esta hormiga llegue a su nodo origen es eliminado para posteriormente dar inicio, nuevamente, al generador de hormigas artificiales la cual crea nuevas hormigas Forward. A diferencia de las primeras hormigas Forwards están no eligen la ruta a seguir basado en probabilidad de la ecuación (1) sino que lo hacen basándose en la ecuación (2). Esta ecuación es utilizada ya que aumenta la probabilidad de elección de la ruta que contenga una mayor intensidad de feromona (señal que fue depositada por la hormiga Backward que le tomo menos tiempo en llegar al nodo origen). Este proceso se repite con cada hormiga Forward generando que la ruta más corta tenga una intensidad de feromona elevada haciendo a su vez que la probabilidad de que las hormigas elijan siempre esa ruta sea muy elevada. De esta forma se logra obtener el camino más corto desde un nodo origen hacia un nodo destino.

La estructura del algoritmo mencionado se puede observar en la **Figura 3.2**. En dicho diagrama se muestra el comportamiento de las hormigas artificiales para poder obtener la ruta más corta usando la lógica descrita anteriormente. Cabe señalar que, además de lo ya mencionado, el algoritmo propuesto guarda la información de cada generador y hormiga creada para así poder obtener un reporte estadístico el cual es será analizado en el siguiente capítulo.

Fuente: Elaboración propia

Figura 3.2: Diagrama de Bloques del Algoritmo propuesto

3.3.- Interfaz propuesta

Ya se describió el funcionamiento del algoritmo base de este proyecto, pero también es importante describir el funcionamiento de la interfaz que utiliza dicho algoritmo. Para ello, se apoya en el diagrama de la **Figura 3.3** el cual describe la lógica utilizada para la implementación de la interfaz propuesta.

Fuente: Elaboración propia

Figura 3.3: Diagrama de Bloques de la Interface Propuesta

En la **Figura 3.1** se puede observar que el algoritmo de la interfaz inicia con un escaneo de nodos de red el cual se realiza en un segmento de red ya establecido. Después, de ello actualiza la tabla de Servidores ONLINE y a su vez se crea un gráfico de la topología de red. Una vez obtenido la información en mención se puede ejecutar el algoritmo basado en ACO el cual tiene el comportamiento descrito anteriormente. De la información obtenida por dicho algoritmo se crea un reporte estadístico.

La interfaz gráfica del algoritmo propuesto se puede observar en la **Figura 3.4**. Dicha interfaz muestra una tabla en donde se está realizando el escaneo de los nodos conectados en ese momento. En una segunda tabla se muestra la lista de los nodos conectados con su respectivo tiempo de distancia desde el nodo en donde se está haciendo el monitoreo. La interfaz del aplicativo, también, muestra un gráfico el cual representa a la topología de red a la cual se realiza el algoritmo de optimización de hormigas. Finalmente, se muestra dos reportes estadísticos de los resultados de la aplicación del algoritmo propuesto en la topología mostrada. Dicho reporte consta de las estadísticas de los nodos de red en la cual se muestra los tiempos de viajes desde un nodo origen hacia un nodo destino. También, se muestra un reporte de las Hormigas artificiales utilizadas por el Algoritmo por Optimización por Colonias de Hormigas. Hay que tener en cuenta que tanto el algoritmo de monitoreo como el algoritmo basado en ACO trabajan en paralelo y no de forma secuencial.

Fuente: Elaboración propia

Figura 3.4: Interfaz Gráfica del Algoritmo propuesto

La realización del monitoreo consta de enviar mensajes de control desde el Nodo de Monitoreo hacia un segmento de IPs las cuales representan a Nodo de Red de cada Servidor. Estos mensajes están basados en función al Protocolo de Mensaje de Control y Error de Internet, Internet Control Message Protocol (ICMP), el cual tiene características similares al protocolo UDP, pero con un formato más simple. Tener en cuenta que el envío de mensajes de control se realiza IP por IP y no como un broadcast, ya que se desea tener los tiempos de demora desde el Nodo de Monitoreo hacia cada Nodo de Red. **En la Figura 3.5** se puede observar que en la tabla de la izquierda se está realizando un escaneo de las IPs mientras que en la tabla de la derecha se muestra ya los servidores encontrados como conectados. Se debe tener en cuenta que primero se hace un escaneo completo de un segmento de red y los resultados del escaneo son comparados con la tabla de “Servidores ONLINE” y, de haber cambios, estos son actualizados. Lo mencionado se puede observar en la **Figura 3.6** en la cual se muestra el diagrama de bloques del algoritmo de monitoreo.

Scanning IP:9.48.68.91			Servidores ONLINE	
IP	HostName	Time ICMP	IP	Time ICMP
9.48.68.82	sig-9-48-...	606 ms	9.48.68.82	584 ms
9.48.68.83	sig-9-48-...	354 ms	9.48.68.83	344 ms
9.48.68.85	sig-9-48-...	314 ms	9.48.68.85	486 ms
9.48.68.87	sig-9-48-...	4130 ms	9.48.68.87	6425 ms
9.48.68.89	sig-9-48-...	577 ms	9.48.68.89	376 ms
9.48.68.90	sig-9-48-...	297 ms	9.48.68.90	298 ms
9.48.68.91	sig-9-48-...	314 ms	9.48.68.91	310 ms
			9.48.68.93	340 ms
			9.48.68.94	354 ms
			9.48.68.95	314 ms

Fuente: Elaboración propia

Figura 3.5: Interface Monitoreo

Fuente: Elaboración propia

Figura 3.6: Diagrama de Bloques del Algoritmo de Monitoreo

Una vez obtenida la tabla de “Servidores ONLINE” se crea un gráfico de la topología de red tal como se muestra en la **Figura 3.7**. Se debe tener en cuenta que esta topología se actualiza según los cambios que pueda suceder. Es decir, si por lo menos uno de los nodos se apagará o se agregará uno o varios nodos a la topología de red, el Host de monitoreo detectará el cambio. Una vez detectado el cambio la tabla “Servidores ONLINE” se actualizará. También, estos cambios harán que harán visibles en el gráfico de la topología de Red, ya que los nodos desconectados se muestran de color rojo mientras que los conectados se muestran de color verde (ver **Figura 3.7**).

Fuente: Elaboración Propia

Figura 3.7: Grafica de la Topología de Red

Las IPs de los servidores online se simularán como si fueran vértices de un grafo al cual se le aplicara el ACO. Para ello, primero se debe crear un gráfico según la topología a utilizar de acuerdo al número de nodos conectados (nodos detectados por el Host de monitoreo). Luego se crea las hormigas artificiales junto al generador de hormigas para después generar un algoritmo para enviar dichas hormigas a todos los nodos de forma aleatoria. Finalmente, se crea un reporte estadístico de lo sucedido en cada vértice.

Después de ejecutado el algoritmo basado en ACO se muestra un reporte de la cantidad de hormigas que cruzan cada vértice y los tiempos que demoran en comunicarse un vértice hacia otro. En la **Figura 3.8** se muestra el reporte que se genera después de aplicarse el algoritmo basado en ACO. En dicha Figura se muestra en el lado izquierdo el reporte estadístico por nodo de red. En dicho reporte se muestra las conexiones de cada nodo, los tiempos de conexión entre ellos con sus respectivos *throughputs*. En el lado derecho se muestra las hormigas artificiales enviadas desde un nodo origen hacia un nodo destino.

Fuente: Elaboración Propia

Figura 3.8: Reporte Estadístico del Algoritmo Propuesto

En el algoritmo propuesto se puede generar las conexiones virtuales entre los nodos. De esas conexiones virtuales creadas se arma una topología la cual se mostrará en el gráfico y a ese grafico se le aplicará el algoritmo basado en ACO para generar un reporte estadístico donde se muestre los tiempos más cortos de conexión entre los nodos y los *throughputs* en cada nodo. También, se mostrar la cantidad de hormigas cruzada en cada nodo.

CAPÍTULO 4: PRUEBAS, RESULTADOS Y VALIDACIÓN

En este capítulo se describirá las pruebas que se realizará al algoritmo basado en ACO dentro del aplicativo propuesto. Para ello se utilizará tres tipos de topología diferentes con las limitaciones mencionadas el Capítulo 3. Sin embargo, para la primera prueba no se utilizará ocho nodos de red sino doce, pero estos conectados en una red LAN por WiFi. Las otras dos topologías si se realizarán en una red conectada por cables de red UTP y ambas con ocho nodos de red, pero con topologías de redes distintas. En cada prueba se medirá los tiempos obtenidos desde el nodo de monitoreo hacia los demás nodos con el algoritmo basado en ACO y sin este algoritmo. Se hará una comparativa de dichos y se realizará un análisis los resultados.

4.1.- Prueba 1

En la primera prueba se validará los resultados en una topología en la cual todos los nodos estén conectados entre ellos, ya que este sería el caso más difícil pues hay una mayor cantidad de caminos que puede elegir la hormiga artificial. Para la evaluación de los resultados se tomará los resultados obtenidos en el escaneo de la **Figura 4.1**. En ella se puede observar que hay 12 nodos de red o servidores conectados formando una topología a la cual se le llamara Topología de Red 1 (gráfica de la derecha de la **Figura 4.1**). De esos nodos se crea una topología en la cual todos se conectan entre si y se genera un reporte por nodo de red. Cabe señalar que esta prueba se realizará en una conexión WiFi.

Fuente: Elaboración Propia

Figura 4.1: Topología de Red 1

Como se mencionó en el capítulo anterior, el escaneo se realiza enviando un mensaje a cada IP enviando un mensaje mediante el protocolo ICMP el cual, en caso de estar conectado, nos retorna un tiempo en milisegundos. Este tiempo será tomado como referencia para compararlo con el tiempo que se toma en detectar la conexión de todas las IP por el algoritmo basado en ACO. Cabe señalar, que la comparación se realizara desde la IP del monitoreo (10.11.128.211) hacia las demás IPs conectadas a la red. En la **Figura 4.3** se muestra la IP de host de monitoreo y en la **Figura 4.2** se muestra los tiempos tomados desde la IP del host de monitoreo hacia los demás nodos. Tanto como en la **Figura 4.1** (en la tabla de escaneo y servidores Online ubicada en el lado superior izquierdo) y la **Figura 4.2** se puede validar que para la IP 10.11.128.211 se tiene un tiempo de respuesta de 0 milisegundos.

IP	Time ICMP
10.11.128.190	380 ms
10.11.128.191	8 ms
10.11.128.194	38 ms
10.11.128.197	8 ms
10.11.128.200	1261 ms
10.11.128.201	402 ms
10.11.128.204	71 ms
10.11.128.206	556 ms
10.11.128.211	0 ms
10.11.128.212	41 ms
10.11.128.213	2829 ms
10.11.128.214	357 ms

Fuente: Elaboración Propia

Figura 4.2: Tiempos desde la IP del Host de monitoreo hacia los demás nodos de red

Fuente: Elaboración Propia

Figura 4.3: IP del host de monitoreo

En el reporte se muestra las conexiones que tiene cada nodo, o en otras palabras, los nodos vecinos. Por ejemplo, en la **Figura 4.4** se muestra los nodos vecinos a la IP 10.11.128.211 (monitoreo) y la IP 10.11.128.190. De esta misma forma se puede validar el reporte estadístico de todos los nodos de red con sus respectivos nodos vecinos en el ANEXO 1

ESTADISTICAS DE: 10.11.128.190	ESTADISTICAS DE: 10.11.128.211
Lista de IP conectadas al Nodo de Red:	Lista de IP conectadas al Nodo de Red:
10.11.128.191	10.11.128.190
10.11.128.194	10.11.128.191
10.11.128.197	10.11.128.194
10.11.128.200	10.11.128.197
10.11.128.201	10.11.128.200
10.11.128.204	10.11.128.201
10.11.128.206	10.11.128.204
10.11.128.211	10.11.128.206
10.11.128.212	10.11.128.212
10.11.128.213	10.11.128.213
10.11.128.214	10.11.128.214

Fuente: Elaboración Propia

Figura 4.4: Lista de nodos vecinos de las IPs 10.11.128.190 y 10.11.128.211

Agregado a ello, el reporte también muestra los tiempos promedios, el tiempo mínimo y la variación entre todos los tiempos tomados desde un nodo de origen a un nodo destino. En la **Figura 4.5** se muestra los tiempos tomados desde el nodo origen 10.11.128.190 (lado izquierdo) hacia los demás nodos. De la misma forma se realiza el reporte

estadístico de los tiempos teniendo como origen al nodo 10.11.128.211 (lado derecho). Los demás tiempos tomados en cada nodo de red se puede ver en el ANEXO 1.

Como se mencionó anteriormente, el *Throughput* es la cantidad de datos correctamente transmitidos en una unidad de tiempo. Es por ello, que para el calcular dicho valor se toma en cuenta la cantidad de hormigas artificiales generadas y enviadas en cada nodo de red menos la cantidad de hormigas artificiales que llegan al nodo de red origen después del viaje de cada hormiga en busca de un nodo vecino. Esa diferencia viene a ser los bits correctamente transmitidos desde un nodo de red. Para calcular la unidad de tiempo se tomada de la información de tiempo que lleva cada hormiga artificial de su viaje (nodo de origen a un nodo destino y su retorno). El algoritmo propuesto es capaz de calcular los *throughputs* de cada nodo de red. Por ello, estos valores son mostrados, también, en el reporte estadístico que genera el aplicativo propuesto. Por ejemplo, en la **Figura 4.6** se muestra los *throughputs* teniendo como origen al nodo 10.11.128.190 y al nodo 10.11.128.211. Los cálculos de los demás *throughputs* se pueden ver en el ANEXO 1

<p>Tiempo de viaje para el destino: 10.11.128.191</p> <p>Mejor tiempo [ms]: 0.1999999999999999</p> <p>Variación del tiempo de viaje [ms]: 3.72860037233369E-32</p> <p>Tiempo promedio de viaje [ms]: 0.2</p>	<p>Tiempo de viaje para el destino: 10.11.128.190</p> <p>Mejor tiempo [ms]: 0.09999999999999943</p> <p>Variación del tiempo de viaje [ms]: 0.000467937673666108</p> <p>Tiempo promedio de viaje [ms]: 0.108675406567827</p>
<p>Tiempo de viaje para el destino: 10.11.128.194</p> <p>Mejor tiempo [ms]: 0.09999999999999996</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 0.09999999999999996</p>	<p>Tiempo de viaje para el destino: 10.11.128.191</p> <p>Mejor tiempo [ms]: 0.07999999999999996</p> <p>Variación del tiempo de viaje [ms]: 1.440000000000016E-05</p> <p>Tiempo promedio de viaje [ms]: 0.0813333333333333</p>
<p>Tiempo de viaje para el destino: 10.11.128.197</p> <p>Mejor tiempo [ms]: 0.06666666666666629</p> <p>Variación del tiempo de viaje [ms]: 1.01881694058085E-30</p> <p>Tiempo promedio de viaje [ms]: 0.0666666666666661</p>	<p>Tiempo de viaje para el destino: 10.11.128.194</p> <p>Mejor tiempo [ms]: 0.03333333333333314</p> <p>Variación del tiempo de viaje [ms]: 2.53600041187846E-30</p> <p>Tiempo promedio de viaje [ms]: 0.0333333333333332</p>
<p>Tiempo de viaje para el destino: 10.11.128.200</p> <p>Mejor tiempo [ms]: 0.03333333333333314</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 0.03333333333333314</p>	<p>Tiempo de viaje para el destino: 10.11.128.197</p> <p>Mejor tiempo [ms]: 0.03333333333333314</p> <p>Variación del tiempo de viaje [ms]: 0.000155917941544861</p> <p>Tiempo promedio de viaje [ms]: 0.0360329025846561</p>
<p>Tiempo de viaje para el destino: 10.11.128.201</p> <p>Mejor tiempo [ms]: 0.06666666666666629</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 0.06666666666666629</p>	<p>Tiempo de viaje para el destino: 10.11.128.200</p> <p>Mejor tiempo [ms]: 0.11999999999999999</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 0.11999999999999999</p>
<p>Tiempo de viaje para el destino: 10.11.128.204</p> <p>Mejor tiempo [ms]: 0.1333333333333326</p> <p>Variación del tiempo de viaje [ms]: 1.84580190554312E-06</p> <p>Tiempo promedio de viaje [ms]: 0.133488966791793</p>	<p>Tiempo de viaje para el destino: 10.11.128.201</p> <p>Mejor tiempo [ms]: 0.09999999999999943</p> <p>Variación del tiempo de viaje [ms]: 3.92264305783982E-05</p> <p>Tiempo promedio de viaje [ms]: 0.105739562799997</p>
<p>Tiempo de viaje para el destino: 10.11.128.206</p> <p>Mejor tiempo [ms]: 0.23333333333332</p> <p>Variación del tiempo de viaje [ms]: 3.7946481753872E-05</p> <p>Tiempo promedio de viaje [ms]: 0.238489474614531</p>	<p>Tiempo de viaje para el destino: 10.11.128.204</p> <p>Mejor tiempo [ms]: 0.266666666666652</p> <p>Variación del tiempo de viaje [ms]: 0.000129218575750946</p> <p>Tiempo promedio de viaje [ms]: 0.276861130095861</p>
<p>Tiempo de viaje para el destino: 10.11.128.211</p> <p>Mejor tiempo [ms]: 1</p> <p>Variación del tiempo de viaje [ms]: 1.14039073997558E-06</p> <p>Tiempo promedio de viaje [ms]: 0.0334290530650239</p>	<p>Tiempo de viaje para el destino: 10.11.128.206</p> <p>Mejor tiempo [ms]: 0.333333333333314</p> <p>Variación del tiempo de viaje [ms]: 3.98700143999505E-05</p> <p>Tiempo promedio de viaje [ms]: 0.345105333333334</p>
<p>Tiempo de viaje para el destino: 10.11.128.212</p> <p>Mejor tiempo [ms]: 0.133333333333326</p> <p>Variación del tiempo de viaje [ms]: 5.38433239425394E-06</p> <p>Tiempo promedio de viaje [ms]: 0.133798237920129</p>	<p>Tiempo de viaje para el destino: 10.11.128.212</p> <p>Mejor tiempo [ms]: 0.1999999999999989</p> <p>Variación del tiempo de viaje [ms]: 2.18181744000326E-05</p> <p>Tiempo promedio de viaje [ms]: 0.2021719999999994</p>
<p>Tiempo de viaje para el destino: 10.11.128.213</p> <p>Mejor tiempo [ms]: 0.1999999999999989</p> <p>Variación del tiempo de viaje [ms]: 2.74642600932117E-28</p> <p>Tiempo promedio de viaje [ms]: 0.1999999999999989</p>	<p>Tiempo de viaje para el destino: 10.11.128.213</p> <p>Mejor tiempo [ms]: 1</p> <p>Variación del tiempo de viaje [ms]: 1.96224921153323E-07</p> <p>Tiempo promedio de viaje [ms]: 0.0666797586120153</p>
<p>Tiempo de viaje para el destino: 10.11.128.214</p> <p>Mejor tiempo [ms]: 0.1999999999999989</p> <p>Variación del tiempo de viaje [ms]: 4.52595897671992E-28</p> <p>Tiempo promedio de viaje [ms]: 0.200000000000013</p>	<p>Tiempo de viaje para el destino: 10.11.128.214</p> <p>Mejor tiempo [ms]: 29</p> <p>Variación del tiempo de viaje [ms]: 1.1412146150537E-05</p> <p>Tiempo promedio de viaje [ms]: 0.0677176201899834</p>

Fuente: Elaboración Propia

Figura 4.5: Estadísticas de tiempos 10.11.128.190(izquierda) y 10.11.128.211(derecha)

Desde el nodo de origen: 10.11.128.190 se obtiene los throughputs:	Desde el nodo de origen: 10.11.128.211 se obtiene los throughputs:
Nodo destino 10.11.128.191 => 2200 [fps]	Nodo destino 10.11.128.190 => 2608.20975762784 [fps]
Nodo destino 10.11.128.194 => 199.999999999999 [fps]	Nodo destino 10.11.128.191 => 162.666666666666 [fps]
Nodo destino 10.11.128.197 => 199.999999999999 [fps]	Nodo destino 10.11.128.194 => 1099.99999999999 [fps]
Nodo destino 10.11.128.200 => 66.6666666666629 [fps]	Nodo destino 10.11.128.197 => 900.822564616401 [fps]
Nodo destino 10.11.128.201 => 266.666666666665 [fps]	Nodo destino 10.11.128.200 => 839.99999999999 [fps]
Nodo destino 10.11.128.204 => 6941.42627317321 [fps]	Nodo destino 10.11.128.201 => 951.656065199976 [fps]
Nodo destino 10.11.128.206 => 4054.32106844703 [fps]	Nodo destino 10.11.128.204 => 3599.19469124619 [fps]
Nodo destino 10.11.128.211 => 2373.4627676167 [fps]	Nodo destino 10.11.128.206 => 2415.73733333338 [fps]
Nodo destino 10.11.128.212 => 4549.14008928439 [fps]	Nodo destino 10.11.128.212 => 2021.71999999999 [fps]
Nodo destino 10.11.128.213 => 5399.99999999994 [fps]	Nodo destino 10.11.128.213 => 4200.82479255696 [fps]
Nodo destino 10.11.128.214 => 3400.00000000022 [fps]	Nodo destino 10.11.128.214 => 3995.33959120902 [fps]

Fuente: Elaboración Propia

Figura 4.6: Estadísticas de los *throughputs*: nodos 10.11.128.190 y 10.11.128.211

En el segundo reporte se muestra, valga la redundancia, los reportes estadísticos de los generadores de hormigas artificiales en la cual se muestra el conteo de las hormigas artificiales enviadas desde un nodo origen hacia un nodo destino; sin embargo, hay que tener en cuenta que no todas estas hormigas logran llegar a su destino el cual, como se comentó anteriormente, es aprovechado para calcular los *throughputs* para cada nodo. En la **Figura 4.7** se muestra el reporte del generador de hormigas en el nodo 10.11.128.190 y 10.11.128.211

<p>GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.128.190</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.191 => 85 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.194 => 19 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.197 => 30 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.200 => 19 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.201 => 19 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.204 => 74 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.206 => 52 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.211 => 173 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.212 => 52 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.213 => 85 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.214 => 63 hormigas</p>	<p>GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.128.211</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.190 => 96 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.191 => 19 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.194 => 63 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.197 => 74 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.200 => 30 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.201 => 30 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.204 => 41 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.206 => 52 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.212 => 52 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.213 => 85 hormigas</p> <p>Hormigas artificiales transmitidas al nodo: 10.11.128.214 => 129 hormigas</p>
---	---

Fuente: Elaboración Propia

Figura 4.7: Estadísticas del generador de hormigas artificiales en los nodos de red 10.11.128.190 y 10.11.128.211

De la **Figura 4.3** y la **Figura 4.5** se obtiene los tiempos tomados desde la IP de monitoreo (10.11.128.211) hacia los demás nodos de red usando el protocolo de red ICMP y el algoritmo propuesto. Dichos valores se observan en la **Tabla 4.1** en donde se puede observar la gran diferencia de los tiempos tomados para detectar la conectividad de los nodos de red, ya que para el algoritmo basado en ACO le toma menos de un milisegundo superando ampliamente al protocolo ICMP.

IP	Tiempo en ms	
	ICMP	ACO
10.11.128.190	380	0.1086
10.11.128.191	8	0.813
10.11.128.194	38	0.0333
10.11.128.197	8	0.036
10.11.128.200	1261	0.1199
10.11.128.201	402	0.1057
10.11.128.204	71	0.2768
10.11.128.206	556	0.3451
10.11.128.212	41	0.2022
10.11.128.213	2829	0.0667
10.11.128.214	357	0.0678

Fuente: Elaboración Propia

Tabla 4.1: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 1

Los reportes completos se pueden ver en los Anexos: ANEXO 1 (Reporte Estadístico de los nodos de Red) y ANEXO 2 (Reporte Estadístico de los Generadores de Hormigas Artificiales). En ellos se puede ver detalladamente los valores estadísticos para cada nodo de red para la topología planteada. Cabe señalar que dicho reporte lo genera la aplicación desarrollada cuando se ejecuta el algoritmo propuesto en tiempo real para cada ejecución del algoritmo basado en ACO para cualquier topología de red.

4.2.- Prueba 2

Se realiza una segunda prueba obteniendo los resultados mostrados en la **Figura 4.8**. En ella se puede observar que se usa una topología donde no estén conectados todos los nodos entre ellos. Para esta prueba se realizará la topología mostrada en la **Figura 4.9**. En la **Figura 4.10** se muestra los tiempos tomados desde la IP del host de monitoreo (para este caso 10.11.11.80) hacia los demás nodos de red.

Fuente: Elaboración Propia

Figura 4.8: Resultados de la Prueba 2

Al igual que en las pruebas en la Topología de Red 1, se genera un reporte en las que se muestra una lista de los nodos vecinos a un servidor. En la **Figura 4.11** se observa que para el nodo 10.11.11.80 se encuentra los nodos vecinos 10.11.11.81 y 10.11.11.84 la cual se puede corroborar en la **Figura 4.9** la cual se muestra la Topología Grafica. Del mismo modo se observa los nodos vecinos para la IP 10.11.11.81 en la **Figura 4.10**

Fuente: Elaboración Propia

Figura 4.9: Topología de Red 2

IP	Time ICMP
10.11.11.80	0 ms
10.11.11.81	1 ms
10.11.11.82	2 ms
10.11.11.83	1 ms
10.11.11.84	1 ms
10.11.11.85	2 ms
10.11.11.86	1 ms
10.11.11.87	2 ms

IP del host de monitoreo

Fuente: Elaboración Propia

Figura 4.10: Tiempos desde la IP del Host de monitoreo hacia los demás nodos de red

ESTADÍSTICAS DE LOS NODOS DE RED	ESTADÍSTICAS DE LOS NODOS DE RED
ESTADÍSTICAS DE: 10.11.11.80 Lista de IP conectadas al Nodo de Red: 10.11.11.81 10.11.11.84	ESTADÍSTICAS DE: 10.11.11.81 Lista de IP conectadas al Nodo de Red: 10.11.11.80 10.11.11.82 10.11.11.84 10.11.11.85

Fuente: Elaboración Propia

Figura 4.11: Nodos vecinos para las IPs 10.11.11.80 y 10.11.11.81

En el reporte de los tiempos tomados por las hormigas artificiales en busca de un nodo de red desde los nodos 10.11.11.80 (izquierda) y 10.11.11.81 (derecha) se muestran en la **Figura 4.12**. En dichos valores se puede observar las hormigas artificiales generadas en el nodo 10.11.11.80 ninguna logro llegar al nodo de red 10.11.11.87. Esto se debe a que cada hormiga tiene un tiempo máximo de vida. Si al pasar dicho tiempo no ha retornado es eliminada para así no generar datos “basuras” en la red. Es por ello que en la **Figura 4.13** que se muestra los cálculos de los *throughputs* desde el nodo de origen 10.11.11.80 hacia el nodo 10.11.11.87 da el valor de “infinity”.

El reporte de los generadores de hormigas artificiales para los nodos de red 10.11.11.80 y 10.11.11.81 se muestra en la **Figura 4.14**.

El reporte completo generado para la Topología de Red 2 se encuentra en el ANEXO 3 (Reporte Estadístico de los nodos de Red) y en el ANEXO 4 (Reporte Estadístico de los Generadores de Hormigas Artificiales).

<p>Tiempo de viaje para el destino: 10.11.11.81</p> <p>Mejor tiempo [ms]: 0.033333333333332</p> <p>Variación del tiempo de viaje [ms]: 2.54054233788981E-33</p> <p>Tiempo promedio de viaje [ms]: 0.033333333333333</p>	<p>Tiempo de viaje para el destino: 10.11.11.80</p> <p>Mejor tiempo [ms]: 0.0333333333333314</p> <p>Variación del tiempo de viaje [ms]: 4.06891841645586E-05</p> <p>Tiempo promedio de viaje [ms]: 0.035970068013333</p>
<p>Tiempo de viaje para el destino: 10.11.11.82</p> <p>Mejor tiempo [ms]: 0.0666666666666629</p> <p>Variación del tiempo de viaje [ms]: 1.01881694058085E-30</p> <p>Tiempo promedio de viaje [ms]: 0.066666666666661</p>	<p>Tiempo de viaje para el destino: 10.11.11.82</p> <p>Mejor tiempo [ms]: 0.0333333333333314</p> <p>Variación del tiempo de viaje [ms]: 4.77757091365125E-05</p> <p>Tiempo promedio de viaje [ms]: 0.0378335151708419</p>
<p>Tiempo de viaje para el destino: 10.11.11.83</p> <p>Mejor tiempo [ms]: 0.166666666666666</p> <p>Variación del tiempo de viaje [ms]: 3.60000000001456E-06</p> <p>Tiempo promedio de viaje [ms]: 0.167333333333334</p>	<p>Tiempo de viaje para el destino: 10.11.11.83</p> <p>Mejor tiempo [ms]: 0.133333333333333</p> <p>Variación del tiempo de viaje [ms]: 2.97756000000968E-06</p> <p>Tiempo promedio de viaje [ms]: 0.133873333333338</p>
<p>Tiempo de viaje para el destino: 10.11.11.84</p> <p>Mejor tiempo [ms]: 0.033333333333332</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 0.033333333333332</p>	<p>Tiempo de viaje para el destino: 10.11.11.84</p> <p>Mejor tiempo [ms]: 0.066666666666664</p> <p>Variación del tiempo de viaje [ms]: 2.76300226238413E-05</p> <p>Tiempo promedio de viaje [ms]: 0.0696259866666665</p>
<p>Tiempo de viaje para el destino: 10.11.11.85</p> <p>Mejor tiempo [ms]: 0.0666666666666629</p> <p>Variación del tiempo de viaje [ms]: 0.00363362856815258</p> <p>Tiempo promedio de viaje [ms]: 0.113157125346678</p>	<p>Tiempo de viaje para el destino: 10.11.11.85</p> <p>Mejor tiempo [ms]: 0.0333333333333314</p> <p>Variación del tiempo de viaje [ms]: 0.000145394782593755</p> <p>Tiempo promedio de viaje [ms]: 0.0389375784319326</p>
<p>Tiempo de viaje para el destino: 10.11.11.86</p> <p>Mejor tiempo [ms]: 0.166666666666657</p> <p>Variación del tiempo de viaje [ms]: 1.02236373316643E-28</p> <p>Tiempo promedio de viaje [ms]: 0.166666666666661</p>	<p>Tiempo de viaje para el destino: 10.11.11.86</p> <p>Mejor tiempo [ms]: 0.133333333333326</p> <p>Variación del tiempo de viaje [ms]: 1.43796634965645E-28</p> <p>Tiempo promedio de viaje [ms]: 0.133333333333333</p>
<p>Tiempo de viaje para el destino: 10.11.11.87</p> <p>Mejor tiempo [ms]: 1.79769313486232E+308</p> <p>Variación del tiempo de viaje [ms]: 0</p> <p>Tiempo promedio de viaje [ms]: 1.79769313486232E+308</p>	<p>Tiempo de viaje para el destino: 10.11.11.87</p> <p>Mejor tiempo [ms]: 0.09999999999999943</p> <p>Variación del tiempo de viaje [ms]: 2.18181744000318E-05</p> <p>Tiempo promedio de viaje [ms]: 0.102171999999999</p>

Fuente: Elaboración Propia

Figura 4.12: Estadísticas de tiempos 10.11.11.80 (izquierda) y 10.11.11.81 (derecha)

Desde el nodo de origen: 10.11.11.80 se obtiene los throughputs:	Desde el nodo de origen: 10.11.11.81 se obtiene los throughputs:
Nodo destino 10.11.11.81 => 166.666666666667 [fps]	Nodo destino 10.11.11.80 => 719.401360266659 [fps]
Nodo destino 10.11.11.82 => 533.333333333329 [fps]	Nodo destino 10.11.11.82 => 1475.50709166284 [fps]
Nodo destino 10.11.11.83 => 1171.33333333334 [fps]	Nodo destino 10.11.11.83 => 2945.21333333344 [fps]
Nodo destino 10.11.11.84 => 1666.66666666666 [fps]	Nodo destino 10.11.11.84 => 2576.16150666666 [fps]
Nodo destino 10.11.11.85 => 4073.6565124804 [fps]	Nodo destino 10.11.11.85 => 1791.1286078689 [fps]
Nodo destino 10.11.11.86 => 1999.99999999993 [fps]	Nodo destino 10.11.11.86 => 4666.66666666667 [fps]
Nodo destino 10.11.11.87 => Infinity [fps]	Nodo destino 10.11.11.87 => 1736.92399999997 [fps]

Fuente: Elaboración Propia

Figura 4.13: Estadísticas de los *throughputs* de los nodos 10.11.11.80 y 10.11.11.81

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.80	GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.81
Hormigas artificiales transmitidas al nodo: 10.11.11.81 => 19 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.80 => 30 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.82 => 30 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.82 => 96 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.83 => 52 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.83 => 85 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.84 => 151 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.84 => 63 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.85 => 140 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.85 => 85 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.86 => 96 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.86 => 96 hormigas
Hormigas artificiales transmitidas al nodo: 10.11.11.87 => 19 hormigas	Hormigas artificiales transmitidas al nodo: 10.11.11.87 => 52 hormigas

Fuente: Elaboración Propia

Figura 4.14: Estadísticas del generador de hormigas en los nodos 10.11.11.80 y 10.11.11.81

De la **Figura 4.10** y la **Figura 4.12** se obtiene los tiempos tomados desde la IP de monitoreo (10.11.11.80) hacia los demás nodos de red usando el protocolo de red ICMP y el algoritmo propuesto. Dichos valores se observan en el **Tabla 4.2** en donde se puede observar la gran diferencia de los tiempos tomados para detectar la conectividad de los nodos de red, ya que para el algoritmo basado en ACO le toma menos de un milisegundo superando ampliamente al protocolo ICMP, pero aun así los tiempos obtenidos por el protocolo ICMP en una red cableada es menor que la obtenida en una red de WiFi

IP	Tiempo en ms	
	ICMP	ACO
10.11.11.81	1	0.0333
10.11.11.82	2	0.0667
10.11.11.83	1	0.1673
10.11.11.84	1	0.0333
10.11.11.85	2	0.1132
10.11.11.86	1	0.1667
10.11.11.87	2	-

Fuente: Elaboración Propia

Tabla 4.2: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 2

Con esta misma topología se apagará al propósito un nodo de red y se verá el comportamiento del algoritmo frente a este cambio. Para ello, se elige aleatoriamente desconectar el nodo de red 10.11.11.81 el cual es detectado inmediatamente por el escaneo de red y este a su vez por el algoritmo basado en ACO. En la **Figura 4.15** se observa como el aplicativo propuesto detecta dicho cambio de red poniendo de color rojo el nodo que fue desconectado (ver gráfico del lado superior derecho de la **Figura 4.15**).

Fuente: Elaboración Propia

Figura 4.15: Topología de Red 2 con el nodo 10.11.11.81 desconectado

El algoritmo basado en ACO inmediatamente actualiza su tabla de rutas y de sus nodos vecinos. Esto se puede validar en la **Figura 4.16** en la cual se muestra como los nodos vecinos al host desconectado ya sacaron de su lista de nodos vecinos a la IP 10.11.11.81.

Fuente: Elaboración Propia

Figura 4.16: Actualización de los nodos de red frente a un cambio en la topología de Red

Analizando el grafico de la **Figura 4.15** se observa que la IP 10.11.11.80 solo tiene comunicación con la IP 10.11.11.84. El cual es reflejado en el reporte estadístico creado por el algoritmo basado en ACO pues en los resultados del cálculo del *Throughput* solo se tiene resultados para la comunicación con la IP 10.11.11.84 tal como se puede apreciar en la **Figura 4.17**.

Fuente: Elaboración Propia

Figura 4.17: *Throughput* de la IP 10.11.11.80

4.3.- Prueba 3

En estas pruebas se utilizará una nueva topología de red, pero también con los 8 nodos de red planteados al inicio. A esta topología se la llamará Topología de Red 3 y se le puede observar en la **Figura 4.18**.

Fuente: Elaboración Propia

Figura 4.18: Topología de Red 3

Al igual que en la prueba 2 se presenta el reporte de los tiempos para las IPs 10.11.11.80 (izquierda) y 10.11.11.81 (derecha)

<div>Tiempo de viaje para el destino: 10.11.11.81 Mejor tiempo [ms]: 0.0666666666666664 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 0.0666666666666664</div> <div>Tiempo de viaje para el destino: 10.11.11.82 Mejor tiempo [ms]: 0.09999999999999943 Variación del tiempo de viaje [ms]: 8.0777550912899E-29 Tiempo promedio de viaje [ms]: 0.1</div> <div>Tiempo de viaje para el destino: 10.11.11.83 Mejor tiempo [ms]: 1.79769313486232E+308 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 1.79769313486232E+308</div> <div>Tiempo de viaje para el destino: 10.11.11.84 Mejor tiempo [ms]: 0.0666666666666664 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 0.0666666666666664</div> <div>Tiempo de viaje para el destino: 10.11.11.85 Mejor tiempo [ms]: 0.03333333333333314 Variación del tiempo de viaje [ms]: 2.54704235145212E-31 Tiempo promedio de viaje [ms]: 0.033333333333333</div> <div>Tiempo de viaje para el destino: 10.11.11.86 Mejor tiempo [ms]: 0.1666666666666657 Variación del tiempo de viaje [ms]: 3.98418341627184E-05 Tiempo promedio de viaje [ms]: 0.173752546666666</div> <div>Tiempo de viaje para el destino: 10.11.11.87 Mejor tiempo [ms]: 1.79769313486232E+308 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 1.79769313486232E+308</div>	<div>Tiempo de viaje para el destino: 10.11.11.80 Mejor tiempo [ms]: 0.1 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 0.1</div> <div>Tiempo de viaje para el destino: 10.11.11.82 Mejor tiempo [ms]: 0.03333333333333314 Variación del tiempo de viaje [ms]: 8.60179901708199E-30 Tiempo promedio de viaje [ms]: 0.03333333333333353</div> <div>Tiempo de viaje para el destino: 10.11.11.83 Mejor tiempo [ms]: 0.0666666666666664 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 0.0666666666666664</div> <div>Tiempo de viaje para el destino: 10.11.11.84 Mejor tiempo [ms]: 0.06666666666666629 Variación del tiempo de viaje [ms]: 3.59491587414114E-29 Tiempo promedio de viaje [ms]: 0.06666666666666732</div> <div>Tiempo de viaje para el destino: 10.11.11.85 Mejor tiempo [ms]: 0.03333333333333314 Variación del tiempo de viaje [ms]: 1.1149192215596E-29 Tiempo promedio de viaje [ms]: 0.03333333333333345</div> <div>Tiempo de viaje para el destino: 10.11.11.86 Mejor tiempo [ms]: 0.09999999999999943 Variación del tiempo de viaje [ms]: 9.88854377992718E-29 Tiempo promedio de viaje [ms]: 0.1000000000000003</div> <div>Tiempo de viaje para el destino: 10.11.11.87 Mejor tiempo [ms]: 1.79769313486232E+308 Variación del tiempo de viaje [ms]: 0 Tiempo promedio de viaje [ms]: 1.79769313486232E+308</div>
--	---

Fuente: Elaboración Propia

Figura 4.19: Estadísticas de tiempos 10.11.11.80 (izquierda) y 10.11.11.81 (derecha)

Los nodos vecinos para dichas IPs mencionadas se muestra en la **Figura 4.20** la cual se puede corroborar con la topología de Red mostrada en la **Figura 4.18**.

Fuente: Elaboración Propia

Figura 4.20: Nodos vecinos para las IPs 10.11.11.80 y 10.11.11.81

Los valores de *Throughput* para los nodos de red 10.11.11.80 y 10.11.11.81 se muestran en la **Figura 4.21**.

Fuente: Elaboración Propia

Figura 4.21: Estadísticas de los *throughputs* de los nodos 10.11.11.80 y 10.11.11.81

De la **Figura 4.18** y la **Figura 4.19** se hace una comparación de los tiempos obtenidos desde el nodo de red 10.11.11.80 (Nodo monitor) hacia las demás IPs. Esta comparación se muestra en el **Tabla 4.3** y se toman los tiempos promedios tomados por las hormigas artificiales. En dicho cuadro se puede observar las mejoras que se ha obtenido con el algoritmo basado en ACO.

IP	Tiempo en ms	
	ICMP	ACO
10.11.11.81	1	0.0667
10.11.11.82	4	0.1
10.11.11.83	3	0.1797
10.11.11.84	1	0.0667
10.11.11.85	3	0.0333
10.11.11.86	2	0.1738
10.11.11.87	1	-

Fuente: Elaboración Propia

Tabla 4.3: Comparación de tiempos entre el protocolo de red ICMP y el propuesto para la Topología de Red 3

El reporte completo generado para la Topología de Red 3 se encuentra en el ANEXO 5 (Reporte Estadístico de los nodos de Red) y en el ANEXO 6 (Reporte Estadístico de los Generadores de Hormigas Artificiales).

4.4.- Resultados Comparativos

Se realiza un cuadro comparativo de las pruebas realizadas para la Topología de Red 2 y la Topología de Red 3. Los resultados son mostrados en la **Figura 4.22**. También se realizó un cuadro comparativo en donde solamente están incluidos los valores obtenidos con el algoritmo propuesto y sin el algoritmo basado en ACO los cuales se muestran en la **Figura 4.23** y la **Figura 4.24** respectivamente. Tener en cuenta que dichos valores son los obtenidos al detectar las IPs conectadas en un segmento de red específico.

Fuente: Elaboración Propia

Figura 4.22: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red para la topología de Red 2 y Red 3

De los valores mostrados en la **Figura 4.22**, **Figura 4.23** y **Figura 4.24** se han obtenido el porcentaje de mejora logrado cuando se usa el algoritmo basado en ACO. Este

porcentaje está por encima del 90% demostrando así la mejora del algoritmo; incluso, de la **Figura 4.22** y la **Figura 4.23** se puede deducir que la mejora es mayor con la Prueba 1 que fue hecha en una conexión por WiFi. También, se puede observar que los tiempos promedios para la topología de Red 3 aumenta con respecto a los tiempos obtenidos en la topología de Red 2; sin embargo, el incremento es en menor proporción cuando se usa el algoritmo basado en ACO (tener en cuenta que dicha topología tienen la misma cantidad de nodos). Agregado a ello, se observa que los tiempos obtenidos con una topología de red conectadas por WiFi aumenta drásticamente pero aun así la proporción de aumento es menor con el algoritmo basado en ACO.

Fuente: Elaboración Propia

Figura 4.23: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red con el algoritmo basado en ACO para la topología de Red 1, Red 2 y Red

Fuente: Elaboración Propia

Figura 4.24: Cuadro comparativo de tiempos y porcentajes promedios de detección de nodos de red sin el algoritmo basado en ACO para la topología de Red 1, Red 2 y Red 3

4.5.- Costo de la Implementación

Los costos de implementación usada para realizar estas pruebas se presentan en la **Tabla 4.4** el cual llega a ser un total de S/. 38,854.00; sin embargo, sólo fue costado por los integrantes de este proyecto S/. 50, ya que el resto lo proporciono la Universidad Peruana de Ciencias Aplicadas (UPC) y material de trabajo del alumno.

Item	Cant.	Costo	Costo Total	Dispuesto UPC	Dispuesto Alumno	Inversion
Computadora Dell Precision T1700	8	4,099.00	32,796.00	-32,796.00		0.00
Patch Panel AMP Category 6	1	150.00	150.00	-150.00		0.00
HP Envy 15-q002la	1	3,799.00	3,799.00		-3,799.00	0.00
Toshiba Satellite C45-ASP4310FL	1	1,999.00	1,999.00		-1,999.00	0.00
CentOS v7.0	1	0.00	0.00			0.00
Visual Studio 2015 - Estudiantil	1	0.00	0.00			0.00
Cableado estructurado para PCs	8	6.25	50.00			50.00
Switch D-Link 8 puertos	1	60.00	60.00	-60.00		0.00
					Total	S/.50.00

Fuente: Elaboración Propia

Tabla 4.4: Costos Reales

Conclusiones y comentarios finales

- A lo largo de los años, la cantidad de usuarios se ha incrementado exponencialmente; sin embargo, la infraestructura de las telecomunicaciones no lo ha hecho en la misma proporción.
- Se obtuvo información de proyectos relacionados con la optimización por algoritmo de hormigas, los cuales demuestran mejoras significativas con el uso del ACO.
- Se comprueba falencias en los algoritmos clásicos de enrutamiento, los cuales no operan con precisión en ambientes con cambios continuos de topología.
- Las redes IP son dinámicas, ya que presentan diversos tipos de cambios en su topología de red y tráfico de datos. Por ello, se implementó un algoritmo de encaminamiento con mayor flexibilidad a estos tipos de cambios como lo es el ACO.
- Este proyecto se implementó en una red LAN.
- El tamaño del paquete puede incrementar los retardos de cola, ya que tomará más tiempo en procesarlo. Por ello, el tamaño del paquete transmitido influye en el tiempo de llegada al host final. Dicho valor de tiempo se está tomando en cuenta para medir el desempeño del algoritmo propuesto.
- Los algoritmos de los protocolos de red clásicos se basan en buscar el camino más corto, pero no toman en cuenta el tráfico de datos; es decir, no buscan mejorar el *throughput*.
- Existen dos tipos de hormigas artificiales: Forward y Backward, pero sólo la hormiga Backward es la que deja la feromona artificial en el camino que recorre.
- Inicialmente las hormigas Forward son enviadas aleatoriamente hacia un nodo destino. Sin embargo, la probabilidad de elección de una ruta inicial está dada por la ecuación (1) descrita en el Capítulo 2. A partir de la segunda interacción la probabilidad que hormiga Forward elija un camino está basado según la ecuación (2) del Capítulo 2.
- La actualización de la feromona artificial se basa en la ecuación (3) del Capítulo 2.

- El aplicativo primero realiza un escaneo en un segmento de red y actualiza su tabla de “SERVIDORES ONLINE”. De esa información tabla se genera la topología y la ejecución del algoritmo basado en ACO
- En el algoritmo se puede programar una determinada topología y esta a su vez actualiza el grafico y el algoritmo basado en ACO
- El aplicativo trabajo en paralelo el escaneo y la generación del algoritmo ACO. Es decir, va escaneando mientras el ACO se inicializa al mismo tiempo. Gracias a ello el ACO se puede actualizar en tiempo real
- Para la primera ejecución del ACO se necesita que se haga por lo menos un escaneo completo del segmento de red ingresado
- El escaneo es mucho más rápido cuando se hace en una red con Host conectados por cable UTP que por WiFi.
- Para efectos de análisis de los resultados generados por el algoritmo propuesto se creó un botón que haga que se ejecute dicho algoritmo cuando uno lo desee.
- El reporte muestra las IPs conectadas a cada host y los tiempos tomados para detectar cada uno de ellos. En ese reporte de tiempos se muestra el mejor tiempo obtenido y la diferencia entre el mejor tiempo y el peor tiempo; también, muestra un promedio de todos los tiempos. Para los cálculos de esos tiempos se realiza basado en el tiempo que demora cada hormiga artificial en ir y regresar a cada nodo de origen.
- El reporte también muestra los valores de *throughputs* generados para cada nodo de origen.
- El algoritmo se adapta según las IPs conectadas a la red. Es decir, si en tiempo real se agrega o quita un host la tabla “Servidores ONLINE” se actualiza y este a su vez actualiza la topología de Red. Por ello, el ACO también, se actualizará los reportes generados por el algoritmo basado en ACO.
- El tiempo de actualización de la tabla “Servidores ONLINE” tiene un retardo, ya que se tiene que esperar que se haga un escaneo completo del segmento de red especificado. En las pruebas realizadas, se observó que el retardo es mucho menor cuando están conectados los hosts por medio de cable UTP que cuando están conectados por medio de WiFi. En este último, el retardo puede tomar varios segundos dependiendo del tamaño del segmento de red a escanear.

- Las pruebas mostradas en este informe se realizaron en una conexión WiFi (Prueba 1) y conexión por cable UTP (Prueba 2 y Prueba 3). De las cuales se observa que el tiempo para detectar las IPs es mucho menor cuando los hosts están conectados con cable UTP. Esto se puede validar con valores de tiempos mostrados en la Figura 4.23 en la cual se verifica que con el algoritmo propuesto el tiempo promedio para detectar un nodo de red con una conexión WiFi es de 0.198 milisegundos y en una conexión por cable UTP es de 0.097 y 0.103 milisegundos para las pruebas 2 y 3 respectivamente.
- El porcentaje de mejora para la detección de nodos de red con el algoritmo propuesto es mayor al 90% llegando a tener un 98% de mejora en la Prueba 1.
- En la Prueba 2 se hace cambios en la topología de red para demostrar adaptabilidad del algoritmo propuesto frente a cambios en la red.
- En cada nodo generador de hormigas artificiales se envía en periodos de tiempo hormigas aleatoriamente en busca de los nodos vecinos. Estas regresan con la información del nodo encontrado y el tiempo que les llevo llegar al nodo y retornar; sin embargo, existen hormigas que no regresan ya que cada una de ellas tienen un periodo máximo de vida antes de desaparecer. Este caso sucede en la Topología de Red 2 y Topología de Red 3 el cual se puede validar en el reporte generado mostrado en el ANEXO 3 y ANEXO 5
- El algoritmo registra la cantidad de hormigas artificiales generadas por cada nodo y a su vez la cantidad de hormigas que retornan (bits transmitidos correctamente) con el tiempo que les ha tomado todo su viaje (tiempo utilizado para transmitir los bits correctos). Con dicha información se logra medir la cantidad de datos transmitidos correctamente en una unidad de tiempo (*throughput*).
- El escaneo de las redes conectadas en toda la red lo realiza enviando un mensaje ICMP IP por IP y si hay respuesta es debido a que dicho nodo de red se encuentra conectado en la red. Sin embargo, esto genera un retardo ya que cuando una IP no está conectada se queda esperando un determinado tiempo que responda (ese tiempo de espera es programable por el algoritmo de escaneo propuesto). Pasado ese tiempo sino hay respuesta el algoritmo asume que no hay conexión hacia dicho nodo de red.
- A diferencia del escaneo, el algoritmo propuesto realiza en paralelo el envío de hormigas artificiales en cada nodo de red para así detectar que IPs se encuentran

conectadas de una forma más eficiente. Agregado a ello, dicho algoritmo detecta la topología ya que da un reporte de que IPs están conectadas a cada nodo de red; es decir, da un reporte de vecindad por nodo de red.

- El algoritmo propuesto ha tenido un correcto desempeño para encontrar rutas de encaminamiento de una manera eficiente ya que se tuvo resultados favorables
- Con el algoritmo propuesto se puede diseñar una topología de red virtual. Es decir, para una misma conexión física se puede diseñar diversas topologías de red.
- Por limitaciones de dispositivos de red se realizaron las pruebas en un ambiente de 8 hosts. Sin embargo, el algoritmo soporta una red mucho mayor pudiendo llegar hasta los 255 nodos de red.
- Por temas de demostración del algoritmo basado en ACO se ejecuta mediante un botón. De no ser así no se podría analizar los reportes estadísticos generados por el algoritmo.

Recomendaciones para trabajos futuros

- Sería ideal poder probar el comportamiento de este algoritmo en redes de datos más grandes. Para ello, se debe tener en cuenta que todos los hosts deben estar conectados en una misma red interna; es decir, en el caso de una empresa de tener varias sedes, estas deben estar interconectadas en una misma red mediante VPNs u otros medios. Sin embargo, dentro de las redes grandes normalmente hay usuarios que se conectan y desconectan de dicha red constantemente sin ser esto necesariamente un problema. Es por eso que se debe configurar el algoritmo de monitoreo para que cuando un grupo de IPs específicos (por ejemplo, las IPs de los servidores) se desconecten se muestre una alarma y no con el resto de IPs. En cambio, por el lado del algoritmo basado en ACO su funcionamiento es adaptable al ingreso o salida de cualquier host en la red de datos, ya que lo que se busca con este algoritmo es mejorar el encaminamiento de datos y su adaptabilidad a estos cambios constantes.
- Se puede adaptar el algoritmo para transmisiones en tiempo real de video y de esta forma ver el comportamiento de este tipo de algoritmos de encaminamiento.

Referencias bibliográficas

- [1].- RODRIGUEZ GARCIA, Jesus (2010) Análisis de algoritmos basados en colonia de hormigas problemas de camino mínimo (Proyecto de fin de Carrera de Ingeniería Informatica). Madrid: Universidad Carlos III de Madrid
- [2].- ALBA, Enrique y CHICANO, Francisco (2003). Una nueva Versión de ACO para Problemas con Grafos de muy Gran Extensión. España
- [3].- DI CARO, Gianni y DORIGO, MARCO (1998) AntNet: Distributed Stigmergetic Control for Communications Networks. Belgica: Université Libre de Bruxelles
- [4].- DOMINGUEZ MEDINA, Christian Horacio (2011) Algoritmos bioinspirados para el encaminamiento de datos en redes inalámbricas de sensors (Tesis de Maestria en Ciencias de la Computación). Mexico: Instituto Politecnico Nacional Centro de Investigación en Computación
- [5].- ALEJANDRO ARITO, Franco Luis (2010) Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados (Trabajo final para alcanzar el grado de Licenciado en Ciencias de la Computación). Argentina : Universidad Nacional de San Luis, Facultad de Ciencias Físico Matemáticas y Naturales
- [6].- APARICIO GUIRAO, Daniel (2012) Aplicación de los algoritmos de hormigas para la resolución de un problema de equilibrado de líneas de montaje con robotizados (Tesis de final de Carrera). España: Universidad Politecnica de Cataluña
- [7].- CISCO NETWORKING ACADEMY (2012) CCNA Exploration 4.0, Conceptos y Protocolos de enrutamiento (<https://www.netacad.com/es/courses/ccna/>)

[8].- CISCO NETWORKING ACADEMY (2012) CCNA Exploration 4.0, Aspectos Básicos de Networking (<https://www.netacad.com/es/courses/ccna/>)

[9].- IMAS RODRIGUEZ, Arturo (2013). Algoritmos Inspirados en Swarm Intelligence para el Enrutamiento en Redes de Telecomunicaciones (Tesis de Master Universitario en Inteligencia Artificial). Madrid: Universidad Politécnica de Madrid

[10].- FOURNIER, Joseph y PIERRE, Samuel (2003) Assigning cells to switches in mobile networks using an ant colony optimization heuristic. Canada: Ecole Polytechnique of Montreal

[11].- VERSTRAETE, Vicent y otros AntNet: ACO routing algorithm in practice. Bélgica: Ghent University

[12].- DEPARTMENT OF COMPUTER SCIENCE (2010) (<http://www.cs.fsu.edu/>). Sitio web oficial; centro para la investigación en ciencias de la computación, análisis numérico, la biología computacional, computación cuántica, la lingüística computacional y sistemas de información (consulta: 18 de Setiembre del 2016)

[13].- INSTITUTO NACIONAL DE ESTADISTICAS E INFORMATICA (INEI) (2016) (<https://www.inei.gob.pe/estadisticas/indice-tematico/transport-and-communications/>) Sitio web oficial del INEI (consulta: 01 de Agosto del 2016)

[14].- MINISTERIO DE TRANSPORTES Y COMUNICACIONES (MTC) (2016) (https://www.mtc.gob.pe/portal/proyecto_banda_ancha/TELMEX%20prob%20instalacion%20redes.pdf) Sitio web oficial del MTC (consulta: 01 de Agosto del 2016)

[15].- CAMARAS DE SEGURIDAD PANAMÁ (2016). En: Blog de Noticias de Seguridad Panamá (Consulta: 19 de Octubre del 2016) (<http://camarasdeseguridadpanama.blogspot.pe/2016/08/conoces-la-diferencia-de-la-red-lan-y.html>)

[16].- FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA DE LA UNIVERSIDAD NACIONAL DEL NORDESTE (UNNE) (2016) (http://exa.unne.edu.ar/depar/areas/informatica/teleproc/Comunicaciones/Presentaciones_Proyector/Conm-Paquetes.pdf) Sitio web oficial (Consulta: 19 de Octubre del 2016)

[17].- KUROSE, James F. y ROSS, Keith (2010) Redes de Computadoras: Un Enfoque Descendente 5a ed. Madrid: Pearson Education S.A.

[18].- HALBERG, Bruce A. (2010) Fundamentos de Redes 4a ed. Mexico: McGRAW-HILL Interamericana Editores SA

[19].-REAL ACADEMIA DE INGENIERÍA (RAI) (2016) (<http://diccionario.raing.es/es/lema/throughput>) Sitio web oficial de la RAI (consulta: 20 de Noviembre del 2016)

[20].- R. P. PEÑABAENA NIEBLES, «Diseño y optimización de un modelo matemático,» 2015. [En línea]. Available: <http://repositorio.unican.es/xmlui/bitstream/handle/10902/8111/TesisRPN.pdf?sequence=1&isAllowed=y>. [Último acceso: 23 Setiembre 2016].

[21].- J. S. ARIAS ROJAS, «APLICACIÓN DE UN MODELO DE OPTIMIZACIÓN EN LA PLANEACIÓN DE RUTAS DE LOS BUSES ESCOLARES DEL COLEGIO LICEO DE CERVANTES NORTE,» 2010. [En línea]. Available: <http://repository.javeriana.edu.co/bitstream/handle/10554/7367/tesis403.pdf;jsessionid=017586B00F283C2B73AE9974D04CDC62?sequence=1>. [Último acceso: 22 09 2016].

Anexos

Anexo 1: Reporte Estadístico de los nodos de Red - Topología de Red 1

ESTADÍSTICAS DE: 10.11.128.190

Lista de IP conectadas al Nodo de Red:

10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.191
Mejor tiempo [ms]: 0.19999999999999999
Variación del tiempo de viaje [ms]: 3.72860037233369E-32
Tiempo promedio de viaje [ms]: 0.2

Tiempo de viaje para el destino: 10.11.128.194
Mejor tiempo [ms]: 0.09999999999999999
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.09999999999999999

Tiempo de viaje para el destino: 10.11.128.197
Mejor tiempo [ms]: 0.066666666666666629
Variación del tiempo de viaje [ms]: 1.01881694058085E-30
Tiempo promedio de viaje [ms]: 0.06666666666666661

Tiempo de viaje para el destino: 10.11.128.200
Mejor tiempo [ms]: 0.03333333333333314
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.03333333333333314

Tiempo de viaje para el destino: 10.11.128.201
Mejor tiempo [ms]: 0.06666666666666629
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.06666666666666629

Tiempo de viaje para el destino: 10.11.128.204
Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 1.84580190554312E-06
Tiempo promedio de viaje [ms]: 0.133488966791793

Tiempo de viaje para el destino: 10.11.128.206
Mejor tiempo [ms]: 0.233333333333332
Variación del tiempo de viaje [ms]: 3.7946481753872E-05
Tiempo promedio de viaje [ms]: 0.238489474614531

Tiempo de viaje para el destino: 10.11.128.211
Mejor tiempo [ms]: 1
Variación del tiempo de viaje [ms]: 1.14039073997558E-06
Tiempo promedio de viaje [ms]: 0.0334290530650239

Tiempo de viaje para el destino: 10.11.128.212
Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 5.3843239425394E-06
Tiempo promedio de viaje [ms]: 0.133798237920129

Tiempo de viaje para el destino: 10.11.128.213
Mejor tiempo [ms]: 0.19999999999999999
Variación del tiempo de viaje [ms]: 4.525942600932117E-28
Tiempo promedio de viaje [ms]: 0.19999999999999998

Tiempo de viaje para el destino: 10.11.128.214
Mejor tiempo [ms]: 0.19999999999999999
Variación del tiempo de viaje [ms]: 4.52595897671992E-28
Tiempo promedio de viaje [ms]: 0.2000000000000013

Desde el nodo de origen: 10.11.128.190 se obtiene los throughputs:

Nodo destino 10.11.128.191 => 2200 [fps]
Nodo destino 10.11.128.194 => 199.999999999999 [fps]
Nodo destino 10.11.128.197 => 199.999999999999 [fps]
Nodo destino 10.11.128.200 => 66.6666666666629 [fps]
Nodo destino 10.11.128.201 => 266.666666666652 [fps]
Nodo destino 10.11.128.204 => 6941.42627317321 [fps]
Nodo destino 10.11.128.206 => 4054.32106844703 [fps]
Nodo destino 10.11.128.211 => 2373.4627676167 [fps]
Nodo destino 10.11.128.212 => 4549.14008928439 [fps]
Nodo destino 10.11.128.213 => 5399.999999999994 [fps]
Nodo destino 10.11.128.214 => 3400.00000000022 [fps]

ESTADÍSTICAS DE: 10.11.128.191

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190
Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 4.22609114190593E-29
Tiempo promedio de viaje [ms]: 0.133333333333333

Tiempo de viaje para el destino: 10.11.128.194
Mejor tiempo [ms]: 0.09999999999999943
Variación del tiempo de viaje [ms]: 2.84289238600921E-05
Tiempo promedio de viaje [ms]: 0.103081035443775

Tiempo de viaje para el destino: 10.11.128.197
Mejor tiempo [ms]: 0.0666666666666629
Variación del tiempo de viaje [ms]: 8.19000000000013E-05
Tiempo promedio de viaje [ms]: 0.0666666666666661

Tiempo de viaje para el destino: 10.11.128.200
Mejor tiempo [ms]: 0.0666666666666664
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.0666666666666664

Tiempo de viaje para el destino: 10.11.128.201
Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 4.07526776232339E-30
Tiempo promedio de viaje [ms]: 0.133333333333332

Tiempo de viaje para el destino: 10.11.128.204
Mejor tiempo [ms]: 1
Variación del tiempo de viaje [ms]: 1.97514265433744E-32
Tiempo promedio de viaje [ms]: 0.0333333333333315

Tiempo de viaje para el destino: 10.11.128.206
Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 6.59099212978212E-06
Tiempo promedio de viaje [ms]: 0.133907289613329

Tiempo de viaje para el destino: 10.11.128.211
Mejor tiempo [ms]: 0.09999999999999943
Variación del tiempo de viaje [ms]: 2.46239999999946E-05
Tiempo promedio de viaje [ms]: 0.1107999999999993

Tiempo de viaje para el destino: 10.11.128.212
Mejor tiempo [ms]: 0.19999999999999999
Variación del tiempo de viaje [ms]: 5.55622155986812E-07
Tiempo promedio de viaje [ms]: 0.2001968299999995

Tiempo de viaje para el destino: 10.11.128.213
Mejor tiempo [ms]: 0.2666666666666652
Variación del tiempo de viaje [ms]: 1.86097499999956E-05
Tiempo promedio de viaje [ms]: 0.268016666666668

Tiempo de viaje para el destino: 10.11.128.214
Mejor tiempo [ms]: 0.2666666666666652
Variación del tiempo de viaje [ms]: 1.70502975001291E-05
Tiempo promedio de viaje [ms]: 0.2681516666666673

Desde el nodo de origen: 10.11.128.191 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 1066.66666666666 [fps]
Nodo destino 10.11.128.194 => 1649.2965671004 [fps]
Nodo destino 10.11.128.197 => 487.666666666662 [fps]
Nodo destino 10.11.128.200 => 399.999999999999 [fps]
Nodo destino 10.11.128.201 => 266.666666666664 [fps]
Nodo destino 10.11.128.204 => 3133.33333333316 [fps]
Nodo destino 10.11.128.206 => 4151.12597801321 [fps]
Nodo destino 10.11.128.211 => 664.799999999999 [fps]
Nodo destino 10.11.128.212 => 3403.34610999991 [fps]
Nodo destino 10.11.128.213 => 3752.23333333336 [fps]
Nodo destino 10.11.128.214 => 2145.21333333338 [fps]

ESTADÍSTICAS DE: 10.11.128.194

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.066666666666629

Variación del tiempo de viaje [ms]: 6.46209289127237E-05

Tiempo promedio de viaje [ms]: 0.0709238330949514

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 2.57534034827379E-32

Tiempo promedio de viaje [ms]: 0.0999999999999998

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.113333333333333

Variación del tiempo de viaje [ms]: 0.0010404

Tiempo promedio de viaje [ms]: 0.215333333333333

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.09999999999999943

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.066666666666629

Variación del tiempo de viaje [ms]: 3.16094399999993E-05

Tiempo promedio de viaje [ms]: 0.0763866666666618

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.066666666666629

Variación del tiempo de viaje [ms]: 7.19783494362446E-06

Tiempo promedio de viaje [ms]: 0.067309227941429

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.166666666666657

Variación del tiempo de viaje [ms]: 2.39131224179213E-05

Tiempo promedio de viaje [ms]: 0.169105531490226

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.133333333333326

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.166666666666657

Variación del tiempo de viaje [ms]: 1.64034547578445E-05

Tiempo promedio de viaje [ms]: 0.168305714432287

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.23333333333332

Variación del tiempo de viaje [ms]: 1.68963974999615E-05

Tiempo promedio de viaje [ms]: 0.234548333333337

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.23333333333332

Variación del tiempo de viaje [ms]: 1.55238909750875E-05

Tiempo promedio de viaje [ms]: 0.234669833333346

Desde el nodo de origen: 10.11.128.194 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 4964.6683166466 [fps]
Nodo destino 10.11.128.191 => 2999.99999999999 [fps]
Nodo destino 10.11.128.197 => 430.666666666666 [fps]
Nodo destino 10.11.128.200 => 299.999999999998 [fps]
Nodo destino 10.11.128.201 => 534.706666666663 [fps]
Nodo destino 10.11.128.204 => 5048.19209560717 [fps]
Nodo destino 10.11.128.206 => 4396.74381874588 [fps]
Nodo destino 10.11.128.211 => 399.999999999977 [fps]
Nodo destino 10.11.128.212 => 4880.86571853632 [fps]
Nodo destino 10.11.128.213 => 4690.96666666673 [fps]
Nodo destino 10.11.128.214 => 2581.36816666681 [fps]

ESTADÍSTICAS DE: 10.11.128.197

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 4.22981074938913E-30

Tiempo promedio de viaje [ms]: 0.0333333333333335

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.166666666666666

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.166666666666666

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.066666666666629

Variación del tiempo de viaje [ms]: 2.23425399511413E-30

Tiempo promedio de viaje [ms]: 0.0666666666666655

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 1.45754251167558E-05

Tiempo promedio de viaje [ms]: 0.0378482733333325

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 2.62384852963494E-06

Tiempo promedio de viaje [ms]: 0.033555695755994

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.199999999999999

Variación del tiempo de viaje [ms]: 1.02282415677933E-05

Tiempo promedio de viaje [ms]: 0.200915168462593

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.299999999999983

Variación del tiempo de viaje [ms]: 3.57285932640176E-05

Tiempo promedio de viaje [ms]: 0.308845200000004

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 0.00012896580019816

Tiempo promedio de viaje [ms]: 0.116159079999995

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.23333333333332

Variación del tiempo de viaje [ms]: 7.4438999999688E-07

Tiempo promedio de viaje [ms]: 0.233603333333323

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.299999999999983

Variación del tiempo de viaje [ms]: 0.000995040603899104

Tiempo promedio de viaje [ms]: 0.345926999999994

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.299999999999983

Variación del tiempo de viaje [ms]: 0.000269557559999282

Tiempo promedio de viaje [ms]: 0.329459999999993

Desde el nodo de origen: 10.11.128.197 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 266.666666666668 [fps]
Nodo destino 10.11.128.191 => 333.333333333332 [fps]
Nodo destino 10.11.128.194 => 399.999999999993 [fps]
Nodo destino 10.11.128.200 => 454.17927999999 [fps]
Nodo destino 10.11.128.201 => 1006.67087267982 [fps]
Nodo destino 10.11.128.204 => 6429.28539080298 [fps]
Nodo destino 10.11.128.206 => 2779.60680000003 [fps]
Nodo destino 10.11.128.211 => 929.272639999957 [fps]
Nodo destino 10.11.128.212 => 1868.82666666659 [fps]
Nodo destino 10.11.128.213 => 2421.48899999996 [fps]
Nodo destino 10.11.128.214 => 1976.75999999996 [fps]

ESTADÍSTICAS DE: 10.11.128.200

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.033333333333333314

Variación del tiempo de viaje [ms]: 2.82404696071975E-30

Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.033333333333333314

Variación del tiempo de viaje [ms]: 0.000103441904172101

Tiempo promedio de viaje [ms]: 0.0374786917821885

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.1400000000000001

Variación del tiempo de viaje [ms]: 0.000238822697184154

Tiempo promedio de viaje [ms]: 0.156325415999997

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0333333333333332

Variación del tiempo de viaje [ms]: 0.00086051483999999

Tiempo promedio de viaje [ms]: 0.0425133333333337

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0999999999999996

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.3266666666666666

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.3266666666666666

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.2666666666666652

Variación del tiempo de viaje [ms]: 3.93320605792078E-05

Tiempo promedio de viaje [ms]: 0.274194817466669

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.12578689911536E-05

Tiempo promedio de viaje [ms]: 0.0682271186148488

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.2666666666666652

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.2666666666666652

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 7.18758917811495E-07

Tiempo promedio de viaje [ms]: 0.100043721851736

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 1.49800084075085E-05

Tiempo promedio de viaje [ms]: 0.101393037206197

Desde el nodo de origen: 10.11.128.200 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 233.333333333333 [fps]
Nodo destino 10.11.128.191 => 1011.92467811909 [fps]
Nodo destino 10.11.128.194 => 1406.92874399998 [fps]
Nodo destino 10.11.128.197 => 255.080000000002 [fps]
Nodo destino 10.11.128.201 => 199.999999999999 [fps]
Nodo destino 10.11.128.204 => 653.333333333333 [fps]
Nodo destino 10.11.128.206 => 3564.53262706669 [fps]
Nodo destino 10.11.128.211 => 3479.58304935729 [fps]
Nodo destino 10.11.128.212 => 799.999999999995 [fps]
Nodo destino 10.11.128.213 => 5702.49214554894 [fps]
Nodo destino 10.11.128.214 => 4968.25882310366 [fps]

ESTADÍSTICAS DE: 10.11.128.201

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.033333333333333314

Variación del tiempo de viaje [ms]: 5.47502981714523E-07

Tiempo promedio de viaje [ms]: 0.0333791157842688

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.133333333333333

Variación del tiempo de viaje [ms]: 4.44763938372286E-32

Tiempo promedio de viaje [ms]: 0.133333333333333

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 2.57398025637256E-05

Tiempo promedio de viaje [ms]: 0.0675338801150474

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0666666666666664

Variación del tiempo de viaje [ms]: 0.001990476

Tiempo promedio de viaje [ms]: 0.1706

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.03333333333333314

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 1.62840220539114E-05

Tiempo promedio de viaje [ms]: 0.168200002386779

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 4.86340619309097E-06

Tiempo promedio de viaje [ms]: 0.100418414128116

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 0.000734281560000115

Tiempo promedio de viaje [ms]: 0.0847933333333311

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 1.70084939105561E-06

Tiempo promedio de viaje [ms]: 0.0334766104065648

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 5.85018068046202E-31

Tiempo promedio de viaje [ms]: 0.03333333333333315

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 1.14121461505386E-05

Tiempo promedio de viaje [ms]: 0.0343842868566511

Desde el nodo de origen: 10.11.128.201 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 2436.67545225163 [fps]
Nodo destino 10.11.128.191 => 3066.66666666666 [fps]
Nodo destino 10.11.128.194 => 1620.81312276114 [fps]
Nodo destino 10.11.128.197 => 511.8 [fps]
Nodo destino 10.11.128.200 => 266.666666666652 [fps]
Nodo destino 10.11.128.204 => 8073.6001145654 [fps]
Nodo destino 10.11.128.206 => 3615.06290861217 [fps]
Nodo destino 10.11.128.211 => 1271.89999999997 [fps]
Nodo destino 10.11.128.212 => 2209.45628683327 [fps]
Nodo destino 10.11.128.213 => 2299.99999999988 [fps]
Nodo destino 10.11.128.214 => 2269.36293253897 [fps]

ESTADÍSTICAS DE: 10.11.128.204

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.033333333333314

Variación del tiempo de viaje [ms]: 5.22492840182464E-30

Tiempo promedio de viaje [ms]: 0.033333333333334

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.066666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.066666666666664

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.246666666666666

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.246666666666666

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.16094400000183E-05

Tiempo promedio de viaje [ms]: 0.0763866666666657

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 1.43999999999968E-05

Tiempo promedio de viaje [ms]: 0.0679999999999961

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0999999999999943

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.1999999999999989

Variación del tiempo de viaje [ms]: 2.69676320306421E-05

Tiempo promedio de viaje [ms]: 0.202861353426617

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.166666666666663

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.166666666666663

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.23333333333337

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.23333333333337

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 1.28764070253301E-06

Tiempo promedio de viaje [ms]: 0.133411431785083

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 4.64597841738516E-06

Tiempo promedio de viaje [ms]: 0.134227627333334

Desde el nodo de origen: 10.11.128.204 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 766.666666666668 [fps]
Nodo destino 10.11.128.191 => 199.999999999999 [fps]
Nodo destino 10.11.128.194 => 493.333333333332 [fps]
Nodo destino 10.11.128.197 => 534.706666666666 [fps]
Nodo destino 10.11.128.200 => 203.999999999998 [fps]
Nodo destino 10.11.128.201 => 299.999999999998 [fps]
Nodo destino 10.11.128.206 => 4868.67248223881 [fps]
Nodo destino 10.11.128.211 => 333.333333333385 [fps]
Nodo destino 10.11.128.212 => 700.000000000109 [fps]
Nodo destino 10.11.128.213 => 6537.16015746907 [fps]
Nodo destino 10.11.128.214 => 5100.64983866668 [fps]

ESTADÍSTICAS DE: 10.11.128.206

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.211
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.166666666666666

Variación del tiempo de viaje [ms]: 2.49600520792586E-32

Tiempo promedio de viaje [ms]: 0.166666666666667

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.066666666666664

Variación del tiempo de viaje [ms]: 1.44000000000007E-05

Tiempo promedio de viaje [ms]: 0.078666666666667

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 2.82069087839818E-05

Tiempo promedio de viaje [ms]: 0.136380133333331

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 2.30564629222262E-30

Tiempo promedio de viaje [ms]: 0.0999999999999991

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 1.43999999999968E-05

Tiempo promedio de viaje [ms]: 0.0786666666666616

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 1.19102399999974E-05

Tiempo promedio de viaje [ms]: 0.0677466666666628

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 1.128213713745E-05

Tiempo promedio de viaje [ms]: 0.101044095613906

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0666666666666629

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 2.59818741905284E-06

Tiempo promedio de viaje [ms]: 0.0668868172560688

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 2.94839999999663E-05

Tiempo promedio de viaje [ms]: 0.168466666666666

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.200000000000031

Variación del tiempo de viaje [ms]: 1.31040000000041E-05

Tiempo promedio de viaje [ms]: 0.201200000000031

Desde el nodo de origen: 10.11.128.206 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 1666.66666666667 [fps]
Nodo destino 10.11.128.191 => 236 [fps]
Nodo destino 10.11.128.194 => 1500.18146666664 [fps]
Nodo destino 10.11.128.197 => 399.999999999996 [fps]
Nodo destino 10.11.128.200 => 471.999999999999 [fps]
Nodo destino 10.11.128.201 => 338.733333333314 [fps]
Nodo destino 10.11.128.204 => 6567.86621490388 [fps]
Nodo destino 10.11.128.211 => 266.666666666652 [fps]
Nodo destino 10.11.128.212 => 4013.20903536413 [fps]
Nodo destino 10.11.128.213 => 842.33333333333 [fps]
Nodo destino 10.11.128.214 => 804.800000000126 [fps]

ESTADÍSTICAS DE: 10.11.128.211

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.212
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.09999999999999994

Variación del tiempo de viaje [ms]: 0.000467937673666108

Tiempo promedio de viaje [ms]: 0.108675406567827

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.07999999999999996

Variación del tiempo de viaje [ms]: 1.44000000000000016E-05

Tiempo promedio de viaje [ms]: 0.0813333333333333

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.03333333333333334

Variación del tiempo de viaje [ms]: 2.53600041187846E-30

Tiempo promedio de viaje [ms]: 0.0333333333333332

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.03333333333333334

Variación del tiempo de viaje [ms]: 0.000155917941544861

Tiempo promedio de viaje [ms]: 0.0360329025846561

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.11999999999999999

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.11999999999999999

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.09999999999999994

Variación del tiempo de viaje [ms]: 3.92264305783982E-05

Tiempo promedio de viaje [ms]: 0.105739562799997

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.26666666666666652

Variación del tiempo de viaje [ms]: 0.000129218575750946

Tiempo promedio de viaje [ms]: 0.276861130095861

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.33333333333333334

Variación del tiempo de viaje [ms]: 3.98700143999505E-05

Tiempo promedio de viaje [ms]: 0.345105333333334

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.19999999999999999

Variación del tiempo de viaje [ms]: 2.18181744000326E-05

Tiempo promedio de viaje [ms]: 0.202171999999994

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 1

Variación del tiempo de viaje [ms]: 1.96224921153323E-07

Tiempo promedio de viaje [ms]: 0.0666797586120153

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 29

Variación del tiempo de viaje [ms]: 1.1412146150537E-05

Tiempo promedio de viaje [ms]: 0.0677176201899834

Desde el nodo de origen: 10.11.128.211 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 2608.20975762784 [fps]
Nodo destino 10.11.128.191 => 162.66666666666666 [fps]
Nodo destino 10.11.128.194 => 1099.9999999999996 [fps]
Nodo destino 10.11.128.197 => 900.822564616401 [fps]
Nodo destino 10.11.128.200 => 839.9999999999995 [fps]
Nodo destino 10.11.128.201 => 951.656065199976 [fps]
Nodo destino 10.11.128.204 => 3599.19469124619 [fps]
Nodo destino 10.11.128.206 => 2415.737333333333 [fps]
Nodo destino 10.11.128.212 => 2021.71999999994 [fps]
Nodo destino 10.11.128.213 => 4200.82479255696 [fps]
Nodo destino 10.11.128.214 => 3995.33959120902 [fps]

ESTADÍSTICAS DE: 10.11.128.212

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.213
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.09999999999999994

Variación del tiempo de viaje [ms]: 0.000523060378739046

Tiempo promedio de viaje [ms]: 0.109219517199998

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 3.22046749665207E-05

Tiempo promedio de viaje [ms]: 0.0680838426666665

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.19999999999999999

Variación del tiempo de viaje [ms]: 1.4400000000000007E-05

Tiempo promedio de viaje [ms]: 0.2013333333333333

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.06666666666666664

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.03333333333333332

Variación del tiempo de viaje [ms]: 0.000270863999999979

Tiempo promedio de viaje [ms]: 0.04146666666666661

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.06666666666666629

Variación del tiempo de viaje [ms]: 5.38433239425702E-06

Tiempo promedio de viaje [ms]: 0.0671315712534644

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.2333333333333332

Variación del tiempo de viaje [ms]: 2.18363166198813E-05

Tiempo promedio de viaje [ms]: 0.235507567330926

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.06666666666666629

Variación del tiempo de viaje [ms]: 6.89117621919079E-30

Tiempo promedio de viaje [ms]: 0.06666666666666636

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.1333333333333326

Variación del tiempo de viaje [ms]: 9.8495999999878E-05

Tiempo promedio de viaje [ms]: 0.1549333333333326

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 1.97825436333453E-06

Tiempo promedio de viaje [ms]: 0.166799610738457

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 3.22643226028192E-06

Tiempo promedio de viaje [ms]: 0.166941188176138

Desde el nodo de origen: 10.11.128.212 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 4587.21972239992 [fps]
Nodo destino 10.11.128.191 => 817.006111999998 [fps]
Nodo destino 10.11.128.194 => 1610.6666666666666 [fps]
Nodo destino 10.11.128.197 => 599.9999999999998 [fps]
Nodo destino 10.11.128.200 => 787.866666666657 [fps]
Nodo destino 10.11.128.201 => 872.710426295037 [fps]
Nodo destino 10.11.128.204 => 4945.65910294945 [fps]
Nodo destino 10.11.128.206 => 2733.333333333321 [fps]
Nodo destino 10.11.128.211 => 464.799999999979 [fps]
Nodo destino 10.11.128.213 => 5504.38715436908 [fps]
Nodo destino 10.11.128.214 => 4674.35326893187 [fps]

ESTADÍSTICAS DE: 10.11.128.213

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.214

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.0666666666666667

Variación del tiempo de viaje [ms]: 0.00179890890072149

Tiempo promedio de viaje [ms]: 0.109329793333332

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.1666666666666666

Variación del tiempo de viaje [ms]: 4.218569501858E-32

Tiempo promedio de viaje [ms]: 0.166666666666667

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.1999999999999999

Variación del tiempo de viaje [ms]: 2.462400000000077E-05

Tiempo promedio de viaje [ms]: 0.202533333333332

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0666666666666666

Variación del tiempo de viaje [ms]: 0.00177908399999982

Tiempo promedio de viaje [ms]: 0.158466666666667

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.066666666666629

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.133333333333326

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.2999999999999983

Variación del tiempo de viaje [ms]: 1.67442501973004E-05

Tiempo promedio de viaje [ms]: 0.301583388000017

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.3799999999999999

Variación del tiempo de viaje [ms]: 1.44000000000007E-05

Tiempo promedio de viaje [ms]: 0.381333333333332

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.113333333333326

Variación del tiempo de viaje [ms]: 0.000291599999999953

Tiempo promedio de viaje [ms]: 0.119333333333326

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 4.29214126937039E-06

Tiempo promedio de viaje [ms]: 0.100367825615359

Tiempo de viaje para el destino: 10.11.128.214

Mejor tiempo [ms]: 0.333333333333314

Variación del tiempo de viaje [ms]: 0.000129599999999971

Tiempo promedio de viaje [ms]: 0.36933333333331

Desde el nodo de origen: 10.11.128.213 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 5138.5002866666 [fps]
Nodo destino 10.11.128.191 => 2833.3333333333 [fps]
Nodo destino 10.11.128.194 => 2430.3999999999 [fps]
Nodo destino 10.11.128.197 => 633.8666666667 [fps]
Nodo destino 10.11.128.200 => 733.3333333333 [fps]
Nodo destino 10.11.128.201 => 1466.6666666666 [fps]
Nodo destino 10.11.128.204 => 11158.5853560006 [fps]
Nodo destino 10.11.128.206 => 1144 [fps]
Nodo destino 10.11.128.211 => 835.3333333333 [fps]
Nodo destino 10.11.128.212 => 5319.49475761403 [fps]
Nodo destino 10.11.128.214 => 738.6666666666 [fps]

ESTADÍSTICAS DE: 10.11.128.214

Lista de IP conectadas al Nodo de Red:

10.11.128.190
10.11.128.191
10.11.128.194
10.11.128.197
10.11.128.200
10.11.128.201
10.11.128.204
10.11.128.206
10.11.128.211
10.11.128.212
10.11.128.213

Tiempo de viaje para el destino: 10.11.128.190

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 7.01217865524028E-05

Tiempo promedio de viaje [ms]: 0.105220546666664

Tiempo de viaje para el destino: 10.11.128.191

Mejor tiempo [ms]: 0.106666666666666

Variación del tiempo de viaje [ms]: 1.4244177868688E-31

Tiempo promedio de viaje [ms]: 0.106666666666666

Tiempo de viaje para el destino: 10.11.128.194

Mejor tiempo [ms]: 0.033333333333314

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.033333333333314

Tiempo de viaje para el destino: 10.11.128.197

Mejor tiempo [ms]: 0.0666666666666771

Variación del tiempo de viaje [ms]: 1.439999999999776E-05

Tiempo promedio de viaje [ms]: 0.0786666666666678

Tiempo de viaje para el destino: 10.11.128.200

Mejor tiempo [ms]: 0.033333333333314

Variación del tiempo de viaje [ms]: 2.46239999999946E-05

Tiempo promedio de viaje [ms]: 0.044133333333302

Tiempo de viaje para el destino: 10.11.128.201

Mejor tiempo [ms]: 0.033333333333314

Variación del tiempo de viaje [ms]: 3.79721365924686E-05

Tiempo promedio de viaje [ms]: 0.038498939853312

Tiempo de viaje para el destino: 10.11.128.204

Mejor tiempo [ms]: 0.1999999999999983

Variación del tiempo de viaje [ms]: 3.14689521684456E-05

Tiempo promedio de viaje [ms]: 0.203587877613338

Tiempo de viaje para el destino: 10.11.128.206

Mejor tiempo [ms]: 0.033333333333314

Variación del tiempo de viaje [ms]: 1.14518801253963E-30

Tiempo promedio de viaje [ms]: 0.033333333333317

Tiempo de viaje para el destino: 10.11.128.211

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 0.000113874524793848

Tiempo promedio de viaje [ms]: 0.104923769296862

Tiempo de viaje para el destino: 10.11.128.212

Mejor tiempo [ms]: 0.23333333333337

Variación del tiempo de viaje [ms]: 1.44000000000045E-05

Tiempo promedio de viaje [ms]: 0.234666666666703

Tiempo de viaje para el destino: 10.11.128.213

Mejor tiempo [ms]: 0.403333333333315

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.403333333333315

Desde el nodo de origen: 10.11.128.214 se obtiene los throughputs:

Nodo destino 10.11.128.190 => 1683.55287466663 [fps]
Nodo destino 10.11.128.191 => 213.33333333332 [fps]
Nodo destino 10.11.128.194 => 399.99999999997 [fps]
Nodo destino 10.11.128.197 => 393.33333333339 [fps]
Nodo destino 10.11.128.200 => 661.999999999954 [fps]
Nodo destino 10.11.128.201 => 808.477736919955 [fps]
Nodo destino 10.11.128.204 => 8347.10298214684 [fps]
Nodo destino 10.11.128.206 => 1599.99999999992 [fps]
Nodo destino 10.11.128.211 => 2937.86554031215 [fps]
Nodo destino 10.11.128.212 => 2346.66666666703 [fps]
Nodo destino 10.11.128.213 => 1613.33333333326 [fps]

Anexo 3: Reporte Estadístico de los nodos de Red - Topología de Red 2

ESTADISTICAS DE: 10.11.11.80

Lista de IP conectadas al Nodo de Red:

10.11.11.81
10.11.11.84

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0333333333333332
Variación del tiempo de viaje [ms]: 2.54054233788981E-33
Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.06666666666666629
Variación del tiempo de viaje [ms]: 1.01881694058085E-30
Tiempo promedio de viaje [ms]: 0.0666666666666661

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.166666666666666
Variación del tiempo de viaje [ms]: 3.60000000001456E-06
Tiempo promedio de viaje [ms]: 0.1673333333333334

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.0333333333333332
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.0333333333333332

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.06666666666666629
Variación del tiempo de viaje [ms]: 0.00363362856815258
Tiempo promedio de viaje [ms]: 0.113157125346678

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.16666666666666657
Variación del tiempo de viaje [ms]: 1.02236373316643E-28
Tiempo promedio de viaje [ms]: 0.1666666666666661

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.80 se obtiene los throughputs:

Nodo destino 10.11.11.81 => 166.666666666667 [fps]
Nodo destino 10.11.11.82 => 533.333333333329 [fps]
Nodo destino 10.11.11.83 => 1171.33333333334 [fps]
Nodo destino 10.11.11.84 => 1666.66666666666 [fps]
Nodo destino 10.11.11.85 => 4073.6565124804 [fps]
Nodo destino 10.11.11.86 => 1999.99999999993 [fps]
Nodo destino 10.11.11.87 => Infinity [fps]

ESTADISTICAS DE: 10.11.11.81

Lista de IP conectadas al Nodo de Red:

10.11.11.80
10.11.11.82
10.11.11.84
10.11.11.85

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.03333333333333314
Variación del tiempo de viaje [ms]: 4.06891841645586E-05
Tiempo promedio de viaje [ms]: 0.035970068013333

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.03333333333333314
Variación del tiempo de viaje [ms]: 4.77757091365125E-05
Tiempo promedio de viaje [ms]: 0.0378335151708419

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.133333333333333
Variación del tiempo de viaje [ms]: 2.97756000000968E-06
Tiempo promedio de viaje [ms]: 0.1338733333333338

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.0666666666666664
Variación del tiempo de viaje [ms]: 2.76300226238413E-05
Tiempo promedio de viaje [ms]: 0.0696259866666665

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.03333333333333314
Variación del tiempo de viaje [ms]: 0.000145394782593755
Tiempo promedio de viaje [ms]: 0.0389375784319326

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.1333333333333326
Variación del tiempo de viaje [ms]: 1.43796634965645E-28
Tiempo promedio de viaje [ms]: 0.133333333333333

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 0.09999999999999943
Variación del tiempo de viaje [ms]: 2.18181744000318E-05
Tiempo promedio de viaje [ms]: 0.102171999999998

Desde el nodo de origen: 10.11.11.81 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 719.401360266659 [fps]
Nodo destino 10.11.11.82 => 1475.50709166284 [fps]
Nodo destino 10.11.11.83 => 2945.21333333344 [fps]
Nodo destino 10.11.11.84 => 2576.16150666666 [fps]
Nodo destino 10.11.11.85 => 1791.1286078689 [fps]
Nodo destino 10.11.11.86 => 4666.66666666667 [fps]
Nodo destino 10.11.11.87 => 1736.92399999997 [fps]

ESTADISTICAS DE: 10.11.11.82

Lista de IP conectadas al Nodo de Red:

10.11.11.81
10.11.11.83

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.61012464000063E-05

Tiempo promedio de viaje [ms]: 0.0754146666666663

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 8.86018983735528E-05

Tiempo promedio de viaje [ms]: 0.0369896474133328

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.166666666666666

Variación del tiempo de viaje [ms]: 5.75079599906118E-29

Tiempo promedio de viaje [ms]: 0.166666666666666

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0999999999999996

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.0800000000000001

Variación del tiempo de viaje [ms]: 0.000418131000000134

Tiempo promedio de viaje [ms]: 0.0904333333333352

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 9.88854377992718E-29

Tiempo promedio de viaje [ms]: 0.100000000000003

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.59491587414114E-29

Tiempo promedio de viaje [ms]: 0.0666666666666667

Desde el nodo de origen: 10.11.11.82 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 754.146666666663 [fps]
Nodo destino 10.11.11.81 => 554.844711199992 [fps]
Nodo destino 10.11.11.83 => 1000.000000000001 [fps]
Nodo destino 10.11.11.84 => 1999.999999999999 [fps]
Nodo destino 10.11.11.85 => 633.033333333346 [fps]
Nodo destino 10.11.11.86 => 4200.000000000013 [fps]
Nodo destino 10.11.11.87 => 1400 [fps]

ESTADISTICAS DE: 10.11.11.83

Lista de IP conectadas al Nodo de Red:

10.11.11.82
10.11.11.87

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 1.0205502775308E-32

Tiempo promedio de viaje [ms]: 0.1

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 7.45167600000061E-05

Tiempo promedio de viaje [ms]: 0.0816066666666659

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 3.21029632779002E-30

Tiempo promedio de viaje [ms]: 0.0333333333333329

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.133333333333333

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.133333333333333

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.1000000000000016

Variación del tiempo de viaje [ms]: 1.53899999998243E-06

Tiempo promedio de viaje [ms]: 0.1027

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 4.53902233858225E-29

Tiempo promedio de viaje [ms]: 0.0666666666666703

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 1.0262858586833E-29

Tiempo promedio de viaje [ms]: 0.0333333333333339

Desde el nodo de origen: 10.11.11.83 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 400 [fps]
Nodo destino 10.11.11.81 => 571.246666666661 [fps]
Nodo destino 10.11.11.82 => 366.666666666661 [fps]
Nodo destino 10.11.11.84 => 1333.33333333333 [fps]
Nodo destino 10.11.11.85 => 1232.4 [fps]
Nodo destino 10.11.11.86 => 3466.66666666686 [fps]
Nodo destino 10.11.11.87 => 1100.00000000002 [fps]

ESTADÍSTICAS DE: 10.11.11.84

Lista de IP conectadas al Nodo de Red:

10.11.11.80
10.11.11.81

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0333333333333332

Variación del tiempo de viaje [ms]: 1.08333559371782E-33

Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0666666666666664

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0999999999999996

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.2199999999999988

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.2199999999999988

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 7.53063360876247E-05

Tiempo promedio de viaje [ms]: 0.0743844516332386

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.1666666666666693

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1666666666666693

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.84 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 133.333333333333 [fps]

Nodo destino 10.11.11.81 => 199.999999999999 [fps]

Nodo destino 10.11.11.82 => 699.999999999998 [fps]

Nodo destino 10.11.11.83 => 1319.999999999993 [fps]

Nodo destino 10.11.11.85 => 2900.99361369631 [fps]

Nodo destino 10.11.11.86 => 2833.333333333378 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADÍSTICAS DE: 10.11.11.85

Lista de IP conectadas al Nodo de Red:

10.11.11.81
10.11.11.86

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0666666666666664

Variación del tiempo de viaje [ms]: 6.4944000000001E-05

Tiempo promedio de viaje [ms]: 0.0894666666666666

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0333333333333334

Variación del tiempo de viaje [ms]: 4.21144852409748E-30

Tiempo promedio de viaje [ms]: 0.0333333333333329

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 4.88107489263703E-05

Tiempo promedio de viaje [ms]: 0.0752157818197474

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.0999999999999943

Variación del tiempo de viaje [ms]: 2.70342360001153E-06

Tiempo promedio de viaje [ms]: 0.1004860000000003

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.113333333333333

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.113333333333333

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 2.462400000000567E-05

Tiempo promedio de viaje [ms]: 0.1358666666666662

Desde el nodo de origen: 10.11.11.85 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 536.8 [fps]

Nodo destino 10.11.11.81 => 599.999999999993 [fps]

Nodo destino 10.11.11.82 => 1579.5314182147 [fps]

Nodo destino 10.11.11.83 => 2914.0940000001 [fps]

Nodo destino 10.11.11.84 => 1020 [fps]

Nodo destino 10.11.11.86 => Infinity [fps]

Nodo destino 10.11.11.87 => 1630.39999999994 [fps]

ESTADÍSTICAS DE: 10.11.11.86

Lista de IP conectadas al Nodo de Red:

10.11.11.85
10.11.11.87

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.09999999999999996

Variación del tiempo de viaje [ms]: 1.4400000000000008E-05

Tiempo promedio de viaje [ms]: 0.112

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.06666666666666664

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.099999999999999943

Variación del tiempo de viaje [ms]: 3.2399999999999997E-05

Tiempo promedio de viaje [ms]: 0.10199999999999997

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.06666666666666629

Variación del tiempo de viaje [ms]: 2.45221311601176E-06

Tiempo promedio de viaje [ms]: 0.067104066666666703

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.1466666666666667

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1466666666666667

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 4.65814513106731E-30

Tiempo promedio de viaje [ms]: 0.03333333333333323

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1666666666666657

Desde el nodo de origen: 10.11.11.86 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 336 [fps]
Nodo destino 10.11.11.81 => 333.333333333332 [fps]
Nodo destino 10.11.11.82 => 2243.999999999993 [fps]
Nodo destino 10.11.11.83 => 2617.058600000014 [fps]
Nodo destino 10.11.11.84 => 439.999999999999 [fps]
Nodo destino 10.11.11.85 => 2133.33333333327 [fps]
Nodo destino 10.11.11.87 => 666.666666666629 [fps]

ESTADÍSTICAS DE: 10.11.11.87

Lista de IP conectadas al Nodo de Red:

10.11.11.83
10.11.11.86

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.13333333333333333

Variación del tiempo de viaje [ms]: 1.73333694994851E-32

Tiempo promedio de viaje [ms]: 0.13333333333333333

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.11999999999999999

Variación del tiempo de viaje [ms]: 3.239999999999986E-05

Tiempo promedio de viaje [ms]: 0.12199999999999999

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0666666666666629

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 2.0127921854187E-06

Tiempo promedio de viaje [ms]: 0.0336876273333364

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.16666666666666666

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.16666666666666666

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.06666666666666629

Variación del tiempo de viaje [ms]: 1.50965180681953E-06

Tiempo promedio de viaje [ms]: 0.0672843400629501

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 1.36875745428914E-07

Tiempo promedio de viaje [ms]: 0.0333562245588004

Desde el nodo de origen: 10.11.11.87 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 533.333333333333 [fps]
Nodo destino 10.11.11.81 => 243.9999999999998 [fps]
Nodo destino 10.11.11.82 => 1333.33333333326 [fps]
Nodo destino 10.11.11.83 => 1650.69373933348 [fps]
Nodo destino 10.11.11.84 => 999.999999999996 [fps]
Nodo destino 10.11.11.85 => 2825.9422826439 [fps]
Nodo destino 10.11.11.86 => 2068.08592264562 [fps]

Anexo 4: Reporte Estadístico de los Generadores de Hormigas - Topología de Red 2

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.80

Homigas artificiales transmitidas al nodo: 10.11.11.81 => 19 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.82 => 30 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.83 => 52 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.84 => 151 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.85 => 140 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.86 => 96 homigas
Homigas artificiales transmitidas al nodo: 10.11.11.87 => 19 homigas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.81

Homigas artificiales transmitidas al nodo:	10.11.11.80 => 30 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.82 => 96 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.83 => 85 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.84 => 63 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.85 => 85 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.86 => 96 homigas
Homigas artificiales transmitidas al nodo:	10.11.11.87 => 52 homigas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.82

Homígas artificiales transmitidas al nodo: 10.11.11.80 => 30 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.81 => 85 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.83 => 30 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.84 => 96 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.85 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.86 => 162 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.87 => 85 homígas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.83

Hormigas artificiales transmitidas al nodo: 10.11.11.80 => 19 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.81 => 30 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.82 => 74 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.84 => 41 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.85 => 85 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.86 => 140 hormigas
 Hormigas artificiales transmitidas al nodo: 10.11.11.87 => 118 hormigas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.84

Homigas artificiales transmitidas al nodo: 10.11.11.80 => 19 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.81 => 30 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.82 => 41 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.83 => 41 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.85 => 228 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.86 => 118 homigas
 Homigas artificiales transmitidas al nodo: 10.11.11.87 => 19 homigas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.85

Homígas artificiales transmitidas al nodo: 10.11.11.80 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.81 => 118 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.82 => 96 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.83 => 96 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.84 => 63 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.86 => 30 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.87 => 96 homígas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.86

Homígas artificiales transmitidas al nodo: 10.11.11.80 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.81 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.82 => 107 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.83 => 96 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.84 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.85 => 228 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.87 => 19 homígas

GENERADOR DE HORMIGAS ARTIFICIALES EN: 10.11.11.87

Homígas artificiales transmitidas al nodo: 10.11.11.80 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.81 => 19 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.82 => 74 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.83 => 118 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.84 => 30 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.85 => 118 homígas
Homígas artificiales transmitidas al nodo: 10.11.11.86 => 129 homígas

Anexo 5: Reporte Estadístico de los nodos de Red - Topología de Red 3

ESTADISTICAS DE: 10.11.11.80

Lista de IP conectadas al Nodo de Red:

10.11.11.84
10.11.11.85

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.06666666666666664

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 8.07777550912899E-29

Tiempo promedio de viaje [ms]: 0.1

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.06666666666666664

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 2.54704235145212E-31

Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.1666666666666657

Variación del tiempo de viaje [ms]: 3.98418341627184E-05

Tiempo promedio de viaje [ms]: 0.173752546666666

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.80 se obtiene los throughputs:

Nodo destino 10.11.11.81 => 399.999999999999 [fps]

Nodo destino 10.11.11.82 => 19700 [fps]

Nodo destino 10.11.11.83 => Infinity [fps]

Nodo destino 10.11.11.84 => 199.999999999999 [fps]

Nodo destino 10.11.11.85 => 99.9999999999991 [fps]

Nodo destino 10.11.11.86 => 23456.5937999991 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADISTICAS DE: 10.11.11.81

Lista de IP conectadas al Nodo de Red:

10.11.11.82
10.11.11.85
10.11.11.86

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.1

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 8.60179901708199E-30

Tiempo promedio de viaje [ms]: 0.03333333333333353

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.06666666666666664

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.06666666666666664

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.59491587414114E-29

Tiempo promedio de viaje [ms]: 0.06666666666666732

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 1.1149192215596E-29

Tiempo promedio de viaje [ms]: 0.03333333333333345

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.09999999999999943

Variación del tiempo de viaje [ms]: 9.88854377992718E-29

Tiempo promedio de viaje [ms]: 0.1000000000000003

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.81 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 1000 [fps]

Nodo destino 10.11.11.82 => 1166.666666666674 [fps]

Nodo destino 10.11.11.83 => 1400 [fps]

Nodo destino 10.11.11.84 => 466.666666666713 [fps]

Nodo destino 10.11.11.85 => 533.333333333352 [fps]

Nodo destino 10.11.11.86 => 3800.00000000012 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADISTICAS DE: 10.11.11.82

Lista de IP conectadas al Nodo de Red:

10.11.11.81
10.11.11.83

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.1333333333333333

Variación del tiempo de viaje [ms]: 0.000705600000000001

Tiempo promedio de viaje [ms]: 0.1426666666666667

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 6.41973355061714E-30

Tiempo promedio de viaje [ms]: 0.0333333333333343

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.03333333333333314

Variación del tiempo de viaje [ms]: 2.60318133154931E-30

Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.0666666666666671

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0666666666666671

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 4.03133699781307E-29

Tiempo promedio de viaje [ms]: 0.0666666666666676

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.82 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 1426.6666666667 [fps]

Nodo destino 10.11.11.81 => 866.66666666692 [fps]

Nodo destino 10.11.11.83 => 500 [fps]

Nodo destino 10.11.11.84 => Infinity [fps]

Nodo destino 10.11.11.85 => 866.666666666802 [fps]

Nodo destino 10.11.11.86 => 1133.33333333335 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADISTICAS DE: 10.11.11.83

Lista de IP conectadas al Nodo de Red:

10.11.11.82
10.11.11.86
10.11.11.87

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.166666666666667

Variación del tiempo de viaje [ms]: 1.52640000000001E-05

Tiempo promedio de viaje [ms]: 0.168533333333333

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 4.56458732222336E-05

Tiempo promedio de viaje [ms]: 0.0692035866666662

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0466666666666669

Variación del tiempo de viaje [ms]: 4.0930055771112E-30

Tiempo promedio de viaje [ms]: 0.0466666666666676

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.1000000000000016

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1000000000000016

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 8.32898759757311E-30

Tiempo promedio de viaje [ms]: 0.0333333333333332

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.83 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 3707.73333333333 [fps]

Nodo destino 10.11.11.81 => 1107.25738666669 [fps]

Nodo destino 10.11.11.82 => 326.66666666673 [fps]

Nodo destino 10.11.11.84 => Infinity [fps]

Nodo destino 10.11.11.85 => 1700.00000000027 [fps]

Nodo destino 10.11.11.86 => 966.66666666662 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADÍSTICAS DE: 10.11.11.84

Lista de IP conectadas al Nodo de Red:

10.11.11.80
10.11.11.85

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0333333333333333

Variación del tiempo de viaje [ms]: 5.21927014928941E-35

Tiempo promedio de viaje [ms]: 0.0333333333333333

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.1333333333333326

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1333333333333326

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.1333333333333333

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.1333333333333333

Tiempo de viaje para el destino: 10.11.11.85

Mejor tiempo [ms]: 0.0333333333333332

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0333333333333332

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.2266666666666588

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.2266666666666588

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.84 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 433.333333333333 [fps]

Nodo destino 10.11.11.81 => Infinity [fps]

Nodo destino 10.11.11.82 => 26266.6666666652 [fps]

Nodo destino 10.11.11.83 => 799.999999999997 [fps]

Nodo destino 10.11.11.85 => 66.666666666664 [fps]

Nodo destino 10.11.11.86 => 30826.666666656 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADÍSTICAS DE: 10.11.11.85

Lista de IP conectadas al Nodo de Red:

10.11.11.80
10.11.11.81
10.11.11.84

Tiempo de viaje para el destino: 10.11.11.80

Mejor tiempo [ms]: 0.0666666666666667

Variación del tiempo de viaje [ms]: 2.08770805971576E-34

Tiempo promedio de viaje [ms]: 0.0666666666666667

Tiempo de viaje para el destino: 10.11.11.81

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 5.58563498778533E-31

Tiempo promedio de viaje [ms]: 0.0333333333333327

Tiempo de viaje para el destino: 10.11.11.82

Mejor tiempo [ms]: 0.0666666666666629

Variación del tiempo de viaje [ms]: 3.07622908238286E-29

Tiempo promedio de viaje [ms]: 0.06666666666666702

Tiempo de viaje para el destino: 10.11.11.83

Mejor tiempo [ms]: 0.0999999999999996

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 0.0999999999999996

Tiempo de viaje para el destino: 10.11.11.84

Mejor tiempo [ms]: 0.0333333333333314

Variación del tiempo de viaje [ms]: 5.52455695114944E-30

Tiempo promedio de viaje [ms]: 0.0333333333333338

Tiempo de viaje para el destino: 10.11.11.86

Mejor tiempo [ms]: 0.133333333333326

Variación del tiempo de viaje [ms]: 3.98418341627554E-05

Tiempo promedio de viaje [ms]: 0.140419213333325

Tiempo de viaje para el destino: 10.11.11.87

Mejor tiempo [ms]: 1.79769313486232E+308

Variación del tiempo de viaje [ms]: 0

Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.85 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 733.333333333333 [fps]

Nodo destino 10.11.11.81 => 433.333333333325 [fps]

Nodo destino 10.11.11.82 => 1933.33333333344 [fps]

Nodo destino 10.11.11.83 => 1499.99999999999 [fps]

Nodo destino 10.11.11.84 => 333.33333333338 [fps]

Nodo destino 10.11.11.86 => 5757.18774666634 [fps]

Nodo destino 10.11.11.87 => Infinity [fps]

ESTADÍSTICAS DE: 10.11.11.86

Lista de IP conectadas al Nodo de Red:

10.11.11.81
10.11.11.83

Tiempo de viaje para el destino: 10.11.11.80
Mejor tiempo [ms]: 0.14
Variación del tiempo de viaje [ms]: 0.000291599999999999
Tiempo promedio de viaje [ms]: 0.194

Tiempo de viaje para el destino: 10.11.11.81
Mejor tiempo [ms]: 0.099999999999999943
Variación del tiempo de viaje [ms]: 2.305646292222262E-30
Tiempo promedio de viaje [ms]: 0.09999999999999991

Tiempo de viaje para el destino: 10.11.11.82
Mejor tiempo [ms]: 0.08000000000000001
Variación del tiempo de viaje [ms]: 1.25115342535195E-29
Tiempo promedio de viaje [ms]: 0.08000000000000013

Tiempo de viaje para el destino: 10.11.11.83
Mejor tiempo [ms]: 0.03333333333333314
Variación del tiempo de viaje [ms]: 2.60318133154931E-30
Tiempo promedio de viaje [ms]: 0.03333333333333333

Tiempo de viaje para el destino: 10.11.11.84
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.85
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.87
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.86 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 5820 [fps]
Nodo destino 10.11.11.81 => 699.9999999999994 [fps]
Nodo destino 10.11.11.82 => 400.0000000000007 [fps]
Nodo destino 10.11.11.83 => 200 [fps]
Nodo destino 10.11.11.84 => Infinity [fps]
Nodo destino 10.11.11.85 => Infinity [fps]
Nodo destino 10.11.11.87 => Infinity [fps]

ESTADÍSTICAS DE: 10.11.11.87

Lista de IP conectadas al Nodo de Red:

10.11.11.83

Tiempo de viaje para el destino: 10.11.11.80
Mejor tiempo [ms]: 0.2133333333333333
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.2133333333333333

Tiempo de viaje para el destino: 10.11.11.81
Mejor tiempo [ms]: 0.1133333333333333
Variación del tiempo de viaje [ms]: 1.440000000000016E-05
Tiempo promedio de viaje [ms]: 0.1146666666666666

Tiempo de viaje para el destino: 10.11.11.82
Mejor tiempo [ms]: 0.08000000000000001
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 0.08000000000000001

Tiempo de viaje para el destino: 10.11.11.83
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.84
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.85
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Tiempo de viaje para el destino: 10.11.11.86
Mejor tiempo [ms]: 1.79769313486232E+308
Variación del tiempo de viaje [ms]: 0
Tiempo promedio de viaje [ms]: 1.79769313486232E+308

Desde el nodo de origen: 10.11.11.87 se obtiene los throughputs:

Nodo destino 10.11.11.80 => 1493.333333333333 [fps]
Nodo destino 10.11.11.81 => 573.3333333333331 [fps]
Nodo destino 10.11.11.82 => 400 [fps]
Nodo destino 10.11.11.83 => Infinity [fps]
Nodo destino 10.11.11.84 => Infinity [fps]
Nodo destino 10.11.11.85 => Infinity [fps]
Nodo destino 10.11.11.86 => Infinity [fps]

[illegible]