

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



**ARQUITECTURA MULTI-AGENTE DE
AUTOGESTIÓN DE REDES EN EL PLANO DE
CONTROL DE FLUJO**

TESIS

PARA OBTENER EL GRADO DE

DOCTOR EN CIENCIAS

Presenta:

KARINA RAYA DÍAZ

Bajo la dirección de:

DRA. CARELIA G. GAXIOLA PACHECO

Co-dirigido por:

DR. MANUEL CASTAÑÓN PUGA

TIJUANA, BAJA CALIFORNIA, MÉXICO

DICIEMBRE DE 2017

Dedico ésta tesis a todas las personas que me apoyaron en el transcurso y cumplimiento de una meta más en mi formación profesional.

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 229

Tijuana, B. C., a 24 de noviembre de 2017

C. Karina Raya Díaz
Pasante de: Doctor en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por los C. Dres. Carelia Guadalupe Gaxiola Pacheco y Manuel
Castañón Puga

Quienes serán los responsables de la calidad de trabajo que usted presente,
referido al tema ARQUITECTURA MULTI-AGENTE DE AUTOGESTIÓN DE REDES
EN EL PLANO DE CONTROL DE FLUJO

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCION
- II.- MARCO TEÓRICO
- III.- ARQUITECTURAS DE REDES AUTONÓMICAS
- IV.- REDES COMPLEJAS
- V.- MODELO BASADO EN AGENTES DE AUTOGESTIÓN DE RED
PARA CONTROL DE FLUJO EN UNA RED
- VI.- CASOS DE ESTUDIO
- VII.- TRABAJO FUTURO Y CONCLUSIONES

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA

Dra. Carelia Guadalupe Gaxiola Pacheco
Directora de Tesis

Dr. José Luis González Vázquez
Sub-Director Secretario

Dr. Manuel Castañón Puga
Co-Director de Tesis

Dr. Luis Enrique Palafox Maestre
Director

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el financiamiento para la realización de esta investigación.

Al programa de Maestría y Doctorado en Ciencias e Ingeniería de la prestigiosa Universidad Autónoma de Baja California. Así mismo a la Facultad de Ciencias Químicas e Ingeniería por haberme sustentado dentro de sus instalaciones.

A los integrantes del Cuerpo Académico de Complejidad y Computación por compartir sus conocimientos y guiarme en el transcurso de mi investigación.

A mi esposo Gabriel y familiares por todo el respaldo durante el transcurso del doctorado.

A mi Comité de Tesis sin su guía, dedicación y contribución no habría logrado ésta meta.

A Dios que me hizo fuerte en los momentos más complejos.

Karina Raya Díaz

Resumen

En éste documento se propone un modelo basado en agentes para la autogestión de los flujos de red. La autogestión de la red utiliza técnicas de inteligencia artificial para la toma de decisiones. La incorporación de agentes inteligentes en la autoconfiguración de una red mediada por un algoritmo de negociación puede tomar acciones proactivas automáticamente para mantener una gestión de flujo eficiente. Esta tesis describe el análisis de la estructura de red con el enfoque de sistemas complejos y propone un algoritmo basado en preferencias y autoridades ocultas para negociar la entrega de los flujos cuando surge la congestión en un nodo.

Abstract

This document proposes an agent-based model for self-management of network flows. Network self-management uses artificial intelligence tools for decision-making. The incorporation of intelligent agents in the auto-configuration of a network mediated by a trading algorithm can perform proactive actions automatically to maintain an efficient flow management. The thesis describes the analysis of the network structure with the approach of complex systems and proposes an algorithm based on preferences and hidden authorities to negotiate the delivery of the flows when congestion arises in a node.

Índice general

1. Introducción	2
1.1. Antecedentes	3
1.1.1. Justificación	7
1.1.2. Planteamiento del problema	8
1.1.3. Hipótesis	10
1.1.4. Objetivo General	10
1.1.5. Objetivo Específico	10
1.1.6. Metodología	11
1.1.7. Estructura de la tesis	11
2. Marco teórico	12
2.1. Introducción	12
2.2. Gestión de Redes	12
2.2.1. Paradigma Gestor-Agente	14
2.3. Gestión Autónoma de Redes	15

3. Arquitecturas de Redes Autónomas	19
3.1. Introducción	19
3.2. Clasificación de Arquitecturas Autónomas	19
3.3. Internet Autónomo (AutoI)	20
3.4. Redireccionamiento y Gestión Dinámica para el Ejército (DRAMA)	22
3.5. Sensibilidad al Contexto para la Gestión Autónoma de MANETs (CA-MANETs)	24
3.6. Unity	26
3.7. Arquitectura de Gestión Descentralizada Autónoma (ADMA)	28
3.8. Arquitectura de Red Autónoma (ANA)	30
3.9. Gestión en Red (INM)	31
4. Redes Complejas	35
4.1. Métricas de Redes Complejas	35
4.1.1. Matriz de adyacencia	36
4.1.2. Grado de un nodo	37
4.1.3. Coeficiente de agrupamiento	37
4.1.4. Centralidad de un nodo	38
4.1.5. Centralidad de Intermediación	38
4.1.6. Longitud de la ruta	39
5. Modelo Basado en Agentes de Autogestión de Red para Control de Flujo en una Red	40

5.1. Introducción	40
5.2. Arquitectura Multi-Agente para la Autogestión de Flujos en la Red (Teleka)	41
5.3. Modelado Computacional de los Agentes y su Entorno	42
5.3.1. Descripción del agente	43
5.4. Algoritmo de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO)	44
5.5. Descripción del Modelo Basado en Agentes en NetLogo	45
5.5.1. Fase de Inicialización	46
5.5.2. Determinación de Métricas de Redes Complejas	47
5.5.3. Asignación de las Preferencias de Flujos	47
5.6. Resumen	48
6. Casos de Estudio	49
6.1. Introducción	49
6.2. Caso de Estudio 1	49
6.2.1. Simulación utilizando topología con distribución de libre escala	50
6.3. Análisis de Resultados	51
6.4. Caso de Estudio 2	55
6.4.1. Simulación utilizando topología generada basada en el grado del nodo	55
6.5. Caso de Estudio 3	59
6.5.1. Simulación utilizando Behavior Space de NetLogo	59

7. Trabajo Futuro y Conclusiones	63
7.1. Conclusiones	63
7.2. Trabajo Futuro	64
A. Tablas de Configuración	65
A.1. Apéndice	65

Índice de figuras

1.1. Esquema propuesto como base para el diseño de una Arquitectura Multi-agente.	7
2.1. Ciclo de gestión de redes [4].	14
2.2. Flujo de información de gestión [4].	15
2.3. Propiedades de la Auto-Gestión.	16
2.4. Modelo de referencia del Ciclo de Control Autónomo [15]	17
3.1. Arquitectura de AutoI [21].	21
3.2. MAPE ciclo de gestión.	22
3.3. Jerarquía de gestión de DRAMA [6].	23
3.4. Roles y Módulos de CA-MANET [22].	24
3.5. Arquitectura de CA-MANET [22].	25
3.6. Escenario de Unity [8].	28
3.7. Arquitectura de un nodo ADMA [2].	29
3.8. Abstracciones fundamentales de ANA [32].	30
3.9. Arquitectur INM.	31

4.1. Topología de red distinción de grados de cada nodo.	36
5.1. Arquitectura Multi-Agente para la Gestión de Flujos de Red (Teleka).	42
5.2. Diagrama de flujo del algoritmo ESAO.	44
5.3. Interfaz de usuario del modelo simulación en NetLogo	46
5.4. Histogramas de las Métricas de Redes Complejas del Modelo.	47
5.5. Topología generada basada en libre escala.	48
6.1. Topología generada considerando escala libre.	51
6.2. Histogramas con las métricas de redes complejas obtenidas del modelo.	52
6.3. Estado de congestiónamiento de la red durante la simulación aplicando el Algoritmo ESAO.	52
6.4. Estado de congestiónamiento de la red utilizando una Distribución Aleatoria.	53
6.5. Promedio de congestiónamiento por nodo durante la simulación aplicando el algoritmo ESAO.	54
6.6. Promedio de congestiónamiento por nodo utilizando Distrución Aleatoria.	54
6.7. Utilidades obtenidas por los agentes aplicando ESAO.	55
6.8. Topología con distribución basado en grado.	56
6.9. Histogramas obtenidos del cálculo de las métricas de redes complejas.	56
6.10. Promedio de estado de congestiónamiento de la red durante la simulación utilizando el Algoritmo ESAO.	57
6.11. Promedio de congestiónamiento de la red durante la simulación utilizando Distribución Aleatoria.	57

6.12. Promedio de congestionamiento de los nodos durante el tiempo de simulación aplicando el Algoritmo ESAO.	58
6.13. Promedio de congestionamiento por nodo durante el tiempo de simlación apli- cando una Distribución Aleatoia.	58
6.14. Utilidades de cada agente obtenidos al aplicar el algoritmo ESAO.	59
6.15. Inicialización de Behavior Space de Netlogo.	60
6.16. Comparación de rendimiento del modelo aplicando el Algoritmo ESAO y con Distribución Aleatoria.	62

Índice de tablas

3.1. Comparación de arquitecturas de red autonómicas parte 1.	33
3.2. Comparación de arquitecturas de red autonómicas parte 2.	34
4.1. Métricas de grado y centralidad de la red de la Figura 4.1.	37
6.1. Parámetros de inicialización del modelo.	50
6.2. Resultados obtenidos después de 600 simulaciones con <i>Behavior Space</i> de NetLogo.	61
A.1. Costo de los enlaces que conforman la red.	66
A.2. Valores de inicialización de los nodos de red.	67
A.3. Utilidades de los agentes después de las negociaciones.	68
A.4. Utilidades de los agentes.	69

Capítulo 1

Introducción

Actualmente el dinamismo de una red de comunicaciones es muy grande, debido a ello es complicado que una sola persona pueda gestionar una red. Tomando en cuenta lo anterior, hoy en día existen nuevas herramientas que permiten la automatización de la toma de decisiones, un ejemplo de ellas son los agentes inteligentes, los cuales pueden decidir qué hacer cuando existe un conflicto en el aprovisionamiento de servicios dentro de una red, sin embargo, para que las decisiones tomadas por el agente sean las óptimas se debe considerar el entorno que rodea a estos servicios dentro de la red. Shoham define un agente de software como una entidad de software la cual funciona continuamente y autónomamente en un ambiente en particular, a menudo habitada por otros agentes y procesos [5].

El entorno está definido como el conjunto de circunstancias que condicionan un hecho; considerando ésta definición se inferiere que medir el número de ocurrencias de un hecho sin tomar en cuenta su entorno ofrece información limitada por ejemplo, si al cuantificar el retardo extremo a extremo de los paquetes transmitidos en un enlace de red y el resultado es muy alto, el administrador de la red puede observar que existe un problema y querrá solucionarlo aumentando el ancho de banda del enlace de salida a Internet, pero no es sino

hasta que identifica que éste fenómeno sólo se presenta en determinadas horas del día las cuales corresponden a los horarios de entrada del personal y la hora del regreso del descanso de mediodía, lo que ayuda al administrador a llegar a la conclusión de que no es necesario el incremento del ancho de banda, ya que el aumento del retardo no es constante; en el momento que se considera el entorno del fenómeno se logra una administración más eficiente.

1.1. Antecedentes

Los agentes inteligentes son entidades autónomas capaces de almacenar conocimiento sobre sí mismos y su entorno, el cual le permite tomar decisiones para completar las tareas del usuario [35].

Las características de un agente inteligente son las siguientes:

1. **Autonomía:** Debe actuar sin ningún tipo de intervención humana directa y tener control sobre sus propios actos.
2. **Sociabilidad:** Un agente debe ser capaz de interactuar con otros agentes por medio de algún lenguaje de comunicación.
3. **Reacción:** El agente posee la habilidad de percibir y reaccionar de acuerdo al entorno. Esto deriva en una capacidad de adaptación.
4. **Proactividad:** Emprende las acciones necesarias para resolver un problema, no solo reacciona.
5. **Movilidad:** El agente tiene la capacidad de transportarse de una máquina a otra y entre sistemas de arquitecturas diferentes.
6. **Aprender:** Los agentes deben ser capaces de aprender, actualizando su conocimiento y

nivel de inteligencia. Cuando un agente trata de adaptarse a un nuevo medio ambiente o nuevas condiciones se dice que el agente actualiza su conocimiento.

Considerando las características anteriores los agentes inteligentes son la herramienta idónea para la autogestión de diversas funciones dentro del área de telecomunicaciones, específicamente en las redes autónomas. Éste tipo de redes cuentan con nodos inteligentes, los cuales incorporan agentes para la detección automática de cambios en los parámetros de calidad de servicio, así como para la toma de decisiones necesarias para aislar o anticiparse a los fallos; logrando así la proactividad en la red.

La gestión autónoma en redes es una aplicación que integra inteligencia computacional para aumentar el rendimiento de una red; haciendo uso de agentes inteligentes explotando su característica de autonomía, con el fin de operar sin intervención directa de seres humanos [11] .

Para lograr la autogestión de una red se requiere de más de un agente y es por ello que se utilizan sistemas multi-agentes, los cuales tendrán una respuesta más rápida para la resolución de fallos, congestión y administración de servicios [7],[28].

A diferencia de los sistemas de monitoreo y gestión de redes actualmente requieren de la intervención del administrador de la red para determinar qué acciones llevar a cabo en un fallo, ahora el agente debe reaccionar al evento y determinar automáticamente cómo solucionar el problema no de forma aislada sino trabajando de forma coordinada con otros agentes.

Los siguientes son algunos de ejemplos de los dominios de aplicación de sistemas multi-agentes:

1. **Evaluación de situaciones distribuidas:** con énfasis en cómo los agentes con distintos niveles de consciencia y control, pueden compartir sus interpretaciones locales

para responder a una situación, es decir, generar un diagnóstico en equipo.

2. **Programación y planeación de distribución de recursos:** éste tipo de sistemas se enfatizan en cómo los agentes pueden coordinarse para evitar conflictos sobre los recursos de un sistema, aumentando la eficiencia del mismo; estos sistemas pueden aplicarse a tareas que van desde programación de tareas en las fábricas, gestión de redes de telecomunicaciones, hasta el control de recursos en ambientes inteligentes.
3. **Sistemas expertos distribuidos:** considerando que los agentes comparten información y negocian para diseñar soluciones en conjunto, éste tipo de sistemas requiere de información de expertos en el área expresar un criterio [20].

La sensibilidad al entorno es sin duda alguna un atributo fundamental para la gestión de redes dinámicas mediante agentes inteligentes. La sensibilidad al entorno debe estar presente en todos los aspectos del ciclo de vida del servicio que requiere el usuario en el plano de control de flujo de datos.

A. K. Dey define entorno como cualquier información que pueda ser usada para caracterizar la situación de una entidad, entendiendo como entidad una persona, lugar u objeto que es considerado relevante para la interacción entre un usuario y una aplicación, incluyendo a los usuarios y aplicaciones mismas [1]. Un sistema es sensible al entorno si se utiliza para obtener información relevante a los servicios ofrecidos al usuario.

Los servicios que requieren los usuarios dentro de una red pueden clasificarse de acuerdo a la calidad de servicio (QoS; del inglés *Quality of Service*) requerida, sobre todo las aplicaciones sensibles al retardo como son video, voz y aplicaciones en tiempo real.

El asegurar que esa QoS se cumpla requiere de un gran esfuerzo humano, muchas veces se deben bloquear puertos, tráfico menos relevante o dar prioridad al tráfico de determinada dirección de Internet; sería más conveniente que un conjunto de agentes se comuniquen y

adapten el tráfico en la red automáticamente de acuerdo a éstos requerimientos de QoS.

Una arquitectura de agentes describe los diferentes módulos que forman parte de un agente y cómo éstos se interconectan para que el agente pueda exhibir una determinada conducta. Los tipos de arquitecturas de agentes se enlistan a continuación:

1. **Arquitecturas Deliberativas:** los agentes con una arquitectura deliberativa se caracterizan porque contienen explícitamente un modelo simbólico del entorno, y las decisiones son tomadas vía razonamiento lógico basado en equiparación de patrones y manipulación simbólica. Además de una representación simbólica del entorno, los agentes deliberativos tienen capacidad para llevar a cabo decisiones lógicas utilizando el conocimiento que con el que cuentan y modificando su estado interno, que frecuentemente se denomina estado mental. Lo más común es que el estado mental de los agentes deliberativos, que normalmente son llamados agentes BDI (del inglés *belief, desire, intention*) [35].
2. **Arquitecturas Reactivas:** los agentes reactivos se caracterizan porque pueden operar rápida y efectivamente sin la necesidad de procesar una representación simbólica del entorno, ya que representan una categoría de agentes que no posee modelos simbólicos externos del entorno en el que se encuentran. El funcionamiento de un agente reactivo es básicamente como sigue: los sensores recogen información, que es directamente enviada a los módulos de competencia [35].
3. **Arquitecturas Híbridas:** aprovechan las características de ambas arquitecturas antes definidas la deliberativa y reactiva, ya que para la mayoría de los problemas no es adecuada ni una arquitectura puramente deliberativa ni una arquitectura puramente reactiva. Estas arquitecturas se diseñan siguiendo una arquitectura jerárquica en capas, en la que las capas más bajas son principalmente reactivas y las capas más altas son principalmente deliberativas [35].

1.1.1. Justificación

Resolver la autogestión de redes en el plano de control de flujo de datos es un problema complejo. Debido a que los algoritmos de enrutamiento de flujos de información tradicionalmente buscan la ruta con un menor costo, sin considerar la capacidad actual de cada uno de los enlaces presentes en la ruta, como consecuencia de esto en ocasiones la ruta definida para la transmisión de la información contiene enlaces congestionados, lo que ocasiona un aumento en el retardo de transmisión de los paquetes o la pérdida de los mismos, repercutiendo en la QoS al usuario de la red. Una arquitectura multi-agente que implemente algoritmos de colaboración que permitirá desarrollar una autogestión de redes en el plano de control de flujo.

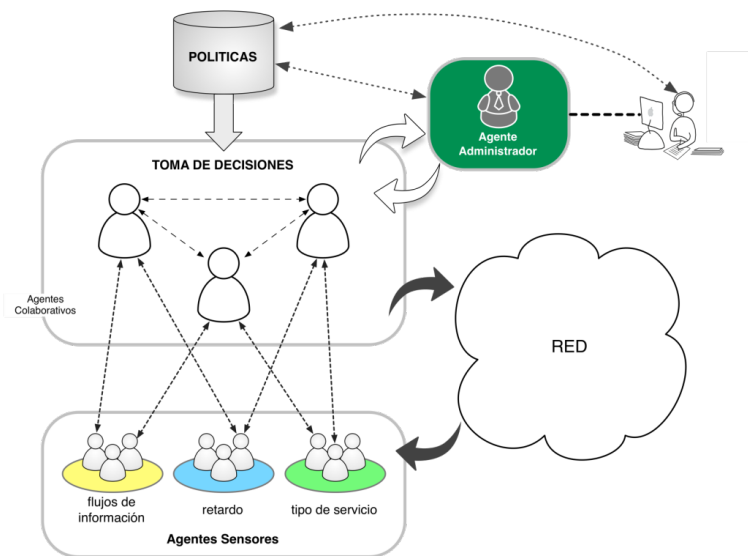


Figura 1.1: Esquema propuesto como base para el diseño de una Arquitectura Multi-agente.

Este proyecto de investigación es un esfuerzo hacia lograr la autogestión en redes en el plano de control de flujo de datos, mediante el análisis, diseño e implementación de una arquitectura multi-agente, que incremente el nivel de automaticidad de la red logrando con eso mejorar su desempeño y optimizar la gestión de recursos, un esquema de la arquitectura

multi-agente propuesta se muestra en la Figura 1.1.

En la Figura 1.1 se observa la arquitectura jerárquica propuesta, basada en la interacción de varios conjuntos de agentes los cuales deben clasificarse de acuerdo al proceso y objetivo que le corresponde. En el nivel bajo existe un conjunto de agentes que recolectan la información de los parámetros de calidad de servicio sensando el estado actual de la red.

Estos agentes alimentan los datos obtenidos al siguiente nivel superior donde se lleva a cabo la toma de decisiones, por medio de un mecanismo de negociación entre los agentes colaborativos mismos que son gobernados por las políticas establecidas por el administrador de la red. Como resultado de la negociación se deben establecer las acciones a realizar por los agentes actuadores del nivel bajo, las cuales se reflejan en la optimización del estado de la red. El administrador de la red es el encargado de alimentar las políticas de alto nivel en un base de datos, la cual conforme el sistema se auto-gestiona puede reestructurarse de forma autónoma.

Finalmente abordar un problema desde la perspectiva de sistemas complejos permitirá analizar cada uno de los patrones de comportamiento emergente dentro de la arquitectura multi-agente propuesta en la Figura 1.1, dando como resultado la implementación de un algoritmo de negociación entre agentes que active la auto-organización de la distribución de los flujos de red cuando se presenta un congestionamiento en la red.

1.1.2. Planteamiento del problema

El mundo de las telecomunicaciones evoluciona a pasos agigantados, lo que repercute en el aumento de servicios y aplicaciones demandando mayor ancho de banda, el cual es el recurso más grande para cualquier administrador de red y por lo tanto hay que saber administrarlo. Sin embargo el ancho de banda se reparte entre los distintos flujos de tráfico en la red, ya sean éstos flujos de datos, video, voz o control; provengan del interior de la red o del exterior.

Todos los flujos de tráfico, sin importar su naturaleza, consumen un determinado ancho de banda por lo que para hacer un uso eficiente de éste recurso es necesario desempeñar una administración minuciosa.

Los sistemas de gestión de redes actuales permiten al administrador de la red configurar manualmente al servidor que va a monitorear los dispositivos que son claves para mejorar el desempeño de la red, desafortunadamente no basta con sensar ciertos nodos de la red o recibir una alerta para implementar acciones en el caso de un posible fallo, si al presentarse alguna situación problemática es necesario que un individuo deba solucionar los problemas.

Desafortunadamente en el dominio de administración y gestión de red a nivel operacional no se ha evolucionado tanto como en el desarrollo de aplicaciones para dispositivos de red, las que trazan su vía de comunicación entrelazando distintos tipos de redes y protocolos, concluyendo gestionar una red se ha vuelto una problema complejo [14], [31]. La tendencia para la siguiente generación de redes es incorporar inteligencia y autonomía en sus dispositivos, para lograrlo es necesario incorporar algoritmos cognitivos que ejecuten procesos de optimización en tiempo real, con el propósito de mejorar el desempeño de la red de acuerdo a las políticas establecidas por el administrador [33]. Resolviendo el problema de la centralización de la información de gestión se tiene la esperanza que los recursos de la red se puedan auto-gestionar por ellos mismos, es decir, se auto-configuren, auto-reparen, auto-optimicen y auto-protejan [7].

La necesidad de optimizar la administración de los recursos de una red ha llevado al ser humano a incorporar dispositivos que automaticen el cumplimiento de las políticas establecidas por el administrador. La optimización implica buscar la mejor manera para realizar una actividad, para esto se requiere del razonamiento y no únicamente de una automatización. Debido a ésta diferencia hoy en día se proponen arquitecturas autonómicas las cuales incorporan un subsistema de monitoreo autonómico externo de la red, que genera una base de

datos de conocimiento la cual se toma de referencia para cumplir con las políticas y actuar proactivamente en la gestión de la red, con la mínima intervención humana.

1.1.3. Hipótesis

Una arquitectura multi-agente de autogestión de redes que incluya mecanismos de colaboración para la toma de decisiones basados en las políticas, integrada al plano de control de una red permitirá optimizar la gestión de flujos de tráfico y garantizar una carga balanceada de los recursos de la red.

1.1.4. Objetivo General

El objetivo general comprende el estudio y aplicación de sistemas multi-agentes conscientes de su entorno como una herramienta esencial para el establecimiento de una arquitectura de autogestión de redes en el plano de control de flujo de datos, considerando los parámetros que miden la calidad de servicio en las aplicaciones sensibles al retardo.

1.1.5. Objetivo Específico

El objetivo específico es generar modelos y desarrollar herramientas que involucren a los agentes inteligentes como monitores, los cuales mediante su programación asignen las rutas para el control de flujo de datos de forma autónoma, éstos agentes deben tomar en cuenta el entorno y adaptarse dinámicamente al él, tomando en cuenta modelos biológicos y teoría de juegos para maximizar la eficiencia y minimizar el retardo en el flujo de datos.

1.1.6. Metodología

La metodología aplicada en el análisis está basada en la teoría de sistemas complejos, la cual permite analizar a un sistema por medio de las interacciones de todas las partes que lo conforman con el propósito de descubrir los patrones emergentes que se generan de acuerdo al entorno que rodea al sistema. Este tipo de metodología basada en sistemas complejos permite además incorporar la teoría de grafos haciendo uso de las métricas de redes complejas, las cuales se basan en la matriz de adyacencia que interconecta a una red, debido a que el sistema multi-agente está ubicado en un entorno de red.

1.1.7. Estructura de la tesis

La organización de la tesis consta de siete capítulos los cuales se describe su contenido brevemente a continuación. El primer capítulo aborda los antecedentes, planteamiento del problema, hipótesis, objetivos de investigación y metodología en la cual se basa el análisis y desarrollo de tesis. En el capítulo dos se describe el marco teórico, el cual incluye una revisión del estado del arte de la tecnología y teorías aplicadas en el proyecto de investigación. En el tercer capítulo se desarrolla una comparación entre los tipos de arquitecturas de redes autónomas existentes. El capítulo cuatro describe las métricas de redes complejas, las cuales se toman como base para la formulación de un algoritmo de negociación descrito en el capítulo cinco, en el que se describe la arquitectura multi-agente propuesta, así como el modelo de agente que se implementa. En el capítulo seis se describen los casos de estudio, se muestran los resultados obtenidos y su análisis. Por último en el capítulo siete se presentan las conclusiones de la investigación y la propuesta de trabajo futuro.

Capítulo 2

Marco teórico

2.1. Introducción

En este capítulo se brinda una introducción al estándar de gestión de redes y una introducción a la gestión autónoma en redes.

2.2. Gestión de Redes

La tarea de gestionar una red implica establecer criterios temporales por parte del administrador de la red con el propósito de garantizar una adecuada QoS. Estos criterios no pueden ser estáticos debido al dinamismo de las aplicaciones y servicios que brinda la red, en consecuencia el administrador debe auxiliarse por arquitecturas de gestión de redes las cuales le permitan optimizar la asignación de recursos de la red [18].

Debido a que las redes son cada vez más complejas e indispensables, el implementar un sistema de gestión de red es imperativo para las organizaciones, algunos de los gestores de red actuales requieren una optimización en diferentes aspectos: automatización,

reencaminamiento de tráfico, seguridad y capacidad de recuperación; éste último tiene una mayor relevancia que el resto ya que repercute en una reducción de tiempo sin servicio y por lo tanto mejora la QoS.

La Organización Internacional de Estándares (ISO; del inglés *International Organization for Standardization*) define un modelo de gestión de redes en el cual especifica cinco áreas funcionales en las que se describen la gestión de fallos, configuración, desempeño, seguridad y contabilidad [4].

- **Gestión de fallos:** Implica la detección, aislamiento, notificación y corrección de fallos de la red.
- **Gestión de configuración:** Describe los aspectos de configuración de dispositivos, archivos, inventarios y software.
- **Gestión de desempeño:** Monitorea y genera mediciones de distintos ámbitos de la red con el propósito de mantener al sistema en con un rendimiento aceptable.
- **Gestión de Seguridad:** Administra la autorización para el acceso de recursos y dispositivos de la red.
- **Gestión de Contabilidad:** Utiliza la información de los recursos de la red para la asignar y contabilizar los costos.

La gestión de red puede llevarse a cabo de forma centralizada o distribuida dependiendo del tipo de plataforma utilizada. Existen distintos análisis en los que se basan las plataformas de gestión de red los más comunes son: descubrimiento de la topología de la red para el mapeo de sus elementos, manejador de eventos o alarmas y graficador de datos de desempeño de los recursos de la red. Estas plataformas funcionan mediante una consola principal en la cual se muestran los problemas de la infraestructura de la red detectados, donde la persona

encargada de la red toma decisiones basada en las notificaciones obtenidas, es decir, su gestión es centralizada [4].

La gestión distribuida permite administrar una de las características principales de las redes que es la escalabilidad, utilizando múltiples estaciones de gestión distribuidas en las distintas redes de área local evitando que un sólo nodo administre toda la información de gestión, con esto se elimina el riesgo de un único punto de ataque de los esquemas centralizados.

2.2.1. Paradigma Gestor-Agente

El paradigma Gestor-Agente es un instrumento que se aplica en la mayoría de los sistemas de gestión de redes, el cual permite el intercambio de información entre nodos gestores y agentes que gestionan a los elementos de red.

Nodo Gestor: Es un dispositivo autónomo el cual tiene la capacidad de traducir los requisitos del administrador de red en la monitorización y control de los elementos remotos de la red, consta de una base de datos de información de gestión y funciona como una interfaz entre el administrador de red y el sistema de gestión de red.

Agente: Responde a las solicitudes de información y realiza acciones efectuadas desde la estación gestora, se ubica en los dispositivos de interconexión de red (nodos gestionados).

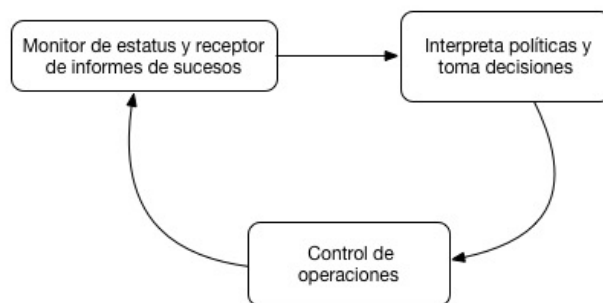


Figura 2.1: Ciclo de gestión de redes [4].

El esquema del funcionamiento que sigue un sistema de gestión se muestra en la Figura 2.1. El flujo de la información de gestión sigue un comportamiento que puede ser observado en la Figura 2.2. Los nodos gestionados a través de agentes, proporcionan la información a los gestores de la red. Los gestores a partir de los parámetros definidos en sus políticas de gestión actúan sobre la red mediante mensajes de control sobre los agentes de los nodos, optimizando el funcionamiento por medio de cambios de configuración, éste control realizado sobre la red modifica las condiciones del tráfico y el ciclo se repite realizando nuevas mediciones sobre los recursos [4].



Figura 2.2: Flujo de información de gestión [4].

2.3. Gestión Autónoma de Redes

Las arquitecturas de gestión de redes autónomas son un caso especial de computación autónoma, las cuales tienen la capacidad de operar y administrar la red de forma autónoma, sin intervención externa inclusive en caso de cambios en su entorno. Su objetivo es tratar de llevar fuera al administrador humano del ciclo de control de red, dejando las tareas de gestión a realizar por la propia red [31], [33], [25], [17]. Un sistema que implementa una arquitectura de gestión de red autónoma tiene las siguientes propiedades:

- **Auto-Configurado:** Se refiere a la capacidad del sistema de configurarse y reconfigurarse por sí mismo, de acuerdo con las políticas de alto nivel en un entorno cambiante.
- **Auto-Optimizado:** Habilita la operación eficiente del sistema incluso en entornos impredecibles. Un sistema de computación autónomo siempre buscará de forma proactiva

la oportunidad de mantener un desempeño eficiente minimizando su costo.

- **Auto-Reparado:** Consiste en la capacidad de descubrimiento y reparación de los problemas potenciales, para asegurar un buen funcionamiento del sistema, es decir, la predicción de problemas y toma de acciones proactivas para prevenir fallos o reducir su impacto.
- **Auto-Protegido:** Se refiere a la característica del sistema para protegerse a sí mismo de cualquier situación que comprometa sus objetivos, por ejemplo de ataques maliciosos, tentativa intrusión o fallos inadvertidos.

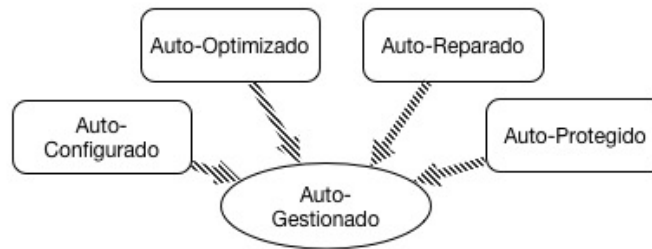


Figura 2.3: Propiedades de la Auto-Gestión.

En la Figura 2.3 se muestra las relaciones entre las propiedades de autogestión.

IBM (del inglés *International Business Machines*) propuso un modelo genérico para describir el Ciclo de Control Autónomo (ACL; del inglés *Autonomic Control Loop*) llamado MAPE-K [15], mostrado en la Figura 2.4. El gestor autónomo alcanza su auto-gestión gracias a un monitoreo de los componentes gestionados y otros módulos que analizan los datos obtenidos, planifican las tareas de adaptación y ejecutan los planes que se decidieron en la red.

Los módulos que constituyen al ACL se enumeran a continuación:

1. **Monitor:** el cual está a cargo de capturar las mediciones necesarias del entorno que son significativas para mantener las auto-propiedades de la red subyacente.

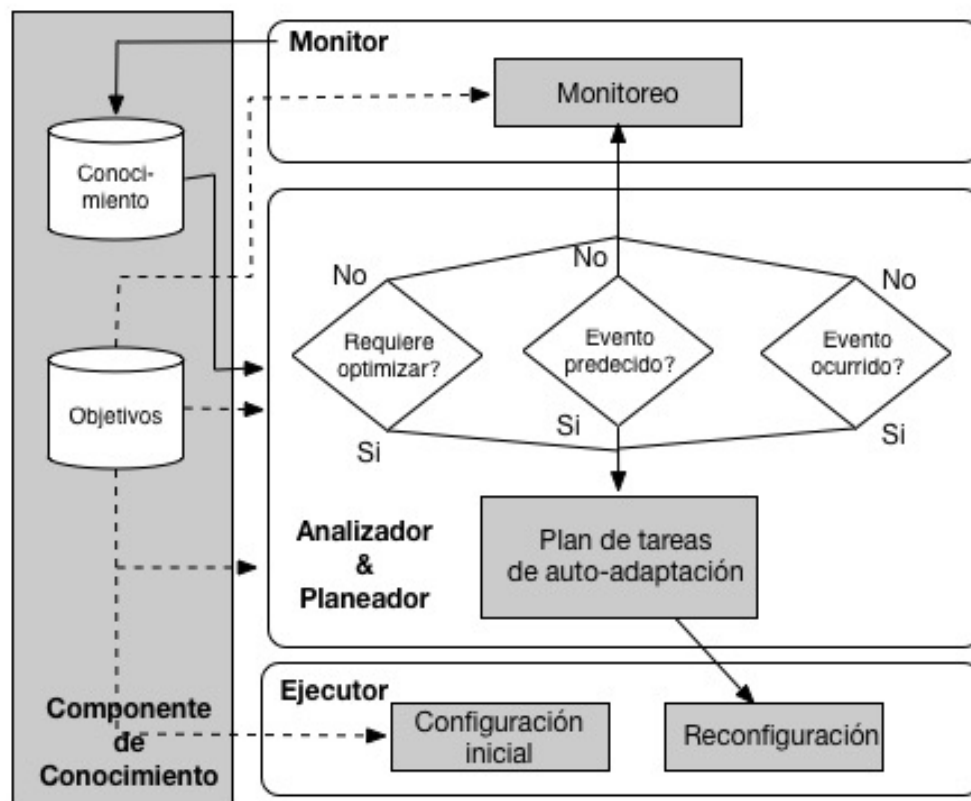


Figura 2.4: Modelo de referencia del Ciclo de Control Autónomo [15]

2. **Base de objetivos:** es un repositorio donde se almacenan las políticas de alto nivel.
3. **Base de conocimiento:** donde se agregan y almacenan los datos recolectados.
4. **Analizador:** es un repositorio el cual analiza el conocimiento de acuerdo a las políticas (objetivos de alto nivel) almacenadas; este verifica el desempeño actual, predice el estado futuro de la red y detecta eventos.
5. **Planificador:** construye los planes de adaptación en base a los resultados del proceso de análisis.
6. **Ejecutor:** constituye la salida del proceso de planificación, reconfigura y gestiona los componentes con otros gestores autónomos.

El grado de automaticidad puede ser medido en cinco niveles de acuerdo a la propuesto por IBM:

1. *Nivel 1*: se refiere al monitoreo continuo de los elementos del sistema por medio herramientas de software manejados por personas que dan soporte de forma manual en caso de algún fallo.
2. *Nivel 2*: consiste en un conjunto de aplicaciones de monitoreo que recolectan información de manera inteligente para disminuir la carga de trabajo del administrador de la red es conocido como nivel de *gestionamiento*.
3. *Nivel 3*: este nivel tiene la capacidad de reconocer ciertos patrones de comportamiento de la red además de sugerir acciones a tomar al personal de soporte, es llamado también como *predictivo*.
4. *Nivel 4*: o nivel *adaptivo* utiliza las herramientas del nivel 3 tratando de minizar la intervención humana ayudándose de los Acuerdos de Nivel de Servicio (SLA; del inglés *Service Level Agreements*).
5. *Nivel 5*: los sistemas de en éste nivel son gestionados de forma dinámica por políticas y objetivos de negocio, su desventaja es que la automaticidad es muy cerrada, únicamente se basan en ciertas reglas, éste nivel es reconocido como *autónomo* otorgando mayor libertad a los administradores de red [25].

Una red comprende analizar, planificar, prevenir y generar la proactividad de la red, de tal forma que el monitoreo de un subconjunto de los elementos que conforman la red genere los datos necesarios para que los SLA sean cumplidos de forma autónoma, minimizando la intervención del administrador.

Capítulo 3

Arquitecturas de Redes Autónomas

3.1. Introducción

En éste capítulo se describen algunas arquitecturas de gestión de redes que incluyen mecanismos de razonamiento derivados de las áreas de inteligencia artificial, las cuales sirven de guía para identificar las métricas de la arquitectura propuesta.

3.2. Clasificación de Arquitecturas Autónomas

La autogestión de redes o redes autónomas es una aplicación actual la cual se aplica de herramientas de inteligencia artificial como lo son agentes inteligentes y minería de datos entre otros. Las arquitecturas autónomas pueden ser clasificadas de acuerdo a su estructura.

1. Arquitecturas Autónomas Jerárquicas:

- a) Internet Autónomo (AutoI; del inglés *Autonomic Internet*) Proyecto Europeo FP7.
- b) Redireccionamiento y Gestión Dinámica para el Ejército (DRAMA; del inglés *Dy-*

namic Readdressing and Management for the Army).

- c) Sensibilidad al Contexto para la Gestión Autónoma de MANETs (CA-MANETs; del inglés *Context-Awareness for the Autonomic Management of MANETs*).
- d) Unity.

2. Arquitecturas Autónomas No Jerárquicas:

- a) Arquitectura de Gestión Descentralizada Autónoma (ADMA; del inglés *Autonomous Decentralized Management Architecture*).
- b) Arquitectura de Red Autónoma (ANA; del inglés *Autonomic Network Architecture*).
- c) Gestión en Red (INM; del inglés *In-Network Management*).

3.3. Internet Autónomo (AutoI)

El objetivo del proyecto AutoI financiado por la Unión Europea, es el de proponer una nueva arquitectura de gestión autónoma por medio de distintos planos los cuales permitirán a la red crear, destruir, implementar y migrar de forma autónoma los servicios virtuales y elementos virtuales de la red [21]. Esta arquitectura se muestra en la Figura 3.1.

- a) *Plano de Virtualización*: Abstrae todas las cuestiones de virtualización de los recursos físicos de la red permitiendo su puesta en marcha y reconfiguración.
- b) *Plano de Gestión*: Se basa en la administración de los distintos Sistemas de Auto-Gestión (AMS; del inglés *Autonomic Management System*) los cuales basan su comportamiento en el ciclo de control llamado MAPE el cual involucra los procesos de monitoreo, análisis, planeación y ejecución, el esquema de éste ciclo se observa en la Figura 3.2. Cada sistema

de auto-gestión administra a un conjunto de dispositivos con políticas o acuerdos de nivel de servicios en común los cuales representan un dominio.

- c) *Plano Habilitador de Servicios*: Su tarea es el establecimiento y habilitación de nuevos servicios empleando los recursos virtuales.
- d) *Plano de Conocimiento*: Distribuye la información entre planos, además de implementar motores de inferencia que le permiten generar conocimiento sobre la información de gestión y su contexto.
- e) *Plano de Orquestación*: Este plano actúa únicamente como mediador entre los sistemas de auto-gestión debido a su origen autónomo detectando y administrando conflictos entre ellos. Utiliza un modelado de información común basado en un conjunto de ontologías.

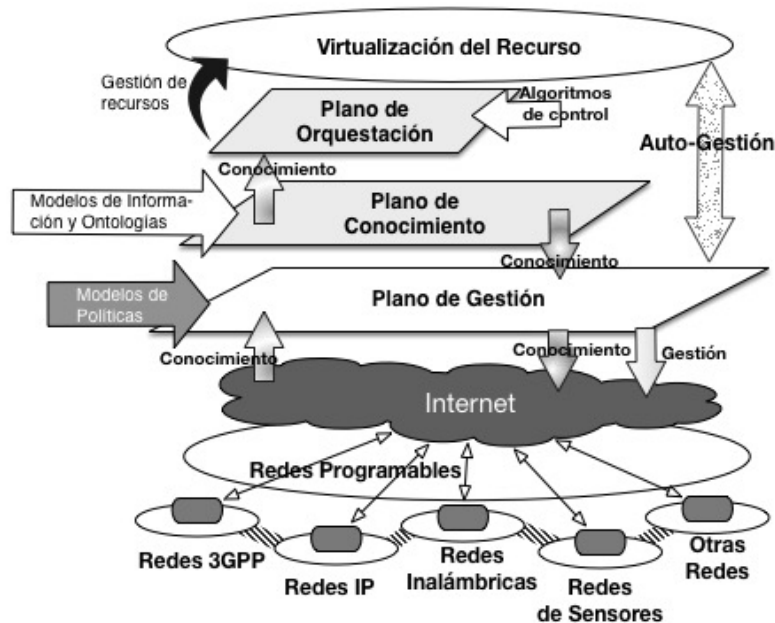


Figura 3.1: Arquitectura de AutoI [21].

Esta arquitectura implementa un protocolo de negociación que utiliza coaliciones [29], el cual se basa en el enfoque de redes sociales con topologías fijas; considera cierto número

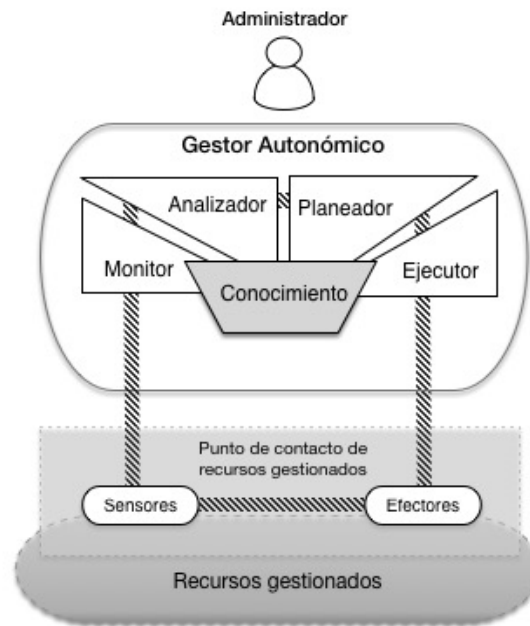


Figura 3.2: MAPE ciclo de gestión.

de proveedores de servicio de red candidatos a formar parte de la coalición dependiendo los servicios que cada uno ofrece y el nivel de desempeño que garantiza.

3.4. Redireccionamiento y Gestión Dinámica para el Ejército (DRAMA)

DRAMA es una arquitectura de gestión de red jerárquica centralizada basada en políticas para redes móviles, comúnmente llamadas MANET (del inglés *Mobile Ad Hoc Networks*) la cual utiliza agentes inteligentes. La arquitectura basada en jerarquías se muestra en la Figura 3.3, donde se puede observar a un conjunto de agentes de política que gestionan a los nodos que conforman la red. En el nivel más alto de ésta arquitectura se tiene un Agente de Política Global (APG) el cual administra a los Agentes de Política de Dominio (APD), los cuales a su vez pueden gestionar a los Agentes de Política Locales (APL) y éstos últimos a los nodos.

Los agentes de políticas reaccionan de acuerdo a los cambios en el estatus de la red de forma automática en distintos niveles con el propósito de reconfigurar la red para optimizar su desempeño [6].

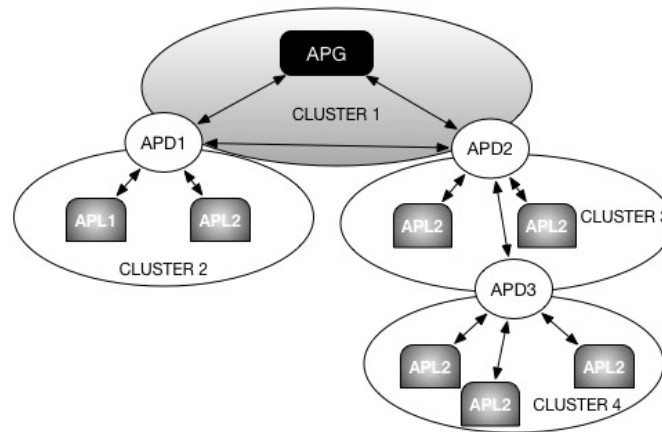


Figura 3.3: Jerarquía de gestión de DRAMA [6].

Las políticas se implementan en el nivel más alto de la jerarquía las cuales son almacenadas en un repositorio, cada política es generada de acuerdo al nivel de desempeño deseado en la red. Estas políticas cuentan con cuatro componentes: evento, condición, acción y alcance; éste último puede ser a nivel global y local. Esta arquitectura crea una jerarquía de gestión de varios niveles, el primer paso es formar un cluster inicial de forma manual así como las asignaciones de su líder, éste transmite un mensaje (latido de corazón) para todos sus clusters asociados. La creación de clusters en distintos niveles permite una gestión escalable del sistema.

3.5. Sensibilidad al Contexto para la Gestión Autónoma de MANETs (CA-MANETs)

CA-MANETs es una arquitectura de autogestión de redes jerárquica distribuida en capas, basada en políticas y utiliza la formación de hiper-cluster [22]. Esta arquitectura divide la red en clusters para la asignación de módulos y roles. En su modelo se definen módulos de software clasificados en CG (Cluster Gestor) y NT (Nodo Terminal). Además propone tres roles los cuales son: NG (Nodo Gestor), CC (Cluster Cabeza) y NC (Nodo de Cluster) en la Figura 3.4.

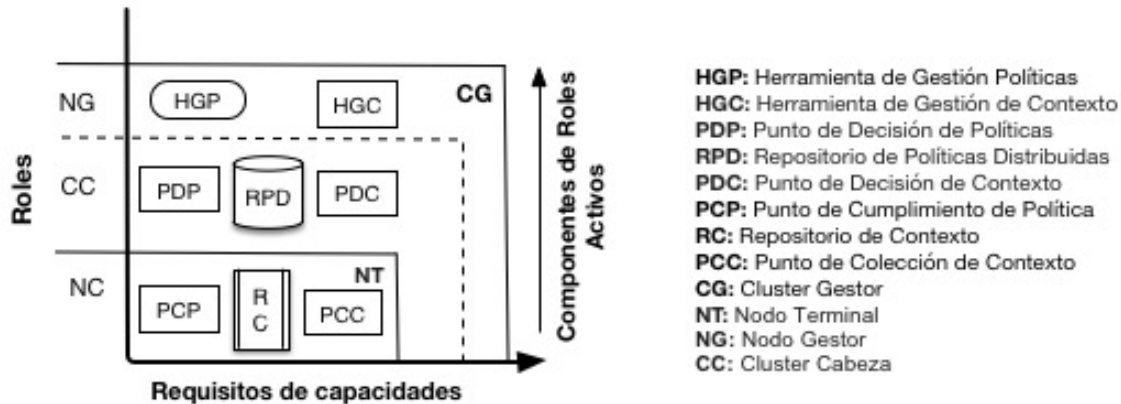


Figura 3.4: Roles y Módulos de CA-MANET [22].

De acuerdo a la clasificación anterior de roles y módulos ésta arquitectura se auxilia de un ciclo de control adaptativo el cual impacta en distintos niveles de la jerarquía, es decir existe un ciclo de control adaptativo a nivel cluster e hiper-cluster. La formación de un hiper-cluster consiste en un conjunto de clusters gestores los cuales tienen asignados roles de nodos gestores o nodos cabeza debido a que éstos cuentan con información del contexto la Figura 3.5 muestra la organización de los nodos de acuerdo al modelo organizacional de la arquitectura CA-MANETs.

El modelado del contexto en CA-MANET es muy importante la propuesta muestra cuatro

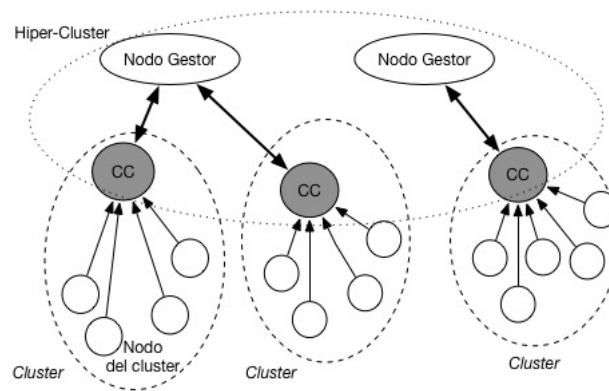


Figura 3.5: Arquitectura de CA-MANET [22].

entidades principales:

- a) **Punto de Colección de Contexto (PCC):** Es responsable de la comunicación con los sensores disponibles en los dispositivos, es el fundamento del nodo terminal y se ubica en el primer nivel de la jerarquía de roles. El PCC está constituido por un plano de comunicación el cual contiene un gestor de cluster, gestor de sensores y una interfaz de comunicación con los sensores. El siguiente es un plano de contexto cuyos elementos son un manejador de semánticas, modelador de contexto y optimizador de contexto. Otro plano es el de datos que colecciona los datos, a su vez los transfiere hacia un optimizador de datos el cual permite optimizar las reglas; todos los planos se comunican con un gestor de contexto local y éste los retroalimenta.
- b) **Punto de Decisión de Contexto (PDC):** Está ubicado en el segundo nivel de la jerarquía de roles, y se compone por varios planos uno de ellos es el de comunicación el cual contiene un gestor de clusters y monitores de los nodos que componen el cluster; el siguiente plano es de contexto el cual contiene tres módulos; el primero es un agregador de contexto, el segundo es el procesador de contexto que alimenta un tercero, un optimizador de contexto. Este último plano se retroalimenta por el repositorio de contexto y optimiza las reglas considerando además al gestor de contexto del cluster el cual es la entidad

principal del PDC.

- c) **Herramienta de Gestión de Contexto (HGC):** Situado el nivel más alto (tercero) de la jerarquía de roles, conformado por dos planos; uno es de comunicación, conteniendo un gestor de cluster y dos tipos de monitores de punto de decisión contexto y herramienta de gestión de contexto; el segundo es el plano de contexto que internamente contiene un colector de contexto MANET, un agregador de contexto de nodo gestor, el cual retroalimenta las reglas de inferencia y por último un optimizador de contexto para optimizar las reglas. Al igual que las entidades descritas en los incisos anteriores contiene un repositorio de contexto y un gestor pero en éste caso gestiona a la herramienta de gestión de contexto. En los nodos gestores existe la oportunidad de la intervención humana para ver la información recolectada por medio de una pantalla, el administrador puede tomar decisiones de gestión de forma directa o modificar las políticas por medio de la herramienta de gestión de políticas.
- d) **Repositorio de Contexto (RC):** El repositorio de contexto existe en todos los nodos de la red MANET, contiene información de acuerdo al rol al que pertenezca; incluye un plano de datos en el cual se ubican el indexador de servicios, el gestor de almacenamiento, responde de retroalimentar a las reglas de archivado y una base de datos con la información de contexto almacenada. El plano de datos se relaciona bidireccionalmente con un gestor de repositorio de contexto el cual es el elemento principal que interactúa con otros módulos.

3.6. Unity

Unity es una arquitectura basada en agentes y orientada a servicios la cual tiene las características de auto-configuración basándose en una meta establecida, utiliza centinelas mediante estrategias de regeneración de grupos para el auto-reparado y por último permite

la auto-optimización basada en funciones de utilidad [8]. Unity fue creada por el Centro de Investigación Thomas J. Watson de IBM para el envío de datos autónomo. En la Figura 3.6 se muestra un escenario de Unity, la cual tiene un árbitro que administra las aplicaciones. Los componentes de Unity son:

- a) **Gestor de entorno de la aplicación:** Es responsable de administrar los recursos que el ambiente requiere para obtener su meta, es decir, debe predecir cuando incrementar o decrementar los recursos asignados al ambiente de la aplicación.
- b) **Arbitro de recursos:** Decide los recursos que se deben asignar a las aplicaciones y calcula el impacto de asignar los recursos a las aplicaciones con el objetivo de llegar a un óptimo.
- c) **Servidor:** Cada servidor es responsable anunciar a los servidores las capacidades de tal manera que los usuarios puedan verlas.
- d) **ContenedorOS:** Representan a las computadoras capaces de soportar elementos autónomos, las cuales aceptan peticiones desde otros elementos para iniciar ciertos servicios. Algunas computadoras pueden tener ambos roles como servidor y ContenedorOS.
- e) **Registro:** Habilita a cada elemento para tener una comunicación basada en un registro de organización virtual (creada a partir del intercambio de recursos entre distintos elementos).
- f) **Repositorio de Políticas:** Permite al administrador humano alimentar las políticas de alto nivel que guiarán la operación del sistema.
- g) **Centinela:** Monitorea el funcionamiento de otro elemento y en caso de ser encontrado irresponsable el centinela notifica al elemento que solicitó la vigilancia. El centinela forma parte de la función de auto-reparado del *cluster*.

- h) **Gestor de Soluciones:** Este elemento representa la solución completa y es responsable del mantenimiento y de cualquier desajuste de la solución.

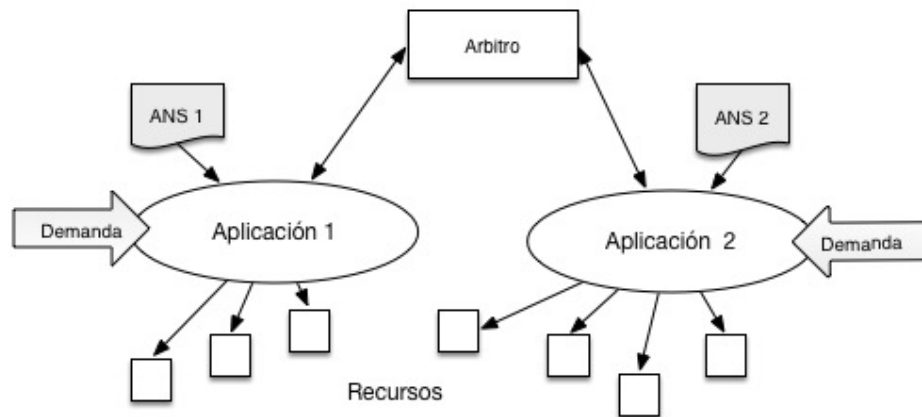


Figura 3.6: Escenario de Unity [8].

Una de las ventajas de utilizar funciones de utilidad en Unity es que se pueden aplicar en la función de auto-reparación de grupos de elementos autónomos, la toma de decisiones se basa en las utilidades las cuales respetan las políticas establecidas por el administrador humano.

3.7. Arquitectura de Gestión Descentralizada Autónoma (ADMA)

La arquitectura ADMA del inglés (Autonomous Decentralized Management Architecture) combina la gestión basada en políticas con principios autónomos dentro de un sistema descentralizado. Las políticas de alto nivel son implementadas por el administrador de la red antes de la formación de la red. Los nodos de la red cooperan entre ellos para lograr alcanzar las metas establecidas anteriormente por el administrador [10], [2].

En la Figura 3.7 se observan los elementos que componen a la arquitectura ADMA. Una breve descripción de cada uno se muestra a continuación:

- a) **Punto de Decisión de Política Local (PDPL):** Es la entidad que administra los recursos y lleva a cabo la toma de decisiones dentro del nodo sin dar cuentas a algún nodo central.
- b) **Punto de Cumplimiento de Política (PCP):** Es el encargado de traducir las decisiones de alto nivel y metas a políticas de bajo nivel, además de encargarse de su cumplimiento respetando al PDPL.
- c) **Monitor:** Recolecta información y la reporta al PDPL de acuerdo a las políticas.
- d) **Repositorio de Políticas (RP):** Es una base datos donde se almacenan el repositorio de políticas locales.

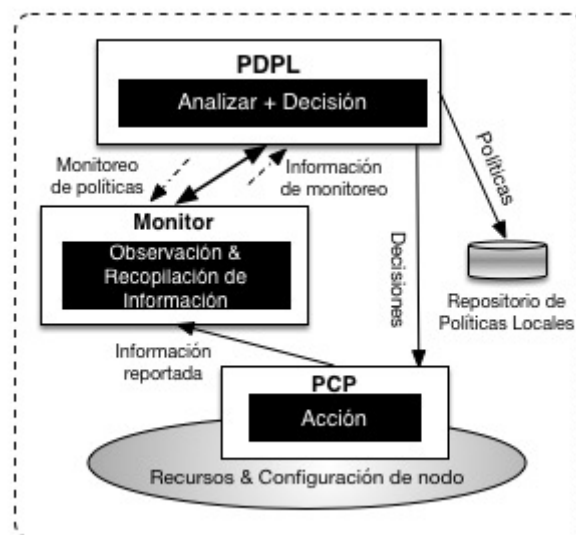


Figura 3.7: Arquitectura de un nodo ADMA [2].

Esta arquitectura no jerárquica tiene como característica principal que cada nodo cuenta con un componente local llamado Gestor de Red Basado en Políticas (GRBP) el cual consta

de un PDPL, PCP, RP y un monitor. ADMA permite que cada nodo busque alcanzar sus metas, mediante la distribución automática de las mismas utilizando el Protocolo Gestor de Distribución de Políticas (PGDP) en cada nodo de la red. Existen cuatro clases de políticas las cuales son: configuración, reconfiguración, monitoreo y meta-políticas (habilitan o deshabilitan políticas) [2]. El aporte de ésta arquitectura radica en la toma de decisiones locales basada en la observación y recolección de información para tomar una acción considerando las políticas locales.

3.8. Arquitectura de Red Autónoma (ANA)

ANA es un proyecto fundado en tecnologías futuras y emergentes. ANA utiliza compartimentos para separar los intereses en la red, éstos tienen sus propios mecanismos de adaptación. ANA define mecanismo de auto-adaptación guiado por políticas que permite a los intra-compartimentos la auto-optimización y ser conscientes de los intra-compartimentos [32]. En la Figura 3.8 se ilustra la abstracción del núcleo de ANA, la cual muestra un modelo de una comunicación unicast.

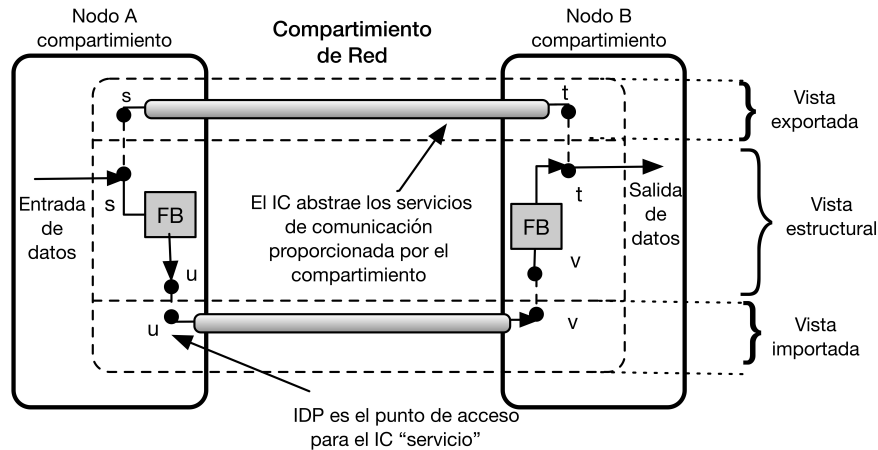


Figura 3.8: Abstracciones fundamentales de ANA [32].

El núcleo funcional de la arquitectura consiste en un componente micro-kernel llamado

Minmex (del inglés *Minimal Infrastructure for Maximal Extendibility*) el cual implementa las funcionalidades de ANA en los nodos. Minmex actúan como corredor de mensajes entre conmutación de servicios para los bloques funcionales (FB). Los bloques funcionales y sus Minmex implementan protocolos y funciones complejas donde se consigue la automaticidad de toda la red.

3.9. Gestión en Red (INM)

La arquitectura INM se fundamenta en cinco principios: intrinsicidad, inherencia, autonomía, abstracción y transición. La gestión está intrínseca a la red, ésto es la red se administra a sí misma [9].

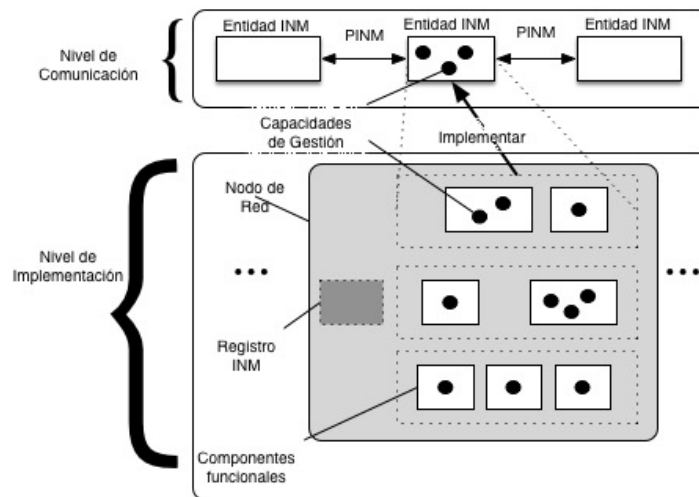


Figura 3.9: Arquitectur INM.

El principio de inherencia es parte de los elementos de la red, protocolos y servicios. La autonomía es completa, lo que indica que no hay intervención externa. Las operaciones externas solo ocurren al inicio de la red de acuerdo al principio de abstracción, el cual esencialmente permite gradualmente el desarrollo de tecnologías de la auto-gestión 4WARD (del

inglés *Wired and Wireless World Wide Architecture and Design*).

En la Figura 3.9 se observan los elementos de la arquitectura los cuales consisten en: entidades EINM dedicadas a la auto-gestión, protocolo PINM el cual es común para todas las operaciones entre las entidades INM, capacidades de gestión las cuales son entidades direccionables en el nivel de comunicación y por último el registro INMR contiene información sobre la estructura del plano de gestión de auto-organización es responsable para el mapeo entre la comunicación y el nivel de aplicación.

En las Tablas 3.1 y 3.2 se muestra un comparativo de las arquitecturas definidas, en las cuales se integran las métricas de jerarquía, número de planos, método de adaptación, ciclo de control, clusters, sistemas complejos, negociación, aprendizaje, tipo de red y agentes.

Tabla 3.1: Comparación de arquitecturas de red autonómicas parte 1.

Arquitecturas Métricas	AutoI [21]	DRAMA [6]	CA- MANET [22]	Unity [8]	INM [9]
Jerarquías	Distribuida	Centralizada	Distribuida	Distribuida	Plana
Planos	Orquestación, Conocimiento, Gestión	Global, Domi- nio, Local	Roles: Gestor, Cabeza, Nodo	Uno	Uno
Método de adaptación	Políticas en cua- tro niveles: Or- questación, Sis- temas, Compone- ntes e Instancia	Políticas en ca- da nivel: Global, Dominio y Local	Basado en políticas	Funciones de utilidad de asignación de recursos basadas en políticas	Red química ba- sada en políticas
Ciclo de Control	MAPE: Moni- tor, Analizador, Planeador y Ejecución	Ciclo de retro- alimentación au- tomática	Ciclo de adapta- ción de red am- plia		Ciclo de control de retroalimen- tación de redi- miento
Clusters	Identificación	Lider de cluster para asignación de políticas en nivel	Niveles de Clus- ters: Amplio, Hi- per, Gestor, Ca- beza y Nodo	Repositorio de políticas para a clusterización	Nodos NetInf

Tabla 3.2: Comparación de arquitecturas de red autonómicas parte 2.

Arquitecturas Métricas	AutoI [21]	DRAMA [6]	CA- MANET [22]	Unity [8]	INM [9]
Clusters	Identificación	Lider de cluster para asignación de políticas en nivel	Niveles de Clusters: Amplio, Hyper, Gestor, Cabeza y Nodo	Repositorio de políticas para a clusterización	Nodos NetInf
Sistemas Complejos	Redes Sociales				
Negociación	Coalición		Paradigma multi-gestor	Árbitro centralizado para la asignación de recursos vía funciones de utilidades	
Aprendizaje	No	No	No	No	No
Tipo de Red	Virtuales	Ad Hoc Móviles	ad Hoc Móviles	Ambiente Grid	Red Estática y Dinámica
Agentes		Si		Si	No

Capítulo 4

Redes Complejas

Una red compleja está compuesta por un conjunto de entidades que se interconectan por medio de enlaces, cuyas características definen un fenómeno en particular de la vida real [7], en donde las entidades pueden representar cualquier tipo de objeto, ya sea individuos, enrutadores, bacterias, entre otras; mientras que los enlaces que las interconectan muestran algún tipo de relación entre ellas, como pueden ser parentesco, capacidad de ancho de banda, dependencia, influencia, etc.

En teoría de grafos las entidades que conforman una red compleja son llamadas nodos y sus enlaces aristas. El uso de la teoría de grafos para analizar las relaciones que guardan los nodos que constituye una red, permite descubrir el grado de importancia de un nodo, la influencia de un nodo respecto a su vecindario, entre otros aspectos [8].

4.1. Métricas de Redes Complejas

Para determinar estas relaciones entre nodos se deben especificar ecuaciones las cuales indican una métrica que al ser aplicada a la matriz de adyacencia del grafo que representa

la red, generan patrones los cuales explican un comportamiento emergente de las entidades que componen a la red compleja.

4.1.1. Matriz de adyacencia

La matriz de adyacencia de una red es un modelo utilizado para analizar su topología. La matriz de adyacencia A_{ij} permite identificar las relaciones de los nodos que integran la red. Para propósitos matemáticos, a menudo representamos una red a través de su matriz de adyacencia. La matriz de adyacencia de una red dirigida de N nodos tiene N filas y N columnas, siendo sus elementos [3]:

- $A_{ij} = 1$ si hay un enlace que apunta del nodo j al nodo i
- $A_{ij} = 0$ si los nodos i y j no están conectados entre sí

Se utiliza la matriz de adyacencia para determinar la longitud de la ruta y la conectividad algebraica de los nodos entre otros aspectos. La Figura 4.1 ilustra una red con la distribución de los nodos que la constituyen; indicando el grado de conectividad por su color y tamaño.

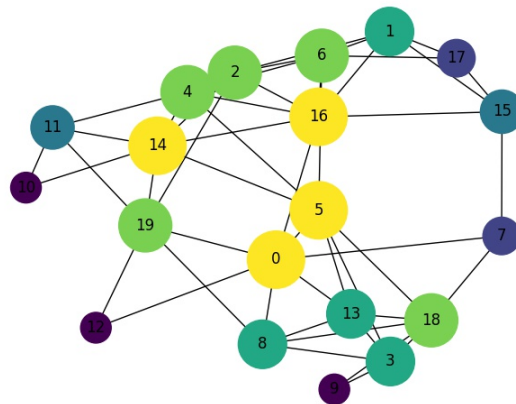


Figura 4.1: Topología de red distinción de grados de cada nodo.

4.1.2. Grado de un nodo

La métrica utilizada para indicar el grado de conectividad k_v del nodo v se determina con la Ecuación 4.1 la cual indica el número de enlaces que inciden con nodo v .

$$k_v = \sum_{i \in N} v_{ij} \quad (4.1)$$

Donde v_{ij} ($i, j = 1, \dots, N$) es igual a uno cuando hay un enlace l_{ij} y cero si no existe. En la Figura 4.1 se muestra una topología donde la dimensión de cada nodo corresponde al grado de conectividad. Los valores de grado y centralidad de red la figura 4.1 se muestran en la Tabla 4.1.

Tabla 4.1: Métricas de grado y centralidad de la red de la Figura 4.1.

Nodo	Grado	Centralidad	Nodo	Grado	Centralidad
0	7	0.268	10	2	0.088
1	5	0.211	11	4	0.172
2	6	0.305	12	2	0.092
3	5	0.175	13	5	0.206
4	6	0.311	14	7	0.318
5	7	0.316	15	4	0.137
6	6	0.285	16	7	0.33
7	3	0.107	17	3	0.114
8	5	0.195	18	6	0.191
9	2	0.066	19	6	0.243

4.1.3. Coeficiente de agrupamiento

El coeficiente de agrupamiento se calcula mediante la aplicación de la Ecuación 4.2, ésta permite identificar que tan extendido está un nodo de una red, donde se refiere al número de vecinos del vértice v . Los científicos Watts y Strogatz en su trabajo sobre redes de mundo pequeño caracterizaron a los nodos altamente interconectados como nodos concentradores

[34],[27],[13].

$$C_v = \frac{k_v(k_v - 1)}{2} \quad (4.2)$$

4.1.4. Centralidad de un nodo

La métrica de centralidad calcula la relación de influencia que guarda un nodo de acuerdo a la centralidad de sus vecinos cuyo valor es determinado por mediante la obtención de los valores propios de cada nodo [24]. La Ecuación 4.3 muestra el valor propio de X_{i+1} del grafo al final de la iteración $(i + 1)$ para el nodo i .

$$X_{i+1} = \frac{AX_i}{\|AX_i\|} \quad (4.3)$$

Donde A representa la matriz de adyacencia del grafo de la red, $\|AX_i\|$ es el valor normalizado del producto de la matriz A y el vector de valores propios tentativos X_i al final de la iteración i [24], el cual representa el valor de la centralidad basada en el valor propio obtenidos de la matriz de adyacencia.

4.1.5. Centralidad de Intermediación

La métrica de centralidad de intermediación de un nodo, es la fracción de las rutas más cortas que pasan por el nodo k cuando se considera sobre todos los pares de nodos i y j [23]. La ecuación 4.4 define la intermediación de un nodo como sigue.

$$B_c(k) = \sum_i \sum_j sp(i, k, j) / sp(i, j) \quad i \neq j \neq k \quad (4.4)$$

Donde $sp(i, j)$ es el número total de caminos más cortos entre los nodos i y j , y $sp(i, k, j)$ es el número de caminos más cortos que pasan a través el nodo k [23], [12]. Esta métrica de centralidad ayuda a detectar el rol de un nodo en la distribución de información dentro de una red.

4.1.6. Longitud de la ruta

Dos aspectos importantes de una red informática son la longitud de la ruta la cual se calcula por el número de saltos necesarios para llegar a un nodo desde otro, dando como resultado la cantidad de nodos que describe la ruta. Debido a la dinámica de una red real las rutas pueden cambiar a lo largo del tiempo dando como consecuencia una variación en la matriz de adyacencia.

Las medidas antes mencionadas han sido ampliamente utilizadas para capturar la dinámica y estructura de una red, clasificar nodos para crear árboles de jerarquías y detectar patrones de interacciones en una amplia variedad de sistemas [24], [26], [16].

Capítulo 5

Modelo Basado en Agentes de Autogestión de Red para Control de Flujo en una Red

5.1. Introducción

El modelado basado en agentes es un enfoque útil para analizar el comportamiento de un sistema complejo bajo condiciones de profunda incertidumbre, debido a la capacidad de conectar comportamientos a nivel micro y macro [19]. En un entorno de red los comportamientos a nivel micro representan las interacciones de diferentes tipos de flujos (datos, video y voz), cada uno proporciona información para comprender el funcionamiento de la misma. En éste capítulo se presenta un modelo basado en agentes en un entorno de red, el cual integra un algoritmo de negociación que permite en situaciones de congestionamiento de los nodos generar una distribución de flujos considerando las políticas establecidas por el administrador de la red.

5.2. Arquitectura Multi-Agente para la Autogestión de Flujos en la Red (Teleka)

La arquitectura propuesta mostrada en la Figura 5.1 representa el modelo multi-agente para la gestión de flujos de red denominada Teleka, la cual está organizada por tres niveles jerárquicos los cuales son explicados a continuación.

- *Red:* Este nivel configura a cada uno de los nodos los cuales son alimentados con los flujos a ser transmitidos integrando la información de origen y destino, además de identificar la jerarquía de cada nodo utilizando su medida de centralidad. Los agentes asignados a cada nodo de la red recolectan información del estado de congestiónamiento sensando su bandera ECG el cual cambia su valor de cero a uno al momento de sobre pasar el umbral de congestiónamiento establecido por el administrador al iniciar la red. Por último éste bit ECG es el que activa el algoritmo de negociación basado en Elección Social con Autoridades Ocultas (ESAO).
- *Control:* En éste nivel los agentes detectan y clasifican los flujos, que llegan a cada nodo, a su vez los agentes proveen información al nivel de negociación donde se aloja el módulo de toma de decisiones.
- *Negociación:* El algoritmo ESAO es activado cuando emerge un estado de congestiónamiento. Como resultado, un conjunto de acciones son alimentadas a los agentes del nivel de Red, con el objetivo de optimizar el estado de la red, bajando el grado de congestiónamiento de la misma.

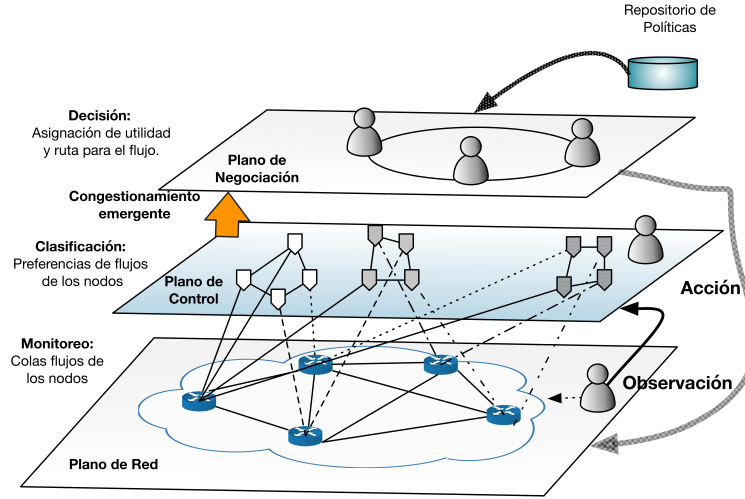


Figura 5.1: Arquitectura Multi-Agente para la Gestión de Flujos de Red (Teleka).

5.3. Modelado Computacional de los Agentes y su Entorno

De acuerdo a lo anterior se propone un mecanismo negociación que coordine a agentes en un medio ambiente cuyo escenario es una red compuesta por nodos N , enlaces L , y flujos F representada por una tupla de tres elementos: $Red = \langle N, L, F \rangle$.

Donde los Nodos N son representados por un conjunto finito de M elementos: $N = \{n_0, n_1, \dots, n_M\}$. En cual cada nodo n_i se describe con una tupla: $n_i = \langle \lambda, g \rangle$.

Donde λ representa el tamaño actual de la cola de espera de paquetes y es el grado del nodo. Cada nodo N_i tiene un estado de congestión ecg cuyo valor depende la función descrita en la Ecuación 5.1.

$$ecg(N) = \begin{cases} 1 & \text{if } \lambda > p * z \\ 0 & \text{if } \lambda \leq p * z \end{cases} \quad (5.1)$$

Donde z corresponde al tamaño máximo de paquetes en la cola y es el umbral (porcentaje de llenado de la cola) que especifica el administrador. Las colas son del tipo FIFO (del inglés

First In First Out). El grado del nodo esta descrito por la centralidad del nodo mediante el cálculo de la Ecuación 4.3 .

Los Enlaces $L=\{l_1, l_2, \dots, l_k\}$ es un conjunto finito de K elementos, los cuales tienen asignado un costo para cada enlace $l_i = < \omega >$.

Los flujos F se describe por un conjunto finito de J flujos navegando por la red. Los flujos están formados por una tupla de tres elementos: $f_i = < \beta, \theta, u >$ Donde β es la clase de flujo y θ es la prioridad de cada tipo de flujo. La clasificación β de los flujos es la siguiente: $\beta_i = \{\beta_1 = Video, \beta_2 = Voz, \beta_3 = Datos\}$. Por último cada tipo de flujo tiene asignada una prioridad θ la cual cuenta con los niveles alta, media y baja. La utilidad u obtiene sus valores de acuerdo a la prioridad del flujo y las preferencias del nodo receptor.

5.3.1. Descripción del agente

Un agente es un sistema computacional situado en un ambiente, el cual tiene la capacidad de tomar acciones de forma autónoma dentro de un ambiente con el propósito de lograr sus objetivos [30].

El agente está integrado en cada uno de los nodos de la red por una tupla $Ag = < \omega, h, u >$ donde: ω son las preferencias de acuerdo a la prioridad del flujo programado en el nodo las cuales determinarán su elección, h es la jerarquía determinada por la centralidad del nodo, u utilidad la cual será positiva al ganar la negociación o negativa en caso contrario y cero cuando no está negociando.

La naturaleza de las preferencias Ω de los agentes se establecen de acuerdo al tipo de flujo β y su prioridad θ . Las preferencias $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ son capturadas por el agente Ag

mediante una función de utilidad definida por la Ecuación 5.2.

$$u_i : \Omega \rightarrow \mathbb{R} \quad (5.2)$$

La utilidad del agente será igual a 1 cuando de servicio a un tipo de flujo preferido y en caso de contrario será de -1. Es importante aclarar que únicamente los nodos congestionados pueden activar la negociación de la transmisión de los flujos.

5.4. Algoritmo de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO)

En la Figura 5.2 se observa el diagrama de flujo del algoritmo de negociación basado en elección social con autoridades ocultas.

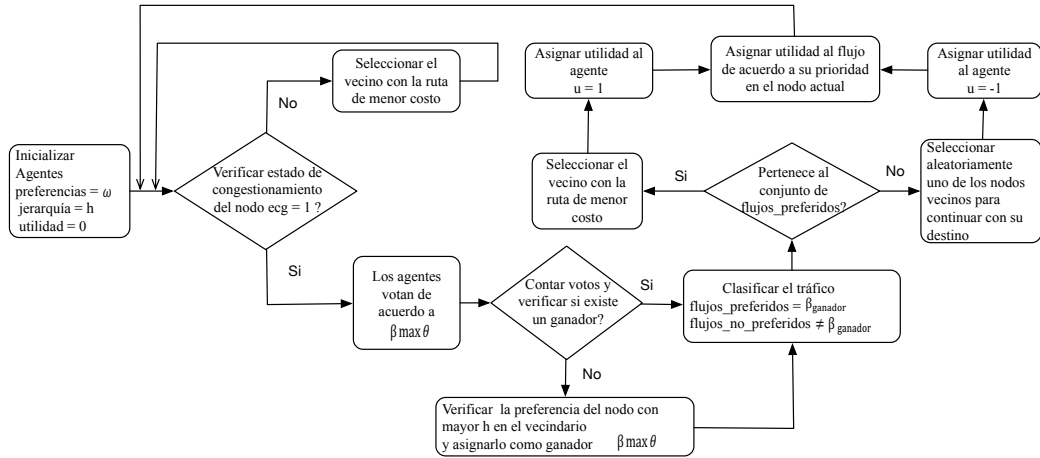


Figura 5.2: Diagrama de flujo del algoritmo ESAO.

De acuerdo a la teoría de elección social, el proceso de la toma de decisiones dentro de un grupo se lleva a cabo mediante la votación de cada uno de los miembros [35]. En el algoritmo propuesto ESAO se determina al ganador mediante la aplicación de las políticas de preferencias de flujos establecidas por el administrador de la red. La activación del algoritmo

1 depende del estado de congestión del nodo en el cual está inmerso el agente. La aplicación de éste algoritmo se prueba dentro de un modelo basado en agentes analizado en [28].

Algorithm 1 Algoritmo de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO)

```
function FINDWINNER(  $\Omega$ ,  $\beta$ ,  $h$ )  
    set choices [ $\omega$  with max  $\beta$ ] of my_neighbors  
    set choices modes of choices  
    if choices length > 1  
        set preferred_choice [ $\omega$  with max  $\beta$ ] of my_neighbors with max  $h$   
    else  
        set preferred_choice choices  
    return preferred_choice  
end function
```

El algoritmo 1 regresa un tipo de flujo f ganador (datos, video o voz); después del proceso de elección mediante la votación de los agentes vecinos del nodo congestionado. El flujo ganador obtiene la prioridad más alta en la cola del nodo congestionado. Al agente ganador obtiene una utilidad de +1, el resto de los flujos contendientes en sus nodos correspondientes su utilidad es puesta a -1.

5.5. Descripción del Modelo Basado en Agentes en NetLogo

El fenómeno complejo a analizar es la congestión de los nodos. En el momento que el grado de congestión excede el umbral establecido por las políticas establecidas la

negociación se lleva a cabo para determinar por cuál de sus nodos vecinos debe continuar el flujo de datos, voz o video considerando sus preferencias preestablecidas por el administrador de la red las cuales son dictadas por sus políticas.

5.5.1. Fase de Inicialización

La red se genera de acuerdo a una distribución a libre de escala [3], la asignación de los costos w de los enlaces bidireccionales se realiza de forma aleatoria en un rango de 0 a 1, en la Figura 5.3 se muestra la interfaz de usuario del modelo propuesto construido en Netlogo.

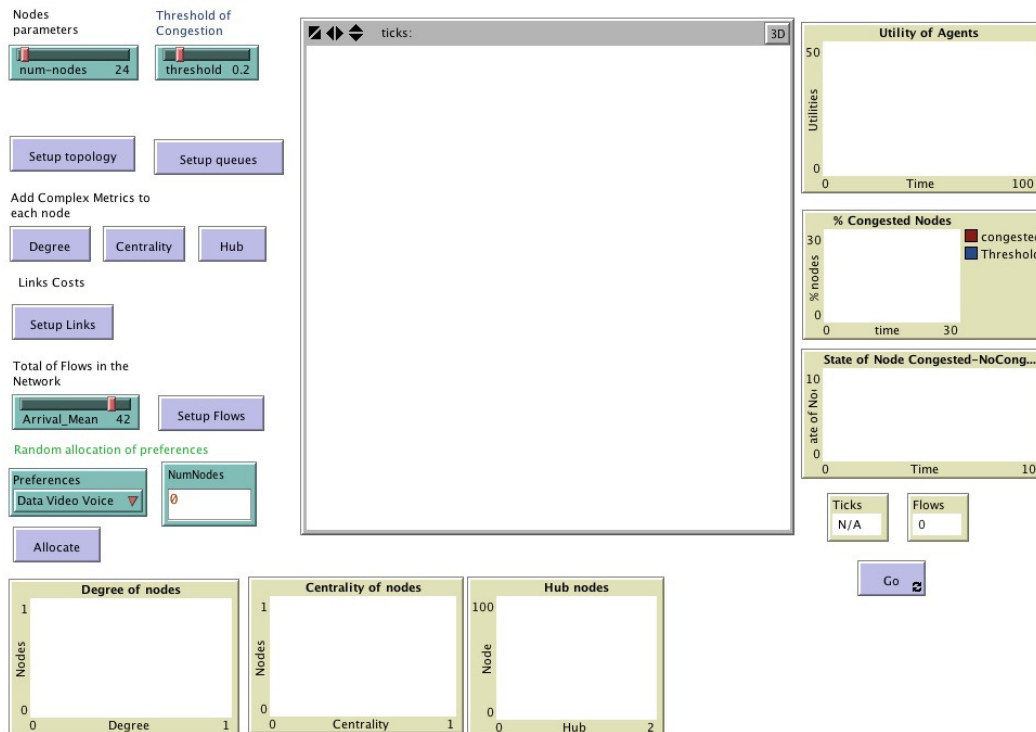


Figura 5.3: Interfaz de usuario del modelo simulación en NetLogo

El umbral de las colas se asigna con incrementos de 0.1 con éste valor se calcula del porcentaje de llenado que no debe excederse para entrar en un estado de congestionamiento (por ejemplo 20 %). El arribo de flujos está determinado con una distribución aleatoria de

Poisson. Los flujos que se transmitirán por la red se clasifican en tres tipos video, voz y datos; para identificar visualmente los flujos se le concede un color con la siguiente distribución: donde el color azul es para datos, el verde para voz y rojo para video mostrada en la Figura 6.1

5.5.2. Determinación de Métricas de Redes Complejas

El modelo al terminar la inicialización de la red permite determinar las métricas de centralidad, grado y si el nodo es hub o no. Esas métricas descritas en el Capítulo 4 las cuales permiten asignar jerarquía a cada nodo.

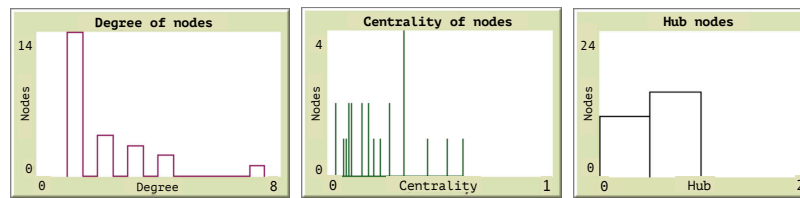


Figura 5.4: Histogramas de las Métricas de Redes Complejas del Modelo.

La Figura 6.2 se muestra las gráficas con los resultados de las métricas obtenidas basado en la estructura de la red.

5.5.3. Asignación de las Preferencias de Flujos

La asignación de las preferencias de cada nodo se establecen de acuerdo a las políticas establecidas por el administrador. En la Figura 6.1 se identifican las preferencias de los nodos por medio del color asignado al nodo.

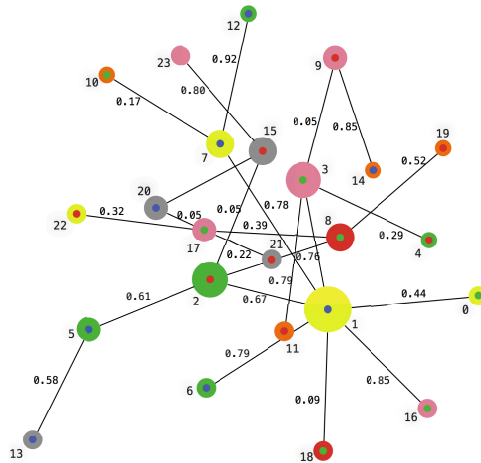


Figura 5.5: Topología generada basada en libre escala.

El grado de agrupamiento de los nodos se representa en el simulador de acuerdo a las dimensiones en el radio del círculo que representa a cada nodo.

5.6. Resumen

En éste Capítulo se describió el modelo basado en agentes propuesto, el cual permite integrar el análisis de una red tomando como referencia su topología, además de mostrar el modelado computacional de cada uno de los elementos que lo integran. En el siguiente capítulo se implementan casos de estudio donde se pone a prueba la efectividad del Algoritmo 1 de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO).

Capítulo 6

Casos de Estudio

6.1. Introducción

El Algoritmo 1 de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO) , cada uno de los cuáles se presenta como un caso de estudio.

6.2. Caso de Estudio 1

El primer caso de estudio propuesto consiste en utilizar una distribución de la topología de red utilizando libre escala y comparar los resultados con el porcentaje de congestionamiento que generaría la distribución de los flujos de los nodos congestionados, sin utilizar el algoritmo ESAO únicamente, es decir, se llevará a cabo la selección de las rutas de forma aleatoria.

6.2.1. Simulación utilizando topología con distribución de libre escala

Los parámetros de inicialización del modelo se muestran en la Tabla 6.1, la cual describe la jerarquía h , preferencias ω de cada nodo, mientras que los costos de los enlaces se pueden consultar en la Tabla A.1 en Apéndice A.1.

Tabla 6.1: Parámetros de inicialización del modelo.

Nodo	h	Preferencias			Nodo	h	Preferencias		
		ω_1	ω_2	ω_3			ω_1	ω_2	ω_3
0	0.34	Voice	Video	Data	12	0.15	Data	Voice	Video
1	1.00	Voice	Video	Data	13	0.079	Video	Data	Voice
2	0.604	Data	Voice	Video	14	0.069	Voice	Data	Video
3	0.533	Data	Video	Voice	15	0.272	Video	Data	Voice
4	0.181	Data	Voice	Video	16	0.34	Data	Video	Voice
5	0.232	Data	Voice	Video	17	0.104	Data	Video	Voice
6	0.34	Data	Voice	Video	18	0.34	Video	Voice	Data
7	0.442	Voice	Video	Data	19	0.092	Voice	Data	Video
8	0.272	Video	Voice	Data	20	0.104	Video	Data	Voice
9	0.204	Data	Video	Voice	21	0.035	Video	Data	Voice
10	0.15	Voice	Data	Video	22	0.035	Voice	Video	Data
11	0.181	Voice	Data	Video	23	0.092	Data	Video	Voice

Los valores antes descritos generan una topología la cual se muestra en la Figura 6.1 en la que se observan las preferencias de los nodos, así como los flujos asignados a cada nodo, los cuales serán entregados a cada uno de sus nodos destino haciendo uso del Algoritmo 1 si

existe una situación de congestionamiento y sin hacer uso de él generando su ruta destino de forma aleatoria.

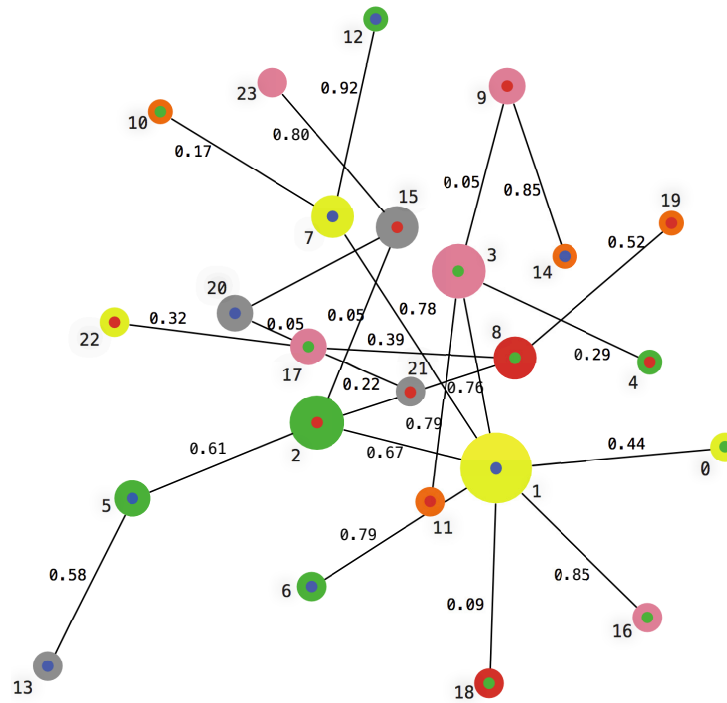


Figura 6.1: Topología generada considerando escala libre.

6.3. Análisis de Resultados

En la Figura 6.2 ilustra los histogramas obtenidos de las medidas de grado, centralidad obtenidas de la red, así como una clasificación de los nodos en nodos hub y no hubs. Un hub se refiere a un nodo muy interconectado, éste concepto lo introdujo Barabási y que con frecuencia son encontrados en una red con nodos distribuidos libre de escala [3].

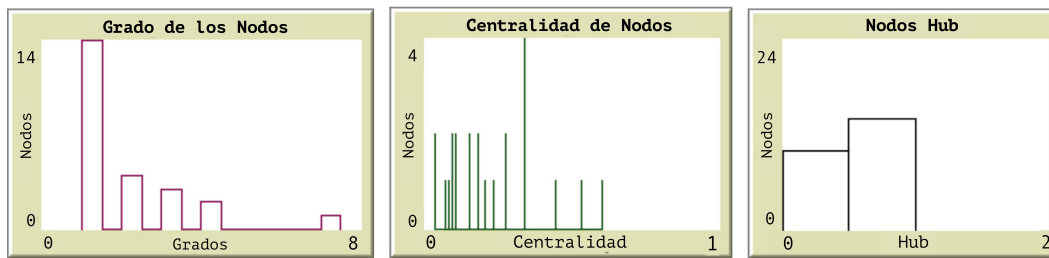


Figura 6.2: Histogramas con las métricas de redes complejas obtenidas del modelo.

El hecho de identificar los nodos con mayor grado de agrupamiento es importante cuando se consideran los nodos hub como prioritarios a atender en caso de sufrir congestionamiento.

En las Figuras 6.3 y 6.4 se muestran los resultados obtenidos utilizando el Algoritmo ESAO y sin utilizarlo respectivamente.

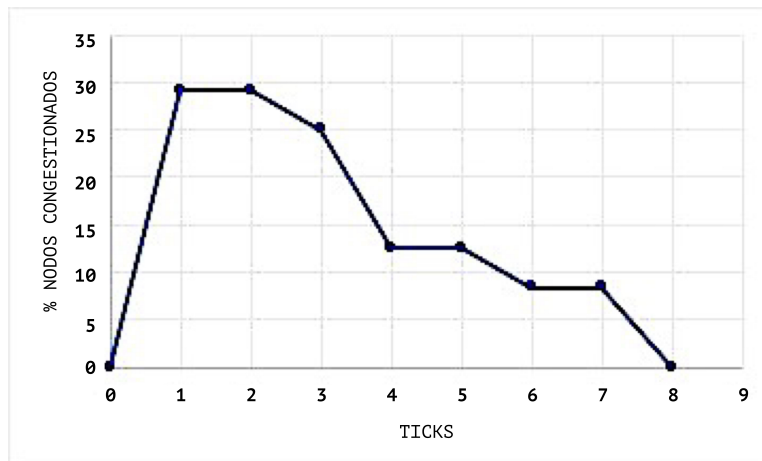


Figura 6.3: Estado de congestionamiento de la red durante la simulación aplicando el Algoritmo ESAO.

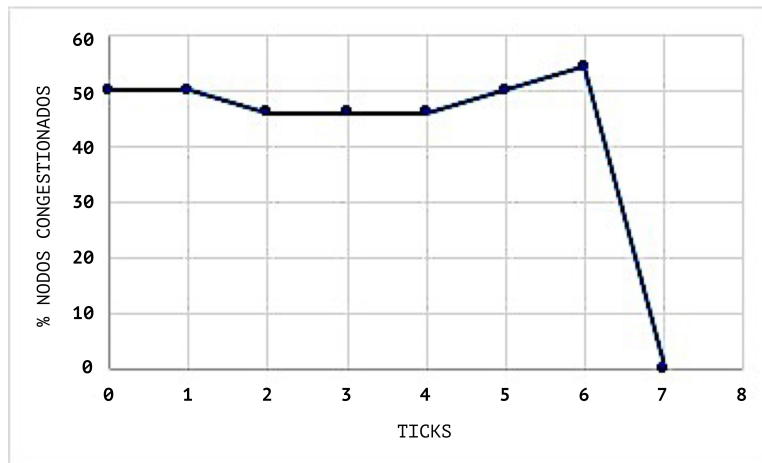


Figura 6.4: Estado de congestionamiento de la red utilizando una Distribución Aleatoria.

Cabe resaltar que el grado de congestionamiento menor correspondió al escenario donde se aplicó el Algoritmo ESAO obteniendo un límite superior de 30 % el cual se observa en la Figura 6.3 mientras tanto los resultados no fueron muy favorables para el caso en el que no se aplicó el algoritmo sino una distribución de flujos aleatoria, en la Figura 6.4 se ilustra que el porcentaje de congestionamiento de la red superó el 50 %.

En la Figura 6.5 y Figura 6.6, se muestran el promedio de congestionamiento de cada nodo durante el tiempo de simulación, donde se detectó que los nodos mantuvieron un alto congestionamiento cuando no se aplica el algoritmo ESAO.

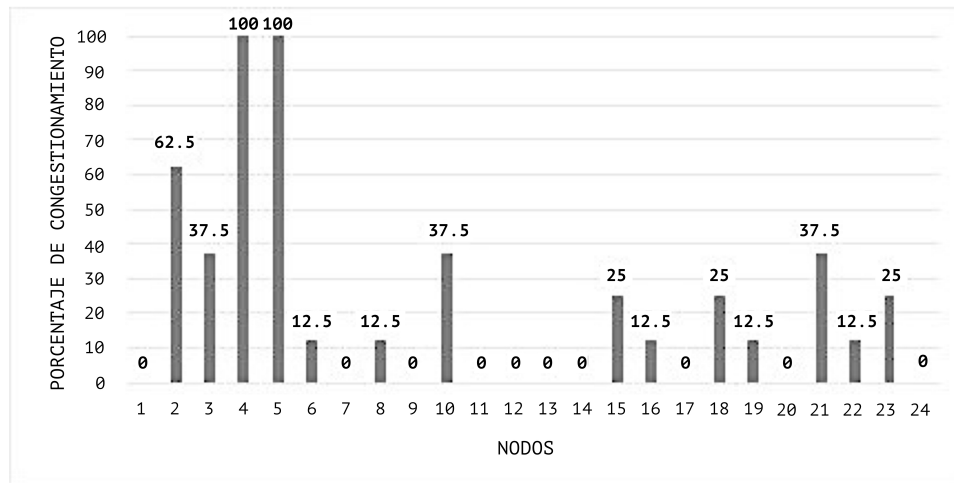


Figura 6.5: Promedio de congestionamiento por nodo durante la simulación aplicando el algoritmo ESAO.

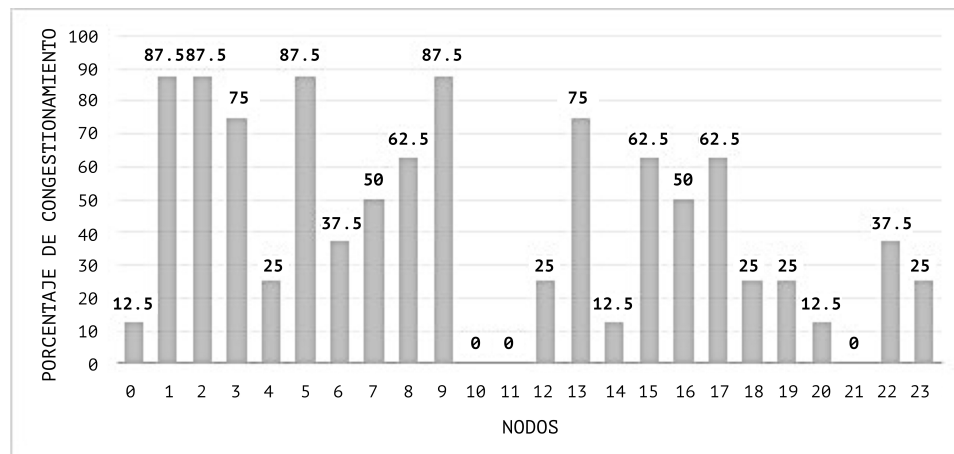


Figura 6.6: Promedio de congestionamiento por nodo utilizando Distrución Aleatoria.

Después de las negociación con el algoritmo cada agente obtiene una utilidad positiva o negativa, las cuales son mostradas en la Figura 6.7.

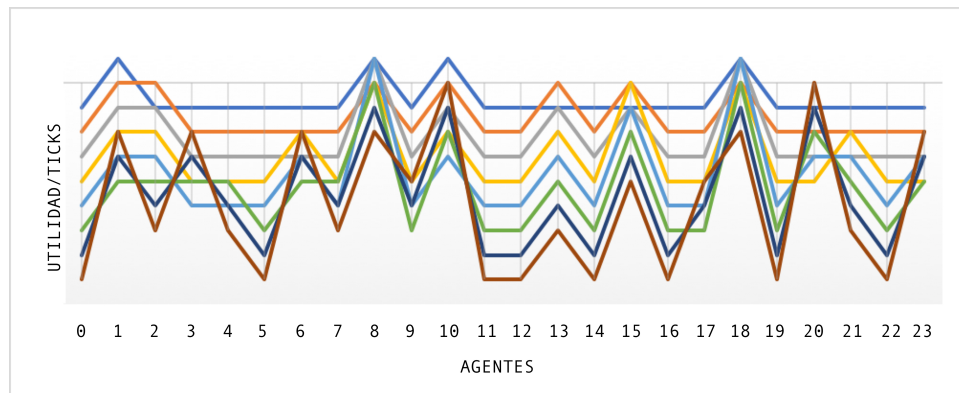


Figura 6.7: Utilidades obtenidas por los agentes aplicando ESOA.

Durante la simulación utilizando distribución aleatoria los agentes no reciben utilidades, debido a esto no se genera ninguna estadística. Ambos modelos distribuyeron cuarenta y dos flujos durante la simulación, generando las rutas necesarias para llegar cada uno a su nodo destino al término de ocho ticks (unidades de tiempo de NetLogo). Los valores utilizados para generar la gráfica de utilidad de los agentes se pueden observar en la Tabla A.3 del Apéndice A.1.

6.4. Caso de Estudio 2

El propósito de éste segundo caso de estudio es comparar si al cambiar la topología a una distribución considerando el grado como métrica para generar la red, la efectividad del Algoritmo 1 ESOA permanece siendo mejor que la selección aleatoria.

6.4.1. Simulación utilizando topología generada basada en el grado del nodo

A diferencia del caso de estudio anterior la topología distribuida de acuerdo al grado, genera menor número de nodos hubs debido que se encuentran muy entrelazados los nodos,

y con el mismo grado de agrupamiento eso se observa en la Figura 6.8 y los resultados de las gráficas mostradas en la Figura 6.9 confirma que la centralidad entre nodos es muy cercana.

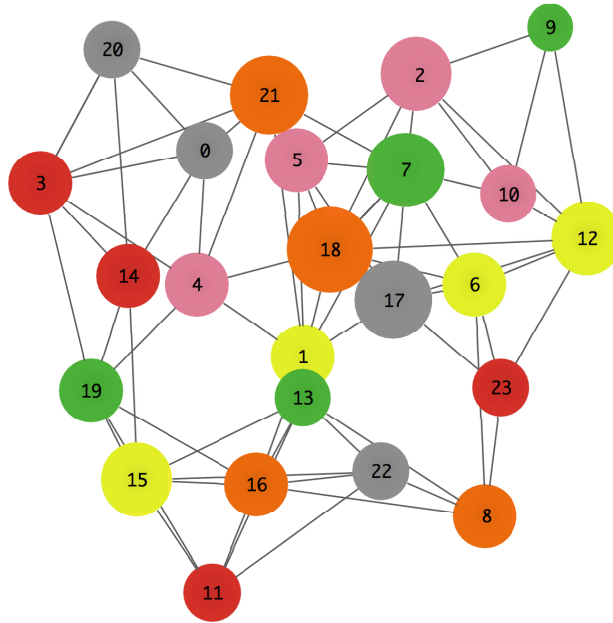


Figura 6.8: Topología con distribución basado en grado.

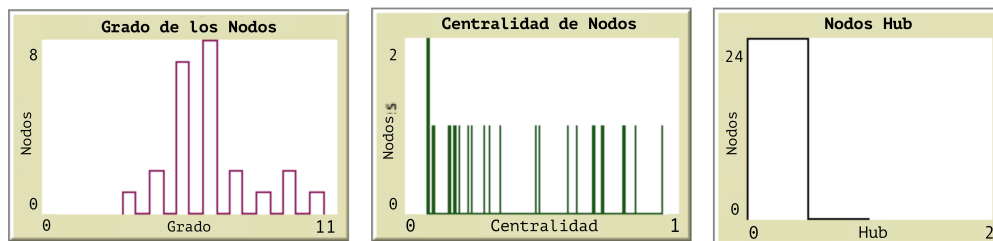


Figura 6.9: Histogramas obtenidos del cálculo de las métricas de redes complejas.

Los resultados obtenidos en el segundo caso de estudio se ilustran en la Figura 6.10 y Figura 6.11. Comparando los resultados el resultado obtenido del promedio de congestión de la red durante el tiempo es menor del 26 % cuando se aplica el algoritmo ESAO, en cambio la gráfica de la Figura 6.11 inicia con un grado de congestión más alto y en promedio es el escenario que sufrió mayor congestión.

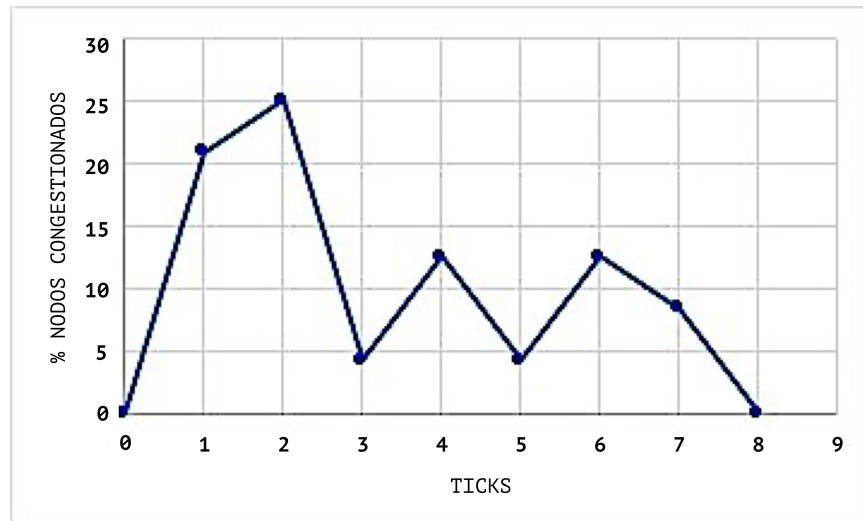


Figura 6.10: Promedio de estado de congestiónamiento de la red durante la simulación utilizando el Algoritmo ESAO.

La información detallada del promedio de congestiónamiento de cada nodo durante la simulación se muestra en la Figuras 6.12 y 6.13.

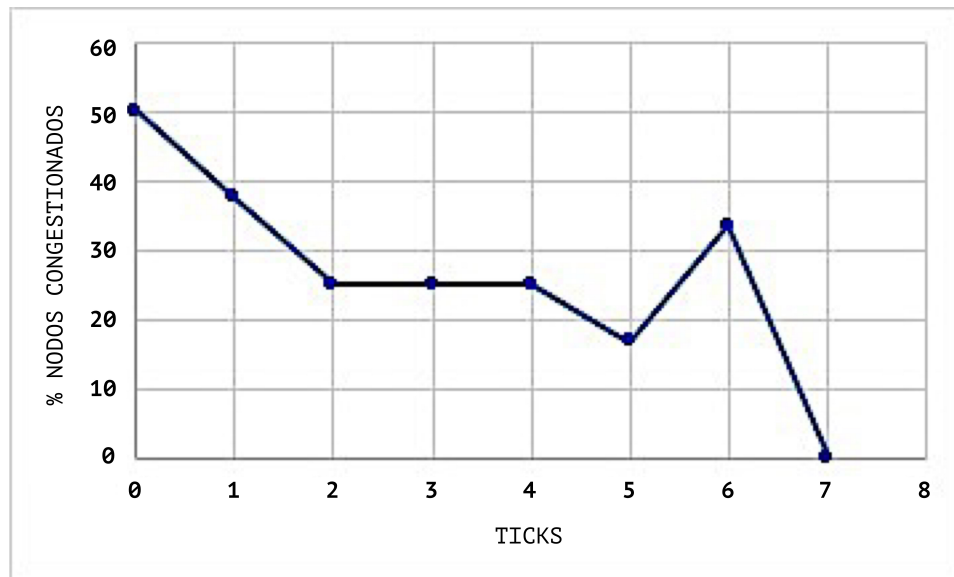


Figura 6.11: Promedio de congestiónamiento de la red durante la simulación utilizando Distribución Aleatoria.

Como se puede observar en la Figura 6.12 el porcentaje de congestiónamiento es menor

que en la simulación donde no se aplica el algoritmo, en la Figura 6.13 se muestra mayor número de nodos congestionados hasta con un 87.5 % mientras que en el máximo en la Figura 6.12 es de 37.5 %.

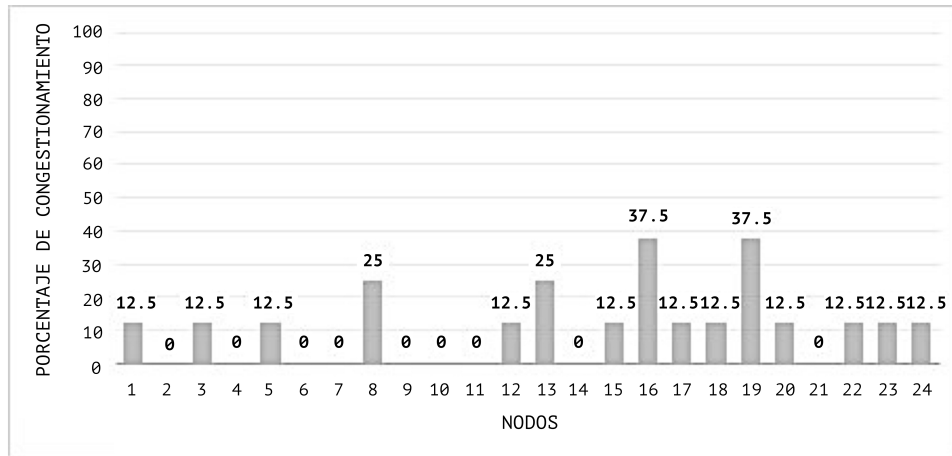


Figura 6.12: Promedio de congestionamiento de los nodos durante el tiempo de simulación aplicando el Algoritmo ESAO.

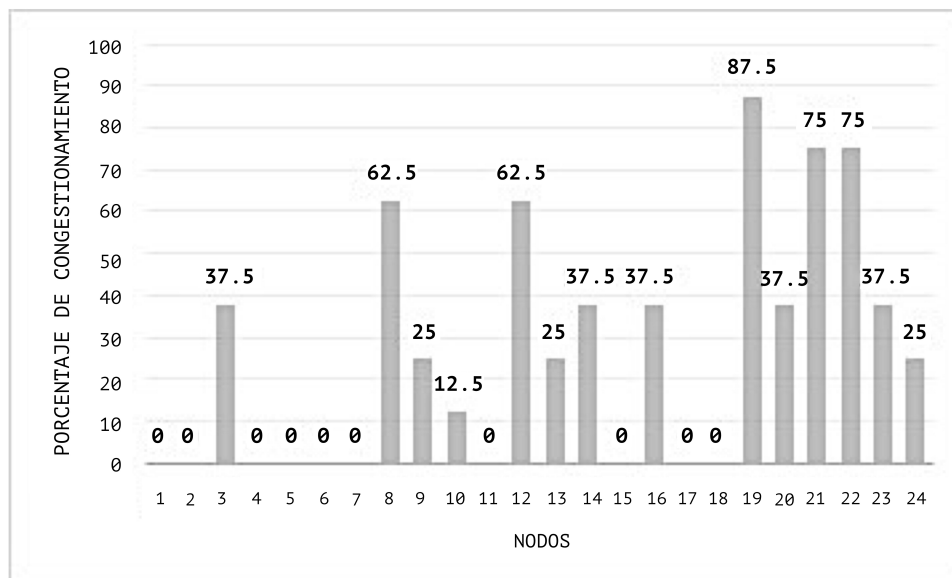


Figura 6.13: Promedio de congestionamiento por nodo durante el tiempo de simulación aplicando una Distribución Aleatoria.

Los resultados antes mostrados confirma que un cambio en la topología afecta el porcentaje de nodos congestionados, pero no afecta en que el menor porcentaje de congestionamiento

en la red continúa siendo cuando se aplica el algoritmo ESAO.

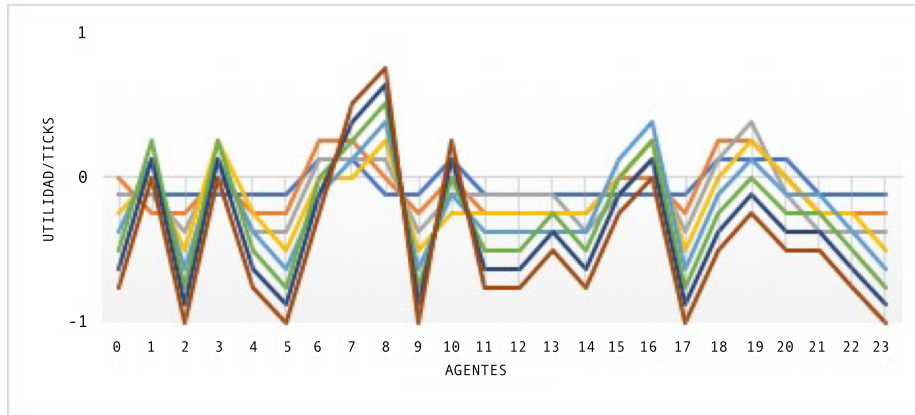


Figura 6.14: Utilidades de cada agente obtenidos al aplicar el algoritmo ESAO.

Finalmente, en la Figura 6.14 se ilustran las utilidades generadas por los agentes durante la simulación. Las utilidades también representan a los nodos más visitados cuando el congestionamiento emerge. Para identificar los nodos con mayores utilidades se pueden observar la Tabla A.4 de resultados ubicado en el Apéndice A.1.

6.5. Caso de Estudio 3

Por último, el caso de estudio 3 se propone hacer cambios en el costo de los enlaces de 0.3, 0.6 y 1.0 haciendo uso de *Behavior Space* herramienta que ofrece NetLogo, se ejecutaron 300 simulaciones de las cuales se variaba, la topología pero no el número de nodos.

6.5.1. Simulación utilizando Behavior Space de NetLogo

Se generaron 600 simulaciones en total, donde en 300 de ellas se generaron escenarios donde se aplica el Algoritmo 1 de Negociación Basado en Elección Social con Autoridades Ocultas (ESAO) y en las otras 300 simulaciones no se aplicó el algoritmo sino una distribución

aleatoria. La configuración de los parámetros a variar en *Behavior Space* de NetLogo se muestran en la Figura 6.15.

Experiment

Experiment name

Vary variables as follows (note brackets and quotation marks):

```
[ "num-nodes" 24 ]
[ "costrand" 0.3 0.6 1 ]
[ "Arrival_Mean" 42 ]
```

Either list values to use, for example:
["my-slider" 1 2 7 8]
or specify start, increment, and end, for example:
["my-slider" [0 1 10]] (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycor, min-pycor, random-seed.

Repetitions
run each combination this many times

Measure runs using these reporters:

```
count nodes with [ecg = 1]
```

one reporter per line; you may not split a reporter across multiple lines

☒ Measure runs at every step
if unchecked, runs are measured only when they are over

Setup commands:

Go commands:

▶ Stop condition: the run stops if this reporter becomes true

▶ Final commands: run at the end of each run

Time limit
stop after this many steps (0 = no limit)

Figura 6.15: Inicialización de Behavior Space de Netlogo.

En la Tabla 6.2 se ilustran los resultados de las 600 simulaciones, con sus respectivos valores de tasa promedio de llegadas, costo, promedio de congestionamiento de cada nodo y la desviación estándar de los resultados obtenidos.

Tabla 6.2: Resultados obtenidos después de 600 simulaciones con *Behavior Space* de NetLogo.

Num. of Corridas	Tasa promedio de interarribo	Costo	Algortimo de Negociación Basado en Elección Social con Autoridades Ocultas		Distribución Aleatoria	
			Promedio de Congestionado Nodos	Desviación Estándar	Promedio de Congestión Nodos	Desviación Estándar
100	42	0.3	2.583	0.447	2.726	0.550
100	42	0.6	2.584	0.402	2.612	0.531
100	42	1.0	2.576	0.446	2.664	0.554

Los resultados muestran que el promedio de nodos congestionados cuando el costo varía en un rango de 0 - 0.3 utilizando el algoritmo ESAO es menor que cuando no se utiliza. La diferencia entre la desviación estándar en el escenario donde se varía el costo de 0 - 0.6 se muestra que es mayor cuando no se aplica el algoritmo, se debe hacer incapié que las topologías fueron variando durante las 100 simulaciones, en ambos casos se probó aplicando el algoritmo ESAO y no aplicándolo.

Por último, el caso final con la variación de costos con un rango de 0 a 1, las desviaciones estándar tuvieron una mayor separación de 2.58 cuando se aplica el algoritmo ESAO y 2.66 cuando no se aplica; los resultados se pueden observar en la Figura 6.16. Finalmente el impacto en la variación del costo de los enlaces si influye en el rendimiento del algoritmo, así como en el porcentaje promedio de congestión de los nodos de la red.

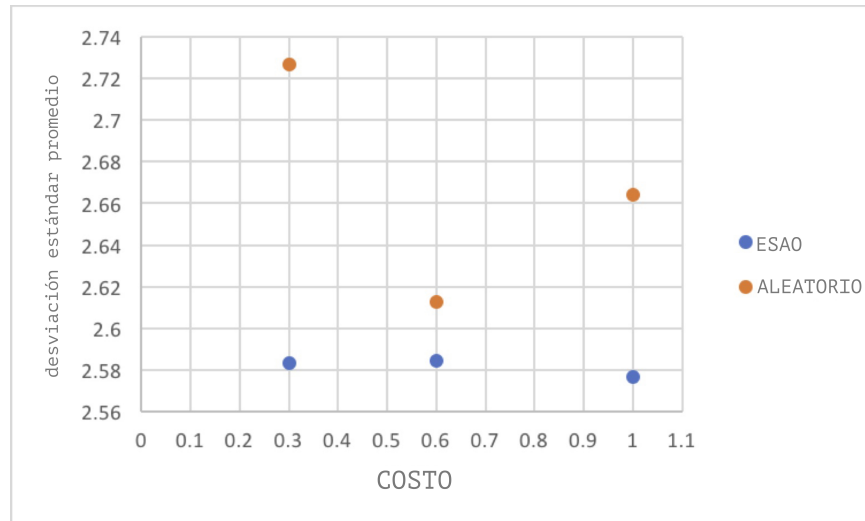


Figura 6.16: Comparación de rendimiento del modelo aplicando el Algoritmo ESAO y con Distribución Aleatoria.

La comparación de los resultados muestran que la Distribución aleatoria es menos efectiva en situaciones de congestionamiento en una red. Los resultados confirmaron que la aplicación del Algoritmo ESAO es efectiva en la distribución de los flujos de una red de forma eficiente.

Capítulo 7

Trabajo Futuro y Conclusiones

7.1. Conclusiones

La integración de un modelo basado en agentes permitió identificar patrones de comportamiento de la red, además las medidas de centralidad obtenidas por el modelo son utilizadas para clasificar los nodos de la red de forma jerárquica. Por otro lado el tener un reporte de cuales son los nodos que sufren mayor congestión genera un diagnóstico para prevenir la caída de los flujos de red más relevantes.

El algoritmo SEHA propuesto establece los criterios de desempate y distribución de flujos cuando un nodo se congestiona, habilitando a la red solucionar ese problema emergente de forma autónoma. La asignación de los flujos de mayor prioridad son establecidos por el administrador en un conjunto de políticas, las cuales son consideradas al momento de activarse el algoritmo SEHA el cual demostró tener un mejor desempeño que la asignación de rutas de los flujos de forma aleatoria.

Considerar el análisis de la topología de una red para establecer criterios de negociación es un aspecto importante, sobre todo cuando se aborda un problema desde el enfoque de

sistemas complejos, en el cual hasta la interacción menos relevante al punto de vista del ser humano puede ser la solución a un problema mayor. Debido a lo anterior el uso de las métricas de redes complejas como la centralidad de un nodo, en el análisis de una red es un término relevante en la identificación de las relaciones que guarda un nodo con respecto a sus nodos vecinos.

Por último la teoría de grafos y el análisis de la matriz de adyacencia de una red es una metodología clave para determinar las relaciones entre nodos independientemente del tipo de red que se analice (social, animal, biológica, etc).

Se generaron 600 simulaciones utilizando la herramienta de *Behavior Space* de NetLogo las cuales confirmaron que la aplicación del Algoritmo en el momento que surgen situaciones de congestionamiento en una red tiene mejor rendimiento la entrega de flujos, que al hacerlo de forma aleatoria.

7.2. Trabajo Futuro

El trabajo futuro a abordar es el reconocimiento de patrones de comportamiento de la red cuando existe congestionamiento, donde se identifiquen cuáles fueron los agentes más beneficiados con utilidades altas y qué tipos de flujos las generaron con mayor frecuencia. Como resultado del análisis de los patrones se podría modificar de forma automática la actualización de las políticas de red, lo cual simplificaría las tareas del administrador y en consecuencia mejoraría el rendimiento de la red en cuestión de control de flujos.

Apéndice A

Tablas de Configuración

A.1. Apéndice

En la Tabla A.1 se muestran los costos asignados a los enlaces son generados de forma aleatoria, considerando un rango de cero a uno.

Las preferencias asignadas a cada agente en el su respectivo nodo están enlistadas en la Tabla A.2. Los valores asignados hacen referencia a la Figura 6.1 dentro de la sección de Casos de estudio y Análisis de Resultados. Por otro lado en la Tabla A.2, los tipos de flujos en la red son: Datos = 0, Voz=1 y Video = 2, el orden en el que fueron colocados impacta indica la prioridad del flujo.

Generalmente un nodo consta de una cola de espera la cual tiene en éste caso tiene un tamaño no infinito para nuestro modelo la medida es de 10 paquetes. Un nodo cambia de estados no congestionado a congestionado, para identificar ese cambio se debe exceder el umbral de congestión el cual está especificado en el 20%, lo que indica si se sobre pasa ese límite el bit de congestionamiento ECG cambia su valor de cero a uno lo cual activa la negociación Ecuación 5.1.

Tabla A.1: Costo de los enlaces que conforman la red.

Cost	Enlace	
	Fin1	Fin2
0.442	nodo 1	nodo 0
0.759	nodo 1	nodo 3
0.788	nodo 1	nodo 6
0.778	nodo 1	nodo 7
0.848	nodo 1	nodo 16
0.668	nodo 2	nodo 1
0.293	nodo 4	nodo 3
0.609	nodo 5	nodo 2
0.577	nodo 5	nodo 13
0.165	nodo 7	nodo 10
0.217	nodo 8	nodo 2
0.392	nodo 8	nodo 17
0.517	nodo 8	nodo 19
0.045	nodo 9	nodo 3
0.850	nodo 9	nodo 14
0.790	nodo 11	nodo 3
0.924	nodo 12	nodo 7
0.053	nodo 15	nodo 2
0.322	nodo 17	nodo 22
0.089	nodo 18	nodo 1
0.678	nodo 20	nodo 15
0.364	nodo 21	nodo 20
0.801	nodo 23	nodo 15

Tabla A.2: Valores de inicialización de los nodos de red.

Nodo Fuente	Color	Preferencias	Num. Flujos	Tipo de Flujos	Nodo Destino
0	amarillo	1 2 0	1	1	21
1	amarillo	1 2 0	4	1,0,2,0	10,21,7,9
2	verde	0 1 2	1	2	14
3	rosa	0 2 1	2	1,1	22,6
4	verde	0 1 2	3	1,2,2	22,3,21
5	verde	0 1 2	5	1,2,2,0,0	8,7,21,15,7
6	verde	0 1 2	3	1,0,0	22,10,21
7	amarillo	1 2 0	1	0	6
8	red	2 1 0	1	1	20
9	rosa	0 2 1	1	2	7
10	naranja	1 0 2	1	1	17
11	naranja	1 0 2	2	0,2	19,5
12	verde	0 1 2	2	1,0	1,2
13	gris	2 0 1	2	1,0	21,8
14	naranja	1 0 2	3	1,2,0	23,23,1
15	gris	2 0 1	1	2	20
16	rosa	0 2 1	1	1	1
17	rosa	0 2 1	4	1,2,0,1	11,20,7,4
18	red	2 1 0	2	2,1	23,20
19	naranja	1 0 2	2	2,2	12,10
20	gris	2 0 1	2	1,0	5,19
21	gris	2 0 1	2	2,2	8,19
22	amarillo	1 2 0	3	2,1,2	13,8,3
23	rosa	0 2 1	0	-	-

Tabla A.3: Utilidades de los agentes después de las negociaciones.

Ticks/Agente en Nodo	1	2	3	4	5	6	7	8
nodo 0	-1	-1	-1	-1	-1	-1	-1	-1
nodo 1	1	-1	-1	-1	-1	-1	1	1
nodo 2	-1	1	-1	-1	-1	-1	-1	-1
nodo 3	-1	-1	-1	-1	-1	1	1	1
nodo 4	-1	-1	-1	-1	-1	1	-1	-1
nodo 5	-1	-1	-1	-1	-1	-1	-1	-1
nodo 6	-1	-1	-1	1	-1	-1	1	1
nodo 7	-1	-1	-1	-1	-1	1	-1	-1
nodo 8	1	-1	1	-1	1	-1	-1	-1
nodo 9	-1	-1	-1	-1	-1	-1	1	1
nodo 10	1	-1	-1	-1	-1	1	1	1
nodo 11	-1	-1	-1	-1	-1	-1	-1	-1
nodo 12	-1	-1	-1	-1	-1	-1	-1	-1
nodo 13	-1	1	-1	-1	-1	-1	-1	-1
nodo 14	-1	-1	-1	-1	-1	-1	-1	-1
nodo 15	-1	1	-1	1	-1	-1	-1	-1
nodo 16	-1	-1	-1	-1	-1	-1	-1	-1
nodo 17	-1	-1	-1	-1	-1	-1	1	1
nodo 18	1	-1	1	-1	1	-1	-1	-1
nodo 19	-1	-1	-1	-1	-1	-1	-1	-1
nodo 20	-1	-1	-1	-1	1	1	1	1
nodo 21	-1	-1	-1	1	-1	-1	-1	-1
nodo 22	-1	-1	-1	-1	-1	-1	-1	-1
nodo 23	-1	-1	-1	-1	1	-1	1	1

Tabla A.4: Utilidades de los agentes.

Ticks /Agente en Nodo	1	2	3	4	5	6	7	8
nodo 0	-1	1	-1	-1	-1	-1	-1	-1
nodo 1	-1	-1	1	1	1	1	-1	-1
nodo 2	-1	-1	-1	-1	-1	-1	-1	-1
nodo 3	-1	1	1	1	-1	1	-1	-1
nodo 4	-1	-1	-1	1	-1	-1	-1	-1
nodo 5	-1	-1	-1	-1	-1	-1	-1	-1
nodo 6	1	1	-1	-1	-1	1	-1	-1
nodo 7	1	1	-1	-1	1	1	1	1
nodo 8	-1	1	1	1	1	1	1	1
nodo 9	-1	-1	-1	-1	-1	-1	-1	-1
nodo 10	1	-1	-1	-1	1	1	1	1
nodo 11	-1	-1	-1	-1	-1	-1	-1	-1
nodo 12	-1	-1	1	-1	-1	-1	-1	-1
nodo 13	-1	-1	1	-1	-1	1	-1	-1
nodo 14	-1	-1	-1	1	-1	-1	-1	-1
nodo 15	-1	1	-1	1	1	-1	-1	-1
nodo 16	-1	1	1	1	1	-1	-1	-1
nodo 17	-1	-1	-1	-1	-1	-1	-1	-1
nodo 18	1	1	-1	-1	-1	-1	-1	-1
nodo 19	1	1	1	-1	-1	-1	-1	-1
nodo 20	1	-1	-1	1	-1	-1	-1	-1
nodo 21	-1	-1	-1	1	1	-1	-1	-1
nodo 22	-1	-1	-1	1	-1	-1	-1	-1
nodo 23	-1	-1	-1	-1	-1	-1	-1	-1

Por último la Tabla A.4 ilustra los resultados de la utilidades obtenidas por los agentes en la segunda simulación utilizando el algoritmo de negociación ESAO considerando una topología generada con una distrución de libre escala.

Bibliografía

- [1] Dey Anind K. Context-aware computing: The CyberDesk project. *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*, pages 51–54, 1998.
- [2] Mouna Ayari and Zeinab Movahedi. ADMA : Autonomous Decentralized Management Architecture for MANETs - A Simple Self-Configuring Case Study Categories and Subject Descriptors. *Architecture*, pages 132–137, 2010.
- [3] Albert-Laszlo Barabasi and M. Pósfai. *Network science graph theory*. Cambridge University Press, Cambridge, U.K., 2016.
- [4] Antoni Barba Martí. *Gestión de red*. Barcelona, España, 1 edition, 2001.
- [5] Jeffrey M Bradshaw. An Introduction to Software Agents. *Software Agents*, 6(4):466–469, 1997.
- [6] R. Chadha, J. Chiang, G. Levin, and a. Poylisher. Policy-based mobile ad hoc network management for drama. *IEEE MILCOM 2004. Military Communications Conference, 2004.*, 3:1317–1323.
- [7] Hong Chen Hong Chen, Wenan Zhou Wenan Zhou, and Lu Liu Lu Liu. An Approach of Agent-Based Architecture for Autonomic Network Management. *2009 5th International Conference on Wireless Communications Networking and Mobile Computing*, pages 1–5, 2007.

-
- [8] David M. Chess, David M. Chess, Alla Segal, Alla Segal, Ian Whalley, Ian Whalley, Steve R. White, and Steve R. White. Unity:experiences with a prototype autonomic computing system. *ICAC '04: Proceedings of the 2004 International Conference on Autonomic Computing*, pages 140–147, 2004.
 - [9] Dominique Dudkowski, Marcus Brunner, Giorgio Nunzi, Chiara Mingardi, Chris Foley, Miguel Ponce De Leon, Catalin Meirosu, and Susanne Engberg. Architectural principles and elements of in-network management. In *2009 IFIP/IEEE International Symposium on Integrated Network Management, IM 2009*, pages 529–536, 2009.
 - [10] Said El brak, Mohammed Bouhorma, and Anouar A. Boudhir. Network Management Architecture Approaches Designed for Mobile Ad hoc Networks. *International Journal of Computer Applications*, 15(6):14–18, 2011.
 - [11] A. Encinosa and D. García. Gestión Autonómica. *Revista Telemática*, 1(10), 2012.
 - [12] Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, Inc., New York, NY, USA, 2011.
 - [13] K. I. Goh, B. Kahng, and D. Kim. Universal Behavior of Load Distribution in Scale-free Networks. pages 1–4, 2001.
 - [14] Etsi Industry Specification Group. Autonomic network engineering for the self-managing Future Internet (AFI): Scenarios, Use Cases and Requirements for Autonomic Self-Managing Future Internet. *AFI Group Specification (GS)*, 1(V1.1.1):1–60, 2011.
 - [15] IBM. *An architectural blueprint for autonomic computing (White Paper)*. June 2005.
 - [16] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, May 2001.

-
- [17] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, Jan 2003.
 - [18] Manzoor Khan and Fikret Sivrikaya. Agent based autonomic network control and management. *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1627–1632, apr 2013.
 - [19] R. Lempert. Agent-based modeling as organizational and public policy simulators. *Proc. Natl. Acad. Sci. (USA)*, (99):7195–7196, 2002.
 - [20] V. R. Lesser. Cooperative multiagent systems: a personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):133–142, Jan 1999.
 - [21] Daniel Fernandes Macedo, Zeinab Movahedi, Javier Rubio-Loyola, Antonio Astorga, Giannis Koumoutsos, and Guy Pujolle. The AutoI approach for the orchestration of autonomic networks. *Annales des Telecommunications/Annals of Telecommunications*, 66(3-4):243–255, 2011.
 - [22] Apostolos Malatras, Antonios M. Hadjiantonis, and George Pavlou. Exploiting context-awareness for the autonomic management of mobile ad hoc networks. *Journal of Network and Systems Management*, 15(1):29–55, 2007.
 - [23] Natarajan Meghanathan. Use of Eigenvector Centrality to Detect Graph Isomorphism. *Computer Science & Information Technology (CS & IT)*, 5(6):01–09, 2015.
 - [24] Natarajan Meghanathan and Raven Lawrence. Centrality Analysis of the United States Network Graph. (including DC):23–28, 2016.
 - [25] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A survey of autonomic network architectures and evaluation criteria. *IEEE Communications Surveys and Tutorials*, 14(2):464–490, 2012.

-
- [26] J. Poncela-Casasnovas, B. Spring, D. McClary, A. C. Moller, R. Mukogo, C. A. Pellegrini, M. J. Coons, M. Davidson, S. Mukherjee, and L. A. Nunes Amaral. Social embeddedness in an online weight management programme is linked to greater weight loss. *Journal of The Royal Society Interface*, 12(104):20140686–20140686, 2015.
- [27] K. Raya-Díaz, C. Gaxiola-Pacheco, and M. Castañón-Puga. Tracing Structures in Networks to Capture its Organization. *International Journal of Applied Evolutionary Computation*, 6(4):1–13, 2015.
- [28] K. Raya-Díaz, C. Gaxiola-Pacheco, M. Castañón-Puga, L. E. Palafox, J. R. Castro, and D.L. Flores. Agent-based model for automaticity management of traffic flows across the network. *Applied Sciences (Switzerland)*, 7(9), 2017.
- [29] Javier Rubio-Loyola, Carlos Mérida-Campos, Steven Willmott, Antonio Astorga, Joan Serrat, and Alex Galis. Service coalitions for future Internet services. *IEEE International Conference on Communications*, 2009.
- [30] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.
- [31] N. Samaan and A. Karmouch. Towards autonomic network management: an analysis of current and future research directions. *IEEE Communications Surveys & Tutorials*, 11(3):22–36, 2009.
- [32] Prométhée Spathis and Marco Bicudo. ANA: Autonomic Network Architecture. *Autonomic Network Management Principles*, 28(1):49–65, 2011.
- [33] Ryan W Thomas and Luiz A. Dasilva. Cognitive networks. 2006.
- [34] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

- [35] Michael J. Wooldridge. *An Introduction to MultiAgent Systems (2. ed.)*. Wiley, 2009.