

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja



MODALIDAD PRESENCIAL

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

**MODELO DE GESTIÓN DE REDES MÓVILES AD HOC INTEGRANDO
TECNICAS DE REDES NEURONALES Y TEORÍA DE JUEGOS**

*Trabajo de fin de carrera previa a la
obtención del título de Ingeniera en
Sistemas Informáticos y Computación*

AUTORA: María José Lazo Reinozo

DIRECTOR: Ing. Rommel Vicente Torres

CO-DIRECTOR: Ing. Greyson Alberca

Loja – Ecuador

2011

CERTIFICACIÓN DE DIRECTOR

Ing. Rommel Vicente Torres

DIRECTOR DE TESIS

CERTIFICA:

*Haber dirigido, corregido y supervisado en todas sus partes, el desarrollo del presente proyecto de tesis previo a la obtención del título de **INGENIERA EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**, titulado “**GESTIÓN DE REDES AD HOC CON TECNICAS DE INTELIGENCIA ARTIFICIAL, REDES NEURONALES Y TEORÍA DE JUEGOS**”, con autoría de la señorita *María José Lazo Reinozo* y una vez que este cumple con todas las exigencias y los requisitos legales establecidos por la *Universidad Técnica Particular de Loja*, autoriza su presentación para los fines legales pertinentes.*

Loja, Octubre de 2010

Ing. Rommel Torres

CERTIFICACIÓN CO-DIRECTOR

Ing. Greyson Alberca

CO - DIRECTOR DE TESIS

CERTIFICA:

*Haber dirigido, corregido y supervisado en todas sus partes, el desarrollo del presente proyecto de tesis previo a la obtención del título de **INGENIERA EN SISTEMAS INFORMÁTICOS Y COMPUTACIÓN**, titulado “**GESTIÓN DE REDES AD HOC CON TECNICAS DE INTELIGENCIA ARTIFICIAL, REDES NEURONALES Y TEORÍA DE JUEGOS**”, con autoría de la señorita *María José Lazo Reinozo* y una vez que este cumple con todas las exigencias y los requisitos legales establecidos por la Universidad Técnica Particular de Loja, autoriza su presentación para los fines legales pertinentes.*

Loja, Octubre de 2010

Ing. Greyson Alberca

AUTORÍA

En el presente proyecto de tesis las ideas, principios, conceptos y todo lo expuesto con cada una de sus observaciones, análisis, evaluaciones, conclusiones y recomendaciones emitidas, es de responsabilidad del autor que firma a continuación, la cual a su vez autoriza a la escuela de Sistemas Informáticos y Computación hacer uso del presente documento en lo conveniente .

Además, la información que fue utilizada en esta investigación perteneciente a otros autores está debidamente especificada en fuentes de referencia y apartados bibliográficos.

.....

María José Lazo Reinozo

CESIÓN DE DERECHOS

Yo, María José Lazo Reinozo, expongo conocer y consentir la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja en el cuál textualmente se expresa lo siguiente: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”.

.....

María José Lazo Reinozo

DEDICATORIA

Con todo mi cariño a mis queridos padres quienes me han apoyado incondicionalmente durante todo el proceso de mi formación profesional inclusive en los momentos más difíciles de mi carrera con el único fin de que siga adelante y así poder alcanzar esta meta tan anhelada.

Además de mis padres también quiero dedicar este trabajo a mis sobrinos que son la luz de mis ojos y a mis hermanos, quienes todos juntos han sido mi pilar fundamental para seguir adelante y me han ayudado a encaminar mi vida, gracias por brindarme siempre toda su confianza, respeto y amor, los quiero mucho.

A amigos y compañeros que de alguna u otra manera siempre me han apoyado y se han preocupado porque todo me salga bien, gracias por compartir conmigo los buenos y malos momentos todo este tiempo.

María José Lazo Reinozo

AGRADECIMIENTO

Ante todo agradecer primeramente de manera muy especial a DIOS, quien durante toda mi vida ha sido mi fortaleza y mi guía iluminando mi vida para tomar las mejores decisiones, a todas y cada una de las personas que de una u otra manera me han ayudado al desarrollo de este proyecto de fin de carrera, de manera muy especial al Director de Tesis, Ing. Rommel Torres quien con su ayuda, brindándome su apoyo y conocimientos me guió en el desarrollo del presente trabajo de investigación.

A todos ustedes

MUCHAS GRACIAS

María José Lazo Reinozo

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN DE DIRECTOR.....	II
CERTIFICACIÓN CO-DIRECTOR.....	III
AUTORÍA.....	IV
CESIÓN DE DERECHOS.....	V
DEDICATORIA.....	VI
AGRADECIMIENTO	VII

CAPITULO I

1. INTRODUCCIÓN.....	2
1.1. DEFINICIÓN DEL PROBLEMA.....	4
1.2. HIPÓTESIS.....	4
1.3. OBJETIVOS.....	5
1.4. ESTRUCTURA DE LA TESIS.....	5

CAPITULO II

2. ESTADO DEL ARTE.....	8
2.1. GESTIÓN DE REDES.....	8
2.1.1. ARQUITECTURA DE GESTIÓN DE RED [24]	8
2.1.2. FUNCIONALIDAD DE LOS SISTEMAS DE GESTIÓN	9
En la <i>Tabla 1</i> , se presenta más detalladamente las funcionalidades de los sistemas de gestión Modelo Gestor-Agente y, Monitorización y Control.....	
2.1.3. MODELOS DE GESTIÓN DE REDES.....	9
2.1.3.1. MODELO DE GESTIÓN DE LA RED ISO	10
2.1.4. SNMP [6]	12
2.1.4.1. RMON [12].....	13
2.1.5. CONCLUSIONES	13
2.2. GESTIÓN DE REDES MÓVILES AD-HOC Ó MANET.....	14
2.2.1. ARQUITECTURA DE REDES MANET [13][23].....	14
2.2.1.1. Capa Física.....	15
2.2.1.2. Capa MAC.....	15

2.2.1.3.	Capa de Red.....	15
2.2.1.4.	Capa de Transporte.....	16
2.2.1.5.	Capa de Aplicación.....	16
2.2.2.	PROTOCOLOS DE ENRUTAMIENTO DE REDES MANET	16
2.2.2.1.	Protocolos Proactivos (o basado en tablas).....	17
2.2.2.2.	Protocolos Reactivos (o sobredemanda).....	18
2.2.2.3.	Protocolos Híbridos [23].....	20
2.2.4.	ARQUITECTURA DE ADMINISTRACIÓN “GUERRILLA” PARA MANET’s [8]	23
2.2.4.1.	Modelos y Mecanismos Esenciales.....	23
2.2.5.	GESTIÓN DE RED MOVILES AD – HOC BASADA EN POLITICAS [27]	27
2.2.6.	CONCLUSIONES	28
2.3.	CLUSTERING	30
2.3.1.	ALGORITMOS DE CLUSTERING	30
2.3.1.1.	Lowest ID[12] (SEAH, 2004).....	30
2.3.1.2.	Highest -degree.....	30
2.3.1.3.	K-hop CONID (k-hop connectivity ID)	31
2.3.1.4.	Algoritmo (α, t) cluster framework.....	31
2.3.1.5.	Algoritmo Max-min d-cluster [12]	33
2.3.1.6.	MobDhop	33
2.3.1.7.	Distributed Mobility – Adaptive Clustering (DMAC)	34
2.3.1.8.	Algoritmo Weighted Clustering Algorithm (WCA).....	36
2.3.2.	COMPARACIÓN DE LAS TÉCNICAS DE CLUSTERING.....	37
2.4.	REDES NEURONALES Y TEORIA DE JUEGOS	39
2.4.1.	AGENTES INTELIGENTES EN REDES MANET [19]	39
2.4.1.1.	Tipos de Agentes Inteligentes.....	39
2.4.1.2.	Características de los Agentes Inteligentes.....	41
2.4.2.	REDES NEURONALES [21]	42
2.4.2.1.	Arquitectura de Redes Neuronales.....	44
2.4.2.2.	Perceptrón Simple	44
2.4.3.	TEORIA DE JUEGOS [26]	47
2.4.3.1.	Juegos Cooperativos [26].....	47
2.4.3.2.	Juegos no Cooperativos [26].....	48

2.4.3.3.	Juegos en forma extensiva (árbol)	48
2.4.3.4.	Juegos en forma de estratégica (normal).....	49
2.4.4.	TRABAJOS RELACIONADOS	50
2.4.4.1.	Sistemas de Gestión de Redes Diseño Basado en Agentes Inteligentes [11]	50
2.4.4.2.	Componentes del Sistema Administrativo de Redes Ad-Hoc basado en Políticas.....	51
2.4.4.3.	MINERVA: An Event Based Model For Extensible Network Management [11].....	53
2.4.5.	CONCLUSIONES	54

CAPITULO III

3.	DEFINICIÓN DEL MODELO DE GESTIÓN DE REDES MÓVILES AD HOC	56
3.1.	INTRODUCCIÓN.....	56
3.2.	JUSTIFICACIÓN DEL MODELO.....	57
3.3.	COMPONENTES Y DEFINICIÓN DEL MARCO DE TRABAJO	57
3.3.1.	Modelo Propuesto.....	58
3.3.2.	Arquitectura basada en Técnicas de Clustering [25]	61
3.3.3.	Arquitectura basada en Redes Neuronales.....	62
3.3.3.1.	Modelo del núcleo basado en SWAN [5]	65
3.3.4.	Arquitectura basada en Teoría de Juegos	69
3.3.5.	Integración de la Arquitectura basada en Técnicas de Clustering, Redes Neuronales y Teoría de Juegos	70
3.4.	CONTRIBUCIONES	74
3.5.	CONCLUSIONES	75

CAPITULO VI

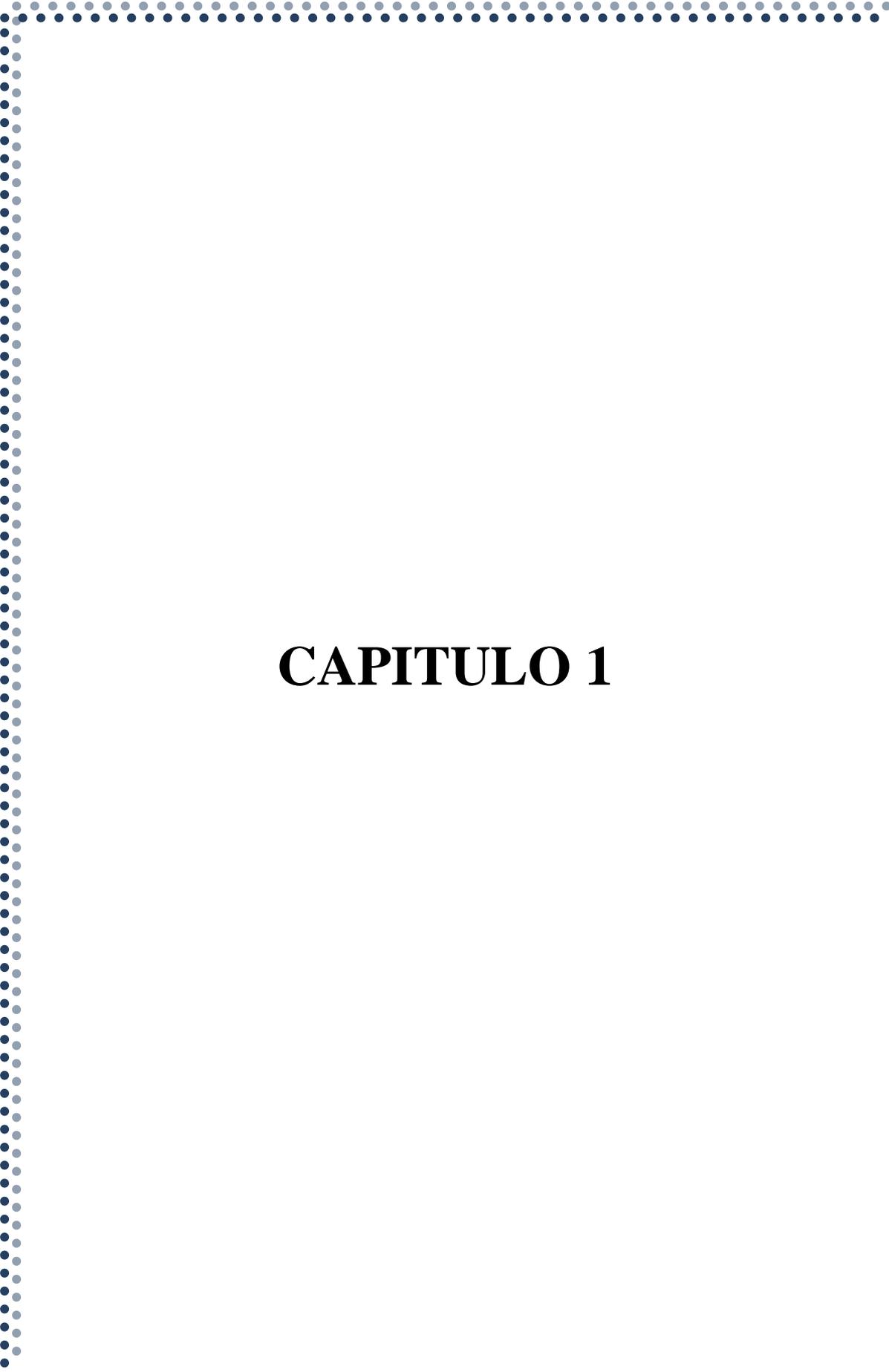
4.	REDES AD HOC CON REDES NEURONALES Y TEORÍA DE JUEGOS	77
4.1.	INTRODUCCIÓN.....	77
4.2.	DETERMINACIÓN DE NODOS EGOISTA A TRAVÉS DE WCA Y TEORIA DE JUEGOS.....	77
4.2.1.	Modelo de Confianza y Dilema del Prisionero [1].....	79
4.3.	EMPLEO DE REDES NEURONALES Y TEORÍA DE JUEGOS	82
4.3.1.	Componentes del Modelo.....	82
4.3.1.1.	Núcleo [7]	82
4.3.1.2.	Plan de Aprendizaje Inteligente.....	94
5.	CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS.....	100

INDICE DE TABLAS

TABLA 1. FUNCIONALIDADES DE LOS SISTEMAS DE GESTIÓN.....	9
TABLA 2. MODELO DE GESTIÓN ISO.....	11
TABLA 3. COMPONENTES DE SNMP.....	12

INDICE DE FIGURAS

<i>Figura 1.</i> Arquitectura de Gestión de Redes [24]	8
<i>Figura 2.</i> Capas del modelo de referencia OSI [13]	10
<i>Figura 3.</i> Arquitectura en capas de un nodo en una MANET y los protocolos que se implementan en cada una [15]	15
<i>Figura 4.</i> Proceso de funcionamiento de DSDV [32]	18
<i>Figura 5.</i> Registro durante el descubrimiento de una ruta [19]	18
<i>Figura 6.</i> Proceso de funcionamiento de AODV [13]	19
<i>Figura 7.</i> Arquitectura jerárquica de ANMP [35]	21
<i>Figura 8.</i> Modelo manager/agente & modelo cliente/agencia [8].....	24
<i>Figura 9</i> Clasificación de nodos en la arquitectura de gestión “La Guerrila” [8]	25
<i>Figura 10.</i> Formación del clúster mediante lowest id [8]	30
<i>Figura 11.</i> Tipos de Agentes Inteligentes [19]	39
<i>Figura 12.</i> Red Neuronal Genérica [21].....	43
<i>Figura 13.</i> Neurona Artificial [21]	43
<i>Figura 14.</i> Arquitectura de una neurona [21]	44
<i>Figura 15.</i> Arquitectura y función de transferencia del perceptrón simple [31]	45
<i>Figura 16.</i> Regiones de decisión obtenidas para el perceptrón simple (arriba), perceptrón multicapa (medio) y perceptron multicapa con dos capas ocultas (abajo) [21]	46
<i>Figura 17.</i> Arquitectura y función de activación para el perceptrón multicapa [31]	46
<i>Figura 18.</i> Arquitectura del perceptrón multicapa [31]	47
<i>Figura 19.</i> Juego en forma extensiva [26]	49
<i>Figura 20.</i> Arquitectura de alto nivel [11].....	51
<i>Figura 21.</i> Componentes del Sistema [8]	52
<i>Figura 22.</i> Modelo Propuesto	60
<i>Figura 23.</i> Esquema de una red ad hoc dividida en clusters [25]	61
<i>Figura 24.</i> Interfaces de comunicaciones	62
<i>Figura 25.</i> Modelo de redes neuronales con soporte multimedia [7].....	64
<i>Figura 26.</i> Modelo del núcleo basado en SWAN [7]	65
<i>Figura 27.</i> Diagrama de operaciones [7]	67
<i>Figura 28.</i> NM1 y NM2 necesitan cooperar entre ellos para llevar información a su destino	70
<i>Figura 29.</i> Esquema de un nodo JC de la red Ad Hoc	71
<i>Figura 30.</i> Determinación de nodos egoístas	72
<i>Figura 31.</i> Esquema de un NM.....	73
<i>Figura 32.</i> Determinación de nodos egoístas. Fase de control de admisión.....	74
<i>Figura 33.</i> Formato RREQ.....	84
<i>Figura 34.</i> Formato RREP	85
<i>Figura 35.</i> Soft-states de un nodo intermedio [7]	92



CAPITULO 1

RESUMEN

En las redes móviles Ad Hoc se maneja una administración descentralizada, donde los nodos pueden ser utilizados como enrutadores, nodos fuente o nodos destino y no existe una entidad central que los administre. Estos nodos poseen recursos limitados como ancho de banda y poder de batería, es por eso que en la actualidad resulta un reto utilizar nuevos mecanismos para optimizar la calidad y servicio, y conservar por más tiempo los recursos de los nodos, es por eso que en la presente investigación se pretende utilizar técnicas de redes neuronales y teoría de juegos con el fin de mejorar el rendimiento de la misma.

El presente trabajo parte de un modelo definido a través de técnicas de clustering en donde se incrementa equipos de respaldo para el JC y el SA; el aporte adicional de la presente investigación a este modelo es el incremento de teoría de juegos y redes neuronales en una de sus interfaces propuestas (*Figura 23*), específicamente en la interfaz uno y dos.

Adherir técnicas de teoría de juegos es con el fin de ver la posibilidad de si existe algún método que permita determinar si un nodo dentro del clúster es egoísta, para así poder aislarlo o tomarlo en cuenta dentro de la red, esto ayuda a optimizar el ahorro de energía dentro de los nodos, y el investigar y proponer una solución a través de redes neuronales multicapa feedforward¹ es para incrementar aprendizaje inteligente que permita mejorar la QoS dentro de la red para ofrecer así garantías de servicio sobre el tráfico de la red.

1. INTRODUCCIÓN

Las redes móviles Ad Hoc son de naturaleza dinámica y auto-configurables, presentan recursos limitados como ancho de banda, poder de batería, capacidad de procesamiento, entre otros. La arquitectura de las redes Ad Hoc está basada en el estándar IEEE 802.11 [15].

Los modelos de gestión para redes inalámbricas Ad Hoc existentes no cubren todas las necesidades que presentan este tipo de redes y conservación de recursos garantizando la calidad de servicio en la red.

¹ Feedforward: es un conjunto de neuronas organizadas en capas.

Con la finalidad de cubrir algunas de las deficiencias notables que presentan los modelos de gestión ya existentes se desarrollo la presente investigación para proponer un modelo de gestión que permita cubrir la mayoría de las necesidades para así maximizar la disponibilidad, mejorar la rapidez, eficacia y eficiencia de la red.

La investigación inicia desde lo propuesto en [25] en donde se utilizan técnicas de clustering para la segmentación de la red; esto es complementado con la propuesta de adherir de técnicas innovadoras basadas en una herramienta de inteligencia artificial como lo es redes neuronales y teoría de Juegos.

Este modelo [25] se basa en técnicas de clustering [2], [10] para controlar la movilidad, el cambio de topología debido a la entrada y salida de equipos en la red, y para segmentar la red llevando la gestión de mejor manera en grupos.

Dentro del modelo [25] se definen varias interfaces entre las cuales consta la comunicación entre un nodo manejado y un nodo jefe de cluster. El jefe de cluster tiene información de todos los nodos manejados, es por eso que en esta investigación se propone adherir en esta interfaz redes neuronales y teoría de juegos.

Las redes neuronales feedforward son una herramienta de inteligencia artificial que se propone adherir a partir de las operaciones realizadas en el núcleo propuesto en [7], esto ayudará a mejorar la transmisión de paquetes en tiempo real de extremo a extremo. La utilización de redes neuronales permite obtener un aprendizaje inteligente a partir de experiencias obtenidas en las operaciones básicas realizadas en [7], esto ayuda a que un nodo tome la mejor decisión para tomar el siguiente salto sin necesidad de buscar la ruta, mejorando así la calidad de servicio en las transmisiones.

Con la finalidad de unificar teoría de juegos y redes neuronales al modelo planteado en [25], se propone incrementar dentro de la fase de control de admisión teoría de juegos para controlar el comportamiento de un nodo, determinando si es egoísta o no para evitar el desgaste de la batería de los nodos y obtener mejores resultados manteniendo la red por más tiempo evitando el desgaste de recursos, esto a través de la utilización de un modelo de confianza que permita decidir a los nodos con que nodo cooperar y con cuál no. El método utilizado en teoría de juegos para realizar este control es el dilema del prisionero [1].

Después de investigar, analizar y estudiar los diferentes métodos de redes neuronales y teoría de juegos que se puedan utilizar en redes Ad Hoc con el fin de mejorar la calidad de servicios para mejorar la gestión de la red y maximizar su rendimiento de los nodos, se propone un modelo unificando estas dos técnicas.

1.1. Definición del problema

Las redes Ad Hoc son redes móviles inalámbricas que no requieren ningún tipo de administración centralizada debido a que son distribuidas con topología dinámica. Los nodos que forman parte de la red tienen características heterogéneas, además poseen limitaciones como: energía (batería), capacidad de procesamiento y ancho de banda.

La gestión de este tipo de redes debe considerar todas las características y necesidades que presentan. El problema surge debido a que no existe un modelo de gestión óptimo que ayude a controlar las limitaciones que presentan estas redes, es por eso que se plantea una nueva propuesta con la finalidad de cubrir gran parte de las necesidades que presentan.

En la propuesta que se presentará más adelante se ha tratado de unificar técnicas de clustering y algunas soluciones basadas en redes neuronales y teoría de Juegos. Asimismo, para la utilización de redes neuronales se ha visto en la necesidad de partir de las operaciones del modelo de calidad SWAN en el cual se incorporan técnicas orientadas a ofrecer calidad de servicio (Quality of Service, QoS) las que permiten, ofrecer garantías de servicio sobre el tráfico de la red que en la actualidad no se toman en cuenta en algunos modelos.

1.2. Hipótesis

Una vez realizada la investigación se asume algunas hipótesis, las cuales se verificarán a través del estudio de los métodos de redes neuronales y teoría de juegos. Se plantean las siguientes hipótesis:

- Existen nuevas técnicas como redes neuronales y teoría de juegos que se pueden aplicar dentro de la gestión de redes Ad Hoc.
- A partir de modelos ya existentes para redes Ad Hoc, teóricamente se pueden proponer nuevos escenarios en base a de redes neuronales y teoría de juegos, que

ayudan a discernir de entre los nodos de la red que nodo es egoísta y a mejorar la calidad y servicio en la red.

- A nivel teórico se puede modelar analíticamente los escenarios de aplicación propuestos para mejorar la gestión de redes Ad Hoc en base a redes neuronales y teoría de juegos.

Con el objetivo de comprobar la hipótesis se realizó lo siguiente:

- ⇒ Investigar y estudiar los fundamentos teóricos de Gestión de Redes fijas y Ad Hoc
- ⇒ Investigar los principios de redes neuronales y teoría de juegos
- ⇒ Crear teóricamente escenarios de aplicación para redes Ad Hoc en donde se pueda incluir redes neuronales y teoría de juegos.

1.3. Objetivos

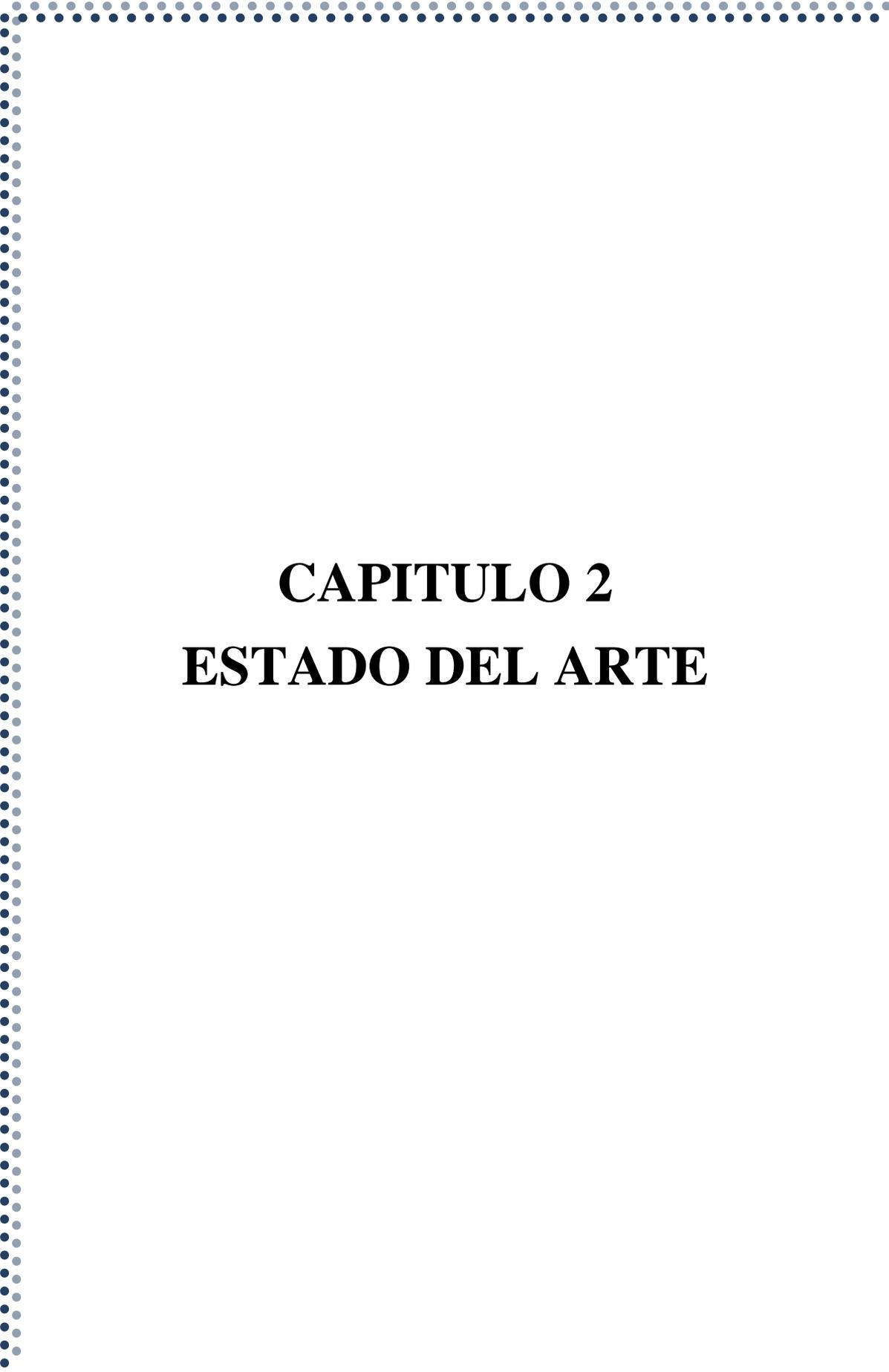
Se definen algunos objetivos con la finalidad de justificar la presente investigación:

- Investigar, estudiar y analizar los modelos de gestión de redes móviles Ad Hoc.
- Investigar, estudiar y analizar los fundamentos teóricos y su aplicabilidad de redes neuronales y teoría de juegos para redes Ad Hoc.
- Proponer teóricamente una solución basada en redes neuronales para mejorar la calidad y servicio de la red.
- Proponer teóricamente una solución para el control de nodos egoístas en la red a través de teoría de juegos.
- Modelar analíticamente la propuesta basada en redes neuronales y teoría de juegos para redes Ad Hoc.

1.4. Estructura de la tesis

La tesis está estructurada de la siguiente manera: En el capítulo 1 se da una introducción a nivel general de la investigación, seguido por la definición del problema, hipótesis y objetivos. En el capítulo 2 se presenta el estado del arte con respecto a la gestión de redes móviles Ad Hoc, en este capítulo se estudia los modelos de gestión existentes tanto para redes fijas como para redes móviles, además se estudia los fundamentos teóricos en base a redes neuronales y teoría de juegos y los trabajos relacionados con estos temas, donde se demuestra la primera hipótesis. En el capítulo 3 se plantea una

propuesta para la gestión en base a técnicas de redes neuronales y teoría de juegos, donde se demuestra la segunda hipótesis. En el Capítulo 4 se modela analíticamente el modelo planteado en el capítulo tres demostrando así la tercera hipótesis y finalmente se concluye lo planteado en el capítulo 5.



CAPITULO 2
ESTADO DEL ARTE

2. ESTADO DEL ARTE

2.1. GESTIÓN DE REDES

La gestión de redes aparece como solución a los diversos problemas que se pueden suscitar en la administración, supervisión y control de la red, esto con el fin de maximizar la eficiencia y productividad de la misma.

La gestión de redes es un conjunto de herramientas, aplicaciones y dispositivos, dedicadas a la vigilancia con el fin de ayudar a los administradores a la supervisión, mantenimiento y control de recursos de la red.

2.1.1. ARQUITECTURA DE GESTIÓN DE RED [24]

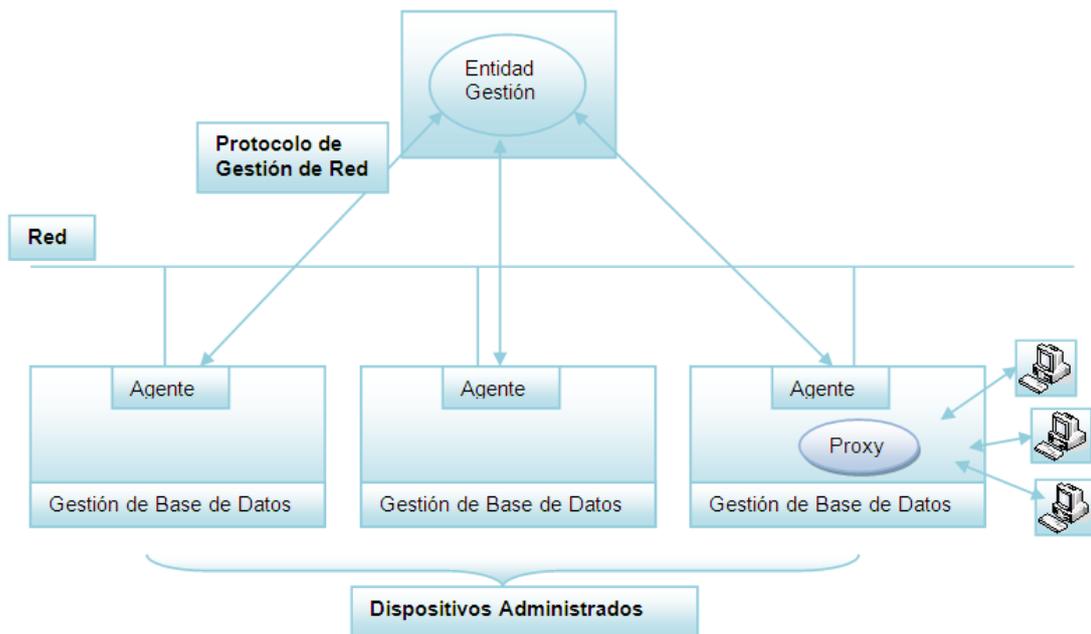


Figura 1. Arquitectura de Gestión de Redes [24]

En la *Figura 1* se presenta la arquitectura para la administración de redes. Debido a que la mayoría de redes presenta esta infraestructura vamos a hablar un poco acerca de cómo es su funcionamiento.

Las estaciones terminales sean estos dispositivos de red o sistemas de computo utilizan un software que les permite enviar mensajes de alerta cuando se detecta algún problema en la red.

Las entidades de administración están programadas para reaccionar cuando reciban un mensaje de alerta, ejecutando una o varias acciones y notificando al administrador, el cierre del sistema y un proceso automático para la posible reparación del sistema.

Además la información de las estaciones terminales pueden ser registradas por las entidades de administración con el fin de verificar el valor de ciertas variables.

2.1.2. FUNCIONALIDAD DE LOS SISTEMAS DE GESTIÓN

En la *Tabla 1*, se presenta más detalladamente las funcionalidades de los sistemas de gestión Modelo Gestor-Agente y, Monitorización y Control.

Tabla 1. Funcionalidades de los Sistemas de Gestión

SISTEMAS	CARACTERÍSTICAS
Modelo Gestor – Agente	<p>Posee una interfaz con el operador o el responsable de la red, además componentes hardware y software en los diferentes componentes de la red permitiendo clasificar a los sistemas de gestión de red en dos grupos:</p> <p>Gestor: atienden las operaciones solicitadas invocando acciones pertinentes para realizar las operaciones solicitadas. Interactúan con los operadores humanos.</p> <p>Agente: realiza las operaciones solicitadas por los gestores de la red.</p>
Monitorización y Control	<p>Monitorización: es observar y analizar el estado y comportamiento de los recursos gestionados en la red.</p> <p>Fases:</p> <ul style="list-style-type: none"> Definición de la información que se va a monitorear Acceso a la información de monitorización Diseño de políticas de monitorización Procesamiento de la información de monitorización <p>Control: es la parte encargada de modificar parámetros e invocar acciones en los recursos gestionados.</p>

2.1.3. MODELOS DE GESTIÓN DE REDES

Los sistemas de gestión de redes tienen una evolución novedosa van desde **Gestión Autónoma, Homogénea, Heterogénea y Integrada.**

- **Gestión Autónoma:** consta de pocos nodos, tiene su propio administrador. En este tipo de redes cuando ocurría un problema los administradores de las distintas redes se comunicaban con el fin de solucionarlo.
- **Gestión Homogénea:** debido al crecimiento de los nodos dentro de las redes surgen aplicaciones que posibilitaban la supervisión remota de los nodos, no obstante estas aplicaciones solo servían para equipos provenientes de un mismo fabricante.

- **Gestión Heterogénea:** surge con el fin de controlar las redes que tienen recursos heterogéneos.
- **Gestión Integrada:** permite la utilización de un único centro de gestión válido para llevar el control de entornos heterogéneos.

Dentro de la gestión integrada existen tres modelos de gestión:

- **Gestión de red OSI** (Interconexión de Sistemas Abiertos), definido por ISO con el objetivo de lograr la gestión de los recursos del modelo de referencia OSI
- **Gestión Internet**, es definido por Internet Society para gestionar el modelo de referencia TCP/IP
- **Arquitectura TMN** (Telecommunications Management Network)

2.1.3.1. MODELO DE GESTIÓN DE LA RED ISO

El modelo de gestión de ISO es fundamental para entender las funciones fundamentales del resto de sistemas de gestión, debido a su gran contribución dentro de la estandarización de redes.

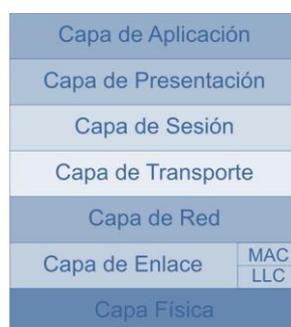


Figura 2. Capas del modelo de referencia OSI [13]

En la *Figura 2*, se muestra el modelo ISO, compuesto por siete capas:

1. **Capa física:** transmite símbolos entre sistemas conectados al mismo medio físico.
2. **Capa de enlace:** posee dos subcapas: MAC (Capa de control de acceso al medio) y LLC (Logical Link Control (Capa de Control de Enlace Lógico)). La MAC proporciona mecanismos de control para acceso de manera ordenada al canal por parte de los usuarios. La capa LLC maneja el intercambio de bloques con control de errores.

3. **Capa de red:** comunica a los sistemas que no se encuentran conectados directamente en un mismo medio físico.
4. **Capa de transporte:** conexión entre usuarios finales y aislamiento de las capas superiores frente a las distintas implementaciones de tecnologías de red.
5. **Capa de sesión:** es la encargada de la organización y estructura de las conexiones entre usuarios finales.
6. **Capa de presentación:** representación sintáctica homogénea de la información.
7. **Capa de aplicación:** interpreta la semántica de la información.

La ISO clasifica las tareas de los sistemas de gestión en cinco áreas funcionales:

1. Gestión de configuración
2. Gestión de funcionamiento
3. Gestión de seguridad
4. Gestión de fallos
5. Gestión de estadística

En la *Tabla 2* se presenta una breve síntesis de las áreas conceptuales del modelo de gestión ISO:

Tabla 2. Modelo de Gestión ISO

Gestión de:	Objetivo	Proceso
Configuración	Obtener los datos de la red y utilizarlos para incorporar, mantener y retirar los diferentes componentes y recursos que la integran. Supervisar la información de configuración de la red y de los sistemas para rastrear y manejar los efectos sobre el desempeño de las versiones de software y hardware de la red.	Consiste de tres tareas fundamentales: 1. Recolección de datos sobre el estado de la red , utilizan dos herramientas de auto-discovery y auto-mapping . 2. Cambio en la configuración de los recursos 3. Almacenamiento de los datos de configuración
Funcionamiento	Poseer la información disponible del desempeño de la red con el fin de mantener el funcionamiento de la red interna.	La información del funcionamiento de la red se recopila en variables, esta información se analiza para determinar los niveles normales de la red y finalmente se determina las variables que exceden los umbrales apropiados de la red y así se envían mensajes de algún posible problema en la red.
Seguridad	Proteger la red y la información que transporta frente a accesos y usos no autorizados ofreciendo mecanismos que faciliten el mantenimiento de políticas de seguridad.	Identificar y localizar la información en donde se encuentre, identificación y protección de los puntos de acceso a la información y mantenimiento de los puntos de acceso protegidos.

Fallos	Mantener la red disponible ante cualquier fallo, detectando, registrando y notificando los problemas de la red y ejecutar un proceso de corrección automática para mantener funcionando red.	Implica los siguientes pasos: <ul style="list-style-type: none"> ▪ Monitorizar la red o estado del sistema ▪ Recibir y procesar alarmas ▪ Diagnosticar las causas de fallos ▪ Determinar la propagación de errores ▪ Medidas de recuperación de errores
Estadística	Registrar el uso de recursos y servicios proporcionados por la red para regular apropiadamente las aplicaciones de un usuario o grupo en la red.	Requiere de las siguientes tareas: Recolección de datos sobre la utilización de recursos, mantenimiento de registro de cuentas, asignación y monitorización de cuotas de uso.

2.1.4. SNMP [6]

La versión 1 del protocolo simple de gestión de red (SNMP), fue definida por el RFC 157 en mayo de 1990. SNMP proporcionó una manera persistente de supervisar y administrar una red informática, convirtiéndose rápidamente en un estándar de facto para la administración de redes, definiendo la comunicación de un administrador con un agente.

En los RFC 1441 a 1452 se definió una versión mejorada del SNMP (SNMPv2) que se volvió un estándar en Internet.

El modelo SNMP de una red administrada consta de cuatro componentes como se presenta en la *Tabla 3*:

Tabla 3. Componentes de SNMP

COMPONENTES	DESCRIPCIÓN
NODOS ADMINISTRADOS	Un nodo para ser administrado por SNMP debe ser capaz de ejecutar un proceso de administración SNMP llamado agente SNMP . Un agente mantiene una base de datos local de variables que describen su estado e historia y que afectan a su operación.
ESTACIONES ADMINISTRATIVAS	Son equipos que contienen un software de administración, el cual contiene uno o más procesos que se comunican con los agentes a través de la red emitiendo comandos y recibiendo respuestas.
INFORMACIÓN DE ADMINISTRACIÓN	Describe la información exacta de cada tipo de agente que tiene que administrar y el formato con el que el agente tiene que proporcionar los datos.
PROTOCOLO DE ADMINISTRACIÓN	Este protocolo permite a la estación administradora consultar y modificar el estado de los objetos locales de un agente.

Otro suceso importante dentro de SNMP es el llamado **interrupción SNMP**, el cual es un informe que se emite cuando un agente nota que ha ocurrido un suceso significativo y de inmediato es informado a todas las estaciones administradoras.

2.1.4.1. RMON [12]

Monitoreo Remoto, extiende la funcionalidad de SNMP sin cambiar el protocolo permitiendo el monitoreo de segmentos de redes.

RMON funciona normalmente en un formato consola-sonda, donde las sondas recogen información dependiendo de los grupos definidos en la especificación, enviándola después a la consola, donde puede ser interpretada por el administrador.

RMON trabaja de las siguientes maneras:

Modo Off-line: el monitor recoge permanentemente información de la red, sin enviar nada al gestor a menos que surja eventos o fallos muy graves en la red.

Monitorización anticipada: el monitor tiene la capacidad de realizar diagnósticos monitorizando y registrando la actividad en la red, si es necesario comunica al gestor.

Detección de problemas e informes: debido a que el monitor continuamente registra el comportamiento de la red, puede realizar informes o registros.

Datos de valor añadido: de los informes generados por el monitor se puede realizar un análisis de los datos y así deducir información importante, como la IP del nodo que más tráfico genera en la subred.

Múltiples gestores: un solo monitor puede ser configurado para trabajar con múltiples gestores, con lo que se intensifica la confiabilidad de la información.

2.1.5. CONCLUSIONES

La gestión de redes tiene como objetivo principal permitir la administración, supervisión y control de la red, lo cual es posible mediante la implementación de herramientas, aplicaciones y dispositivos dedicados a la vigilancia con el fin de ayudar a los administradores al mantenimiento y control de recursos de la red.

La heterogeneidad de los equipos y el crecimiento de los nodos dentro de la red han hecho que los modelos de gestión avancen cada vez más con el fin de mejorar la administración de la red.

Uno de los modelos más idóneos dentro de la gestión de redes es el SNMP, debido a que administra cuatro componentes como: nodos, estaciones, información y protocolo.

2.2. GESTIÓN DE REDES MÓVILES AD-HOC ó MANET

Las redes móviles Ad-Hoc ó también conocidas como MANET, son redes inalámbricas las cuales no requieren ningún tipo infraestructura ni administración centralizada, donde las estaciones ofrecen funcionalidades de estación final y además deben proporcionar servicios de encaminamiento, retransmitiendo los paquetes entre aquellas estaciones que no tienen conexión inalámbrica directa.

Las redes ad hoc pueden combinarse con redes locales inalámbricas o desplegarse de manera autónoma para conectarse a Internet utilizando puntos de acceso inalámbricos.

Debido a su naturaleza dinámica de las redes ad hoc, estas deben poder adaptarse a los cambios continuos de su topología, potencia de señal, tráfico de la red y distribución de carga.

Para poder controlar ó administrar las características singulares de las redes ad hoc es necesario la implementación de algoritmos, protocolos y middleware; que superen las limitaciones y permitan establecer redes independientes y descentralizadas. Los protocolos deben ser completamente adaptativos, anticipando el comportamiento futuro de la red a partir de parámetros tales como nivel de congestión, tasa de errores, cambios de rutas utilizadas, etc.

Los recursos de la red deben poder ser localizados y utilizados de forma automática sin necesidad de una configuración manual anteriormente establecida. Además aspectos relacionados con la seguridad y la privacidad de la información deberían ser considerados para proporcionar un acceso que permita asegurar la privacidad de los dispositivos y usuarios. Asimismo, se deben incorporar técnicas orientadas a ofrecer calidad de servicio (Quality of Service, QoS) que permitan, ofrecer garantías de servicio sobre determinado tráfico de la red.

2.2.1. ARQUITECTURA DE REDES MANET [13][23]

Los mecanismos que proporcionan los servicios de red se distribuyen en capas, formando una pila. A continuación se describe el modo de operación de cada una de las capas:

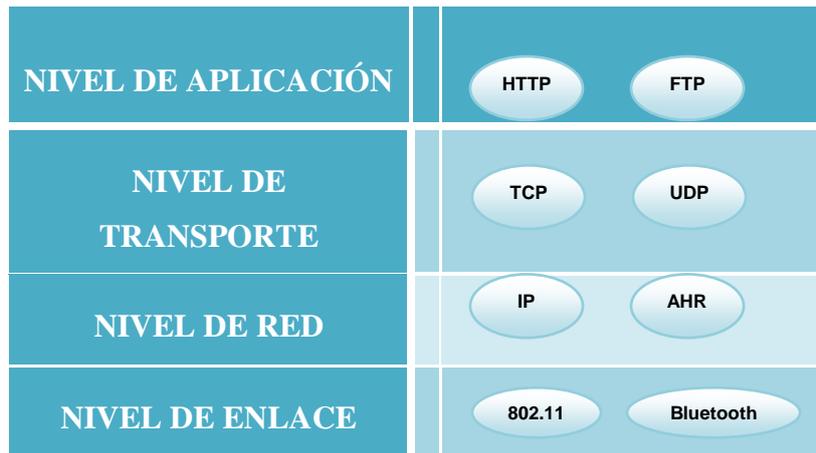


Figura 3. Arquitectura en capas de un nodo en una MANET y los protocolos que se implementan en cada una [15]

2.2.1.1. Capa Física

Los nodos de la red se comunican mediante canales de radiofrecuencia (RF), estos presentan características que dificultan la calidad de servicio en redes inalámbricas, algunos de los efectos que sufre la señal cuando se transmite en un canal RF son: el efecto doppler, la atenuación de la señal, el desvanecimiento por multitrayecto, etc.

Los estándares IEEE 802.11 definen interfaces para canales de RF en las bandas de los 2.4GHz y de los 5GHz, siendo la primera la más extendida.

2.2.1.2. Capa MAC

Existen dos categorías principales de protocolos de control de acceso al medio: los protocolos de acceso aleatorio, en los cuales los nodos compiten entre sí para ganar el acceso al medio de transmisión compartido, y los protocolos de acceso controlado, en los cuales un nodo maestro o de infraestructura decide cuál de los nodos puede acceder al medio de transmisión en cada momento.

Entre los Protocolos MAC de Acceso Controlado se encuentran los siguientes: TDMA (*Time Division Multiple Access*), FDMA (*Frequency Division Multiple Access*), CDMA (*Code Division Multiple Access*) y TSMA (*Time Spread Multiple Access*).

2.2.1.3. Capa de Red

La característica más importante de los protocolos de encaminamiento para redes MANET es que deben poder adaptarse rápidamente a los cambios continuos de la red, con el fin de mantener las rutas entre los nodos que se están comunicando. De manera

general, los protocolos de encaminamiento para redes MANET se clasifican en categorías principales: proactivos, reactivos y híbridos.

2.2.1.4. Capa de Transporte

Las redes MANET tienen como objetivo final la implantación de la Internet inalámbrica omnipresente y ubicua, por lo que está basada en la pila de protocolos TCP/IP.

En los últimos tiempos se ha presentado propuestas encaminadas a mejorar las prestaciones de los principales protocolos de transporte de la Internet, TCP (*Transmisión Control Protocol*) y UDP (*User Datagram Protocol*), en entornos MANET.

2.2.1.5. Capa de Aplicación

Uno de los objetivos es que las aplicaciones diseñadas para la Internet actual puedan funcionar en las MANETs. Esto es válido actualmente para las aplicaciones de transmisión de datos que pueden conformarse con una entrega de datos, sin grandes requerimientos de ancho de banda ni restricciones en cuanto al tiempo de entrega de los paquetes.

2.2.2. PROTOCOLOS DE ENRUTAMIENTO DE REDES MANET

Inicialmente para el estudio de las redes Ad hoc es necesario tener conocimiento de los protocolos de enrutamiento para comprender el desempeño tanto interno como externo de las mismas al momento de establecer una ruta entre los nodos dentro del cluster de la red.

Los protocolos de enrutamiento proporcionan una ruta de transmisión de paquetes del nodo fuente al nodo destino. Los protocolos de enrutamiento que utilizan las redes Ad Hoc, sin duda son diferentes a los protocolos que utilizan las redes fijas.

Debido a que existe ausencia de servidores centrales, conceptos como la identidad y confianza deben estar bien definidos en las redes Ad Hoc, además es necesario que los nodos de la red se puedan comunicar sin riesgo, es decir que nodos ajenos a la red intenten hacerse pasar por nodos con los que en realidad se desea comunicar.

Existen varios protocolos de enrutamiento desarrollados para redes cableadas que no se adaptan al entorno altamente dinámico de las redes Ad Hoc. Los protocolos de redes

cableadas hacen uso de mensajes de actualización periódicos de rutas que ofrecen una elevada sobrecarga incluso en redes con tráfico reducido. Esta metodología de diseño hace que en entornos dinámicos con cambios de topología frecuente, se presente una sobrecarga excesiva, por tal razón aparecen nuevos protocolos especializados para de redes MANET o Ad –Hoc.

Los protocolos de redes Ad Hoc se pueden clasificar en tres grupos:

2.2.2.1. Protocolos Proactivos (o basado en tablas)

Mantienen en cada nodo información actualizada acerca de la topología de la red, la almacena en tablas de enrutamiento las cuales son actualizadas de forma periódica o cuando ocurre algún suceso.

Este tipo de protocolos están basados en los protocolos de vector distancia y estado de enlace.

1. OLSR (Optimized Link State Routing) [18],

Mantiene siempre actualizada la información sobre las rutas hacia todos los nodos, para que esté disponible en caso de que sea necesaria.

Lo que lo diferencia de otros protocolos similares son las optimizaciones que se realizan para que la sobrecarga producida por las actualizaciones periódicas sea mínima. El modo en que se hace llegar la información sobre rutas a toda la red es mediante inundación controlada; designando a ciertos nodos encargados de enviarse entre ellos la información, haciéndola llegar al resto de la red, y comprobando que no se envían datos duplicados.

2. DSDV (Destination Sequenced Distance Vector) [18]

Los paquetes de señalización son intercambiados entre nodos vecinos a intervalos regulares de tiempo o emitidos por eventos.

DSDV proporciona la obtención de rutas sin bucles mediante la introducción de números de secuencia, suministrando una sola ruta para cada destino, siendo esta la ruta con la trayectoria más corta posible como se puede observar en la *Figura 4*.

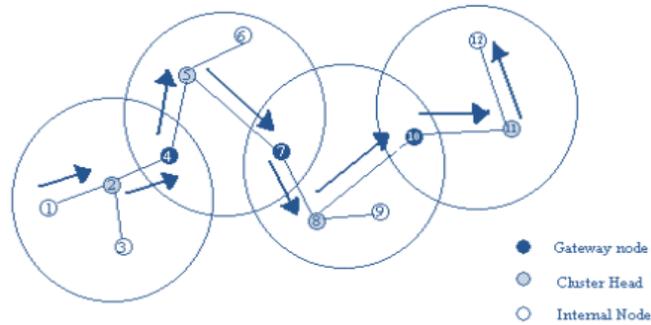


Figura 4. Proceso de funcionamiento de DSDV [32]

2.2.2.2. Protocolos Reactivos (o sobredemanda)

Son algoritmos, donde solo se crean rutas únicamente cuando un nodo fuente desea enviar información hacia un nodo destino, utilizan mecanismos de Descubrimiento y Mantenimiento de Ruta.

Los protocolos reactivos a su vez se pueden clasificar en:

BASADOS EN LA FUENTE

En este tipo de protocolos los paquetes de datos transportan la ruta completa de la fuente al destino.

⇒ **DSR (Dynamic Source Routing).**

DSR [18], crea las rutas únicamente en el caso de que un nodo fuente necesite enviar datos hacia un nodo destino utiliza el descubrimiento de ruta como se observa en la *Figura 5*.

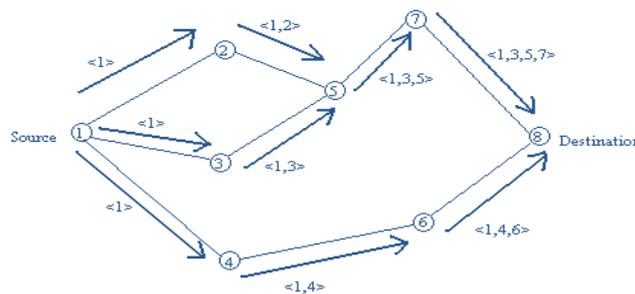


Figura 5. Registro durante el descubrimiento de una ruta [19]

Elimina los mensajes de actualización periódica que usan los protocolos proactivos por lo que optimiza el ancho de banda dentro de una red Ad Hoc.

En este protocolo los nodos no tienen una tabla de enrutamiento, ya que la ruta vieja en un paquete de datos. Utiliza el proceso de Mantenimiento de Ruta en el caso de ocurrir algún problema en la comunicación entre nodos durante la transmisión de datos. Cada nodo asocia un temporizador a la ruta con el fin de poderla eliminar en el caso de que esté inactiva por un cierto periodo tiempo.

SALTO A SALTO

Los protocolos basados en este concepto, únicamente llevan en la cabecera de los paquetes de datos la dirección del destino y la dirección del próximo salto.

⇒ **AODV (Ad-Hoc On-Demand Distance Vector) [13]**

Es un protocolo salto a salto y establece rutas bajo demanda. Características:

- No mantiene actualizaciones periódicas entre nodos
- Descubre y mantiene rutas solamente cuando son necesarias
- Utiliza procesos de Descubrimiento y Mantenimiento de Ruta.
- Usa tablas de enrutamiento en cada nodo para evitar transportar las rutas en los paquetes de datos.



Figura 6. Proceso de funcionamiento de AODV [13]

Utiliza números de secuencia para evitar los lazos de enrutamiento que pudieran darse, y cada cierto tiempo envían paquetes a los nodos vecinos para informar que el nodo sigue activo, de esta forma los nodos actualizan los temporizadores asociados a dicho nodo o deshabilitan las rutas que utilicen el nodo suprimido.

AODV trabaja de la siguiente manera *Figura 6*; un nodo realiza un descubrimiento de ruta haciendo un ‘flooding’ de la red con un mensaje llamado ‘Route Request’. Una vez llega a un nodo que conoce la ruta pedida responde con un ‘Route Reply’ que viaja de vuelta al originador del ‘Route Request’. Después de todo este proceso de trabajo todos los nodos de la ruta descubierta conocen las rutas hacia los dos extremos de la ruta.

Una particularidad de AODV es la reparación a nivel local de un enlace caído que forma parte de una ruta activa. En este caso, el nodo que detecta la caída de un enlace que está siendo utilizado, procede a intentar repararlo comenzando un proceso de descubrimiento de ruta hacia el destino y encola los paquetes de datos recibidos para el destino, hasta localizar una nueva ruta. En el caso de que este intento resulte fallido, se dará lugar al proceso normalmente establecido, con el envío del mensaje de error RERR hacia el nodo origen.

2.2.2.3. Protocolos Híbridos [23]

Es una combinación de los reactivos y los proactivos, se basa en que los nodos más cercanos funcionen entre sí proactivamente y los lejanos reactivamente. La parte reactiva controla la sobrecarga y la proactiva la latencia.

Uno de los protocolos que más interesa dentro de la presente investigación sin duda es el ANMP.

2.2.3. ANMP (Protocolo de Gestión de Redes Ad Hoc) [35]

Es un protocolo de seguridad de gestión de redes móviles Ad Hoc, utiliza agrupamiento jerárquico de los nodos con el fin de reducir el número de mensajes de intercambio entre gestor y agente móvil.

ANMP es totalmente compatible con SNMPv3 y utiliza el mismo PDU para recoger los datos. Para lo cual es conveniente señalar algunos detalles de implementación de ANMO como:

⇒ La PDU (Protocol Data Unit), utiliza una estructura idéntica a la estructura de PDU de SNMP.

⇒ PDU es el protocolo de la capa de transporte utilizado para la transmisión de mensajes ANMP.

⇒ La información es periódicamente actualizada por lo que los datos no se retransmiten por ANMP, pero se puede solicitar al ANMP Manager el reenvío de la información.

Arquitectura de ANMP

Es un protocolo eficiente en el envío de mensajes, su modelo es jerárquico para la colección de datos, es apropiado en los niveles intermedios de la jerarquía para comparar los datos antes de reenviarlos a capas superiores.

Un problema que se da en la utilización de un modelo jerárquico en las redes Ad-Hoc es el costo que implica el mantenimiento de una jerarquía por la movilidad de los nodos. Es por eso que se propone una arquitectura como la que se puede observar en la **¡Error! o se encuentra el origen de la referencia.**, la cuál es de tres niveles.

El nivel inferior consta de nodos individuales denominados agentes. Varios agentes son agrupados en grupos, los cuales están bajo la dirección de la cabeza de grupo. Las cabezas de grupo a su vez están bajo la dirección del administrador de la red como se muestra en la *Figura 7*.

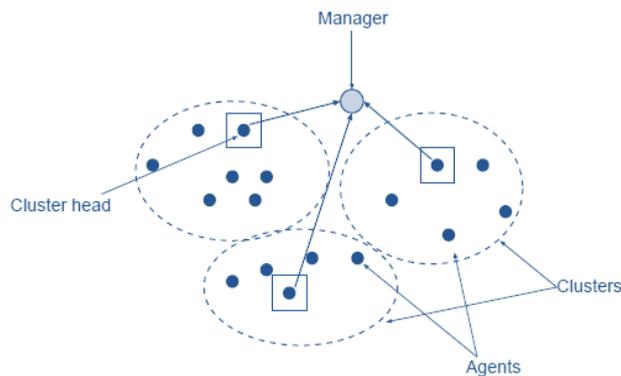


Figura 7. Arquitectura jerárquica de ANMP [35]

Es importante resaltar dos puntos dentro de la arquitectura de ANMP:

- ⇒ La agrupación para la gestión es muy diferente de la agrupación para el enrutamiento.
- ⇒ Un administrador es a menudo más de un salto de distancia de la cabeza de grupo.

La estructura de los nodos es dinámica, es así que los nodos se mueven de un lado a otro, el número y la composición de los cambios de grupo; de igual manera los nodos que sirven de cabezas del grupo y también cambian con el paso del tiempo.

Recolección y Control de Datos

ANMP extiende la MIB usada en SNMP para incluir información específica de redes ad hoc. La información necesaria en redes ad hoc es usualmente muy diferente de la información necesaria en las redes de telefonía fija.

Cada nodo se rellena los campos de las estructuras de la base de gestión de información (MIB) basados en lo que el administrador quiere.

Se utiliza el PDU de SNMP para el intercambio de información entre las entidades ANMP.

Una característica novedosa de ANMP es la capacidad del administrador para ejercer un control total sobre el comportamiento de los agentes.

Por ejemplo: como la mayoría de los nodos en una red ad hoc funcionan mediante una batería, su funcionamiento, debe ser optimizado para maximizar la vida útil de la batería. Esto puede hacerse poniendo el nodo en un ahorro de energía cuando no hay actividad de los usuarios.

Ahora, supongamos que un nodo de la red ad hoc es una región donde hay una alta probabilidad de peligro para el usuario. En tales circunstancias, no es aconsejable para la máquina del usuario, para entrar en modo de espera, incluso si el usuario no está siendo utilizado debido a que la información sobre el peligro si ocurre, debe ser enviado a la dirección.

En este caso, ANMP tiene una planta en la que el administrador puede forzar el dispositivo para permanecer activos y no se apague.

En ANMP, se asocia una función con cada alarma. La función puede ser tan general como queremos que sea y puede alterar el comportamiento del nodo.

Por otra parte, un agente o un administrador pueden descargar dinámicamente una función de asociación diferente con una alarma. En ANMP el alarma se activa cuando el nodo detecta que no hay actividad de los usuarios de X minutos.

En definitiva un administrador en ANMP puede reconfigurar los nodos de la red ad hoc.

Seguridad de ANMP

ANMP implementa la seguridad de unidifusión de SNMPv3. Además ANMP soporta seguridad multidifusión y un modelo de seguridad militar.

Para garantizar que el mensaje no es interceptado por intrusos, es necesario que se transmitan de forma segura (es decir, codificado, con información oportuna para evitar ataques de repetición y con una firma digital). En SNMPv3, esto puede hacerse mediante la transmisión de mensajes individuales desde el gestor de a cada uno de los sensores. Sin embargo, un multidifusión segura es una mejor alternativa, ya que reduce la sobrecarga de mensajes.

ANMP también implementa el modelo de seguridad militar, donde se le asigna las autorizaciones de seguridad a los nodos. De igual manera, la clasificación de asignar niveles de seguridad a los datos. Los cabezas de grupo son incapaces de leer los datos de la MIB que se encuentra a mayor nivel. Sin embargo, pueden leer los datos en los niveles clasificación igual al suyo o inferior.

Un problema que se debe resolver, a fin de aplicar este mecanismo, es asegurarse de que un agente conoce la habilitación de seguridad de la cabeza de grupo que se utiliza para determinar la manera de encriptar los datos.

En ANMP, el administrador asigna los niveles de seguridad y esta información se distribuye de manera más segura a las cabezas de grupo. Los Clúster cabeza transmiten la información de todos sus nodos de clúster cuando se forman los grupos.

2.2.4. ARQUITECTURA DE ADMINISTRACIÓN “GUERRILLA” PARA MANET’s [8]

2.2.4.1. Modelos y Mecanismos Esenciales

a. Modelo Cliente / Agencia para la gestión escalable y de cooperación

Debido a que las redes Ad Hoc crecen constantemente se hace inviable interactuar de manera individual en un sistema centralizado como el que se presenta en el modelo manager/agente, el cual es el típico modelo Cliente/Servidor.

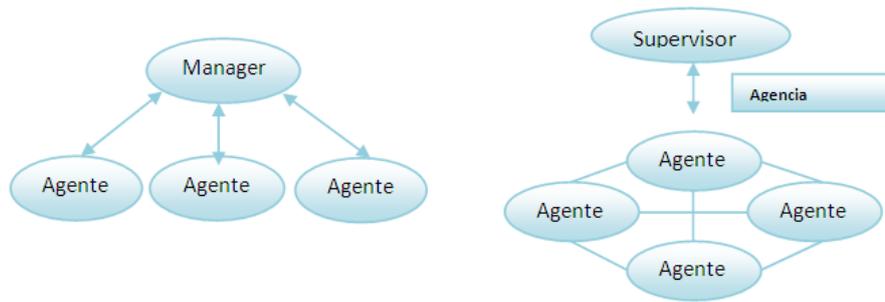


Figura 8. Modelo manager/agente & modelo cliente/agencia [8]

Ahora bien lo que se propone es un modelo Cliente/Agencia, esto con el fin de que los agentes que conforman la agencia trabajen de manera conjunta basándose en un paradigma de colaboración punto a punto.

b. Técnicas de código móvil para gestión nómada y activa

Se propone dos niveles de infraestructura para implementar la capacidad de gestión:

Nivel superior se compone de grupos de directores de nómadas, que se adaptará a la dinámica de la red, poseen una gestión más la inteligencia, colaborar entre sí, y servir como puntos focales para los nodos locales en torno a ellos.

Nivel inferior se compone de ligeros sondas activas para llevar a cabo las operaciones de gestión localizada.

Las sondas activas son programables que se hacen lo suficientemente compactas como para ser de transmisión de tamaño moderado en los paquetes de datos para reducir los gastos de ancho de banda.

c. Enfoque decisión-teórico de gestión inteligente adaptable

Para mantener las operaciones de gestión adaptativa y autónoma, módulos de gestión debe ser capaz de hacer inteligentes decisiones basadas en situaciones de la red.

La Guerrilla adopta una decisión enfoque teórico para adaptación de la gestión y la autonomía.

El software de adaptación, integra la teoría de la utilidad, el aprendizaje de refuerzo, y las técnicas de agente inteligente, dicta el comportamiento de los directores de autónomos.

2.2.4.2. Arquitectura “La Guerrilla” [8]

Se prevé la auto-organización y auto-gestión de los nodos autónomos. Los nodos con mayor capacidad de actúan como administradores mientras que el resto son los nodos administrados.

Los nodos de la red se dividen en grupos, en cada grupo lidera un nodo administrador sin la ayuda de cualquier entidad externa. Debido a la movilidad que presentan los nodos, un nodo puede dejar de formar parte de un grupo para pasar a formar parte de otro grupo.

Un nodo puede dejar de ser administrador debido a la topología, nivel de potencia, densidad del nodo, etc. Si el nivel de un nodo administrador baja de acuerdo a un umbral preestablecido, la inteligencia de administración podrá decidir migrar a otro nodo capaz ó en el caso de que aumente la densidad del nodo administrador este puede decidir la clonación de su inteligencia de gestión a otro nodo que es capaz de acoger a los nodos administrados con el fin de compartir la carga de gestión.

El administrador de los nodos, para minimizar el tráfico de gestión y conservación de ancho de banda inalámbrico puede decidir emitir una sonda activa, que atraviesa una secuencia de nodos administrados de acuerdo a una ruta determinada.

La secuencia de comandos emitidos por la sonda activa realiza la gestión de tareas a través de la interacción con el agente local de SNMP con el fin de lograr el uso económico de ancho de banda, eliminando la necesidad de sondeo constante de la gestión de información de los agentes remotos de SNMP.

Los nodos se clasifican de acuerdo a tres roles diferentes en función de la capacidad y dinámica de la red.

En la *Figura 9* podemos observar los tres roles de los nodos:

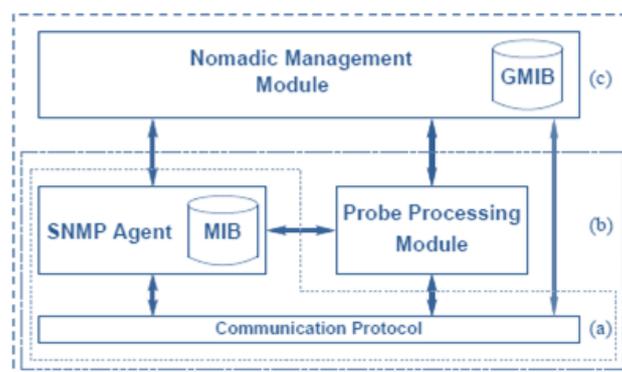


Figura 9 Clasificación de nodos en la arquitectura de gestión “La Guerrilla” [8]

- Nodos que tienen la capacidad de gestión de sólo ejecutar un agente SNMP con el fin de facilitar el acceso remoto a la información de gestión local en un modelo cliente/servidor.
- Nodos con suficiente capacidad están equipados con una sonda Módulo de procesamiento y un agente SNMP.
- Nodos con nivel adecuado de energía y procesamiento para que puedan asumir el papel de nodo administrador. Un nodo administrador mantiene la inteligencia de gestión y los estados, colabora con otros nodos administradores, difunde sondas activas a otros nodos dentro de su ámbito de gestión.

Es importante señalar que la asignación de los nodos se adapta a las diferentes funciones a la dinámica de la red, sin perjuicio de poder y las limitaciones de procesamiento. Un nodo puede cambiar su papel para maximizar la eficacia de gestión definida por las funciones de utilidad. Por ejemplo, un nodo jefe de cluster se le puede agotar la batería y pasa a ser como un solo nodo SNMP para conservar energía, y un nodo de ejecución módulo de procesamiento de la sonda puede convertirse en un jefe de clúster a compartir la carga de trabajo de gestión.

⇒ **Sondas Activas**

Basándose en sus funciones existen dos tipos de sondas activas:

1. Sondas de monitoreo

Un nodo administrador trata de obtener información acerca de sus nodos vecinos a través de sondas de control, estos a su vez se responden a sí mismos para cubrir la zona especificada por el administrador, recogiendo y difundiendo la información hasta que llegue a su nodo administrador local.

2. Sondas de tareas específicas

Recogen información específica ó realizan operaciones específicas, como Enrutamiento de QoS, las sondas pueden ser programadas para realizar dichas tareas.

⇒ **Módulo de Gestión de Nodos (NMM)**

NMM alcanza la autonomía y la capacidad de adaptación de manera inteligente de su comportamiento según el estado actual del medio ambiente percibido a través de sondas activas.

El NMM debe ser ligero, su extensibilidad y el diseño modular facilitan la integración de nuevas o imprevistas funcionalidades sin necesidad de interrumpir las operaciones de capacidad actual de interrumpir las operaciones de la capacidad actual de gestión.

Además del apoyo de las ejecuciones NMM, todos relacionados con la gestión de datos se mantienen dentro de una estructura de datos que son equivalentes al MIB de SNMP.

Sin embargo, es probable que una agregación de gestión de la información recogida de nodos vecinos a través de sondas. Llamamos a la colección de la guerrilla de información de gestión Información de Base (GMIB). Tenga en cuenta que el módulo de procesamiento de investigación también tiene su propia colección de información de gestión, por ejemplo, información del nodo próximo, que es considerado como parte de GMIB también. Además, GMIB se puede acceder tanto NMM y las sondas de entrada a través del agente SNMP como una sucursal MIB en SNMP.

2.2.5. GESTIÓN DE RED MOVILES AD – HOC BASADA EN POLITICAS [27]

El sistema de gestión de redes basado en políticas nos ayuda a automatizar el proceso de administración de la misma. Este sistema de gestión fue creado con el fin de abordar las necesidades que tienen las MANET como auto-formación, auto-configuración y auto-regenerativo, el mismo que provee de capacidad suficiente para expresar requerimientos de alto nivel de la red.

Una política es definida como un conjunto de directivas especificadas por el administrador para gestionar ciertos aspectos de los resultados deseados de las interacciones entre usuarios, entre aplicaciones, y entre usuarios y aplicaciones [23].

El administrador de la red tiene capacidades para especificar políticas de alto nivel como:

- Especificar a largo plazo los objetivos de configuración de ancho de la red
 - Todo el tráfico de la red debe obtener el nivel más alto de prioridad de QoS
 - Todas las comunicaciones deben ser encriptados

- Provee un bucle de retroalimentación de manera automatizada de información que es reportada de la monitorización de agentes la cual se puede utilizar automáticamente para la corrección de problemas basados en políticas.

Una vez establecidas las políticas descritas anteriormente se puede añadir funcionalidades habilitadas por el administrador de la red:

- Cambios dinámicos en el rol de un nodo para actuar como servidor (servidor DNS), basado en capacidades relevantes como poder de computar, capacidad de la batería, intensidad de señal, hardware, etc.
- La alteración entre la determinación del protocolo de recorrido proactivo y reactivo para optimizar la actuación de los nodos basado en su conocimiento.

Las políticas son implementadas por Policy Agents y se guardan en un repositorio de políticas implementado como un servidor de base de datos MySQL.

Las políticas pueden tener cuatro componentes [27]:

1. **Evento:** es un acontecimiento nombrado en el dominio local de política que provoca ejecución de una política. Cada vez que el acontecimiento ocurre, la condición de política es evaluada. Si es verdadero la acción es ejecutada.
2. **Condición:** es una expresión binaria que se evaluó por un Policy Agent para determinar si la acción correspondiente necesita ser ejecutada para un alcance dado de la política.
3. **Acción:** describe una tarea que para ser ejecutada. Una acción puede estar subdividida en dos componentes: nombre de la acción y parámetros de la acción.
4. **Alcance:** indica en cuales nodos se debería implementar. Actualmente soporta dos alcances: Global Policy Agent (GPA) y All Local Policy Agent (All_LPAs). Un alcance GPA señala que la política debe ser implementada sólo en la GPA. El alcance All_LPAs señala que la política debe ser implementada en todos los LPAs.

2.2.6. CONCLUSIONES

La gestión de redes fijas es diferente a la gestión de redes Ad Hoc, debido a diferentes parámetros como los protocolos de enrutamiento, a la seguridad que se le debe prestar en las MANET ya que son más propensas a recibir ataques por personas ajenas a la red.

Otro parámetro que complica de cierta manera a la gestión de redes MANET es que tiene una infraestructura descentralizada y la jerarquía de los nodos cambian continuamente.

Los protocolos de enrutamiento son diferentes para las redes fijas y para las redes MANET sin embargo se puede destacar que los protocolos de redes MANET se basan en el protocolo SNMP que es para redes fijas.

La gestión de este tipo de redes debe ser controlada de manera estricta ya sea por políticas de seguridad o por agentes que estén inmersos en el mundo de las redes ad hoc, esto con el fin garantizar un mejor funcionamiento y administración de recursos de la misma.

2.3. CLUSTERING

Clustering es una técnica que ayuda a la gestión de redes Ad Hoc la cual tiene como objetivo contribuir para llevar de mejor manera la administración de la red, agrupando los nodos de la red en clúster esto con el fin de obtener diversos beneficios. Por lo que presentamos a continuación es una breve síntesis de cada uno de los algoritmos o técnicas de clustering.

2.3.1. ALGORITMOS DE CLUSTERING

2.3.1.1. Lowest ID[12] (SEAH, 2004)

En este algoritmo a cada nodo de la red se le asigna un ID (identificador), estos nodos efectúan un broadcast de su ID a sus vecinos directos y compara su ID con el ID de sus vecinos.

Además en este algoritmo se considera al nodo Gateway el cuál es el que tiene la capacidad de comunicarse con dos o más clusterhead, el resto de nodos que no son ni nodos clusterhead ni nodo Gateway es un nodo ordinario.

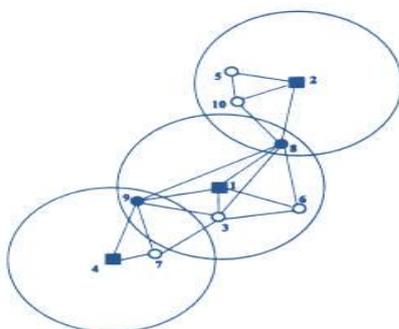


Figura 10. Formación del clúster mediante lowest id [8]

En la *Figura 10*, los cuadrados representa a los clusterhead, los nodos Gateway son los círculos negros y los círculos blancos a los nodos ordinarios. Una vez explicado cómo están representados cada uno de los tipos de nodos dentro de la *Figura 10*, procedemos a explicar la misma; como se puede observar los nodos clusterheads no están directamente conectados ya que el algoritmo establece un conjunto dominante independiente.

2.3.1.2. Highest -degree

Este algoritmo también se lo conoce como Highest-connectivity, se basa en la conectividad que tiene cada nodo con sus vecinos directos. Los nodos independientemente envían periódicamente un broadcast periódico del estado de conectividad a todos sus vecinos.

Un nodo determina si es nodo clusterhead o nodo ordinario comparando su grado de conectividad con el de sus vecinos. Un nodo es clusterhead si tiene mayor grado de conectividad entre sus vecinos.

2.3.1.3. K-hop CONID (k-hop connectivity ID)

Es la combinación de los algoritmos Lowest-ID y Highest-degree en uno solo.

Para la selección del clusterhead se considera dos criterios:

1. El mayor grado de conectividad
2. El valor de ID más pequeño

El propósito de este algoritmo es minimizar el número de clusters formados en la red y obtener conjuntos dominantes de menor tamaño.

La formación de los cluster se realiza mediante la selección de un clusterhead y de todos los nodos que se encuentren a una distancia k-saltos.

A cada nodo se le asigna un par conocido como prioridad del clusterhead y denotado como $did = (d, ID)$; d equivale a la conectividad de un nodo y ID es el identificador del nodo.

2.3.1.4. Algoritmo (α, t) cluster framework

La técnica de clustering (α, t) caracteriza la probabilidad de disponibilidad de un enlace o de un camino en función del modelo de movilidad conocido como random walk.

Otras Características:

- i. Este algoritmo es simple, distribuido y asíncrono
- ii. Los nodos se organizan en clusters de una forma dinámica
- iii. Utiliza un modelo de probabilidad para determinar la disponibilidad de ruta entre los nodos, con el fin de asegurar las decisiones de clustering.

- iv. Organiza los nodos dentro de los clusters de manera dinámica.
- v. La disponibilidad de la ruta es un proceso aleatorio que está en función de la movilidad de los nodos de una cierta ruta.

Las rutas se evalúan por dos parámetros del sistema:

α establece un salto más pequeño en la probabilidad en la probabilidad de una ruta dada por el cluster que permanecerá disponible por un tiempo t .

α controla la estabilidad del cluster mientras el rol de t es manejado por el tamaño del cluster por un nivel dado de estabilidad.

A cada nodo de la red se le asigna un número identificador del clúster CDI (cluster identifier number), además hace uso de un timer conocido como α timer el cual determina el máximo tiempo t para el cual un nodo puede garantizar la disponibilidad de un camino a cada cluster destino con una probabilidad mayor o igual a α .

Expone cinco eventos que pueden producir cambios topológicos en las MANET organizados mediante clusters [13]:

1. **Activación de un nodo:** un nodo de la red se activa, inicializa su CID a un valor predefinido para indicar que no se encuentra unido a un cluster, esto con el propósito de descubrir nodos adyacentes y unirse a su cluster.
2. **Activación de un enlace:** Cuando un nodo dentro de un clúster detecta la activación de un enlace, la topología del clúster se actualiza.
3. **Falla un enlace:** si hay la falla de un enlace el nodo debe determinar si ha perdido caminos con respecto a los destinos en el cluster. Si un destino es inaccesible el nodo lo elimina de su tabla de enrutamiento y no lo sigue teniendo en cuenta para el cálculo de los caminos.
4. **Expiración del α timer:** cuando se expira el α timer las acciones que toma un nodo son las mismas que cuando detecta la falla de un enlace o la desactivación de un nodo.
5. **Desactivación de un nodo:** puede ser por la desactivación voluntaria de un nodo, una falla imprevista, la desconexión de un cluster o el abandono voluntario de un nodo de su cluster.

2.3.1.5. Algoritmo Max-min d-cluster [12]

En este algoritmo se considera distancias d-hops o d-saltos. Los clúster se forman por nodos que están al menos a d-hops de distancia del clusterhead.

Objetivo: seleccionar múltiples líderes que contengan cientos de nodos y que con los clusterhead seleccionados se forme un conjunto dominante mínimo de d saltos.

d se define como el número máximo de saltos de distancia del clusterhead más cercano ($d \geq 1$), además permite controlar el número de clusterhead que serán seleccionados.

Los nodos participan en el algoritmo de elección del clusterhead basándose en sus respectivos ID. Cada nodo cuenta con dos arreglos: WINNER y SENDER.

WINNER corresponde al ID del nodo ganador de una serie en particular, al comienzo del algoritmo cada nodo tiene un valor de WINNER igual a su propio ID.

SENDER corresponde con el nodo que envía el ID del nodo ganador de una serie en particular, ayuda a determinar el camino de regreso más corto desde un nodo hacia el clusterhead. Además decide quién es el próximo nodo al que se le envía el contenido emitido por el Gateway, ayudando así a que el clusterhead pueda crear una base de datos sobre cada uno de los de su cluster.

Después de seleccionar el clusterhead, SENDER ayuda a determinar las rutas más cortas de vuelta al clusterhead.

2.3.1.6. MobDhop

Este algoritmo se basa en la métrica de movilidad y el diámetro de los clusters se adapta de acuerdo a la movilidad de los nodos. [1]

Características:

- a. Intercambio de información mínimo durante la formación de clusters y durante los cambios de clusterheads.
- b. Asume que cada nodo puede medir la potencia de las señales que recibe.
- c. Dos nodos se consideran cercanos, si las señales recibidas por ellos tienen un valor grande de potencia.

Este algoritmo requiere el cálculo de cinco términos:

1. Distancia estimada entre nodos

$$E[D_{AB}] = \frac{k}{\sqrt{P_R}}$$

k = constante

P_R = potencia de la señal

2. Movilidad relativa entre nodos

Se determina midiendo la potencia de la señal de paquetes sucesivos.

3. Variación de la distancia estimada a través del tiempo

Se refiere a su a un nodo se le gasta la batería, transmite paquetes a una potencia baja y por lo tanto se considera como un nodo distanciado de un vecino que se puede encontrar físicamente cerca de él.

4. Estabilidad local

Representa el grado de estabilidad de un nodo respecto a sus vecinos. Un nodo puede volverse clusterhead su es el nodo más estable del vecindario.

5. Distancia media estimada

Indica la distancia de cada vecino al clusterhead seleccionado en el cluster.

2.3.1.7. Distributed Mobility – Adaptive Clustering (DMAC)

El clusterhead se selecciona mediante el criterio basado en peso, en función de los parámetros basados en movilidad relativa de los nodos. Este algoritmo es apropiado para redes móviles altamente dinámicas.

Se asume durante la ejecución del algoritmo que cada nodo tiene:

⇒ Un peso, un número real ≥ 0

⇒ Un ID asociado al nodo

El peso representa los parámetros de movilidad del nodo.

Un nodo elige su rol ya sea como clusterhead o nodo ordinario basándose en el conocimiento de sus actuales vecinos localizados a 1 salto de distancia.

Un nodo es clusterhead si tiene el valor más alto entre sus vecinos a 1 salto de distancia, de lo contrario se une a un clusterhead vecino.

Durante la ejecución del algoritmo se ejecutan cinco procedimientos en cada nodo:

1. Rutina de inicialización para el proceso de formación de clusters
2. Procedimiento por falla de un enlace
3. Procedimiento para el establecimiento de un nuevo enlace
4. Procedimiento tras la recepción de un mensaje CH
5. Procedimiento tras la recepción de un mensaje JOIN

Los nodos empiezan la ejecución del algoritmo mediante la rutina de inicio. Los nodos con el mayor peso entre vecinos envían un mensaje CH. Los nodos restantes simplemente esperan a recibir un mensaje.

Al recibir un mensaje CH un nodo chequea si se afilia o no al clusterhead que le envió este mensaje.

Cuando un clusterhead recibe un mensaje JOIN verifica si el nodo que le manda este mensaje se va a unir a su cluster o a otro.

Si un nodo ordinario recibe un mensaje JOIN de su propio clusterhead, significa que ese clusterhead ha renunciado a su papel.

Este algoritmo permite que cada nodo reaccione ante la falla de un enlace o ante la presencia de un nuevo enlace con otro nodo.

Si falla un enlace entre un clusterhead y uno de sus miembros del cluster, la membresía del nodo en el cluster se remueve y este nodo debe determinar un nuevo papel.

Cuando un nodo detecta la presencia de un nuevo vecino, se realiza el procedimiento para establecer un nuevo enlace entre estos dos nodos. En este caso, el nodo que detecta el nuevo enlace determina si el nuevo vecino es clusterhead y si tiene un peso mayor al de su propio clusterhead. Si esto sucede, el nodo se une al nuevo vecino. Si el nodo que

detecta el nuevo enlace es un clusterhead, este deberá renunciar a su papel si el nuevo clusterhead tiene un peso mayor al suyo.

2.3.1.8. Algoritmo Weighted Clustering Algorithm (WCA)

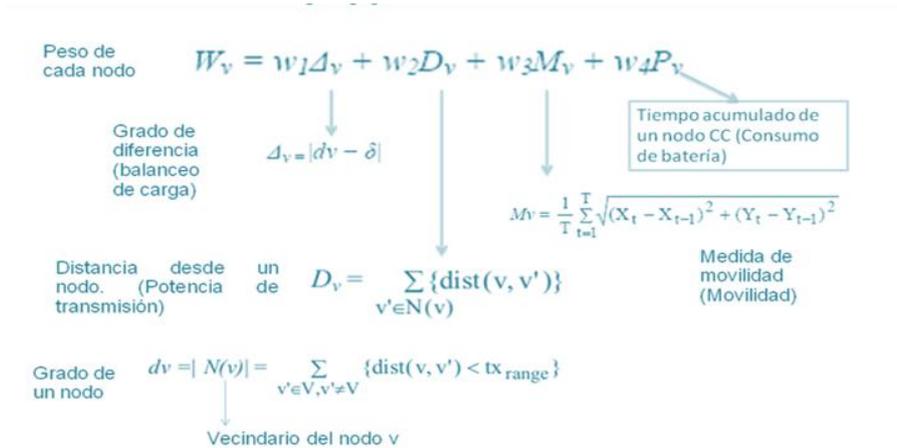
Características:

- Elige el nodo clúster cabeza en base al valor del peso de cada nodo.
- Selecciona los clúster cabeza (CC) tomando en cuenta aspectos relacionados al funcionamiento eficiente de los componentes del sistema.
- Elige un nodo clúster cabeza de acuerdo a:
 - Capacidad de nodos que puede manejar (balanceo de carga)
 - Movilidad
 - Potencia de transmisión
 - Capacidad de batería.
- Evita sobrecarga de actualizaciones(no es periódico)
- El procedimiento de la elección de CC se invoca únicamente en base:
 - Cambio topológico debido a la movilidad de los nodos
 - Si actual CC es incapaz de cubrir todos los nodos.
- Para evitar sobrecarga en los CC se especifica número de nodos que cada CC puede soportar, representado por δ

Para la elección del cluster cabeza se considera las siguientes situaciones:

- Se elige de acuerdo al valor del peso asociado de cada nodo v .
- El nodo con mínimo peso es elegido como CC.
- Los factores de peso se seleccionan de forma que $w_1 + w_2 + w_3 + w_4 = 1$.
- Dos CC no pueden ser vecinos inmediatos

WCA elige los clusterhead de acuerdo al valor del peso asociado a cada nodo. El peso asociado a un nodo v está definido por [2]:



WCA se termina una vez todos los nodos decidan su papel ya sea clusterheads o miembros de un clusterhead.

Cuando un nodo queda elegido como clusterhead no se le permite a él ni a sus vecinos participar más en el algoritmo de selección.

2.3.2. COMPARACIÓN DE LAS TÉCNICAS DE CLUSTERING

Con la comparación de las técnicas de clustering se busca encontrar el más óptimo de acuerdo a las condiciones de la red.

En la tabla se presenta los parámetros que estiman cada algoritmo para la formación del

Parámetros	PESO		MOVILIDAD NODOS		CONECTIVIDAD		ID		DISTANCIA		COMUNICACIÓN		DISPONIBILIDAD RUTA		BALANCEO DE CARGA		POTENCIA TRANSMISIÓN		CONSUMO BATERIA		
	ALTO	BAJO	ALTO	BAJO	ALTO	BAJO	ALTO	BAJO	ALTO	BAJO	ALTO	BAJO	SI	NO	SI	NO	SI	NO	SI	NO	
Highest degree heuristic			x	x				x			x										
Lowest ID			x					x													
Highest degree			x	x					x	x											
k-Hop CONID					x			x	x	x											
(α, t)			x						x	x		x									
Max-Min			x					x	x												
MobDhop			x	x	x				x	x	x				x		x		x		
DMAC	x		x					x	x												
WCA		x	x					x							x		x		x		

cluster, la elección del clusterhead, del nodo ordinario y del nodo Gateway de acuerdo a las características expuestas anteriormente en la síntesis de cada uno.

De acuerdo a lo expuesto en las secciones anteriores y con la comparación de las técnicas de clustering es más fácil la elección de la óptima para la formación del clúster de acuerdo a nuestras necesidades y exigencias.

2.4. REDES NEURONALES Y TEORIA DE JUEGOS

Se presenta un nuevo paradigma en cuanto a Gestión de Redes MANET basándose en Redes Neuronales y Teoría de juegos.

En la actualidad no existen muchos estudios alrededor de estas nuevas alternativas para la gestión de MANET, pero en esta sección se estudia el fundamento teórico para poder analizar dentro de que fase se las puede implementar para el mejor rendimiento de este tipo de redes, el estudio empieza por agentes inteligentes debido a que a partir de esto se puede definir cómo se va a implementar, luego se sigue con el estudio teórico de redes neuronales y teoría de juegos.

2.4.1. AGENTES INTELIGENTES EN REDES MANET [19]

Un agente no es más que un software capaz de realizar una tarea definida por un usuario determinado, con un nivel de inteligencia que le permite desarrollarla de alguna manera autónoma y que puede interactuar con su entorno.

Entre sus principales características están: Autonomía, Inteligencia e Interacción.

2.4.1.1. Tipos de Agentes Inteligentes

Basándose en que redes neuronales es una herramienta de inteligencia artificial, y dentro de inteligencia artificial están los agentes inteligentes, se procede a analizar los tipos de agentes inteligentes *Figura 11*, para ver cuál es el que más se acopla a nuestras necesidades.

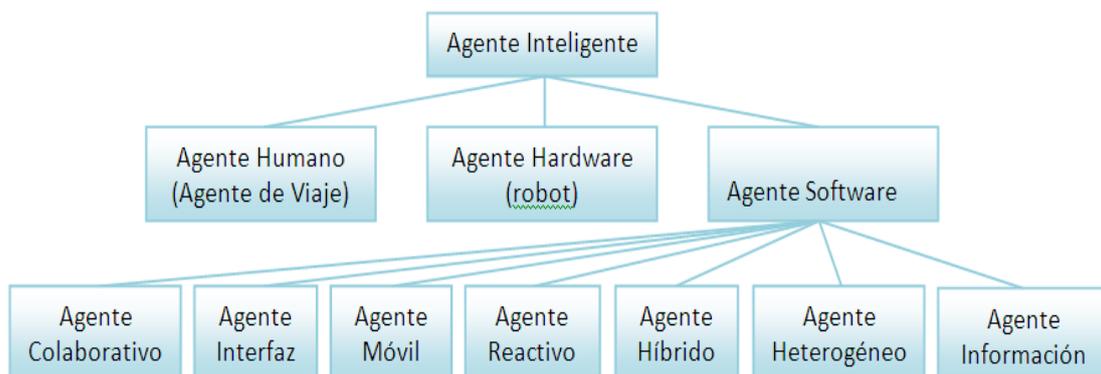


Figura 11. Tipos de Agentes Inteligentes [19]

⇒ **Agente de Interfaz**

Se basa en el análisis de sus experiencias, hábitos y comportamiento, ajustándose de acuerdo a comportamientos anteriores.

Su aprendizaje se fundamenta en:

- Observar e imitar las acciones del usuario
- Retroalimentación positiva o negativa del usuario sobre una acción tomada de manera autónoma
- Recepción clara de instrucciones del usuario I
- Investigación de comportamiento similar a otros agentes.

⇒ **Agente móvil [15]**

Tiene la capacidad de salir de los límites de las redes y acceder a computadores en redes remotas o incluso Internet, además permiten la ejecución de tareas en computadores remotos y la transmisión de los resultados al usuario.

⇒ **Agente de Información**

El objetivo es la recolección, filtro, administración y clasificación de manera autónoma de gran cantidad de información proveniente de fuentes distribuidas.

⇒ **Agentes Reactivos**

Actúan de manera reactiva a un evento producido dentro del entorno del sistema. No agregan mucha autonomía. Por lo regular son conjuntos de agentes los que realizan labores autónomas.

⇒ **Agentes Heterogéneos**

Es la integración de un sistema de información con base en dos o más tipos distintos de agentes.

⇒ **Agentes de Gestión**

Los agentes de gestión son instantáneos cuando las políticas son activadas y incitadas. Estos agentes de la gerencia realizan diferentes tipos de acciones.

Se definen los siguientes tipos de agentes de gestión basados en políticas:

⇒ **Agente de Configuración**

Realizan la configuración en un elemento local administrativo o nodo.

⇒ **Agente de Monitoreo**

Coleccionan información en los intervalos del nodo local configurable y almacenan en el CMDB (Configuración y Base de Datos Monitora).

⇒ **Agente de Reporte**

Reportan dos tipos de información: el monitoreo y la información de configuración. La información es reportada en los intervalos configurables vía YAP.

⇒ **Agente de Agregación**

Realizan agregación de datos localmente antes de que la información sea reportada.

⇒ **Agente de Filtro**

Reporta la información vía YAP sólo cuando ciertas condiciones son encontradas.

⇒ **Agente Multiusos**

Los agentes multiusos son diseñados para realizar funciones directivas especiales.

2.4.1.2. Características de los Agentes Inteligentes

Movilidad

Es la capacidad de un agente para moverse de un nodo a otro, esto también se lo conoce como migración.

La movilidad puede ser débil (ejecución desde el inicio) ó fuerte (manejo del estado en la ejecución):

- Ejecución remota (RPC)
- Código en demanda (Applets y Controles ActiveX)
- Movilidad débil (Aglets)
- Movilidad fuerte (Telescript)

Un agente móvil tiene estado, localización y función predefinida. Después de la ejecución autónoma, el agente define hacia donde moverse y función nueva ejecutar.

Comunicación

Un agente puede establecer comunicación a través de protocolos (TCP, UDP, HTTP, IIOP, RMI), procedimientos (RPC, CORBA, DCOM, Multicast, Broadcast, Blackboards, Message based-Speech based), lenguajes de comunicación entre agentes (ACL), HTML y XML.

Cooperación

Incluye todas las actividades a realizar para solucionar un problema o para ejecutar una tarea de manera distribuida.

Seguridad y Autenticación

Minimiza el riesgo ante problemas como: revelación de información no autorizada, negación del servicio, acceso no permitido a la información.

Filtrado y obtención de Información

Es la búsqueda textual de información del contexto y de la información de varios documentos que pueden ser accedidos por diferentes usuarios.

Aprendizaje y Planeación

El aprendizaje hace parte de la inteligencia del agente, se basa en la utilización del conocimiento de experiencias previas para la toma de decisiones presentes.

Modelamiento del Usuario

Soporte a gestión de perfiles de los usuarios, no solo con información estática, sino también de comportamiento y de patrones de interacción.

2.4.2. REDES NEURONALES [21]

Una Red Neuronal engloba diferentes estructuras de procesamiento paralelo distribuido, estas estructuras son de utilidad en reconocimiento de patrones, modelado y filtrado no lineal de señales, entre muchas otras aplicaciones.

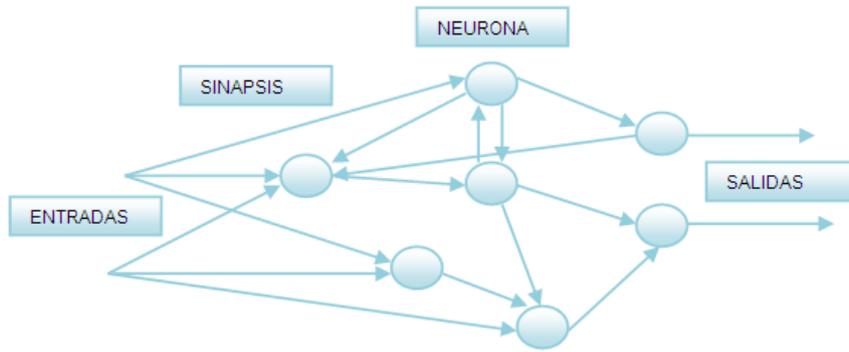


Figura 12. Red Neuronal Genérica [21]

En la *Figura 12*, se observa un esquema general de una red neuronal. La red es un operador no lineal formado por un conjunto de elementos simples de procesamiento o neuronas, que se conectan entre sí y con el medio externo a través de conexiones o sinapsis, que poseen un peso asociado.

En la *Figura 13*, podemos ver como es un modelo habitualmente utilizado para una neurona artificial. Esencialmente, la operación de la neurona involucra el cálculo de una *función de entrada h* , a partir de las señales que ingresan a la misma, y la posterior aplicación de una *función de activación g* , en general no lineal.

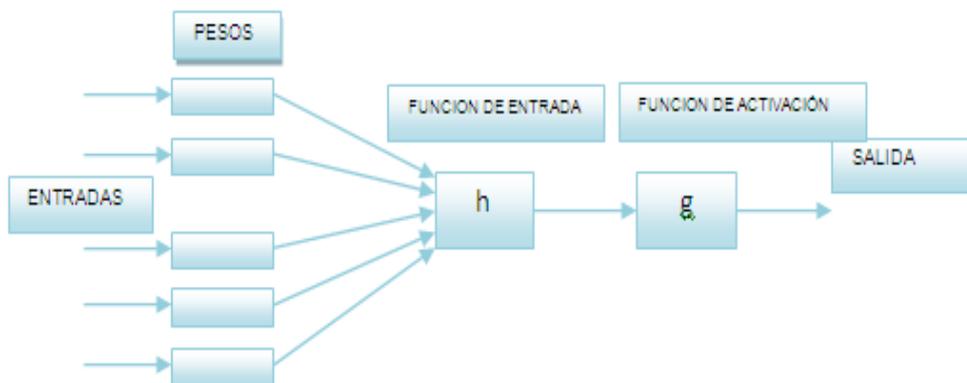


Figura 13. Neurona Artificial [21]

En algunos casos, existe una entrada de umbral (*offset*), que puede ser siempre asociada a una entrada adicional de valor 1 y peso igual al umbral. La salida de cada neurona dependerá de sus señales de entrada del peso asociado a cada entrada y de las características de las funciones de entrada y activación.

2.4.2.1. Arquitectura de Redes Neuronales

Es la topología, estructura o patrón de conexionado de una red neuronal, donde los nodos se conectan por medio de sinapsis, estando el comportamiento de la red determinado por la estructura de conexiones sinápticas. Las conexiones sinápticas son direccionales, es decir, la información solamente puede propagarse en un único sentido.

Las neuronas se agrupan en unidades estructurales denominadas capas.

Una red neuronal se constituye un conjunto de una o más capas. Existen tres tipos de capas: de entrada, de salida y ocultas, como podemos observar en la *Figura 14*.

Capa de Entrada también llamada capa sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno.

Capa de Salida se compone de neuronas que proporcionan la respuesta de una red neuronal.

Capa Oculta no tiene una conexión directa con el entorno, es decir no se conecta directamente ni a los órganos sensores ni a efectores.

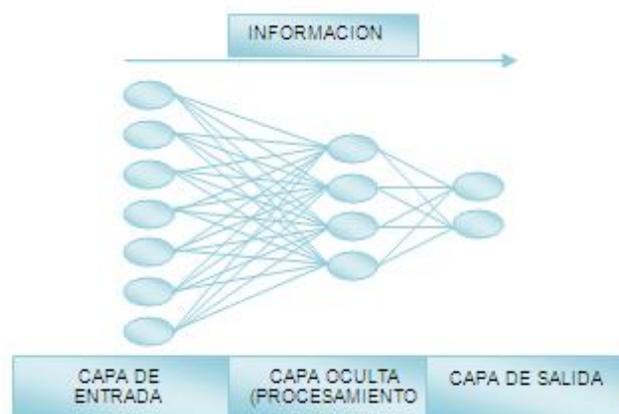


Figura 14. Arquitectura de una neurona [21]

2.4.2.2. Perceptrón Simple

Es un modelo unidireccional compuesto por dos capas de neuronas, una de entrada y otra de salida.

La operación en un perceptrón simple que consta de n neuronas de entrada y m neuronas de salida:

$$y_i = \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right)$$

Con $i=1, \dots, m$.

Como podemos observar en la *Figura 15*, las neuronas de entrada son discretas y la función de activación de las neuronas de capa de salida es de tipo escalón.

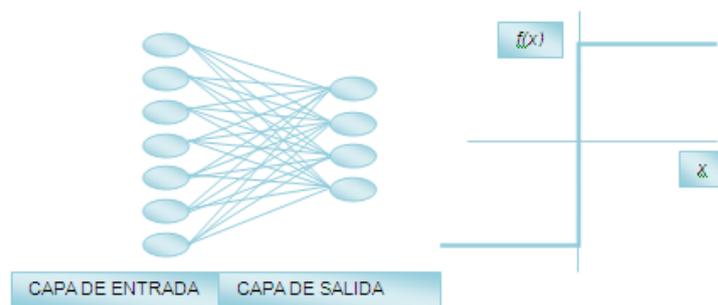


Figura 15. Arquitectura y función de transferencia del perceptrón simple [31]

El perceptrón simple puede utilizarse como clasificador, radicando su importancia histórica en su carácter de dispositivo entrenable, ya que el algoritmo de aprendizaje permite determinar automáticamente los pesos sinápticos que clasifican un conjunto de patrones a partir de un conjunto de ejemplos etiquetados.

El perceptrón simple discrimina patrones que pueden ser separados por un hiperplano, una recta en caso de dos neuronas de entrada siendo esta la región de decisión.

i. **Perceptrón Multicapa**

A diferencia del perceptrón simple, el multicapa incluye capas ocultas.

En la *Figura 16*, se presenta las regiones de decisión que se obtienen para distintas arquitecturas de redes neuronales considerando dos neuronas en la capa inicial.

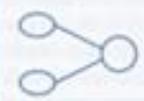
Arquitectura	Región de decisión	Ejemplo 1: XOR	Ejemplo 2: clasificación	Regiones más generales
Sin capa oculta 	Hiperplano (dos regiones)			
Una capa oculta 	Regiones polinomiales convexas			
Dos capas ocultas 	Regiones arbitrarias			

Figura 16. Regiones de decisión obtenidas para el perceptrón simple (arriba), perceptrón multicapa (medio) y perceptrón multicapa con dos capas ocultas (abajo) [21]

El perceptrón multicapa o MLP (Multi-Layer Perceptron) con dos capas de neuronas ocultas es capaz de discriminar regiones de forma arbitraria, además se suele entrenar por medio de un algoritmo de retropropagación de errores o BP (Back Propagation), a esta también se la conoce como red de retropropagación.

- **Estructura MLP con una única capa oculta**

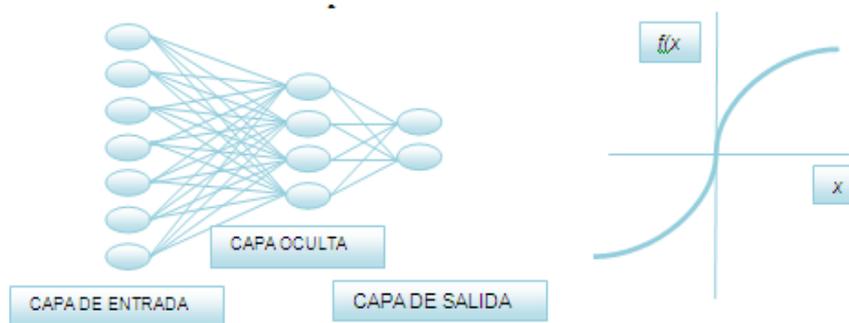


Figura 17. Arquitectura y función de activación para el perceptrón multicapa [31]

Las operaciones efectuadas por un MLP con una única capa oculta y con funciones de activación para la capa oculta y capa final de tipo sigmoide y lineal respectivamente.

Un MLP con una única capa oculta puede aproximar cualquier función continua en un intervalo hasta el nivel deseado, aunque el resultado no informa sobre el número de nodos ocultos necesarios para llevar a cabo la aproximación.

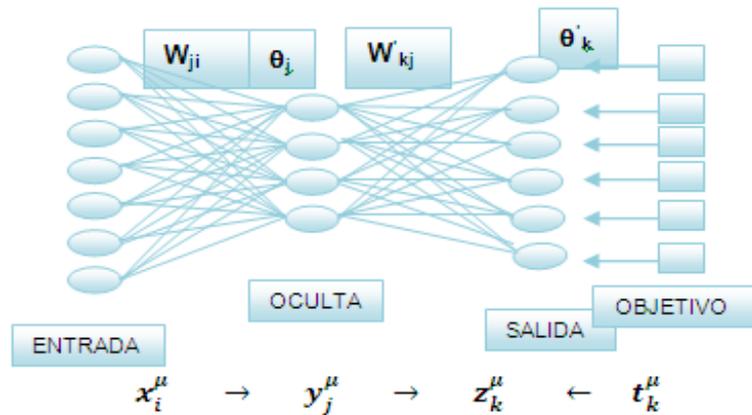


Figura 18. Arquitectura del perceptrón multicapa [31]

En la *Figura 18*, se puede observar una arquitectura MLP de tres capas, con una capa oculta, con las entradas, salidas, pesos y umbrales de las neuronas.

2.4.3. TEORIA DE JUEGOS [26]

Haciendo un poco de historia la teoría de juegos fue creada por Von Neumann y Morgensten, en 1944 lo cuál fue publicado en su libro *The Theory of Games Behavior*.

En sus inicios se la empleo en campos como la economía, biología, sociología, psicología y filosofía, esto con el fin de entender el comportamiento. En la actualidad se la está implementando en el campo de las redes esto con el fin de maximizar su utilidad eligiendo determinados cursos de acción basándose en hechos históricos, con el único objetivo de analizar el comportamiento estratégico de los jugadores en nuestro caso de los nodos.

La teoría de juegos se la emplea para llevar procesos que permitan tomar decisiones, a través de modelos matemáticos para estudiar interacciones en estructuras de incentivos denominados juegos.

Con una problemática diferente se plantean dos clases de juegos las cuales requieren una forma de análisis diferente [14]:

2.4.3.1. Juegos Cooperativos [26]

También se los conoce como juegos con transferencia de utilidad, este tipo de juegos es cuando los jugadores (nodos) se pueden comunicar entre ellos y negociar los resultados, en los cuales la problemática se concentra en el análisis de las posibles coaliciones y su estabilidad.

2.4.3.2. Juegos no Cooperativos [26]

Son juegos sin transferencia de utilidad en los que los jugadores no pueden llegar a acuerdos previos, entre este tipo de juegos esta el “dilema del prisionero”, “modelo halcón paloma”, “la guerra de los sexos”.

Dilema del Prisionero

Se basa en acuerdos mutuos entre dos partes de tal manera que si deciden colaborar ganen los dos o caso contrario pierdan por igual. El dilema está en que si un nodo decide colaborar y el otro nodo no, este obtendría una pérdida en comparación con el segundo nodo, o en el caso de que decidan ambas colaborar por igual ambos van a terminar con una pérdida.

El dilema del prisionero muestra las dificultades para establecer la colaboración en cualquier situación en la que hacer trampa beneficia a las partes.

El equilibrio de Nash en el dilema del prisionero esta en el punto en que ambos jugadores traicionan.

Modelo Halcón Paloma

La definición del nombre de este modelo parte de “halcón” a los políticos partidarios de estrategias más agresivas mientras que identificamos como “paloma” a los más pacifistas. De tal manera que el modelo Halcón-Paloma sirve para analizar situaciones de conflicto entre estrategias agresivas y conciliadoras.

El equilibrio de Nash esta cuando las estrategias elegidas por cada jugador son diferentes; es decir cuando uno elige halcón y el otro paloma.

Se pueden clasificar en varias categorías como las anteriores y como las que se analizara a continuación.

2.4.3.3. Juegos en forma extensiva (árbol)

Para explicar este tipo de juego se basará en la *Figura 19*, en la cual se presentan dos jugadores el nodo 1 y 2 los cuales participarán en el siguiente juego. Primero el jugador 1 decide ir a la izquierda (I) o a la derecha (D). Entonces el jugador 2 decide ir a la derecha o a la izquierda. Los pagos que corresponden al primer (segundo) jugador son la primera (segunda) componente del vector que tiene asignada cada situación.

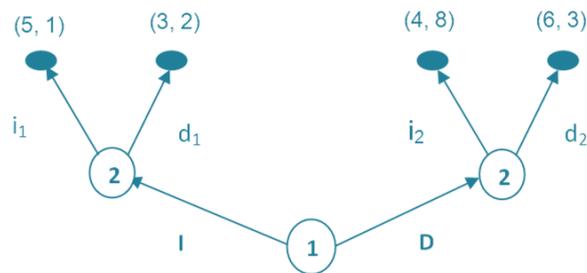


Figura 19. Juego en forma extensiva [26]

Ahora bien se analizará como deben jugar 1 y 2. El jugador 2, teniendo en cuenta los pagos que recibiría al terminar el juego, debe elegir la siguiente estrategia: si el jugador 1 elige I, ir a la derecha elegido d_1 ; y si 1 elige D; elegir i_2 : Esta estrategia denotará d_1i_2 : El jugador 1 conoce el árbol y los pagos, luego puede anticipar la conducta del jugador 2 y debe elegir D: El par de estrategias (D; d_1i_2) da lugar a un escenario en el que el jugador 1 recibe 4 y el jugador 2 recibe 8.

2.4.3.4. Juegos en forma de estratégica (normal)

A partir de la *Figura 19*, que se toma como ejemplo se analiza que el jugador 1 tiene dos estrategias I y D; mientras que el jugador 2 tiene cuatro estrategias por: i_1i_2 , i_1d_2 , d_1i_2 , d_1d_2 . Las cuales se pueden representar los pagos en la siguiente matriz, cuyas entradas son los vectores de pagos,

$$\begin{matrix} I \\ D \end{matrix} \begin{pmatrix} i_1i_2 & i_1d_2 & d_1i_2 & d_1d_2 \\ (5, 1) & (5, 1)(3, 2) & (3, 2) \\ (4, 8) & (6, 3)(4, 8) & (6, 3) \end{pmatrix}$$

Donde las matrices de pago para los jugadores 1 y 2 son, respectivamente,

$$P_1 = \begin{pmatrix} 5 & 5 & 3 & 3 \\ 4 & 6 & 4 & 6 \end{pmatrix} \quad P_2 = \begin{pmatrix} 1 & 1 & 2 & 2 \\ 8 & 3 & 4 & 3 \end{pmatrix}$$

El par de estrategias (D; d_{1i2}) es un equilibrio de Nash porque ninguna desviación unilateral de los jugadores les permite mejorar sus pagos, por (4, 8).

⇒ **Equilibrio de NASH [26]**

El matemático John Nash estableció este equilibrio definiéndolo como una combinación de estrategias (una por jugador) que está en equilibrio de Nash si ningún jugador puede aumentar sus ganancias por un cambio unilateral de estrategia.

Un par de estrategias es un equilibrio de Nash si la elección de A es óptima dada la de B y la de B es óptima, dada la elección de A. El equilibrio de Nash se diferencia del equilibrio de las estrategias dominantes en que, en el equilibrio de las estrategias dominantes, se exige que la estrategia de A sea óptima en el caso de todas las elecciones óptimas de B, y viceversa. El equilibrio de Nash es menos restrictivo que el equilibrio de estrategias óptimas.

⇒ **Estrategia MAXIMIN**

Es aquella en la que se maximiza la ganancia mínima que puede obtenerse.

2.4.4. TRABAJOS RELACIONADOS

2.4.4.1. Sistemas de Gestión de Redes Diseño Basado en Agentes Inteligentes [11]

En la *Figura 20* se puede observar como el sistema administrativo es organizado en un número de dominios de política.

Un dominio de política es una colección de entidades sujetas al mismo grupo de políticas y bajo la dirección de un agente de política, conocido como el Agente de Política de Dominio. Un dominio de política puede contener uno o más dominios de política, organizados en una estructura jerárquica.

Un Agente de Política es una entidad responsable para ingeniarse e implementar las políticas de un dominio de política.

El Agente de Política de un dominio atómico de políticas es el que no contiene cualquier otro dominio de política, también es llamado Agente Local de Política (LPA).

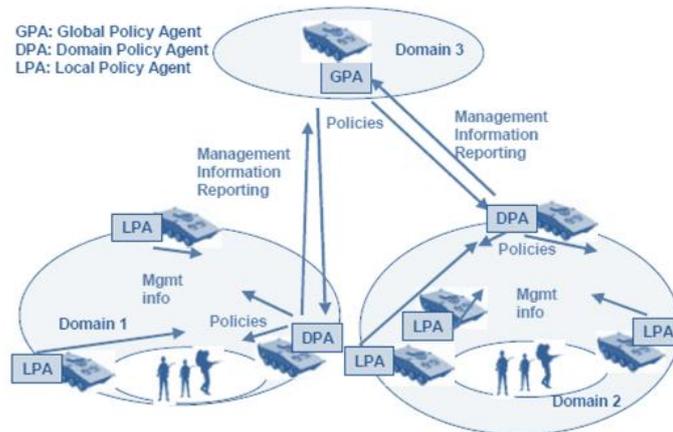


Figura 20. Arquitectura de alto nivel [11]

El Agente de Política del dominio de políticas del más alto nivel es llamado Agente Global de Política (GPA).

Los Agentes intermedios de Política son llamados Agentes de Política de Dominio (DPA).

El Servicio de Distribución de Datos (DDS) provee el mecanismo para distribuir políticas y para reportar información. Las políticas creadas en la GPA son distribuidas para los DPAs que a su vez las distribuye para los LPAs.

El alcance de una política determina si la política es aplicable para un dominio particular de política.

Los LPAs coleccionan información local de la gerencia y le informan la información de agentes de política del nivel más alto usando una versión realizada del protocolo YAP.

2.4.4.2. Componentes del Sistema Administrativo de Redes Ad-Hoc basado en Políticas

- **Plataforma de Agente de Política**

Una GPA, DPA, o LPA tiene la misma estructura básica y consta del mismo código, la misma estructura básica y consta de la misma base de código. Tiene funcionalidad similar pero alcance diferente. Los agentes diferentes pueden ser instalados dentro de una GPA, DPA Y LPA para permitir funcionalidad diferente.

▪ **Implementador de Política**

Es la entidad responsable para implementar políticas. Las políticas constan de acontecimiento, condición, acción y alcance.

Además monitorea acontecimientos y evalúa condiciones para decidir cuál agente debería estar encaminado para realizar sus acciones de la gerencia.

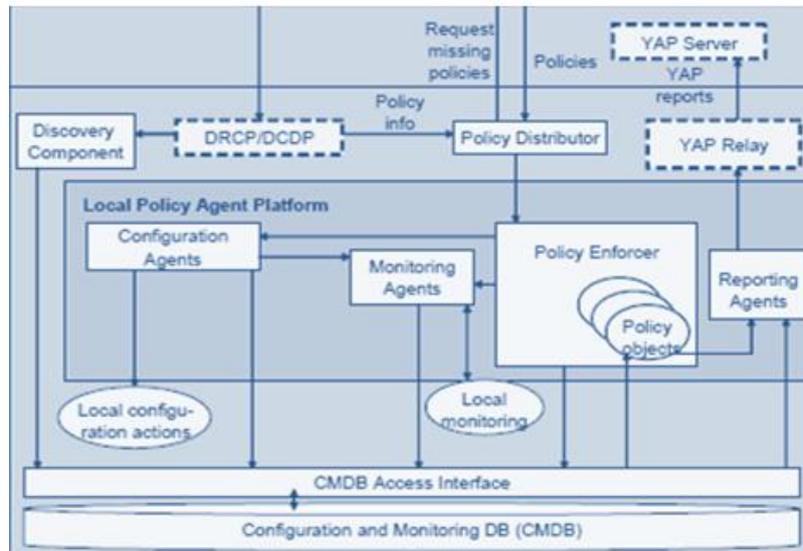


Figura 21. Componentes del Sistema [8]

▪ **Distribuidor de Políticas**

Su función es recibir actualizaciones de política de un nodo remoto y enviarle estas actualizaciones al Agente Local de Política en su nodo. Las políticas son repartidas para todos los nodos que necesitan implementar las políticas de conformidad con el alcance definido de política.

Las políticas típicamente serán creadas por una Interfaz gráfica del Usuario, o por otros sistemas de dirección descritos.

▪ **Servidor y Relevador YAP**

YAP (Yelp Announcement Protocol) es un protocolo diseñado para ambiente MANET y usado para reportar información de la gerencia a un gerente del nivel más alto. YAP es diseñado para pasar información eficaz y robusta de la administración de red acerca

de un nodo llamado YAP server que colecciona y almacena esta información en una base de datos.

- **DRCP/DCDP**

DCDP (Protocolo de Distribución de Configuración Dinámico), es un protocolo dimensionable, ligeramente diseñado para distribuir información de configuración. DCDP confía en el Protocolo de Configuración Dinámico y Rápido (DRCP) para configurar las interfaces. DRCP automáticamente puede detectar la necesidad para reconfigurar las publicaciones periódicas directas.

- **CMDB**

La configuración y Base de Datos Monitora (CMDB) se usan para almacenar configuración y datos monitoreados en un nodo. Esta información se guarda localmente en cada nodo como por ejemplo en un servidor de la base de datos MySQL.

- **Componente de Descubrimiento**

Es una entidad responsable para inicializar la configuración Local del Agente de Política (LPA) y la base de datos monitora (CMDB), para informar al LPA CMDB cuando los valores se varían por componentes externos del software, y para acceder al CMDB en nombre de estos componentes.

2.4.4.3. MINERVA: An Event Based Model For Extensible Network Management [11]

Es un sistema de gestión de red basado en eventos capaz de soportar eventos poco comunes.

El sistema consta de un **reportero** el cual se basa en toda la información recolectada por el administrador de la red la cuál es resultado de un evento.

Cada evento reportado debe contener información suficiente para permitir la correcta interpretación y transformación de la notificación de eventos. Cada notificación de eventos tiene la capacidad de generar una entrada de registro en los estados miembros en la base de datos, actualización de una relación en la base de datos y la generación de una alerta inmediata al personal de gestión de red.

El informe que es reportado de acuerdo a un evento se encuentra asociado a una clase de evento y un índice de eventos. Una clase de evento indica el área específica de operaciones a las que corresponde el evento y el índice de sucesos especifica un evento en particular dentro de una categoría.

Si se recibe un reporte que indica que una interfaz de un nodo en particular ha dejado de funcionar, el registro correspondiente en la base de datos debe ser actualizado para reflejar el nuevo estado del nodo.

En este modelo de gestión de red se hace referencia únicamente a cambios de estado que tiene un nodo, de tal manera que el reportero puede determinar el estado actual de un objeto y también el estado propuesto como resultado de procesar el evento.

El informe que es emitido por el reportero ante un evento de un nodo es descartado si el evento no causa un cambio en el estado actual del objeto.

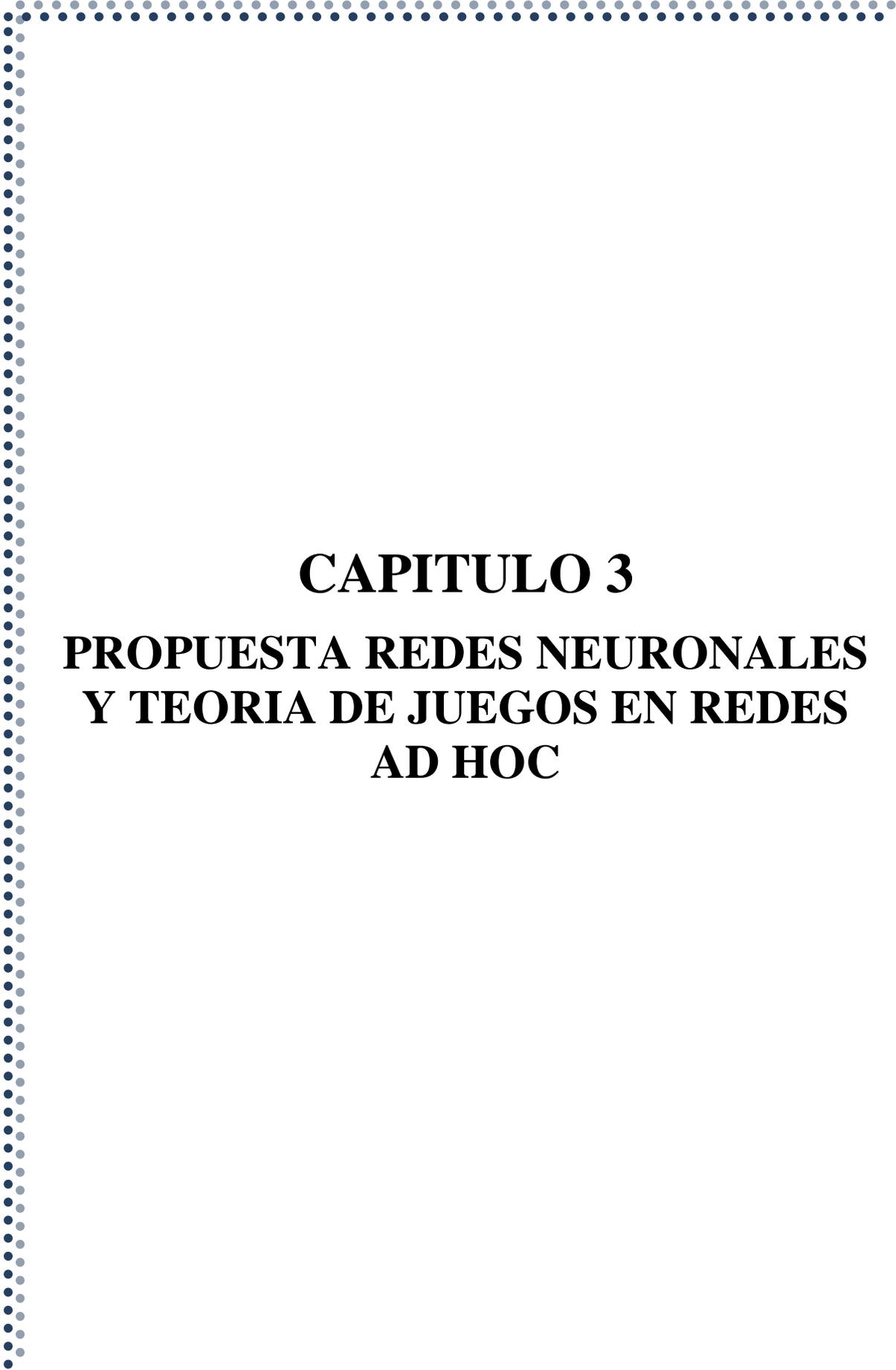
Cada informe está contenido en un solo paquete. El paquete de informe incluye nombre del nodo al que se refiere el informe, una definición del mensaje, la clase del evento y el índice de eventos.

2.4.5. CONCLUSIONES

La implementación de herramientas como redes neuronales y teoría de juegos puede resultar muy factible para la implementación de un nuevo modelo orientado a la gestión.

Luego de haber investigado el fundamento teórico de cómo están conformados los agentes inteligentes, las Redes Neuronales y teoría de juegos dentro de las MANET, se puede empezar un nuevo estudio orientado a dar solución a los diferentes problemas que se presentan en la Gestión de Redes Móviles.

La gestión de MANET basada en políticas combinado con aprendizaje inteligente implementando redes neuronales y teoría de juegos garantiza una optimización de administración de la red.

A decorative border composed of small, dark blue dots arranged in a rectangular shape, starting from the top-left corner and extending towards the bottom-left corner.

CAPITULO 3

**PROPUESTA REDES NEURONALES
Y TEORIA DE JUEGOS EN REDES
AD HOC**

3. DEFINICIÓN DEL MODELO DE GESTIÓN DE REDES MÓVILES AD HOC

3.1. INTRODUCCIÓN

En la actualidad existen varios modelos de gestión de redes inalámbricas Ad Hoc, pero ninguno de estos ha alcanzado a cubrir de manera eficiente las diferentes necesidades que presentan, es por esto que se propone un nuevo modelo, con el fin de cubrir gran parte de estas.

Para definir el modelo se tiene en consideración algunos aspectos como por ejemplo que los nodos móviles tienen funcionalidades y capacidades heterogéneas, además que estos nodos pueden ser laptops, PCs, celulares, etc.

En el modelo que se propone se integran técnicas basadas en una herramienta de inteligencia artificial como lo es redes neuronales feedforward y teoría de Juegos; sobre la red dividida a través de técnicas de clustering.

Inicialmente se parte dividiendo a la red en grupos llamados cluster como se observa en la *Figura 23*, esto a través del algoritmo WCA que es una técnica de clustering. La finalidad de dividir a la red en cluster es controlar la movilidad y el cambio de topología esto debido a la entrada y salida de equipos en la red. Dentro del algoritmo WCA se tiene en consideración algunas características en los nodos para que puedan ser jefes del cluster debido a que no todos los nodos que forman parte de una red Ad Hoc tienen las capacidades optimas para realizar tareas que solo un jefe de cluster lo puede hacer (JC). Un jefe de cluster es el nodo que tiene información de los otros nodos del cluster, independientemente de quien sea el JC todos los nodos del clúster cumplirán un rol en la gestión de acuerdo a sus capacidades de almacenamiento, procesamiento y energía.

Luego de haber dividido a la red en cluster se establece las interfaces en las que se utilizará redes neuronales y teoría de juegos. Las redes neuronales dentro de las redes Ad Hoc según algunas investigaciones [7], [11], [19], ayuda a mejorar la transmisión de paquetes en tiempo real de extremo a extremo partiendo de operaciones básicas que nos ayudarán a adquirir conocimiento en los nodos, este conocimiento se lo adquiere de un núcleo creado a partir del modelo SWAN [5], donde se tiene en consideración la priorización de nodos dentro de una cola. El aprendizaje automático que adquieren los

nodos ayudará a tomar la mejor decisión para el siguiente salto sin necesidad de buscar la ruta, mejorando así la calidad de servicio en las transmisiones, y la utilización de teoría de juegos en el control de nodos egoístas existentes en la red, esto a través del dilema del prisionero, todas estas técnicas innovadoras se aplican en la interfaz no y dos de la *Figura 24*.

Teoría de juegos se emplea dentro de la fase de control de admisión y en la formación del clúster, esto con la finalidad de controlar el comportamiento de los nodos, si es egoísta o no para evitar el desgaste de la batería de los nodos y obtener mejores resultados manteniendo la red por más tiempo evitando el desgaste de recursos, esto a través de la utilización de un modelo de confianza que permita decidir a los nodos con que nodo cooperar y con cuál no, con lo que se mejora la operatividad de la red consumiendo menos energía.

3.2. JUSTIFICACIÓN DEL MODELO

La iniciativa de proponer un modelo de gestión de redes Ad Hoc es con la finalidad de ayudar a mejorar: simplificación de funciones como enrutamiento, asignación de ancho de banda, canal de acceso, reserva de recursos, asignación de recursos, control de poder, ahorro de energía y reducción de tiempo de transmisión.

Además el modelo propone garantizar la calidad de servicio a través de una mejor entrega de paquetes extremo a extremo complementado con técnicas de redes neuronales multicapa a través del aprendizaje automático adquirido a partir de las experiencias, permitiendo tomar la mejor decisión a un nodo para el siguiente salto; y desde el punto de vista de teoría de juegos determinar el egoísmo o no de los nodos egoístas y así tomar la decisión entre colaborar con el resto de nodos para la transmisión de sus paquetes o no, esto a través del manejo del dilema del prisionero, esto se lo plantea analíticamente.

3.3. COMPONENTES Y DEFINICIÓN DEL MARCO DE TRABAJO

La utilización de nuevos mecanismos en redes Ad Hoc es con el fin de optimizar la gestión, utilizando componentes como: técnicas de clustering, redes neuronales multicapa feedforward y teoría de juegos.

La gestión de las redes móviles Ad Hoc contribuye a evitar la no disponibilidad de los recursos y mejorar la comunicación entre los dispositivos, evitando que en caso de que haya falla en un dispositivo o que este salga de la red ya sea por falta de batería, por movilidad o por potencia de transmisión, otro pueda tomar su lugar y establecer comunicación normalmente.

En las siguientes secciones se presenta la propuesta en si explicando cada una de sus partes y al final unificando todo lo que se propone.

3.3.1. Modelo Propuesto

Teóricamente el modelo que se propone parte de [25], el cual emplea algoritmos de clustering a fin de segmentar la red, facilitar la gestión y permitir la asignación de roles específicos a cada dispositivo o nodo; a esto se adhiere redes neuronales feedforward con el fin de mejorar la calidad de servicio en el envío de paquetes entre los nodos tomando decisiones adecuadas por si mismos de acuerdo a su conocimiento adquirido a través de la implementación de una capa oculta dentro de redes neuronales donde se puede aplicar un plan de aprendizaje empleando inteligencia artificial, y además controlar la existencia de nodos egoístas a través de técnicas de teoría de juegos mediante el conocido “dilema del prisionero”.

En la *Figura 22*, se muestra lo que se plantea en la presente investigación, con más detalle se presenta en las siguientes secciones cada una de sus partes.

El proceso que se sigue en la *Figura 22* es:

- I. *Ejecución del algoritmo WCA en cada nodo* que forman parte de la red, los resultados son enviados.
- II. *Recursos de nodo*, luego de ejecutarse el algoritmo WCA en cada nodo se obtiene como resultado los recursos de cada nodo como: capacidad de procesamiento, poder de batería, movilidad y potencia de transmisión.
- III. *Controlador de recursos*, una vez obtenido el resultado de los recursos de los nodos estos pasan a un proceso de control para ver si estos son suficientes para formar parte del cluster y entran a un proceso del dilema del prisionero, ó si son insuficientes se declaran como nodos egoístas por falta de recursos.

- IV. *Controlador de nodos egoístas*, antes de pasar a formar parte del cluster los nodos que han sido clasificados como aptos para formar parte del mismo, es necesario determinar si un nodo es egoísta o no. Los nodos en este paso empiezan a jugar el dilema del prisionero iterado, es decir da la oportunidad a los nodos egoístas como no egoístas en reivindicarse en su decisión en el caso de que ocurra un cambio de topología y se vuelva a ejecutar WCA.
- V. *Forma parte del cluster*, los nodos no egoístas forman parte del cluster y los egoístas pueden formar parte del cluster cuando ellos necesiten enviar algún paquete y esto ocurre cuando se ejecute el juego del dilema del prisionero y ocurra un cambio de topología y se vuelva a ejecutar WCA.
- VI. *Nodo jefe de cluster y nodo manejado*, una vez que se determina cuales son los nodos que forman parte del cluster se determina de acuerdo a las mejores características cual nodo es JC y cual es NM.
- VII. *Admisión de flujo*, tanto el nodo JC como el NM pueden solicitar admisión de flujo cuando así lo requieran, pero antes de admitir el flujo es necesario pasar por un proceso inteligente donde interactúa redes neuronales que se lo puede ver a detalle en la *Sección 3.3.3* y en la *Sección 3.3.5*.
- VIII. *Control de admisión*, una vez que solicitan los nodos admisión de flujo es necesario pasar por un control de admisión con la finalidad de evaluar si es factible o no que el resto de nodos colaboren con este nodo, es decir se evalúa si este es o no egoísta para permitir o no el acceso y poder asignar el siguiente salto para el flujo del paquete. Cuando ingresa a esta etapa se consideran varios aspectos para poder asignar inteligentemente cual es el siguiente salto esto en base a la experiencias que se adquiere por cada nodo en situaciones pasadas, para esto se toma en consideración la prioridad de llegada, ancho de banda que necesita, entre otros aspectos que se presentan en la *Sección 3.3.3*.

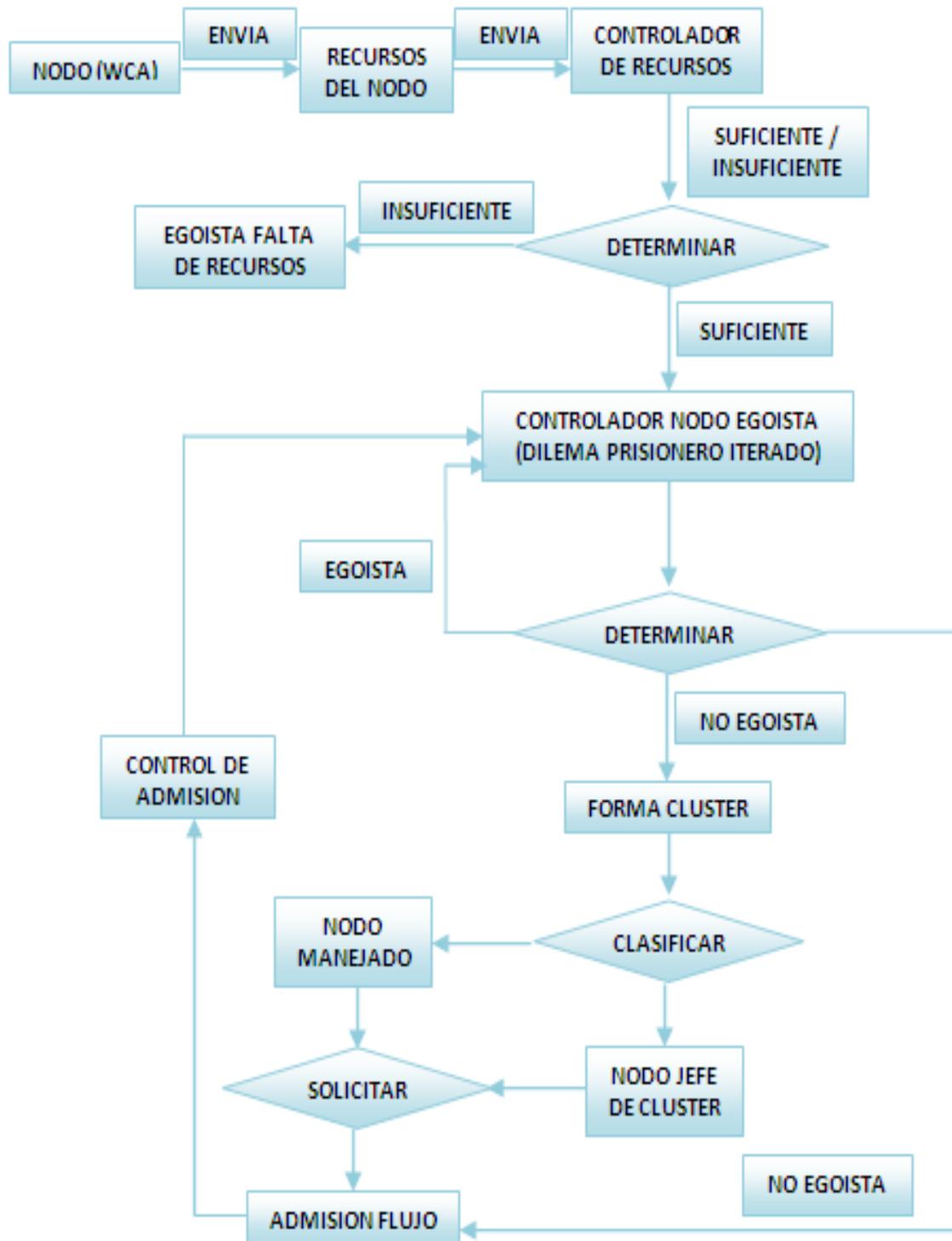


Figura 22. Modelo Propuesto

En pasos uno, dos, cinco y seis se trata de la *Arquitectura basada en Técnicas de Clustering [25]*, los pasos tres y cuatro se trata de *Arquitectura basada en Teoría de Juegos*, y los pasos siete y ocho se trata en la *Arquitectura basada en Redes Neuronales*. Todos estos pasos se pueden ver a detalle en las siguientes secciones.

3.3.2. Arquitectura basada en Técnicas de Clustering [25]

El manejo de una arquitectura basada en clustering es con el fin de manejar jerárquicamente a los nodos en agrupaciones denominadas cluster, donde cada nodo tiene información de sus nodos vecinos.

Como resultado de utilización de una topología jerárquica se obtiene menos registros de enrutamiento ya que un nodo sólo necesita tener conocimiento de la información sobre el enrutamiento de su propio clúster y no de toda la red.

La *Figura 23*, muestra el esquema del marco de trabajo basado en técnicas de clustering, donde se puede observar cómo está dividida la red en clusters.

El marco de trabajo contiene los siguientes elementos: servidor de administración (SA), servidor de administración respaldo (SAR), jefe clúster (JC), jefe de clúster respaldo (JCR), nodo manejado (NM), base de información, protocolos de gestión (AODV, WCA, DMAC, MobDhop).

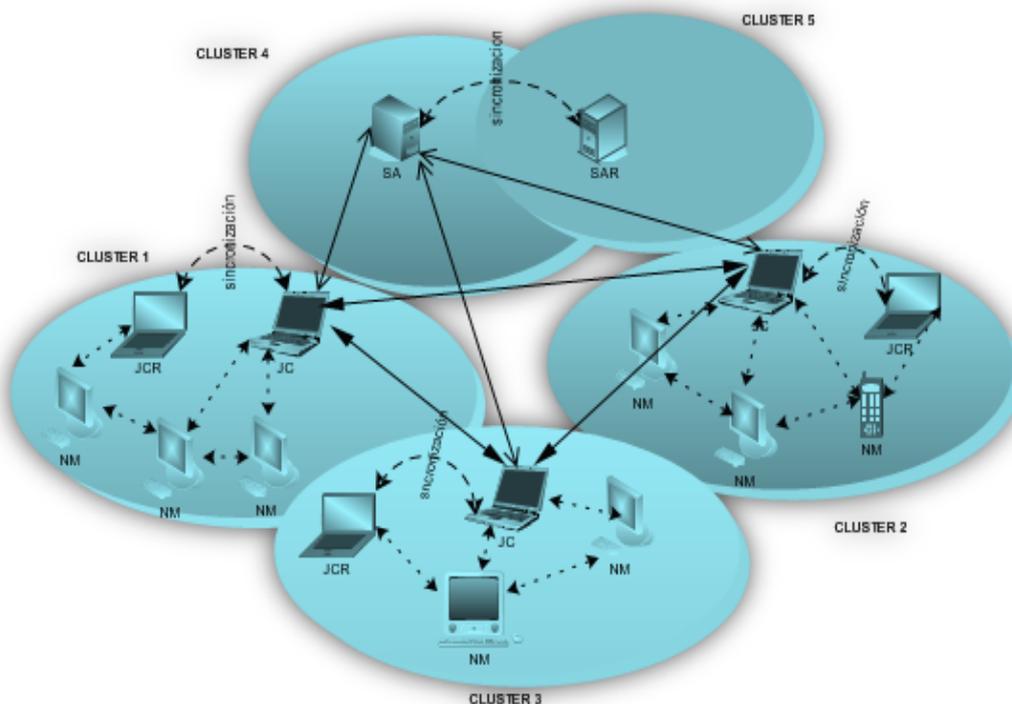


Figura 23. Esquema de una red ad hoc dividida en clusters [25]

Como podemos observar en la *Figura 23*, cada clúster tiene un nodo JC y varios NM. Cada nodo tiene un rol específico dentro de la red, uno de sus principales elementos es

el SA central que será elegido de entre todos los JC siempre y cuando este sea el más óptimo de entre todos.

El uso de equipos de respaldo para el jefe de cada clúster y del servidor administrador en un sistema de gestión de red aplicados en redes móviles ad hoc, mejora la eficiencia y la disponibilidad de los recursos en una red.

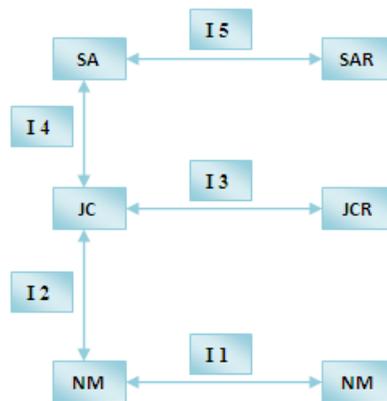


Figura 24. Interfaces de comunicaciones

En la *Figura 24*, se muestran cinco interfaces de comunicación:

Interfaz 1 es la comunicación existente entre nodo manejado a nodo manejado,

Interfaz 2 es la comunicación entre nodo manejado y JC.

Interfaz 3 es la comunicación entre JC y JCR con el fin de mantenerse sincronizados.

Interfaz 4 es la comunicación entre JC y SA

Interfaz 5 es la comunicación entre SA y el SAR para mantenerse sincronizado entre ellos.

Dentro de esta arquitectura se trabaja en dos interfaces dentro del modelo propuesto, en la *Sección 3.3.5*, se explica la relación con el modelo propuesto en la *Sección 3.3.1*.

3.3.3. Arquitectura basada en Redes Neuronales

Dentro del modelo planteado en la *Figura 22*, en la fase de control de admisión empieza a ejecutarse la arquitectura en base a redes neuronales. Está vinculada con la arquitectura basada en técnicas de clustering y dentro de esta incrementar en una de sus interfaces (interfaz de comunicación JC y NM) lo que corresponde a técnicas de redes neuronales e inteligencia artificial.

Las redes neuronales multicapa (MFNN) o feedforward son las más óptimas debido a que posee una capa de entrada, una capa oculta en donde se maneja un plan de aprendizaje y una capa de salida, y permite manejar las reglas de decisión basadas de acuerdo al cambio del medio y estado del sistema, debido a que su arquitectura es paralela.

Los componentes de procesamiento de MFNN se llaman neuronas. Las neuronas están interconectadas entre sí por medio modificable llamado peso que representa la fuerza de enlace de peso [31].

En la interfaz 1 se propone realizar todo lo correspondiente a las operaciones básicas planteadas en el núcleo de [7], el resultado de estas operaciones lo manejará el JC a través de la comunicación existente en la interfaz 2, este decidirá cuál es la mejor opción o la mejor ruta para cada nodo y realizara la reservación de recursos, todo esto a través del aprendizaje adquirido mediante la utilización de redes neuronales multicapa.

El objetivo principal de proponer la utilización de redes neuronales, es reducir al mínimo el retardo del tráfico de extremo a extremo en tiempo real para mejorar la calidad y servicio en la entrega de datos dentro de la red.

Con la utilización de este modelo se pretende garantizar diferentes operaciones básicas como tareas de enrutamiento y control de apoyo de calidad y servicio. Este modelo es complementado con una capa de aprendizaje inteligente, la cual aprende a través de las diferentes operaciones realizadas por las funcionalidades básicas a través de una red neuronal multicapa feedforward de acuerdo a las necesidades del usuario y estado de la red.

La utilización de esta arquitectura se basa en el modelo de redes neuronales con soporte multimedia (NNMS) [7], este modelo frente a otros modelos como SWAN [5], INSIGNIA y FQMM que fueron diseñados para mejorar la calidad y servicio, supera la calidad de entrega de datos en la red, gracias a la funcionalidad básica proporcionada a través del núcleo junto con el aprendizaje inteligente.

El núcleo de [7], no es más que un conjunto de operaciones básicas basadas en el modelo SWAN las cuales vienen a ser la capa de entrada en la red neuronal una vez finalizada las operaciones, por lo tanto el núcleo vendría a ser lo esencial en el modelo

porque a partir de esto se ejecuta la capa oculta donde se maneja el plan de aprendizaje inteligente para luego obtener el resultado a través de la capa de salida que vendría a ser la mejor opción o la mejor ruta para cada nodo con la respectiva reservación de recursos.

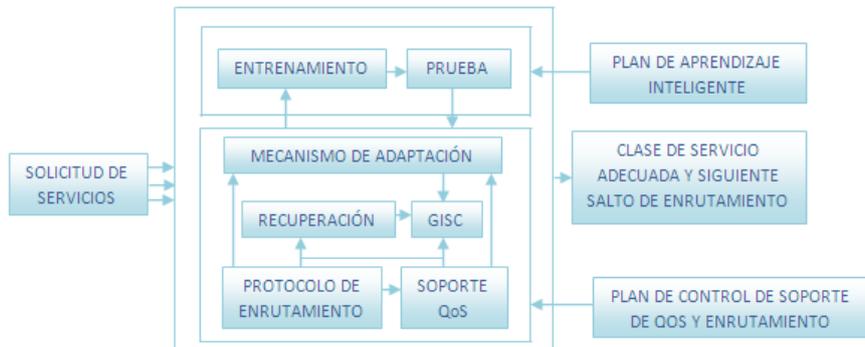


Figura 25. Modelo de redes neuronales con soporte multimedia [7]

Se efectúan dos planes de gestión dentro de este modelo con el fin de mejorar la calidad de entrega de datos en la red.

1. Primer plan, se basa en las clases de múltiples servicios para brindar calidad de servicio en el flujo de extremo a extremo.
2. Segundo plan, es el de aprendizaje inteligente en cual permite una adaptación dinámica de la red a las condiciones de tráfico a través de una herramienta inteligente de redes neuronales.

En la *Figura 25*, se muestra las principales funciones del modelo como: el esquema de enrutamiento el cual permite descubrir las rutas existentes entre el origen y destino. Vea más detalladamente en la sección 4.3.

El plan de control de soporte de calidad de servicio, es responsable de proporcionar garantías de calidad de servicio a las necesidades de los usuarios en términos de diferentes restricciones como: ancho de banda, retardo y número de saltos. Por lo que las funcionalidades requeridas como reserva de recursos y control de admisión están a cargo de este plan.

Los intervalos de garantía de las clases de servicio (GISC) que están dentro del plan de control de soporte de calidad y servicio, son un conjunto de intervalos que representan el estado de la red en términos de diferente métrica. GISC permite una selección de la ruta adecuada de conformidad con los requisitos de flujo. La actualización de los GISC se efectuará ante cualquier variación de los recursos de la red o la recuperación de violación de calidad del servicio.

En el plan de aprendizaje para verificar el nivel de inteligencia o aprendizaje de los nodos, las redes neuronales realizan la fase de entrenamiento de las operaciones del núcleo (*sección 4.3.1.1*) y la prueba del conjunto de requisitos de calidad. Luego que se alcanza la fase de entrenamiento, se realizan las fases de la prueba y validación.

Una vez alcanzado el nivel de aprendizaje, un nodo es capaz de seleccionar el próximo salto para el encaminamiento de los paquetes de datos recibidos sin realizar ningún otro procedimiento, como el descubrimiento de ruta.

3.3.3.1. Modelo del núcleo basado en SWAN [5]

Basándose en el modelo SWAN, se realizan cambios para la realización del modelo del núcleo de NNMS. Los cambios que se realizan es el incremento de dos nuevas técnicas que se desarrollan e integran en el núcleo para la detección y recuperación de violación de calidad de servicio.

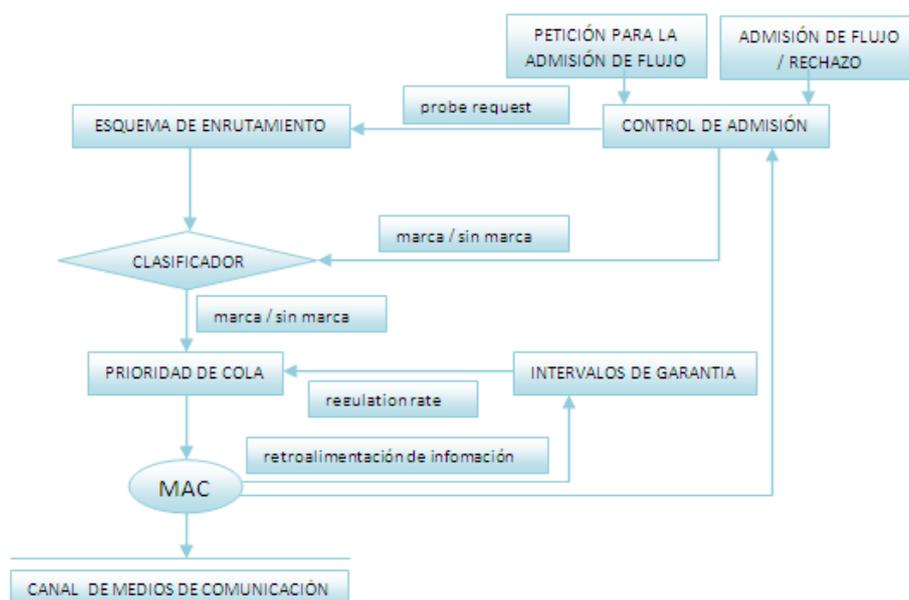


Figura 26. Modelo del núcleo basado en SWAN [7]

Definición de las operaciones del núcleo NNMS de la *Figura 26*.

- **Control de Admisión:** en esta fase se decide que flujo se admite y cuál se rechaza. El control de admisión estima el ancho de banda local disponible en cada nodo, y proporciona información precisa sobre el estado de ancho de banda, retardo de extremo a extremo y número de saltos de las rutas descubiertas entre la fuente y el destino.
- **Esquema de Enrutamiento:** realiza el descubrimiento de rutas y reservación de recursos.
- **Clasificador:** es capaz de diferenciar entre los flujos en términos de requisitos de calidad del servicio como: retardo, ancho de banda y flujos de mejor esfuerzo.
- **Priorización de Cola:** se utiliza para la regularización del tráfico en la red y para que el tráfico que pertenece a clases de menor prioridad se regule de forma más agresiva que no pertenezcan a clases de mayor prioridad.
- **Intervalos de Garantía:** se ajustan dinámicamente de acuerdo con la información de retroalimentación recibida por el nodo sobre el estado de la red.

Una vez establecido el funcionamiento del núcleo el plan de aprendizaje inteligente se realiza a través de las operaciones que se presentan en la *Figura 27*.

Es conveniente aclarar que las redes neuronales son una tecnología robusta de inteligencia artificial la cual puede manejar nodos de naturaleza compleja y dinámica en los distintos procesos.

El proceso a través de redes neuronales es el siguiente, la red está dividida en capas: la capa de entrada que se compone de las entradas a la red; la capa oculta que constará de un número de neuronas colocadas en paralelo, en cada neurona se realiza una suma ponderada de las entradas, y a través de la capa de salida se genera la transferencia de funciones de salida.

Con el fin de asegurar el comportamiento adaptativo de la red, la correlación entre los niveles sucesivos de las neuronas se cambia cuando los valores de los pesos son modificados. El proceso de la conducta adaptativa que se conoce como formación.

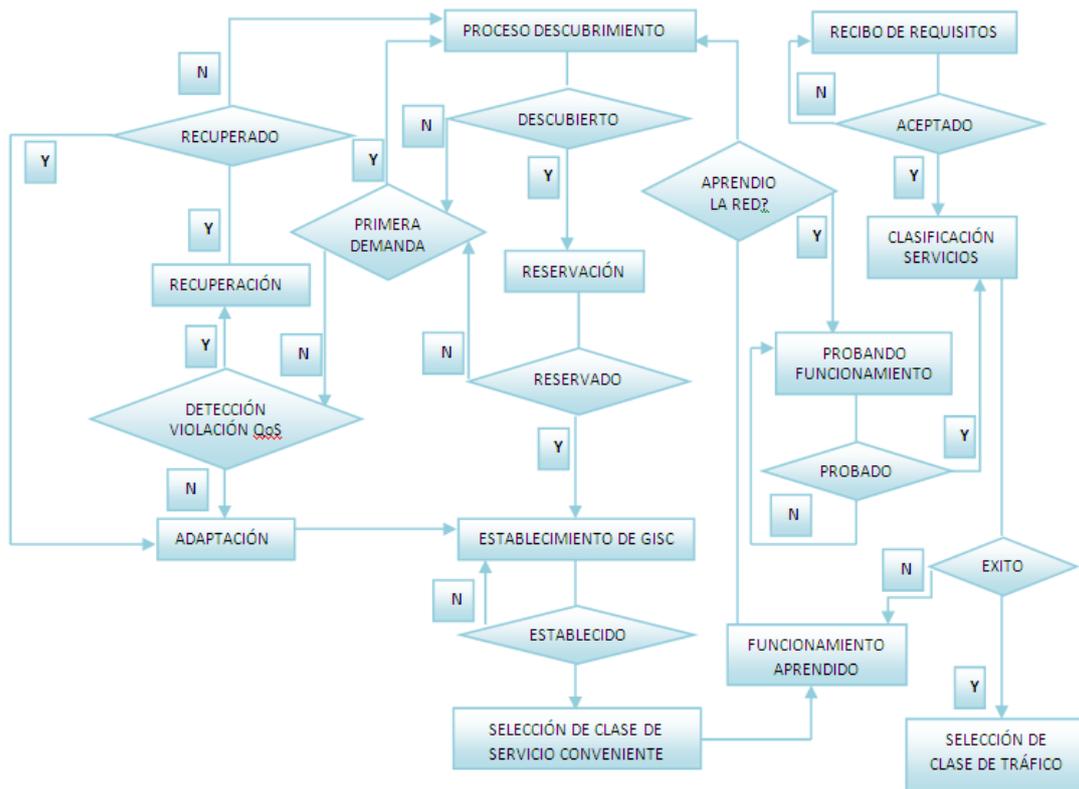


Figura 27. Diagrama de operaciones [7]

La capacitación de la red es el proceso de estimación de los pesos de conexión, minimizando algunas medidas de error global, utilizando mecanismos tales como algoritmo de retropropagación (BP (Back Propagation)). BP es un algoritmo de entrenamiento ampliamente utilizado en una red neuronal.

Para las neuronas de la entrada y las capas ocultas se ha utiliza la función de transferencia tan-sigmoid. La neurona de la última capa tiene la función de transferencia lineal. Las dimensiones de los patrones de entrada que representan a los requisitos de QoS (retardo, ancho de banda y número de saltos requeridos por el usuario) especifica el número de nodos de entrada. En cuanto al número de nodos en su capa oculta, de acuerdo con la teoría de Kolmogorov [20], si el número de nodos de entrada es N , entonces $2N+1$ un nodo oculto puede ser utilizado. Los nodos de salida se determinan por el número de categorías que se clasifican. Se pueden clasificar en tres categorías que corresponden al tráfico con limitación de retraso, al tráfico con limitación de ancho de banda y al tráfico de mejor esfuerzo. Para cada categoría, MFNN da el siguiente salto para el encaminamiento del tráfico recibido.

Para las categorías de los nodos de salida se emplea una clasificación binaria, entonces cada bit de una salida deseada presenta el estado 1 o el estado 0: estado 1 significa "pertenece a" y el estado 0 significa "no pertenece a" una categoría determinada.

Los pesos de cada unidad se deben ajustar para que el error entre la salida deseada y la producción real se reduzca con el fin de capacitar a una red neuronal para realizar alguna tarea. Esto requiere que la red neuronal calcule el error derivado de los pesos.

El BP es un método utilizado para determinar este error llamado MSE (error cuadrático medio). Se calcula como la diferencia media cuadrática entre la salida deseada y la producción real.

Dado que buscamos minimizar el error para cada peso por separado, el error global puede aumentar. Luego, debemos calcular el error de salida total después de cada adaptación, cuando el error es mayor que el error rechazado anteriormente, después de que calculamos las nuevas tasas de aprendizaje.

La modificación de los pesos continúa hasta que la producción real se aproxima a la respuesta deseada. El problema de la convergencia de mucho tiempo de BP se resuelve mediante la incorporación de la expresión dinámica del peso de la BP.

La *Figura 27*, muestra el diagrama esquemático de las operaciones NNMS basado en SWAN. El mismo muestra la relación entre las operaciones del núcleo y su proceso de aprendizaje mediante redes neuronales. Las operaciones es decir las funcionalidades descritas en la *Figura 25* y *Figura 27*, tales como control de admisión, etc., están representados por cuadrados, y la comprobación del éxito de las operaciones está representado por rombos ('Y' y 'N' significan, respectivamente, el éxito y el fracaso de la operación).

La aplicación de MFNN se divide en dos fases. La primera fase es la formación de la red neuronal con los datos de un conjunto de entrenamiento y las pruebas después del entrenamiento. La segunda fase es la predicción y la selección de la clase de servicio y del siguiente salto para reenviar los paquetes de datos.

Como se muestra en la *Figura 27*, la etapa de entrenamiento de la red se realiza después de varios escenarios de ejecución de las operaciones del núcleo como: el proceso de

descubrimiento, proceso de reserva, adaptación del tráfico a través de priorización en las colas, detección y recuperación de violación QoS, etc. Una vez que la etapa de formación del núcleo se logra, se inicia la operación de las pruebas sobre escenarios adicionales para probar la exactitud del resultado de la primera etapa. El éxito de aprendizaje de la red es capaz de satisfacer las necesidades de flujo sin realizar las operaciones requeridas por el núcleo.

3.3.4. Arquitectura basada en Teoría de Juegos

La determinación de los nodos egoístas dentro de la conformación de las redes Ad Hoc es tarea de la teoría de Juegos, ahora bien se pretende determinar cuando un nodo es egoísta en la interfaz 1 y en la formación del clúster en el modelo propuesto en la *Figura 23*.

Lo que se puede aportar adicional en este tema además del modelo es determinar si un nodo es egoísta ya sea por falta de recursos o simplemente porque no quiere colaborar con el resto de los nodos en la conformación de la red.

La cooperación entre los nodos es primordial dentro de una red ad hoc y dentro de la formación del clúster, esto para determinar que rol cumple el nodo dentro de la red y retransmitir los paquetes entre ellos.

Debido a que son dispositivos móviles los que conforman una red Ad Hoc es importante considerar la gran cantidad de energía que se consume al transmitir un paquete lo cual hace difícil tomar la decisión de que un nodo colabore o no para la transmisión del mismo. Teniendo en cuenta esto es probable que un nodo adopte un comportamiento egoísta, lo cual hace necesario desarrollar un modelo de confianza que permita a los nodos decidir con quién cooperar y con quien no, para maximizar así la operatividad de la red con el menor consumo de energía posible, aislando a los nodos egoístas y estimulando la cooperación entre los nodos que estén dispuestos a cooperar.

Con el fin de cumplir este propósito, consideramos la toma de decisiones desde el punto de vista de la teoría de juegos:

Cooperar para ganar confianza de los nodos que después se necesitará, o no coopero para evitar desperdiciar energía atendiendo nodos que no cooperarán. Una versión

simplificada de este modelo corresponde al famoso dilema del prisionero, como muestra la *Figura 28*.



Figura 28. NM1 y NM2 necesitan cooperar entre ellos para llevar información a su destino

El dilema del prisionero consiste en que los nodos no pueden estar dispuestos a cooperar para transmitir un paquete, inclusive si esto sea de interés para ambas partes.

Utilizando el dilema del prisionero la cooperación entre los nodos se puede obtener como un resultado de equilibrio. Ya que el juego se repite reiteradamente, se ofrece a cada nodo la oportunidad de castigar al otro por la no cooperación en juegos o transmisiones anteriores.

En la *Figura 28*, se puede observar que el nodo NM1 llegue a su destino NMB necesita la colaboración del nodo NM2, el cual a su vez necesita de NM1 para llegar a su destino NMA.

La arquitectura basada en inteligencia artificial y redes neuronales es complementada con la arquitectura basada en teoría de juegos.

En la *Figura 26*, podemos observar cómo está compuesto el núcleo, ahora bien lo que se plantea es incrementar políticas para la detección de nodos egoístas al entrar a la fase de control de admisión y en la formación del clúster, estas políticas nos permitirán determinar si un nodo es egoísta o no y así aislarlo y no poder contar con él dentro de la red.

3.3.5. Integración de la Arquitectura basada en Técnicas de Clustering, Redes Neuronales y Teoría de Juegos

Partiendo de las arquitecturas planteadas anteriormente se propone unificar estas arquitecturas con el fin de mejorar la calidad de servicio en la comunicación extremo a extremo y alargar la supervivencia de la red con la incrementación del ahorro de energía y así dicha comunicación sea más estable y duradera.

Indicaciones previas al Modelo

Existen algunas condiciones previas a la conformación del modelo como:

- La movilidad de los nodos debe ser entre media y baja es decir los nodos no deben ser muy inestables
- Utiliza tecnología WIFI con un enlace a través de antena
- El alcance de la red es de 100m
- Los nodos deben ser capaces de soportar SNMP
- El protocolo de transporte es UDP
- Utilización el algoritmo WCA para la formación del clúster
- Utilización del modelo SWAN como base para NNMS
- Utilización de teoría de juegos para la determinación de nodos egoístas

Dentro del modelo planteado en [25] se propone aplicar esta solución en la formación del cluster.

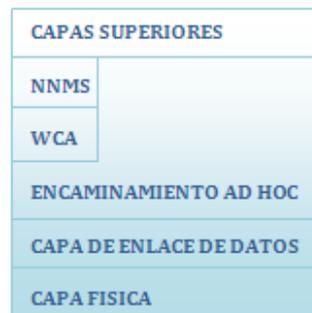


Figura 29. Esquema de un nodo JC de la red Ad Hoc

En la *Figura 29*, podemos observar como estaría conformado un nodo JC de la red Ad Hoc.

De acuerdo con el algoritmo WCA se puede elegir al JC o clusterheads de acuerdo a la capacidad de los nodos con el objetivo de optimizar el consumo de batería, el balanceo de carga y funciones de la capa MAC. Para la elección del JC se considera los siguientes aspectos:

- El número de nodos que JC puede manejar o balanceo de carga
- Movilidad

- La potencia de transmisión
- Consumo de batería

Una vez que se tiene conocimiento de las características de los recursos de cada nodo a través de WCA como se aprecia en la *Figura 30*, se puede determinar si un nodo es egoísta o no para luego aplicar teoría de juegos aplicando el dilema del prisionero en cada nodo.

Un nodo puede ser egoísta por no poseer los recursos necesarios para colaborar dentro de la red o por el simple hecho de ahorrar energía.

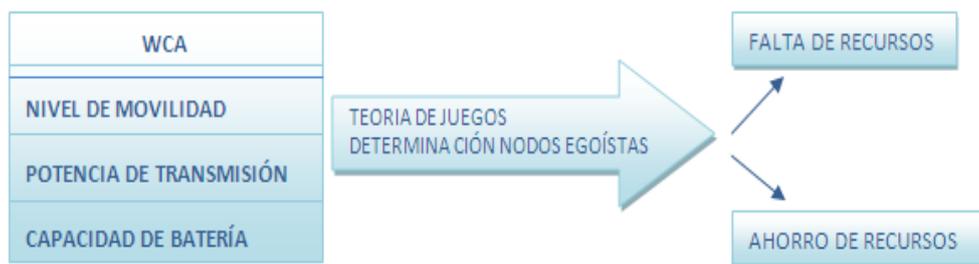


Figura 30. Determinación de nodos egoístas

Con la información adquirida de los recursos de los nodos, estos entran a un juego en donde reciben una ganancia o pérdida por su disposición de colaborar o no colaborar con la red. Las ganancias o castigos que reciben se las realiza a través de estándares predeterminados por el administrador.

La *Figura 30* se puede representa en pseudocódigo de la siguiente manera:

```

Variables flotantes  nmov, ptrans, cbate
                    const1, const2, const3
Si nmov <= const1 {
    Si ptrans >= const2 {
        Si cbate >= const3 {
            presentar -> "El nodo desea ahorrar energía"
        } caso contrario
        Presentar -> "El nodo no es apto para formar parte del
cluster por falta de recursos"
    }
}

```

Inicialmente se declaran las variables nmov que representa el nivel de movilidad, ptrans representa la potencia de transmisión y cbate representa la capacidad de batería, que

vendrían a ser los parámetros que se consideran para la elección del JC. Para establecer si estos parámetros son idóneos para realizar la mejor elección se establecen unas constantes $const1$, $const2$, $const3$ que nos ayudaran para la comparación y elegir el JC. Partiendo del resultado de la comparación los parámetros con las constantes se puede establecer si un nodo es egoísta ya sea por falta de recursos o por ahorro de energía.

Dentro de un nodo manejado como se muestra en la *Figura 31*, se maneja teoría de juegos y el algoritmo WCA pudiendo así identificar si un nodo es egoísta con el fin de tomarlo en cuenta para la formación del clúster.

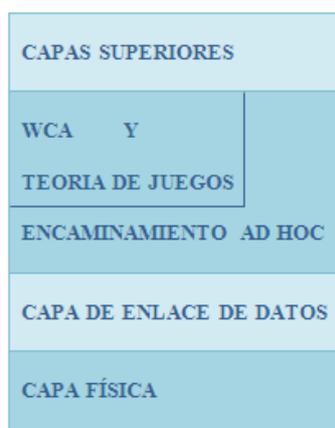


Figura 31. Esquema de un NM

Siguiendo con el proceso *Figura 31*, a partir del modelo SWAN se realizan modificaciones para la creación de NNMS [7], el cual utiliza una capa MAC best-effort, además que es capaz de distinguir entre el tráfico de tiempo real y best-effort, el cual se realiza a través del protocolo de transporte UDP.

El objetivo principal de unificar redes neuronales con el modelo de calidad y servicio en mejorar la gestión de la misma ayudando a que esta sea fiable y eficaz.

Como se había especificado en la sección basada en redes neuronales el proceso de control de admisión se realiza luego de clasificar los paquetes. La aportación en esta parte es la utilización de teoría de juegos en la fase de control de admisión, de tal manera que antes de ingresar el paquete se tome en consideración si un nodo está dispuesto a colaborar con la red.

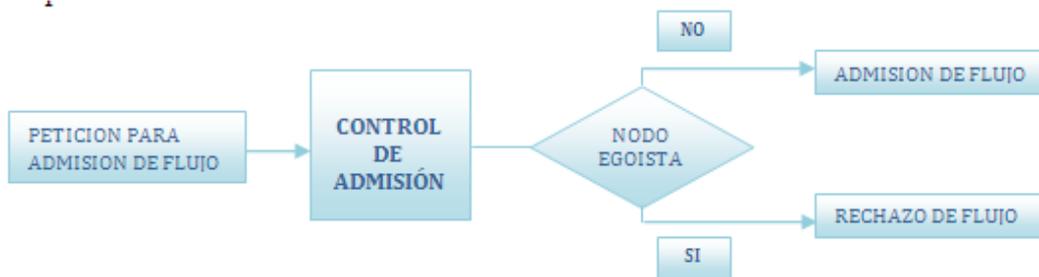


Figura 32. Determinación de nodos egoístas. Fase de control de admisión

Además de la estimación de ancho de banda local disponible en cada nodo y de proporcionar la información sobre el estado de ancho de banda, retardo extremo a extremo y número de saltos de las rutas descubiertas entre la fuente y el destino, se propone emplear el dilema del prisionero para la determinación de nodos egoístas de tal manera que antes de permitir la admisión de flujo de un nodo ya se sepa si se trata de un nodo egoísta o no con el fin de ahorrar energía de los nodos para el paso de paquetes como se muestra en la *Figura 32*.

La *Figura 32*, se puede representar en pseudocódigo de la siguiente manera:

Control_Admisión

Variables enteras n, a=0, b=1;

```

Si n=a {
  ca= aceptar
} caso contrario {
  Si n=b{
    ca= rechazar
  }
}
  
```

Se declaran las variables n que representa a la petición de un nodo, a representa si un nodo no es egoísta y b si un nodo es egoísta. Partiendo de la declaración de las variables se procede a realizar comparaciones de que si un nodo no es egoísta se realiza la admisión del flujo, caso contrario si el nodo es egoísta se rechaza el flujo.

3.4. CONTRIBUCIONES

La finalidad de este trabajo es contribuir con [25], por lo que se ha planteado algunas fases dentro del mismo con la finalidad de ayudar a la gestión de la red.

Las contribuciones que se realizan son las siguientes:

- Emplear Redes Neuronales en el nodo jefe de clúster para determinar el siguiente salto de un nodo con la información adquirida a partir de las operaciones realizadas en el modelo SWAN a través de un plan de aprendizaje inteligente realizado con redes neuronales feedforward con ayuda del algoritmo back propagation.
- Incrementación de teoría de juegos para la determinación de nodos egoístas a partir del algoritmo WCA. Con la información de los recursos de los nodos obtenidos a través de WCA de determina si un nodo egoísta es egoísta por falta de recursos o porque simplemente no desea colaborar con la red.
- Dentro de NNMS una de sus operaciones es el control de admisión en donde se determinan factores importantes para aceptar que un nodo pueda enviar sus paquetes o no, dentro de esta fase se propone incrementar teoría de juegos de tal manera que permita adquirir confianza entre los nodos y determinar si este es egoísta o no para permitir que avance a la siguiente fase que es de admitir o rechazar que pueda transmitir sus paquetes.

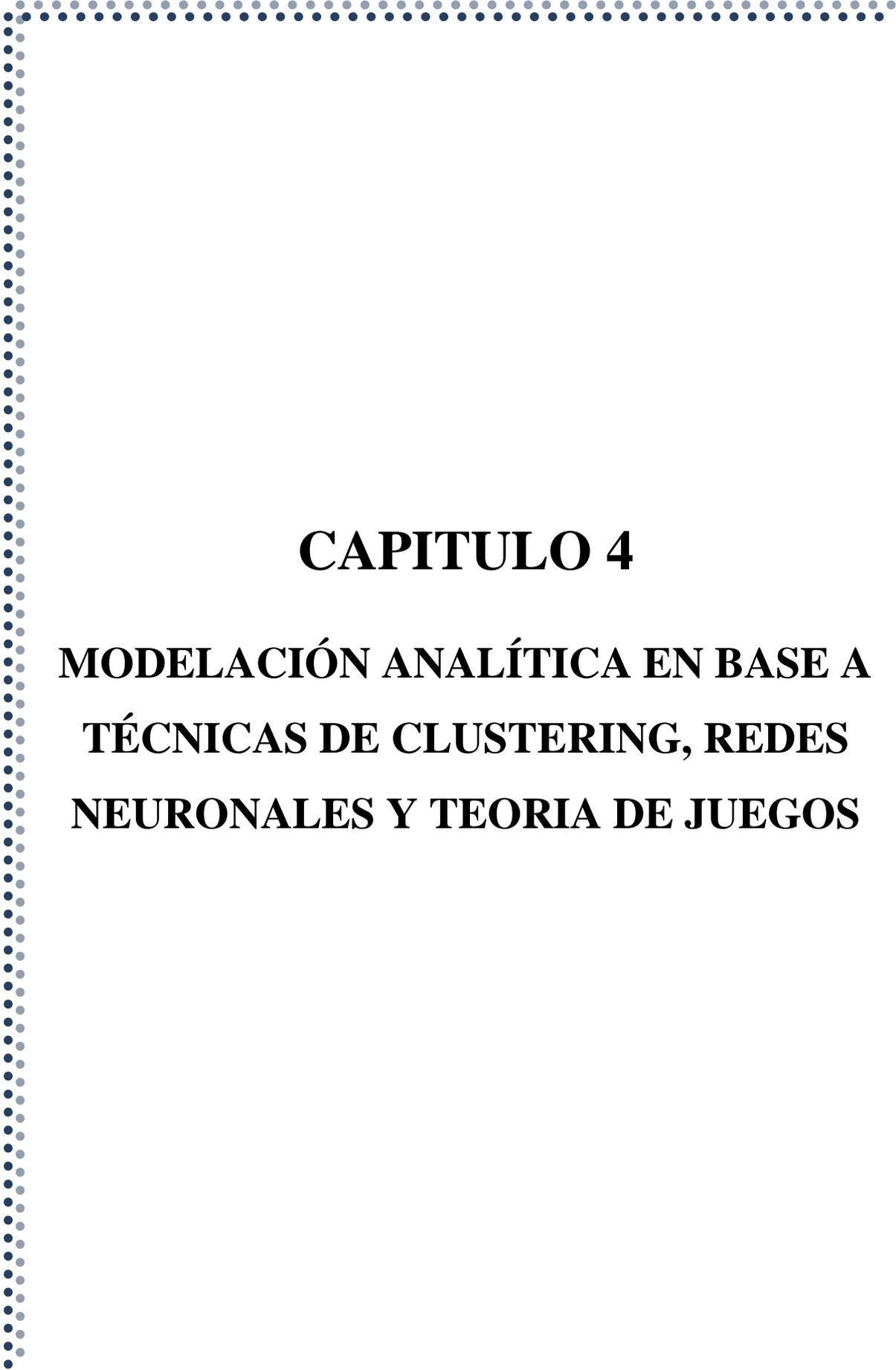
3.5. CONCLUSIONES

El emplear técnicas innovadoras como redes neuronales y teoría de juegos es para optimizar la gestión de redes ad hoc mejorando la calidad y servicio de las mismas.

La energía es uno de los recursos más apreciados en los nodos que conforman la red por lo que la determinación de nodos egoístas es fundamental para no poder colaborar con los que no han adquirido un nivel de confianza necesaria y así ahorrar energía con quienes no van a hacer útiles más adelante.

La adquisición de un aprendizaje inteligente dentro de la gestión de redes ad hoc con el fin de optimizar sus servicios es importante para garantizar su calidad y servicio en la transmisión de paquetes.

Las redes neuronales son una herramienta de inteligencia artificial muy útil para desarrollar un aprendizaje dentro de los nodos para que tomen la mejor decisión en el caso que lo requieran. La aplicación del aprendizaje inteligente en el JC sirve para que tome la mejor decisión indicando a los NM cuál es el siguiente salto que deberían dar para mejorar el paso de paquetes en menor tiempo.

A decorative border composed of small, dark blue dots arranged in a rectangular shape, with the top and left sides being solid and the bottom and right sides being dashed.

CAPITULO 4

**MODELACIÓN ANALÍTICA EN BASE A
TÉCNICAS DE CLUSTERING, REDES
NEURONALES Y TEORIA DE JUEGOS**

4. REDES AD HOC CON REDES NEURONALES Y TEORÍA DE JUEGOS

4.1. INTRODUCCIÓN

Partiendo de que las redes ad hoc tienen recursos finitos ó limitados como ancho de banda, batería y capacidad de procesamiento, es necesario que la administración de los mismos permita mejorar el desempeño y disponibilidad de la red, a través de la utilización de nuevas técnicas como redes neuronales y teoría de juegos.

En este capítulo se presenta la modelación analítica del modelo propuesto en el capítulo anterior. Inicialmente se combina el algoritmo WCA con teoría de juegos en la elección del clusterhead, luego en la interfaz 2 de [25] se combina redes neuronales con SWAN en donde en base a las experiencias que se adquieren a partir de SWAN se emplea redes neuronales para predecir el siguiente salto de un nodo y se controla la existencia de nodos egoístas en la fase de control de admisión que está dentro de SWAN.

4.2. DETERMINACIÓN DE NODOS EGOISTA A TRAVÉS DE WCA Y TEORIA DE JUEGOS

En la *Figura 30*, podemos observar que a partir del algoritmo WCA, se determina que nodo es clusterhead a través del nivel de movilidad, balanceo de carga, distancia entre nodos, potencia de transmisión y capacidad de batería; y además se puede manejar teoría de juegos para la determinación de nodos egoístas a través del dilema del prisionero iterado, creando un modelo de confianza que sería: cooperar para ganar la confianza del resto de nodos de la red que después puede necesitar, o no cooperar para evitar el desgaste de energía atendiendo a nodos que no cooperarán con este. Esto permitirá a los nodos decidir con quién cooperar y con quien no con el fin de maximizar la operatividad de la red con el menor consumo de energía, aislando a los nodos egoístas y estimulando la cooperación entre nodos que estén dispuestos a cooperar y ser parte del cluster.

El proceso de elección del JC es de acuerdo al peso asociado a cada nodo v [3] y al nivel de confianza de cada nodo:

$$W_v = w_1\Delta_v + w_2D\Delta_v + w_3M_v + w_4P_v + w_5N_c$$

El nodo con menor peso es elegido como clusterhead. Los factores de peso se seleccionan de forma que $w_1 + w_2 + w_3 + w_4 = 1$.

Parámetros asociados:

- Medida de movilidad $M_v = \frac{1}{T} \sum_{t=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$, donde (X_t, Y_t) y (X_{t-1}, Y_{t-1}) corresponden a las coordenadas del nodo v en el tiempo t y $(t-1)$ respectivamente. Aquí lo que se calcula es la suma de las velocidades para cada nodo respectivamente después de un tiempo T .
- Con el fin de asegurar el balanceo de carga se calcula Δ_v que es el grado de diferencia de un nodo. Primero se calcula el grado de un nodo d_v , este parámetro se calcula como la suma de los vecinos de v , los cuales deben cumplir una condición de estar dentro de su rango de transmisión.

$$d_v = |N(v)| = \sum_{v' \in V, v' \neq v} \{dist(v, v') < tx_{range}\}$$

Δ_v , se calcula con el objetivo de asegurar el balanceo de carga: $\Delta_v = |d_v - \delta|$

- Distancia entre nodos $D_v = \sum_{v' \in N(v)} \{v, v'\}$, está relacionado con el consumo de energía. A menor distancia la conexión es mejor entre el clusterhead y sus vecinos. A mayor distancia la señal de comunicación se va atenuando, por lo que las comunicaciones de larga distancia requieren mayor energía.
- P_v , es el valor que adquiere el tiempo acumulado que lleva un nodo siendo clusterhead.
- N_c , es el nivel de confianza que adquiere cada nodo a través de la cooperación con el resto de nodos.

Partiendo de la *Figura 30*, se puede determinar dos tipos de nodo egoístas:

a. Nodos Egoístas por ahorro de energía

Partiendo de la elección del JC, el cual tiene en su MIB de SNMP las características fundamentales de los nodos como: estado, ubicación, poder de batería, etc., se puede determinar si un nodo es egoísta con el fin de ahorrar recursos y de acuerdo a su nivel de confianza.

Si un nodo posee las características adecuadas para formar parte del cluster como: distancia apropiada para poder establecer una buena conexión y poder de energía aceptable, y este no desea formar parte del clúster para evitar el desgaste de batería al momento de la transmisión de paquetes, se denomina que es un nodo egoísta por ahorro de energía.

Además si un nodo que no desea formar parte del clúster en sus primeras interacciones y posee los recursos adecuados para ser parte del mismo puede reivindicarse, y luego de que exista cambio de movilidad por parte del JC o exista cambio de JC, este puede pasar a formar parte del clúster debido a que el dilema del prisionero está dispuesto a perdonar y considerar su nuevo comportamiento de colaborar incrementando así su nivel de confianza.

b. Nodos Egoístas por falta de recursos

El proceso es similar a la determinación de los nodos egoístas con la finalidad de ahorrar energía, la diferencia radica en que las características de este nodo no son las más adecuadas de acuerdo a las exigencias para formar parte del clúster.

La razón por la que un nodo puede ser egoísta por falta de recursos es porque no tiene el poder de energía necesaria para colaborar como nodo intermedio para la transmisión de paquetes. Es decir la pérdida de energía al colaborar con un nodo vecino va a ser mayor a la ganancia que adquiriría este nodo.

4.2.1. Nivel de Confianza y Dilema del Prisionero [1]

Con la finalidad de adquirir un nivel de confianza por un nodo con sus nodos vecinos, desde el punto de vista de teoría de juegos para la toma de decisiones se tiene en consideración:

- Coopero con los nodos vecinos para adquirir confianza, debido a que más adelante podre llegar a necesitar de su colaboración.
- No coopero con los nodos vecinos para evitar el desgaste de energía colaborando con nodos que no cooperarán.

Como podemos ver en la *Figura 28*, el juego empieza con la necesidad que tiene un nodo de transmitir sus paquetes a un destino y para ello necesita la colaboración de nodos intermedios y termina cuando llega a su destino o cuando un nodo no decide colaborar con este.

En la *Figura 28*, el nodo *NM1* necesita llevar los paquetes a su destino *NMB*, al igual que *NM2* a *NMA*. Para que los paquetes lleguen a su destino *NM1* necesita la cooperación de *NM2*, quien a su vez necesita la colaboración de *NM1* para llevar los paquetes hasta su destino *NMA*.

El proceso de que los paquetes lleguen o no a su destino es recompensado o castigado de acuerdo a los siguientes parámetros de proceso del juego:

- Si los paquetes llegan a su destino ganan A pesos cada vez que esto suceda.
- La transmisión de los paquetes es costoso en términos de energía y es por eso que pierden B pesos con cada transmisión.
- Para que un nodo se decida a participar en la red es necesario que $A > B$.
- Si los nodos cooperan entre si reciben un premio $P = A - 2B$. Recibe A ganancias porque logró que su paquete llegue a su destino, $-B$ por haber transmitido su propio paquete, y $-B$ por haber retransmitido el paquete de otro nodo.
- En el caso de que el nodo *NM1* decida caer en la tentación de no cooperar con *NM2* con la finalidad de ahorrarse B pesos y *NM2* ya coopero con *NM1*, la ganancia que recibirá *NM1* es $T = A - B$, y *NM2* recibirá $M = -2B$ debido a que no logro su objetivo de llevar a su destino su paquete y únicamente fue utilizado por *NM1*. Este proceso se puede dar también si el *NM2* no decide colaborar con *NM1* una vez que este ya haya colaborado.
- Si los nodos no deciden cooperar entre si reciben un castigo $C = -B$

Estructura de pagos por acceder a la tentación (T), recibir el premio por colaborar (P), recibir el castigo por no colaborar (C), y dejarse ser utilizado (M), obedece a la relación $T > P > C > M$ [1].

	NM2 Descarta	NM2 Colabora	
NM1 Descarta	(C,C)	(T,M)	$M = - 2B$
NM1 Colabora	(M.T)	(P,P)	$C = - B$
			$P = A - 2P$
			$T = A - B$

La utilización del modelo de confianza es para disminuir a desconfianza entre los nodos en que ninguno de ellos esté dispuesto a colaborar con el otro, la estrategia empleada por *NM1* es:

- Si el nodo *NM2* no me colabora, yo no le colaboro para evitar ser utilizado

- Si el nodo NM2 me colabora, yo no colaboro para ahorrar Y pesos de transmisión

El nodo NM2 realiza el mismo análisis de tal manera que independientemente de lo que haga cada nodo, deciden que es más conveniente no colaborar y recibir un castigo $C = -B$, aunque lo más factible es recibir un premio $P = A - 2B$ en caso de que decidieran colaborar. La actitud que toman los nodos de no colaborar y descartar el paquete del otro nodo se constituye como equilibrio de Nash.

La situación del dilema del prisionero iterado se da cuando los nodos necesitan enviar más de un paquete y estarán interactuando constantemente aquí la reciprocidad es importante debido a que es mejor cooperar y se adquiere mejores ganancias. Esta situación del dilema está dispuesta a perdonar, no son envidiosas es decir no importa si él un nodo gane más que el otro mientras se esté satisfecho con las ganancias individuales, además no permite que los nodos sean utilizados para bienes individuales.

En el caso de la red ya segmentada en cluster cada par de nodos se encuentran jugando su propio dilema del prisionero iterado, pero con la condición de que para que cada paquete llegue exitosamente a su destino es necesaria la cooperación simultánea de todos los nodos que participan en la ruta entre la fuente y destino.

En el modelo de confianza se incluye un mecanismo de evaluación de confianza mediante el cual cada nodo observa a sus compañeros y mantiene para cada uno la tasa de retransmisiones, esto es, la fracción de paquetes que le ha visto retransmitir entre aquellos que le ha visto recibir para retransmisión.

El cálculo de la tasa de transmisiones que el nodo NM2 ha calcula para el nodo NM1 en base a n observaciones se muestra en la ecuación. $d_i \in \{0, 1\}$ es la decisión que NM2 vio tomar a NM1 en la i -ésima observación.

$$f_r(NM2, NM1; n) = \begin{cases} \frac{1}{n} \sum_{i=1}^n d_i & n < m \\ \frac{1}{m} \sum_{j=0}^{m-1} d_{n-j} & n \geq m \end{cases}$$

La confianza que el nodo NM2 tiene en el nodo NM1, $T\{NM2, NM1\}$, será 0, 1, 2 ó 3 según la tasa de retransmisiones $f_r(NM2, NM1)$ esté en los rangos $[0, 0.3)$, $[0.3, 0.6)$,

[0.6, 0.9) ó [0.9, 1.0]. El modelo también incluye un mecanismo de operación de la red basado en teoría de juegos, en el que un juego inicia cuando un nodo fuente genera un paquete para ser transmitido sobre una ruta multisalto y termina cuando el paquete llega a su destino o es descartado por algún nodo intermedio.

4.3. EMPLEO DE REDES NEURONALES Y TEORÍA DE JUEGOS

En el esquema de la *Figura 29*, se puede observar la pila de protocolos que interactúan en un JC, de tal manera que se procede a la explicación del desarrollo de esta fase donde se emplea NNMS y Teoría de juegos como se puede observar en la *Figura 32*.

En la *Figura 26*, se observa las operaciones que se realizan en el núcleo de NNMS desarrollado en base a SWAN, una de las operaciones es el control de admisión en donde se emplea adicionalmente teoría de juegos para la determinación de nodos egoístas. Estas operaciones siguen ciertos procesos que vamos a ver a continuación.

4.3.1. Componentes del Modelo

Lo que se propone es partir del núcleo de NNMS emplear la alimentación de varias capas de la red neuronal hacia adelante para la optimización de la decisión del núcleo de forma adaptativa en función del estado de la red.

Una de las ventajas del núcleo es que los recursos asignados durante la fase de reserva se eliminan automáticamente de forma independiente y totalmente distribuidos cuando cambia de trayectoria del flujo debido a la topología dinámica.

Además, la violación de la calidad de servicio causada por la topología dinámica de la movilidad media y baja se recupera y varias capas de la red neuronal feedforward (MFNN) permite un rápido aprendizaje de las distintas operaciones realizadas por el núcleo y una mejor selección de clase de tráfico y la entrega de acuerdo a las necesidades de flujo.

4.3.1.1. Núcleo [7]

El núcleo del modelo proporciona funciones básicas de enrutamiento y control de calidad de servicio como: descubrimiento del trayecto y reserva de recursos, control de admisión, la adaptación de tráfico, detección, control de admisión, la adaptación de tráfico, detección y recuperación de violación de calidad de servicio.

En la *Figura 26*, se muestra el diagrama esquemático del núcleo basado en SWAN. Como se puede observar está compuesto por: el control de admisión de múltiples criterios de eficacia estima el ancho de banda disponible local en cada nodo, determina si un nodo es egoísta o está dispuesto a colaborar con la transmisión de paquetes con sus nodos vecinos, y proporciona información precisa sobre el estado de ancho de banda y la existencia de nodos egoístas para el aislamiento, retardo de extremo a extremo y número de saltos de las rutas descubiertas entre la fuente y el destino. La decisión de admitir a un nuevo flujo se realiza por este mecanismo.

I. ESQUEMA DE ENRUTAMIENTO

Las principales operaciones del sistema de enrutamiento son el descubrimiento ruta de acceso y reserva de recursos.

Un estado del nodo en este esquema puede adoptar cualquiera de los siguientes estados: comprobar el descubrimiento de ruta, reservación recursos, asignación en la recepción de paquetes de datos, liberación de recursos que no se utilizan por la recuperación de la calidad del servicio como violación.

Proceso de descubrimiento de ruta:

Se realiza antes de que la fuente S transmita los paquetes de datos hacia su destino D.

- Inicia transmitiendo una demanda de la ruta (RREQ) de mensajes cortos por S a sus vecinos.
- Se propaga la petición de S a D, que lleva la información sobre el estado de flujo que sirve para calcular el retardo de extremo a extremo, ancho de banda y número de saltos para una ruta dada.
- El nodo intermedio lleva a cabo la política de control de admisión en el momento de recibir los paquetes RREQ (*Figura 33*).
- Si el RREQ es aceptado, se agrega al nodo una ruta de entrada en su tabla de enrutamiento con el estado de verificado y retransmite la solicitud al siguiente salto

Este proceso es válido por un periodo corto de tiempo. Si no llega a tiempo ningún paquete de respuesta a este nodo, la entrada de la ruta se eliminarán en el nodo y los paquetes de respuesta que viene después de este tiempo serán ignorados.

El formato de RREQ es identificado por (source_addr, secuencia, Broadcast_ID). Los campos hop_rep, del_req, y band_req indican respectivamente la cuenta de saltos requerida, el retraso y ancho de banda.

Estos campos son útiles para el proceso de control de admisión basado en número de saltos y / o retardo de extremo a extremo y / o ancho de banda disponible y/o nodos egoístas.

Los RREQs ya procesados son desechados por el nodo haciendo un seguimiento de identificador de RREQ, esto con la finalidad de limitar la duración y la longitud de descubrimiento de ruta, en términos de número de saltos, variable time_to_live (TTL), se decrementa en cada salto. Si la cuenta es atrás hasta cero, el RREQ se ha caído y no serán tratados posteriormente.

<i>packet_type</i>	<i>source_addr</i>	<i>dest_addr</i>	<i>sequence</i>	<i>Broadcast_ID</i>
<i>previous_addr</i>	<i>TTL</i>	<i>hop_req</i>	<i>del_req</i>	<i>band_req</i>

Figura 33. Formato RREQ

Proceso de Reserva de Rutas:

El nodo de destino inicia el proceso de reserva después de recibir el primer RREQ generando una reproducción de ruta (RREP) de mensajes cortos. RREP es unicasted en el origen de la ruta inversa.

El nodo intermedio que recibe un RREP (Figura 34), comprueba su disponibilidad de recursos y actualiza su estado de reserva de ruta si el paquete es aceptado.

Cualquiera de los nodos intermedios puede iniciar el proceso de reserva si se conoce la ruta hacia el destino con el número de secuencia correspondiente mayor o igual a la contenida en el RREQ, si se añade información a la ruta parcial en un RREP, y enviarlo de vuelta al de origen.

Los campos hop_rep, del_rep y band_rep correspondientes a los criterios de calidad del servicio se inicializan al comienzo del proceso de reserva, y se actualiza cuando un nodo intermedio recibe RREP.

Por lo tanto, el nodo de origen se detiene hasta recibir el RREP, para ver el estado de la red que le permite tomar una decisión acerca de la transmisión.

<i>packet_type</i>	<i>source_addr</i>	<i>dest_addr</i>	<i>sequence</i>
<i>hop_req</i>	<i>del_req</i>	<i>band_req</i>	

Figura 34. Formato RREP

II. POLÍTICA DE CONTROL DE ADMISIÓN DE MULTI-CRITERIOS [7]

El plan de control de calidad de servicio trata de encontrar las rutas que satisfagan al máximo las limitaciones entre la fuente S y el destino D.

Control de ancho de banda:

El control ancho de banda es de gran utilidad para la optimización de los recursos que utilizan.

Proceso de cálculo de ancho de banda disponible (β) en un nodo i está en función de la capacidad de ancho de banda en el nodo BC_i y el tráfico en este nodo T_i .

Definición del dominio de ancho de banda $\varphi(i)$ de un nodo i como el dominio que abarca el primero y segundo vecino de i :

$$\varphi(i) = \left\{ \frac{T_{ij}}{\omega(ij)}, \omega(\omega(i, k), j) \right\}$$

Cuando $\omega(i, j)$, $\omega(\omega(i, k), j)$ son las relaciones de vecinos definen como:

$$\omega(i, j) = (i \text{ tiene un enlace con } j)$$

El ancho de banda disponible en el nodo β_i me viene dado por (1):

$$\beta_i = BC_i - \sum_j T_{ij}, \forall T_{ij} \in \varphi(i) \quad (1)$$

Por lo tanto, el ancho de banda disponible en el nodo i depende del tráfico generado en este nodo, el tráfico que viaja a través de este nodo, y el tráfico generado en los nodos vecinos. Usamos (1) para determinar la ruta que ofrece el ancho de banda posible entre los caminos n .

Suponiendo que $N_i\beta_k$ es el ancho de banda en el nodo k en el camino l, entonces (2) da el ancho de banda mínimo disponible dentro de cada ruta.

$$X_b^{(n,1)} = \begin{cases} \text{MIN}(N_1b_0 + \dots + N_1b_k + \dots + N_1b_m) \\ \text{MIN}(N_ib_0 + \dots + N_ib_k + \dots + N_ib_m) \\ \text{MIN}(N_nb_0 + \dots + N_nb_k + \dots + N_nb_m) \end{cases} \quad (2)$$

$$\delta_{B^1} = \text{MAX}(X_b^{(n,1)}) \quad (3)$$

δ_{B^1} denota el ancho de banda de la ruta que estamos buscando.

Que band_rep ser el valor del ancho de banda requerido en el mensaje RREQ llegar al nodo k en el camino de l.

La ecuación (3) se resuelve mediante el siguiente proceso: Durante el viaje de los RREQ lo largo de una ruta de acceso l, cada nodo k calcula MIN (band_req, $N_i\beta_k$) y toma una decisión que transmita RREQ si el ancho de banda solicitado está disponible, de lo contrario la solicitud se descarta.

Al recibir cada RREQ, el destino se inicializará al azar el campo band_rep a un valor superior al valor de ancho de banda. A continuación, un RREP se genera y envía de nuevo a la fuente.

En cuanto es recibido RREP por un nodo intermediario k en un ruta l, el cálculo del MIN (band_rep, $N_i\beta_k$) se realiza y se almacena en la tabla de enrutamiento del nodo y luego se ajusta el nuevo RREP generado.

Al recibir el primer RREP por el nodo origen, se activa un algoritmo de ordenación para calcular el ancho de banda máximo, y se realiza en la recepción de otros paquetes de la ruta de respuesta.

δ_{B^1} es el valor máximo de ancho de banda entre los n RREPs recibidos. En punto P1 señala la ruta de acceso asociada.

Ahora tenemos que identificar el intervalo de garantía de clase de servicio (GISC). Para tal fin, se extrae el segundo de ancho de banda posible δ_{B^2} usando (4).

$$\delta_{B^2} = \text{MAX}(X_b^{(n,1)} - \delta_{B^1}) \quad (4)$$

$GI(B^1) = [\delta_{B^2}, \delta_{B^1}]$ Es el primer intervalo de garantía para la restricción de ancho de banda, el camino viable asociados se observa P1.

De igual manera se realiza para identificar el segundo GSIC:

$$\delta_{B^3} = MAX(X_b^{(n,1)} - \delta_{B^2}) \quad (5)$$

$GI(B^2) = [\delta_{B^3}, \delta_{B^2}]$ Representa el segundo intervalo de garantía, con el P2 es el camino viable asociados.

Podemos calcular otros intervalos de garantía de la siguiente manera:

$$\delta_{B^k} = MAX(X_b^{(n,1)} - \delta_{B^{k-1}}) \quad (6)$$

y P^k es la ruta generada por δ_{B^k} , que será la ruta de acceso asociados a $GI(B^{k-1})$.

Control de Retardo extremo a extremo:

Lo que se busca a través de este proceso es obtener las rutas que ofrezcan un mínimo retardo entre la fuente S y el destino D.

Inicialmente se supone que $N_i d_j$, es el retardo en el nodo j en la ruta i, que es el tiempo requerido por j para recibir y procesar los paquetes que se detallan en la ecuación (7).

A continuación, el plazo mínimo de extremo a extremo entre S y D es $MIN(X_d^{(n,1)})$, y la ruta asociada será la ruta óptima en términos de retraso.

$$X_d^{(n,1)} = \begin{cases} del_{path} = (N_1 d_0 + \dots + N_1 d_k + \dots + N_1 d_m) \\ del_{path} = (N_i d_0 + \dots + N_i d_k + \dots + N_i d_m) \\ del_{path} = (N_n d_0 + \dots + N_n d_k + \dots + N_n d_m) \end{cases} \quad (7)$$

Con el desarrollo del esquema de enrutamiento no se trata de solo encontrar el menor tiempo de retraso extremo a extremo, sino también los otros valores de retraso para poder construir los intervalos de clases de servicio.

Al recibir una petición de ruta por el nodo j, y antes de enviarlo al nodo j +1, el cálculo $(del_{req} - N_i d_j)$ se lleva a cabo, y se almacena en la tabla de enrutamiento del nodo. Luego se ajusta el nuevo RREQ generado que será enviado al siguiente nodo $(del_{req}$ es el retraso solicitado). Este cálculo es necesario para el nodo para que pueda tomar una

decisión sobre la satisfacción de las necesidades retraso. Si $(\text{del_del_req} - N_i d_j) < 0$, entonces la petición de ruta se descarta.

Una vez que el destino D recibe una solicitud, se inicializa el cálculo ruta de acceso donde se pone $\text{del_rep} = 0$, y luego $(\text{del_req} + N_i d_j)$ se calcula en cada nodo k intermedios en el camino de vuelta a la fuente S, almacenado en la tabla de enrutamiento del nodo i, se envía a repetir para una nueva ruta.

Usando el algoritmo de clasificación utilizado en el kernel NNMS la demora en la determinación de una ruta en el nodo fuente es mínima. Entre algunos n del_rep recibidos, se realiza la selección del menor plazo posible, señalado en δ_{D^1} , el segundo retraso mínimo δ_{D^2} , el tercero δ_{D^2} , etc. En la ecuación (8) MIN^k da un mínimo de orden k.

$$\delta_{D^k} = \text{MIN}^k(X_d^{(n,1)}) \quad (8)$$

δ_{D^1} es el menor plazo posible, teniendo a P1 como la ruta de acceso asociados. Otros mínimos se determinan de la siguiente manera:

$$\delta_{D^2} = \text{MIN}(X_d^{(n,1)} - \delta_{D^1}) \quad (9)$$

$GI(D^1) = [\delta_{D^1}, \delta_{D^2}]$, define el primer intervalo de garantía para la restricción de demora, y la ruta de acceso asociado es P1.

Del mismo modo, se identifican otros intervalos de garantía:

$$\delta_{D^k} = \text{MIN}(X_d^{(n,1)} - \delta_{D^{k-1}}) \quad (10)$$

Donde la ruta de acceso generados por δ_{D^k} , P^k , se asocia con la $GI(D^{k-1})$ definido en la ecuación (11):

$$GI(D^{k-1}) = [\delta_{D^{k-1}}, \delta_{D^k}] \quad (11)$$

Control de cuenta de saltos:

El cálculo de los caminos viables basado en número de saltos y sus intervalos de garantía correspondiente es similar al cálculo del retardo de extremo a extremo, porque el control de número de saltos es también una métrica aditivo que ha de ser mínimo.

Control de nodos egoístas:

Adicionalmente a lo propuesto en [7] en la fase de control de admisión se propone el control de nodos egoístas dentro de la conformación de la red para reservación de recursos y asignación de la mejor ruta a estos nodos.

El control de nodos egoístas se lo puede realizar siguiendo el mismo procedimiento de la sección 4.2.1, partiendo de la *Figura 32*.

III. Garantía De Intervalos

Los intervalos de garantía de clase de servicio (GISC) son usados por el nodo origen para seleccionar la ruta de acceso adecuada para la transmisión de datos. Los GISC se componen de los valores del intervalo y sus caminos viables asociados.

Si la calidad de servicio requerida se encuentra dentro del intervalo de GISC, a continuación, esta clase se consideran adecuada y también para ser la mejor. De lo contrario, si los requisitos de flujo son menores que el valor inferior del intervalo de algunas clases de servicios disponibles, todas estas clases se consideran adecuadas, y el protocolo selecciona una clase de servicio adecuada sobre la base de formación de algunos criterios.

Criterios de selección de una clase de servicio adecuada:

- a. La categoría adecuada con el mayor intervalo de menor valor
- b. La clase adecuada con el menor intervalo de menor valor
- c. Simplemente una clase adecuada al azar.

Teorema: Sea considera A_1, A_2, \dots, A_n , los caminos de confianza de un nodo fuente S al nodo de destino D y $f_m(A_1), f_m(A_2), \dots, f_m(A_n)$ corresponde a las funciones de medición en términos métricas m de calidad de servicio. Entonces, el problema de determinar el camino que ofrece un mejor rendimiento dentro de métricas de garantía de intervalo $[\alpha, \beta]$ es un problema solucionable.

Prueba: Podemos decir que A_j tiene mejor rendimiento que A_i para el m métricas con parámetro S si $f_m(A_i) < f_m(A_j)$: $\int_{\alpha}^{\beta} [f_m A_i(s)]' ds / (\beta - \alpha)$ es la derivada de $f_m(A_i)$ durante el intervalo $[\alpha, \beta]$.

Entonces, la comparación de la calidad del servicio y la satisfacción de A_i A_j más $[\alpha, \beta]$ viene dada por la ecuación (12):

$$\int_{\alpha}^{\beta} \frac{(f_m(A_i(s)) - f_m(A_j(s)))' ds}{(\beta - \alpha)} = \frac{(f_m(A_i(\beta)) - f_m(A_j(\beta))) - (f_m(A_i(\alpha)) - f_m(A_j(\alpha)))}{(\beta - \alpha)} \quad (12)$$

A_j ofrece mejor rendimiento que A_i si:

$$(f_m(A_i(\beta)) - f_m(A_i(\alpha))) < (f_m(A_j(\beta)) - f_m(A_j(\alpha)))$$

$$\text{Permitir } \Omega(A_i) = f_m(A_i(\beta)) - f_m(A_i(\alpha))$$

$$\Omega(A_j) = f_m(A_j(\beta)) - f_m(A_j(\alpha))$$

Entonces,

A_i tiene mejor rendimiento que si A_j if: $\Omega(A_i) > \Omega(A_j)$

A_i tiene menos rendimiento que si A_j if: $\Omega(A_i) < \Omega(A_j)$

A_i tiene el mismo rendimiento que si A_j if: $\Omega(A_i) = \Omega(A_j)$

IV. Prioridad haciendo cola

El esquema de prioridades es una de las herramientas utilizadas para lograr una diferenciación de servicios en las redes inalámbricas ad hoc.

El modelo SWAN utiliza la herramienta de prioridad de cola mediante la limitación de la cantidad de tráfico en tiempo real con el fin de procesar los paquetes de menor prioridad. FQMM modelo utiliza una cola simple prioridad para garantizar que los paquetes de alta prioridad se procesan antes que los de baja prioridad. NNMS utiliza una cola separada para cada clase.

Los GISC comparten un espacio de búfer, cuando el buffer no está ocupado por una clase de servicio, se le asignan a otras clases si es que fuera necesario.

Mediante el uso priorización de colas, la predicción y la prevención de la congestión se hace posible a nivel del nodo: antes de ocupar todo el espacio de búfer, el nodo informa

a sus vecinos para frenar el tráfico de la hora de enviar paquetes ACK (la recepción de los paquetes ACK se recibió con éxito).

La disminución de las tasas (RD) del tráfico se calcula de acuerdo a la prioridad de cola del Estado (pequeños o grandes congestión) del nodo con el envío del ACK. En consecuencia, la tasa de nuevos tráficos (NT) de transmisión sería:

$NT = OT - (OT * RD) / 100$, donde OT es el tiempo de la tasa de tráfico. Por lo tanto, el tráfico se adapta, según el estado de la red.

V. Detección y Recuperación de violación de QoS

En las redes ad hoc, los mecanismos que permiten la detección y recuperación de violación de calidad del servicio son de vital importancia para apoyar las aplicaciones multimedia.

Detección De Qos Violación:

Para resolver este problema, se proponen dos métodos:

- El primero se basa en diferentes estados de reserva de recursos en el nodo intermedio
- Segundo, que se basa en el momento de la recepción de paquetes de datos.

Reserva de Recursos Soft-states para la detección de violación QoS:

Los recursos disponibles en un nodo intermedio puede tomar cualquiera de los siguientes estados: marcado, reservado, asignado (en la recepción de paquetes de datos), y liberado (una vez que los recursos no se utilizan).

Cada estado tiene un intervalo corto llamado "intervalo de soft-state" (*Figura 35*). Durante el proceso de enrutamiento, los recursos disponibles se reservan temporalmente en un corto período de tiempo δ . Cuando un nodo intermedio recibe un paquete de datos que asigna los recursos reservados en el siguiente intervalo. Por lo tanto, si un paquete nuevo no se recibe dentro del intervalo de soft-state, entonces los recursos se liberan y el estado del flujo se elimina de manera totalmente descentralizada.

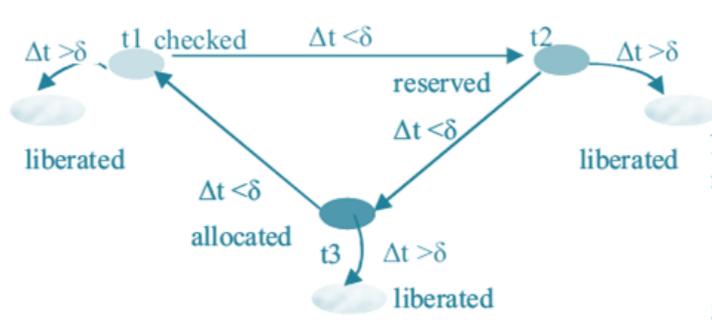


Figura 35. Soft-states de un nodo intermedio [7]

Vamos a considerar:

Δt el intervalo de soft-states (diferencia de tiempo entre dos estados i y j) de un nodo intermedio.

μ el deseo de satisfacción de destino define el porcentaje deseado de paquetes que se envían dentro de la restricción de QoS, donde $\mu = 1$ corresponde a la mejor garantía y calidad de servicio $\mu = 0$ corresponde a la transmisión de mejor esfuerzo.

En la ecuación (13) se verifica la probabilidad de que Δt es menor que el tiempo requerido y el valor de δ de solicitud de sesión para ser aceptado.

$$P[\Delta t \leq \delta] \geq \mu \quad (13)$$

La evaluación exacta de la ecuación (13) requiere el destino que desea conocer la descripción estadística de retraso de cada nodo a lo largo del camino.

Sin embargo, en muchos casos, la distribución estadística de dicho parámetro se puede aproximar mediante una distribución de Gauss.

Según esta hipótesis, y suponiendo independencia estadística entre los nodos, el intervalo de soft-states resulta ser una variable gaussiana. Que m_{sp} y σ_{sp}^2 igual a la media estadística y la varianza de la variable aleatoria Sp , respectivamente (Sp es el momento de la soft-state de un nodo dado).

Por lo tanto, las estadísticas intervalo de soft-estados se puede expresar de la siguiente manera:

$$P[\Delta t \leq \delta] = 1 - Q\left(\frac{\delta - m_{sp}}{\sigma_{sp}}\right) \quad (14)$$

Donde Q representa la función de distribución complementaria de la variable gaussiana con media 0 y varianza 1.

Vamos a considerar δ y μ es el tiempo máximo y el objetivo la satisfacción de una solicitud de sesión aceptada.

Que v cumpliera el tiempo real proporcionado por el nodo intermedio dado por la ecuación (14). Por lo tanto, un flujo de solicitudes de QoS se cumplirá incluso el intervalo de estado medio suave, incrementando a m_{Sp} valor dado por:

$$m_{Sp} = \delta - \sigma_{Sp} Q^{-1}[1 - v] \quad (15)$$

Como resultado, la violación de QoS se detecta en un nodo intermedio, si el tiempo soft-state es mayor que m_{Sp} (m_{Sp} es el plazo determinado del intervalo soft-estado).

Esta herramienta de detección violación QoS también es útil para optimizar el uso de ancho de banda disponible, ya que permite liberar el ancho de banda reservado si no se asigna a ninguna de flujo.

Retraso de detección de violación de QoS:

Usamos este mecanismo para detectar la violación QoS durante la transmisión de paquetes de datos. El proceso de recuperación se dispara automáticamente cuando el tiempo estimado para recibir un paquete es superior. Este exceso es causado ya sea por vínculos rotos o congestión como resultado del aumento del tráfico en la red.

Vamos a designar:

T el tiempo necesario para generar un RREQ en el nodo i , del_node es el retraso en curso disponible en su tabla de enrutamiento.

ΔT_T el tiempo estimado de procesamiento de paquetes en el nodo origen.

ΔT_G el tiempo transcurrido entre el envío de dos paquetes de datos consecutivos por la aplicación que genera el tráfico.

Ahora, se puede calcular el tiempo estimado $TECX$ para la recepción de paquetes j por nodo i .

$$TEXC(j) = T_i + \Delta T_T + \Sigma^{j-1} \Delta T_G \quad (16)$$

Por lo tanto, la violación de QoS se puede detectar fácilmente en un nodo dado por el control de retraso de los paquetes de datos que llegan como se expresa en expresa en la ecuación (16).

Si un nodo recibe paquetes de datos j cuyo retraso es superior al tiempo máximo requerido $TEXC(j)$, la violación de calidad del servicio por tiempo excedido es detectada y el mecanismo de recuperación de calidad de servicio puede dispararse.

Recuperación de QoS violación:

La priorización de cola es un medio eficaz para prevenir la violación de QoS causado por la congestión de la red. Esto se logra al disminuir la tasa de tráfico cuando se detecta un exceso de tráfico.

Por otra parte, en NNMS se aplica un mecanismo de recuperación que se puede utilizar para volver a establecer la ruta a lo largo de un nuevo camino cuando una ruta de acceso está rota. El enfoque se adapta las rutas de enrutamiento en función del estado nueva red causada por la movilidad de los nodos o estado ruta degradada. Cuando se detecta violación QoS en un nodo x , un nodo vecino z enviará un mensaje de actualización corto hasta el destino D . A continuación, D emitirá de nuevo a la fuente S un mensaje de actualización.

Los cálculos igual que en el proceso de descubrimiento de ruta en términos de control de admisión se llevarán a cabo durante el viaje de los mensajes de actualización a S .

Se evitan los bucles al disminuir el valor del número de secuencia del nodo z antes de iniciar el proceso de recuperación.

Después de haber descrito las diferentes funciones del núcleo, vamos a presentar en lo sucesivo, la aplicación de redes neuronales multicapa feedforward al núcleo NNMS.

4.3.1.2. Plan de Aprendizaje Inteligente

Dentro de este plan se garantiza el aprendizaje a partir de las operaciones realizadas por el kernel de NNMS usando una herramienta de inteligencia artificial específicamente redes neuronales multicapa.

La herramienta de redes neuronales es una tecnología robusta inteligencia artificial que puede manejar una naturaleza compleja y dinámica de los distintos procesos

Una de las características funcionales principales de las redes neuronales es su capacidad de distribución debido a su arquitectura especial paralelo.

Se posee varias entradas y trata el problema en forma paralela. Uso de la forma adecuada de los datos de formación hace que el tiempo de procesamiento sea independiente de la carga de tráfico en la red.

En la presente investigación se utiliza las redes neuronales feedforward para la optimización de las decisiones del núcleo NNMS según es estado adaptativo de la red.

Redes neuronales multicapa Feedforward (MFNN) y su aplicación a NNMS [7]

La red está dividida en capas: la capa de entrada que se compone de las entradas a la red; la capa oculta que constará de un número de neuronas colocadas en paralelo, en cada neurona realiza una suma ponderada de las entradas, y a través de la capa de salida se genera a transferencia de funciones de salida.

Con el fin de asegurar el comportamiento adaptativo de la red, la correlación entre los niveles sucesivos de las neuronas se cambia cuando los valores de los pesos son modificados. El proceso de la conducta adaptativa que se conoce como formación.

La capacitación es el proceso de estimación de los pesos de conexión, minimizando algunas medidas de error global, utilizando mecanismos tales como algoritmo de propagación (BP (Back Propagation)). BP es un algoritmo de entrenamiento ampliamente utilizado en redes neuronales. Detalles matemáticos relacionados con la BP se describen en muchas redes neurales, como en la referencia [21].

Con el fin de capacitar a una red neuronal para realizar alguna tarea, los pesos de cada unidad se deben ajustarse de tal manera que el error entre la salida deseada y la producción real se reduce.

Este proceso requiere calcular el error derivado de los pesos de una red neuronal. El algoritmo de retropropagación es ampliamente utilizado para determinar el error llamado MSE (error cuadrático medio).

Se calcula como la diferencia media cuadrática entre la salida deseada y la producción real. Dado que buscamos minimizar el error para cada peso por separado, el error global puede aumentar. Luego, debemos calcular el error de salida total después de cada adaptación, cuando el error es mayor que el error anterior es rechazado, después de que calculamos las nuevas tasas de aprendizaje.

La modificación de los pesos continúa hasta que la producción real se aproxima a la respuesta deseada. El problema de la convergencia de mucho tiempo de BP se resuelve mediante la incorporación de la expresión dinámica del peso de la BP.

Aplicación de MFNN (Redes Neuronales Multicapa Feedforward)

Un diagrama esquemático de las operaciones NNMS se presenta en la *Figura 27*. Se muestra la relación entre las operaciones del núcleo NNMS y su proceso de aprendizaje mediante redes neuronales.

Las operaciones (es decir, las funcionalidades descritas en la *Figura 25* y *Figura 26*, tales como control de admisión, etc) están representados por cuadrados, y la comprobación del éxito de las operaciones está representado por rombos ('Y' y 'N' significan, respectivamente, el éxito y el fracaso de la operación).

El perceptrón multicapa o MLP (Multi-Layer Perceptron) con dos capas de neuronas ocultas es capaz de discriminar regiones de forma arbitraria, además se suele entrenar por medio de un algoritmo de retropropagación de errores o BP (Back Propagation), a esta también se la conoce como red de retropropagación.

MFNN sobre GQoS se divide en dos fases.

- La primera fase es la formación de la red neuronal con los datos de conjunto de entrenamiento y la prueba de la que después del entrenamiento.
- La segunda fase es la predicción y la selección de la clase de servicio y del siguiente salto para reenviar los paquetes de datos.

Para el desarrollo de las dos fases mencionadas se sigue el siguiente algoritmo:

1 La creación de la red neuronal.

2 La división de los datos de entrenamiento en grupos al azar

3 La formación de la red neuronal con cada conjunto de datos y en consecuencia que recibimos la salida de la predicción.

4 La prueba de cada salida de predicción por el conjunto de datos experimental y la selección de la salida que tiene el máximo nivel de exactitud.

5 La predicción de la clase de servicio potencial y el siguiente salto para reenviar los paquetes de datos de las pruebas de conjunto de datos por la mejor salida.

Los pasos consisten en 1-4 de la primera fase, el paso 5 es la etapa de fase.

Una vez que la red se aprende (pasos 1-4) con los datos ofrecidos por los mecanismos del núcleo, sería posible seleccionar el servicio adecuado y próximo salto (paso 5) con el fin de satisfacer las necesidades de flujo. La ventaja de esta técnica como se muestra en la simulación es la reducción del tiempo requerido por los diferentes dispositivos del núcleo NNMS.

Como se muestra en la *Figura 27*, la etapa de entrenamiento de la red se realiza después de varios escenarios de la ejecución de las operaciones del núcleo, como proceso de descubrimiento, el proceso de reserva, la adaptación del tráfico a través de colas de prioridad, la detección y recuperación de violación QoS, etc.

Una vez que la etapa de formación del kernel se logra, NNMS inicia la operación de las pruebas sobre escenarios adicionales para probar la exactitud del resultado de la primera etapa. El éxito de la red de aprendizaje sería capaz de satisfacer las necesidades de flujo sin realizar las operaciones requeridas por el núcleo.

Se utiliza un modelo MFNN de tres capas. Para el entrenamiento de la red neuronal se ha utilizado el algoritmo de capacidad de recuperación posterior formación de propagación.

La función de rendimiento es del error cuadrado, que se ha utilizado para calcular las diferencias entre la salida de la meta y la salida de la red.

Para las neuronas de entrada y las capas ocultas se ha utilizado la función de transferencia tan-sigmoid. La neurona de la última capa tiene la función de transferencia lineal.

Las dimensiones de los patrones de entrada que representan a los requisitos de QoS (retardo, ancho de banda y número de saltos requeridos por el usuario) especifica el número de nodos de entrada.

En cuanto al número de nodos en su capa oculta, de acuerdo con la teoría de Kolmogorov [20], si el número de nodos de entrada es N , entonces $2N+1$ un nodo oculto puede ser utilizado.

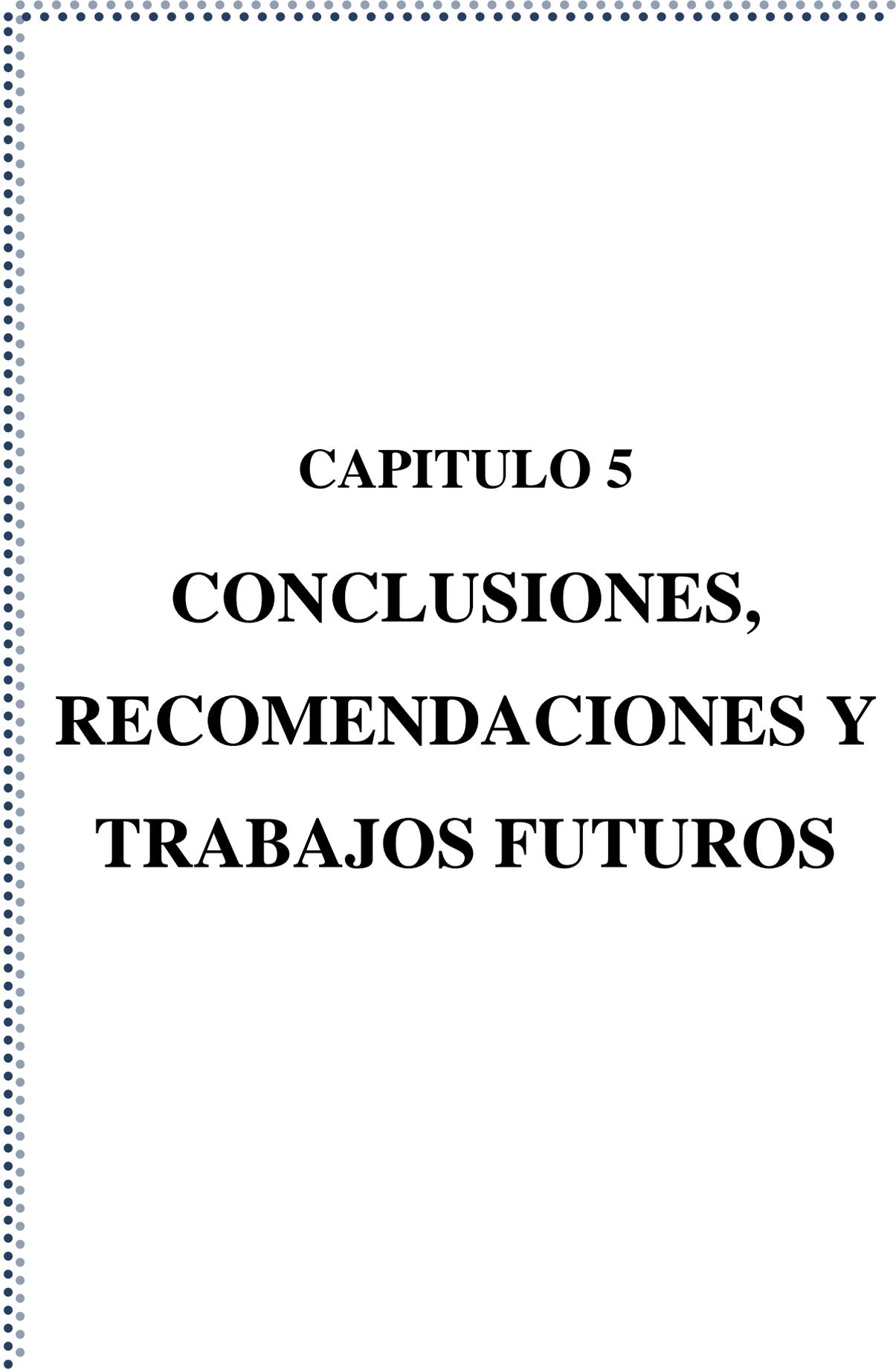
Los nodos de salida se determinan por el número de categorías que se clasifican. En nuestro caso, podemos clasificar tres categorías que se corresponden con el tráfico con limitación demora, el tráfico con limitación de ancho de banda y el tráfico de mejor esfuerzo. Para cada categoría, MFNN da el siguiente salto para el encaminamiento del tráfico recibido.

Se opta por una clasificación binaria, entonces cada bit de una salida deseada presenta el estado 1 o el estado 0: estado 1 significa "pertenecer a" y el estado 0 significa "no pertenece a" una categoría determinada (por ejemplo, 001 significa que la deseada patrón de producción pertenecen a la categoría 1).

4.4. CONCLUSIÓN:

Con la modelación analítica de la propuesta se busca dar una idea de cómo realizar la simulación del modelo a futuro.

A partir de la modelación analítica se puede concluir que trae consigo diversos beneficios como asignación de recursos y de la mejor ruta a cada nodo, ocasionando así disminución de retardo extremo a extremo en la transmisión de paquetes.



CAPITULO 5

**CONCLUSIONES,
RECOMENDACIONES Y
TRABAJOS FUTUROS**

5. CONCLUSIONES, RECOMENDACIONES Y TRABAJOS FUTUROS

5.1. CONCLUSIONES

A partir del modelo propuesto para Redes Ad Hoc basado con técnicas de clustering y el emplear técnicas en base a Redes Neuronales y Teoría de Juegos, se puede concluir teóricamente lo siguiente y afirmar las hipótesis planteadas:

- Se puede determinar teóricamente cuál es el siguiente salto sin necesidad de empezar el proceso de descubrimiento de ruta desde el inicio, esto a través de la información que es manejada por el nodo JC incluido el proceso en base a redes neuronales a través de un aprendizaje inteligente.
- Dentro de algunas investigaciones los autores [1][7], nos muestran que las redes neuronales feedforward y el dilema del prisionero se pueden emplear dentro de redes Ad Hoc, redes neuronales para mejorar la calidad y servicio de la red y teoría de juegos en la determinación de nodos egoístas dentro de la red.
- En la formación del cluster se puede determinar si un nodo es egoísta o no, ya sea por falta de recursos o por ahorro de energía, esto a partir de la ejecución del algoritmo WCA el cual recoge las características de los recursos de los nodos y la implementación de teoría de juegos.
- La propuesta de teoría de juegos en la fase de control de admisión dentro de NNMS ayudará a determinar si un nodo es egoísta o no, en el caso de no ser un nodo egoísta se admite el flujo y pasa a la siguiente fase, y si es un nodo egoísta el flujo se rechaza.
- Para que un nodo forme parte del cluster debe de tener los recursos necesarios, debido a que si no poseen los recursos necesarios este no puede ayudar al paso de paquetes y se lo declararía como nodo egoísta por falta de recursos a través del dilema del prisionero.
- La utilización del dilema del prisionero iterado dentro del modelo propuesto brinda la oportunidad a los nodos egoístas de reivindicarse en su decisión de no formar parte del cluster en caso de que llegara a necesitar ser parte del cluster para enviar

algún paquete.

- Reduce el tiempo de transmisión de paquetes y ahorro de energía de los nodos al no colaborar con nodos egoístas.
- El modelo propuesto puede ser comprobado a partir de la modelación matemática planteada mediante una simulación, que se la puede realizar en trabajos futuros.

A continuación se presenta las conclusiones generales obtenidas del trabajo de investigación sobre redes Ad Hoc:

- La gestión de redes tiene como objetivo principal permitir la administración, supervisión y control de la red, lo cual es posible mediante la implementación de herramientas, aplicaciones y dispositivos dedicados a la vigilancia con el fin de ayudar a los administradores al mantenimiento y control de recursos de la red
- La heterogeneidad de los equipos y el crecimiento de los nodos dentro de la red han hecho que los modelos de gestión avancen cada vez más con el fin de mejorar la administración de la red.
- Uno de los modelos más idóneos dentro de la gestión de redes es el SNMP, debido a que administra cuatro componentes como: nodos, estaciones, información y protocolo.
- La gestión de redes fijas es diferente a la gestión de redes Ad Hoc, debido a diferentes parámetros como los protocolos de enrutamiento, a la seguridad que se le debe prestar en las MANET ya que son más propensas a recibir ataques por personas ajenas a la red.
- La infraestructura descentralizada y la jerarquía de los nodos cambian continuamente lo que complica de cierta manera a la gestión de redes MANET.
- Los protocolos de enrutamiento son diferentes para las redes fijas y para las redes MANET sin embargo se puede destacar que los protocolos de redes MANET se basan en el protocolo SNMP que es para redes fijas.

- La gestión de este tipo de redes debe ser controlada de manera estricta ya sea por políticas de seguridad o por agentes que estén inmersos en el mundo de las redes ad hoc, esto con el fin garantizar un mejor funcionamiento y administración de recursos de la misma.
- Emplear herramientas como redes neuronales y teoría de juegos resultan muy factibles al momento de crear un nuevo modelo orientado a la gestión.
- Luego de haber investigado el fundamento teórico de cómo están conformados los agentes inteligentes, las Redes Neuronales y teoría de juegos dentro de las MANET, se puede empezar un nuevo estudio orientado a dar solución a los diferentes problemas que se presentan en la Gestión de Redes Móviles.
- La gestión de MANET basada en políticas combinado con aprendizaje inteligente en redes neuronales y teoría de juegos, resultan ser técnicas innovadoras para futuras investigaciones con el fin de garantiza una optima administración de la red.

5.2. Recomendaciones

A partir de la investigación realizada se puede recomendar lo siguiente:

- Estudiar a fondo el fundamento teórico de gestión de redes ad hoc, redes neuronales y teoría de juegos para poder entender el modelo propuesto.
- Identificar bien los escenarios en los que se vaya a integrar redes neuronales y teoría de juegos.
- Dentro de redes neuronales se considera como mejor alternativa para implementar dentro de la investigación las redes neuronales multicapa feedforward, debido a que gracias a que posee varias capas y es de tipo cuadrática podemos obtener un mejor aprendizaje.
- Si se desea simular el modelo propuesto estudiar bien la factibilidad de las herramientas existentes para este tipo de simulaciones, eligiendo bien los escenarios que se va a demostrar, a fin de no cambiar de herramienta en el transcurso de la simulación.

- NS2, es una buena herramienta que se puede utilizar esta es utilizada de manera experimental en la simulación de redes móviles para fines académicos e investigaciones.
- Se recomienda utilizar las versiones actuales de la herramienta por las funcionalidades que cada vez se incrementan y obtener mejores resultados obtenidos.

5.3. Trabajos Futuros

Existen algunos trabajos que se pueden desarrollar a futuro con el fundamento del modelo propuesto con respecto al enrutamiento y retardo extremo a extremo, y control de nodos egoístas:

- Implementar redes neuronales y teoría de juegos en otras interfaces de comunicación.
- Adherir redes neuronales en la elección del cluster cabeza partiendo de lo planteado de lo planteado con WCA y teoría de juegos.
- Definir escenarios de propuesta y simulación más complejas y con otro tipo métodos de teoría de juegos y la utilización de otro tipo de redes neuronales.
- Realizar la simulación del modelo planteado con la finalidad de legalizar su factibilidad.
- Analizar el retardo extremo a extremo de los paquetes con la implementación de aprendizaje inteligente frente a otros modelos de gestión de QoS, esto a través de los resultados obtenidos en la simulación.
- Definir un modelo que presente mayor movilidad en sus nodos, planteándose los mismos retos de implementación de redes neuronales y teoría de juegos.

BIBLIOGRAFÍA:

- [1] Alzate Marco, M. A. (2009). Ingeniería de sistemas complejos aplicada a redes de comunicación: Emergencia de cooperación en redes móviles Ad Hoc. Bogota.
- [2] BASU, P., & KHAN, N. y. (2001). A mobility metric for clustering in mobile ad hoc networks. Arizona, USA: IEEE .
- [3] BETTSTETTER, C. (2004). The cluster density of a distributed clustering algorithm in Ad Hoc networks. IEEE. Paris, France.
- [4] Bao, L., and Garcia-Luna-Aceves, J. J. Topology Management in Ad Hoc Networks. In Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc 03) (New York, NY, USA,(2003), ACM, pp. 129–140.
- [5] Carmen, D. A. (2005). *Diferenciación de servicios y mejora de supervivencia en redes Ad Hoc*. Madrid.
- [6] Caryuly, R. B. (2005). *Protocolo SNMP (Protocolo simple de administración de redes)*. venezuela.
- [7] Cherkaoui, L. K. (2009). *Toward Neural Networks Solutios for Multimedia Support in Mobile Ad Hoc Networks*. Recuperado el 8 de 03 de 2010
- [8] Chien-Chung Shen, C. J. (s.f.). *The Guerrilla Management Architecture for Ad Hoc Networks*. Recuperado el 28 de 01 de 2010, de <http://eprints.kfupm.edu.sa/23709/1/23709.pdf>
- [9] Cisco. (04 de 03 de 2010). Network Management Basics. <http://www.cisco.com/en/US/docs/internetworking/technology/handbook/NM-Basics.pdf>.
- [10] Correa Montes, Blanca Alicia; Ospina Alzate, Laura Elena, “Análisis de procesos de clustering en modelos de movilidad de redes móviles Ad Hoc”, Universidad Pontificia Bolivariana, 2007, <http://eav.upb.edu.co/banco/files/04CAPITULOS.pdf>
- [11] David J. Hughes, W. Z. (s.f.). Minerva: An event based model for extensible network management. Australia.
- [12] E, I. (01 de 03 de 2010). Remote Monitoring. Network Cisco.
- [13] Eduardo, M. R. (2009). *Prestaciones de las Redes Inalámbricas Ad Hoc: Teoría a través de capas*. Madrid-España.

- [14] *EUMED*. (s.f.). Recuperado el 12 de 07 de 2010, de <http://www.eumed.net/coursecon/juegos/index.htm>
- [15] Francisco, R. (Diciembre de 2004). *Interconexión a Internet para Redes Móviles Ad Hoc Híbridas*. Murcia.
- [16] Gerla, M. y. (s.f.). *Multicluster, mobile, multimedia radio network*. USA.
- [17] Grigoras Dan, Riordan Mark, Cork, Ireland, 2005. *Service Driven Mobile Ad Hoc Networks Formation and Management*, IEEE Computer Society.
- [18] Guerrero, M. (09 de 05 de 2007). *Securing and Enhancing Routing Protocols for Mobile Ad Hoc Networks*.
- [19] *Inteligencia en Telecomunicaciones*. (2010). Recuperado el 01 de 04 de 2010, de http://www.teleco.com.br/tutoriais/tutorialprotocolo/pagina_2.asp
- [20] Juan, B. R. (s.f.). *ECPUNR*. Recuperado el 29 de 07 de 2010, de Teoría de Juegos: http://www.ecpunr.com.ar/Docs/Teoria_de_Juegos%20II.pdf
- [21] K.A. Jain, J. M. (1996). *Artificial Neural Networks*. IEEE Computer.
- [22] Mainak Chatterjee, S. K. *WCA: A Weighted Clustering Algorithm form Mobile Ad Hoc Networks*. Texas-Arlington.
- [23] Marti, J. G. (s.f.). *Arquitectura de gestión de res basada en políticas para un ambiente Integrado 3G-WLAN*. Barcelona.
- [24] R, Hernando, “Gestión de redes”, 2002, <http://www.rhernando.net/modules/tutorials/doc/redes/Gredes.html>
- [25] Raquel Solano, R. T. (2010). *Contribución a la Gestión de Redes Ad Hoc*. Loja: UTPL.
- [26] Raspeño, J. B. (29 de 07 de 2010). *Teoría de Juegos*. Obtenido de http://www.ecpunr.com.ar/Docs/Teoria_de_Juegos%20II.pdf
- [27] Ritu Chadha, Y.-H. C.-W. (s.f.). *POLICY-BASED MOBILE AD-HOC NETWORK MANAGEMENT FOR DRAMA*. Recuperado el 20 de 02 de 2010, de <http://www.ece.osu.edu/medhoc04/medhocnetfiles/papers/S10.5.pdf>
- [28] Royer, E. (21 de 02 de 2010). *A Review of Cyrrrent Routing Protocolos for AdHoc Mobile Wireless Networks*.
- [29] SEAH, W. (2004). *Mobility-based d-hop clustering algorithm for mobile ad hoc networks*. Atlanta, Georgia, USA.
- [30] SIVAVAKEESAR, S. y. (Septiembre de 2002). *A prediction-Based Clustering Algorithm to achieve Quality of Service in Multihop Ad Hoc Networks*. . Londres.

- [31] Stuart Russell, P. N. (2006). *Inteligencia Artificial*. España: Pearson.
- [32] Subiela, R. (30 de 04 de 2007). Simulación de Protocolos de encaminamiento en redes móviles Ad Hoc.
- [33] uni.lu. (19 de 01 de 2006). *Cluster-Head Gateway Switch Routing Protocol*. Recuperado el 02 de 03 de 2010, de <http://wiki.uni.lu/secan-lab/Cluster-Head+Gateway+Switch+Routing+Protocol.html>
- [34] Victor Garcia, F. G. (2009). *Secure Address Range Autoconfiguration Protocol for MANET*. Recuperado el 04 de 02 de 2010, de http://eprints.ucm.es/9444/1/Memoria_SS.II._2008_2009_-_Protocolo_SARA.pdf
- [35] Wenli Chen, N. J. (2010). *Ad Hoc Network Management Protocol*. Recuperado el 01 de 04 de 2010, de <http://www.cs.pdx.edu/~singh/ftp/anmp.ps.gz>
- [36] Yangcheng Huang, S. B. (2008). *The impact of topology update strategies on the performance of a proactive MANET routing protocol*. Recuperado el 01 de 03 de 2010, de <http://www.informaworld.com>

GLOSARIO DE TÉRMINOS Y ABREVIATURAS

- **Ad Hoc (a propósito):** es un tipo de red inalámbrica móvil la cuál posee una topología Ad Hoc, está conformada por un grupo de equipos móviles inalámbricos los cuales se comunican entre sí, no poseen infraestructura fija, ni administración centralizada y se comunican a través de señales de radio sin usar un punto de acceso.
- **AODV:** Es un protocolo de enrutamiento Ad Hoc reactivo que no tiene actualizaciones periódicas, pero está basado en mensajes hello que sirven para mantener actualizada la comunicación local.
- **Atenuación de la señal:** Es la pérdida de potencia sufrida por la misma al transitar por cualquier medio de transmisión.
- **Disponibilidad:** Servicio de seguridad que garantiza que los recursos de un sistema abierto sean accesibles y utilizables a petición de una entidad autorizada.
- **DSDV (Destination-Sequenced Distance-Vector Routing):** protocolo de enrutamiento proactivo, en el que los nodos vecinos se van enviando mensajes de enrutamiento los unos a los otros.
- **DSR:** Protocolo de enrutamiento reactivo, que no mantienen mensajes de control.
- **Enrutamiento:** es el camino o ruta establecida entre nodos durante la comunicación.
- **Fiabilidad:** garantizar que los paquetes o información llegue a su destino.
- **Fluctuación (Jitter):** es la variabilidad del tiempo de ejecución de los paquetes, es decir es una desviación de la exactitud de la señal de reloj durante el envío de señales digitales.
- **FQMM (Flexible QoS Model for MANET):** es un modelo de calidad de servicio, caracterizado como modelo híbrido IntServ/DiffServ.
- **GISC (Intervalos de Garantía de Clases de Servicio):** son un conjunto de intervalos que representan el estado de la red en términos de métrica diferente.
- **IEEE:** Institute of Electrical and Electronics Engineers
- **Integridad:** Servicio de seguridad que garantiza que los datos no han sido alterados o destruidos de modo no autorizado.
- **MAC, Dirección de Control:** Dirección hardware única que identifica únicamente cada nodo de una red
- **MANET:** Mobile Ad Hoc Network
- **MAC:** Medium Access Control
- **MHWN:** Multihop Wireless Network
- **NODO MÓVIL:** es un dispositivo terminal móvil, como celulares, computadores,

PDA, etc.

- **OSI:** Open System Interconnection
- **Protocolo:** Normas a seguir en una cierta comunicación: formato de los datos que debe enviar el emisor, cómo debe ser cada una de las respuestas del receptor, etc.
- **QoS:** (Quality Of Service) Calidad de servicio. Designa la posibilidad de medir, mejorar y, en alguna medida, garantizar por adelantado los índices de transmisión y error. Es importante para la transmisión fluida de información multimedia.
- **Router:** Dispositivo que transmite paquetes de datos a lo largo de una red. Un router está conectado al menos a dos redes, generalmente dos LANs o WANs o una LAN y la red de un ISP.
- **SNMP:** Simple Network Management Protocol
- **SWAN:** Stateless Wireless Ad Hoc Networks
- **Soft-State:** la reservación de recursos se efectúa temporalmente y permanece activa la sesión a lo largo del tiempo solamente si esta en uso la ruta y se reciben paquetes de flujo antes de que el temporizador asociado a la reserva expire.
- **Tan-Sigmoid:** es una función de transferencia de redes neuronales, esta función toma los valores de entrada, los cuales oscilan entre más y menos infinit, y restringe la salida a valores entre cero y uno.
- **Throughput:** En redes de comunicaciones, el rendimiento de procesamiento es la cantidad de datos digitales por la unidad del tiempo que se entrega sobre un acoplamiento físico o lógico, o que está pasando con cierto nodo de red. El término corresponde a la consumición digital de la anchura de banda. El rendimiento de procesamiento de sistema o el rendimiento de procesamiento del agregado es la suma de las tarifas de datos que se entregan a todos los terminales en una red.

RESUMEN

En el presente proyecto de investigación se propone un Modelo de Gestión de Redes Ad Hoc integrando técnicas de Redes Neuronales y Teoría de Juegos. La finalidad de integrar estas técnicas es con el fin de controlar la existencia de nodos egoístas en la red, mejorar la calidad y servicio en el paso de paquetes, y conservar por más tiempo el poder de batería de los nodos que es uno de los recursos más preciados para que forme parte de la red por más tiempo.

La existencia de nodos egoístas es controlada en dos fases dentro del modelo, una al momento de elección del cluster cabeza en donde se utiliza el algoritmo WCA combinado con teoría de juegos y en la fase de control de admisión dentro del modelo SWAN que es la base para aplicar redes neuronales y asignar el siguiente salto de manera inteligente, optimizando así la calidad de servicio en la red.