Construcción de un módulo de seguridad Informática en el sistema TLÖN que permita tener características de disponibilidad

Santiago José Molina Sanchez

Universidad Nacional de Colombia
Facultad de ingeniería, Departamento de Sistemas e Industrial
Bogotá, Colombia
2019

Construcción de un módulo de seguridad informática en el sistema TLÖN que permita tener características de disponibilidad

Santiago José Molina Sanchez

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:
Magíster en Ingeniería - Telecomunicaciones

Director:
Ph.D., Jorge Eduardo Ortiz Triviño

Línea de Investigación:
Redes y Sistemas de Telecomunicaciones

Grupo de Investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos - TLÖN

Universidad Nacional de Colombia
Facultad de ingeniería , Departamento de Sistemas e Industrial
Bogotá, Colombia

2019

Resumen

TLÖN es un sistema de cómputo de naturaleza estocástica y dinámica construido sobre una red Ad hoc cuyo propósito es gestionar los recursos (procesamiento, memoria, disco, etc.), que cada nodo (dispositivo) aporta al sistema. El funcionamiento de TLÖN esta inspirado en el concepto de Estado (o país) que cuenta con un «territorio» (Los recursos) y personas (agentes artificiales) que manejan y administran los recursos del Estado por medio de «Instituciones estatales». El presente informe presenta el análisis y diseño del módulo de seguridad del sistema de cómputo TLÖN; cuya solución se abordó a través del concepto de institución (del «Estado»). La institución de seguridad diseñada se compone de tres dependencias: Confidencialidad, Integridad y Disponibilidad, así mismo, se implementó un prototipo de la dependencia disponibilidad cuyo funcionamiento adecuado fue verificado en cuatro escenarios de prueba. Los diseños e implementaciones realizadas corroboran que la computación social es una buena estrategia para la seguridad de este tipo de sistemas distribuidos dinámicos.

Palabras clave: MANET, Virtualización Inalámbrica, Sistemas Distribuidos, Seguridad de la Información, Detector de Fallas, Institución, Social inspiración.

Abstract

TLÖN is a computing system of stochastic and dynamic nature built on an Ad hoc network whose purpose is to manage the resources (processing, memory, disk, etc.), that each node (device) contributes to the system. The operation of TLÖN is inspired by the concept of state (or country) that has a ”territory” (Resources) and people (artificial agents) that manage and administer state resources through ”State institutions”. This report presents the analysis and design of the security module of the computer system TLÖN; whose solution was addressed through the concept of institution (of the ”state”). The security institution designed is composed of three dependencies: Confidentiality, Integrity and Availability), likewise, a prototype of the availability dependency was implemented, whose proper functioning was verified in four test scenarios. The designs and implementations made corroborate that social computing is a good strategy for the security of this type of dynamic distributed systems.

Keywords: MANET, Wireless Virtualization, Distributed systems, Information security, Failure Detector, Institution, Social inspiration

Contenido

Lista de Figuras

Lista de Tablas

Lista de símbolos

Abreviatura	Término
ACK	Acknowledgement
AL	Agente Local
$ALFRED$	Almighty Lightweight Fact Remote Exchange Daemon
$B.A.T.M.A.N.$	Better Approach to Mobile Ad hoc networking
CAP	Consistency Availavility Partition tolerance
CIA	Confidentiality Integrity Availability
CPU	Central Processing Unit
DSR	Dynamic Source Routing
$DDOS$	Distributed Denial Of Service
GPS	Global Positioning System
IDS	Intrusion Detection System
$ICMP$	Internet Control Message Protocol
IoT	Internet of Things
IPS	Intrusion Prevention System
$IPv4$	Internet Protocol Versión 4
ISO	International Organization for Standardization
MAC	Message Authentication Code
MAS	Multi Agent System
$MANET$	Mobile Ad hoc network
OGM	Originator Message B.A.T.M.A.N.
$OLSR$	Optimized Link State Routing
ORQ	Orchestrator
QoS	Quality Of Service
RAM	Random Access Memory
$PRNET$	Packet Radio Network
$S.O.V.O.R.A.$	Sistema Operativo Virtualizado Orientado a Redes Ad hoc
$SURAN$	Survible Adaptative Radio Networks
TCP	Transmission Control Protocol
TQ	Transmission Quality
TTL	Time To Live

Abreviatura	**Término**
UDP	User Datagram Protocol
$VANET$	Vehicular Ad-Hoc Network
$WBAN$	Body Area Network

Introducción

Junto con las nuevas tendencias tecnológicas, las telecomunicaciones han ampliado su gama de aplicaciones en torno a un mundo que es cada vez más exigente. Han surgido nuevas tecnologías que permiten no solo enviar mensajes de correo electrónico o mensajería instantánea, sino que también, utilizar las telecomunicaciones para controlar los dispositivos de una casa, un vehículo, hospitales y tener aulas virtuales. Las redes inalámbricas son de gran utilidad como solución para la operación de equipos que no pueden permanecer en un lugar estático, pero que deben cubrir las demandas de un entorno en el cual los dispositivos están en constante movimiento, como lo son, los vehículos, celulares, redes de sensores, redes inalámbricas *Mesh*, entre otros.

TLÖN es un sistema de cómputo de naturaleza estocástica y dinámica construido sobre una red Ad hoc móvil (MANET) cuyo propósito es gestionar los recursos (procesamiento, memoria, disco, etc.), que cada nodo (dispositivo) aporta al sistema. Una *MANET* es una red de nodos conectada por interfaces inalámbricas, con un nivel de recursos dinámico, capaz de proveer servicios sin importar las condiciones estocásticas de los nodos al transcurrir el tiempo. Tienen características de auto organización y son auto creadas (Reddy y M, 2016). Formalmente, una MANET puede modelarse mediante un grafo aleatorio con un conjunto de vértices, comúnmente llamados nodos, en este caso móviles, unidos por un conjunto de enlaces denominados aristas, que cambian de forma dinámica en función del tiempo y las condiciones del ambiente. Por ejemplo, las peticiones de los usuarios (M. E. Newman, 2003).

Las MANET por sus características son propicias para la implementación de tecnologías como sistemas distribuidos y virtualización computacional. El sistema TLÖN está implementado sobre el concepto de computación distribuida, por esto, varios procesos pueden requerir un mismo conjunto de recursos simultáneamente (Elser, 2012). Un sistema distribuido está compuesto de N procesos, los cuales pueden realizar diferentes tareas de computación, algunas veces funcionando en fracciones muy cortas de tiempo. La suma del conjunto de procesos es lo que constituye el sistema distribuido. Los procesos se comunican intercambiando mensajes, asumiendo que todos los mensajes serán únicos e intercambiados a través de un enlace de comunicación (Cachin y cols., 2014). El sistema TLÖN adopta el modelo del teorema CAP (Consistencia, Disponibilidad y Tolerancia) de computación distribuida para las transacciones realizadas sobre las aplicaciones y base de datos asociadas, el teorema CAP discute la imposibilidad de garantizar aseguramiento (*safety*) y vitalidad (*liveness*) en un ambiente no confiable para un sistema distribuido. La consistencia trata con la respuesta adecuada a una

solicitud dada, y la disponibilidad busca que toda solicitud tenga una respuesta, es importante mencionar que el teorema CAP en (Gilbert y Lynch, 2012) es diferente a la definición de disponibilidad en seguridad de la información. De acuerdo con el teorema CAP ninguna aplicación en un sistema distribuido puede alcanzar estas tres características, una aplicación altamente disponible puede degradar de forma rápida la consistencia. Para (Gilbert y Lynch, 2012) la compensación del teorema CAP debe ser seleccionada dependiendo el tipo de aplicación y sistema a implementar. En un sistema que es altamente dinámico y con características de movilidad se requiere alcanzar suficiente disponibilidad y desempeño, por lo tanto, la consistencia debe ser sacrificada.

El funcionamiento de TLÖN está inspirado en el concepto de Estado (o país) que cuenta con un «territorio» (Los recursos) y personas (agentes artificiales) que manejan y administran los recursos del Estado por medio de «Instituciones estatales». Estos conceptos son aplicados a la distribución y gestión de recursos, los cuales son manejados por agentes artificiales que interactúan entre ellos para ejecutar tareas. Al igual que los sistemas de cómputo convencionales, el sistema TLÖN está expuesto a vulnerabilidades que pueden ser explotadas por entes maliciosos, por esto, el presente trabajo presenta la construcción del módulo de seguridad para el sistema TLÖN que actúa a nivel de red Ad hoc y virtualización, y proporciona capacidades de disponibilidad. Debido a que el diseño del sistema TLÖN es de naturaleza social inspirada, en el cual un conjunto de agentes llamados comunidad, se organizan para brindar servicios al sistema, el concepto de seguridad es abordado a través del concepto de institución.

El funcionamiento de la institución de seguridad depende de tres dependencias: Confidencialidad, Integridad y Disponibilidad, conceptos adoptados del modelo CIA (*Confidentiality, Integrity, and Availability*) , el cual enmarca los requerimientos de seguridad en estas tres características fundamentales (Regalado y cols., 2015); (Lehto y Neittaanmäki, 2018), para trabajar de forma integral el reto de la seguridad de la información, de la infraestructura y de todos los demás recursos que componen el sistema. Como primer paso para el diseño de un sistema seguro, se propone garantizar la disponibilidad de los elementos presentes en la red del sistema TLÖN, como parte de esto, se desarrolló un prototipo detector de fallas que permite detectar la falta de disponibilidad en sistemas distribuidos inalámbricos. La idea de un modelo de cómputo distribuido con características de movilidad y auto organización generan otros retos de seguridad, en donde las soluciones de redes tradicionales resultan ineficaces y exponen el sistema a una amplia gama de amenazas, por ejemplo, ataques de denegación de servicio, espionaje, ataques de agujero gris y negro, entre otros (Kalinin, Zegzhda, Zegzhda, Vasiliev, y Belenko, 2016). Es claro que cuando un sistema garantiza cierto nivel de seguridad, sobre él se genera también un grado de confianza.

En el marco de esta investigación se han aplicado aspectos metodológicos como la investigación exploratoria, de esta manera, se realizó un estudio de tipo exploratorio debido a la poca información e investigación que han tenido los temas relacionados con la seguridad de las MANET y los sistemas distribuidos, además, por ser un tema relativamente nuevo en el entorno académico e industrial. Por

lo anterior, esta investigación exploratoria es la fase inicial de la investigación de seguridad para sistemas estilo TLÖN, así también como la forma idónea de aproximarse a los nuevos fenómenos. La investigación exploratoria se orientó a fin de tratar de dar respuestas a preguntas básicas como ¿para qué?, ¿Cuál es el problema?, entre otras.

Otro aspecto metodológico utilizado fue la investigación descriptiva y comparativa, ya que por medio de la descripción de los diferentes vectores de ataques y vulnerabilidades de las MANET, se pudo exponer y resumir la información de manera cuidadosa y luego analizarla minuciosamente, a fin de extraer generalizaciones que contribuyan al conocimiento, posteriormente se comparó con ataques y vulnerabilidades ya conocidos en otras redes o situaciones, esto permitió pensar en las diferentes soluciones que podrían dar solución al problema.

Finalmente, se llevó a cabo un proceso de investigación experimental por medio de cuatro escenarios de prueba en laboratorio para la implementación del módulo detector de fallas para el sistema TLÖN que tenga propiedades de disponibilidad, además de la recolección de datos e información que contribuyan con posteriores investigaciones.

El presente documento se organiza de la siguiente manera: En el capitulo 1 se presentan las características de las redes Ad hoc, los sistemas distribuidos y sus generalidades en seguridad enmarcadas en el modelo CIA. En el capitulo 2 se presenta el sistema TLÖN, el cual es implementado sobre una MANET y utiliza el concepto de computación distribuida. El capitulo 3 menciona los objetivos desarrollados en este documento. El capitulo 4 muestra el análisis y diseño de seguridad para el sistema TLÖN basado en el concepto de institución. El capitulo 5 muestra los resultados obtenidos de la construcción del prototipo de seguridad propuesto para el sistema. Finalmente, en el capitulo 6 están las conclusiones, recomendaciones y trabajo futuro que se obtuvieron como resultado del desarrollo de este documento.

Capítulo 1

Redes Ad hoc y Sistemas Distribuidos

1.1. Definición redes Ad hoc

La primera generación de redes Ad hoc se remonta a 1972, en ese entonces fueron llamadas PRNET (*Packet Radio Network*), utilizaban un protocolo de enrutamiento de vector distancia y se aplicó en ambientes de combate, la segunda generación llego en 1980 donde fueron implementadas como parte de SURAN (*Survible Adaptative Radio Networks*), su aplicación apuntaba a dispositivos de baja eficiencia, pequeños y de bajo costo. En la década de 1990 las redes Ad hoc llegaron a los dispositivos de comunicación y computadores, finalmente, el termino red Ad hoc fue adoptado por el IEEE 802.11.

Las redes móviles Ad hoc (MANET) consisten de nodos móviles interconectados por caminos de comunicación inalámbricos multisalto (Mishra, 2008), el objetivo es que un nodo se pueda comunicar directamente con otro nodo siempre que el canal de propagación sea adecuado y esté disponible entre ellos (Sarkar, Basavaraju, y Puttamadappa, 2016). A diferencia de las redes inalámbricas convencionales, las redes Ad hoc no poseen una infraestructura fija o un soporte administrativo. La topología de una red Ad hoc cambia dinámicamente a medida que los nodos móviles se unen o salen de la red, son auto-creadas, auto-organizables y auto-administradas (Lin y Lu, 2015).

La palabra Ad hoc puede interpretarse como una red improvisada y sin organización lo cual a menudo tiene un significado negativo, en el contexto de este trabajo el significado de red Ad hoc debe relacionarse con las situaciones dinámicas que se pueden describir. Existen diferentes tipos de MANET como, *Building-To-Building*, *Vehicle-To-Vehicle*, *Peer-To-Peer*, al igual que las redes inalámbricas convencionales, las redes Ad hoc utilizan dos métricas importantes para su comunicación, el rango del espectro y las diferentes frecuencias de radio, por ejemplo, IEEE 802.11g utiliza la frecuencia de radio de 2,4- 2,58 GHz . Cada nodo en la red tiene ambas funciones, *router* y *host*, la naturaleza dinámica de la red hace que la conectividad entre los nodos pueda variar con el tiempo por la salida y entrada de otros nodos en la red, por lo tanto, se necesita la eficiencia de un protocolo de enrutamiento que permita comunicar todos los nodos (Murthy y Manoj, 2004). las redes Ad hoc deben manejar aspectos como

la falta de una administración centralizada, ser capaz de manejar los cambios arbitrarios en la topología y la presencia de nodos maliciosos cuyo comportamiento puedan afectar la red (Loo y cols., 2016).

Entre las características de las MANET se destacan:

- Las MANET tienen característica de operación distribuida, cada nodo opera independientemente con otros nodos para implementar funciones como enrutamiento y seguridad.
- La seguridad es una de las características más importantes porque los nodos no se encuentran protegidos como estarían en una estructura de operación fija, el tráfico puede ser robado o podría viajar inseguramente a través de la red, los ataques más comunes en este tipo de redes son, falsificación, interceptaciones y ataques de denegación de servicio.
- La topología de las MANETs siempre está cambiando aleatoriamente en el tiempo, por lo tanto, el enrutamiento debe ser dinámico, la red debe adaptar su tráfico de acuerdo a los patrones de movilidad de los nodos.

1.1.1. Clasificación de las redes Ad hoc

Existen diferentes formas de clasificar la comunicación entre los nodos, por ejemplo, en la figura **1-1** está el ejemplo más simple de red Ad hoc donde todos los nodos pueden comunicarse directamente sin la ayuda de otros nodo, este tipo de topología no es muy eficiente ya que los nodos tienen características de movilidad que les impide mantener una topología de salto simple. Los nodos no deben ser estáticos, sin embargo, se deben mantener dentro del rango de cobertura de los otros nodos, lo que significa que todos los nodos podrían moverse como un grupo. en la figura **1-2** se encuentra un ejemplo de red Ad hoc multisalto en la cual algunos nodos se encuentran fuera del rango de cobertura y por lo tanto se hace necesario utilizar nodos intermedios para completar la comunicación, este tipo de topología requiere un protocolo de comunicación que sea altamente adaptativo.

Existen formas de clasificar la red Ad hoc dependiendo el tipo de topología, las principales son: plana, jerárquica e híbrida. la topología de red plana se caracteriza por no tener distinción entre los nodos, todos son equivalentes, lo cual es una desventaja en escalabilidad, entre más nodos en la red menor será su desempeño. en la topología jerárquica los nodos están organizados en *cluster* representando una red en donde todos están conectados por medio de *cluster heads* o maestros. Los nodos están clasificados en dos categorías: nodo maestro, nodo normal (Santi, 2005).

- Nodo maestro: el nodo maestro es el encargado de administrar la red y es el responsable de enrutar los mensajes con los otros *cluster*.
- Nodo normal o esclavo: se comunican dentro del *cluster* directamente y con otros nodos a través del nodo maestro.

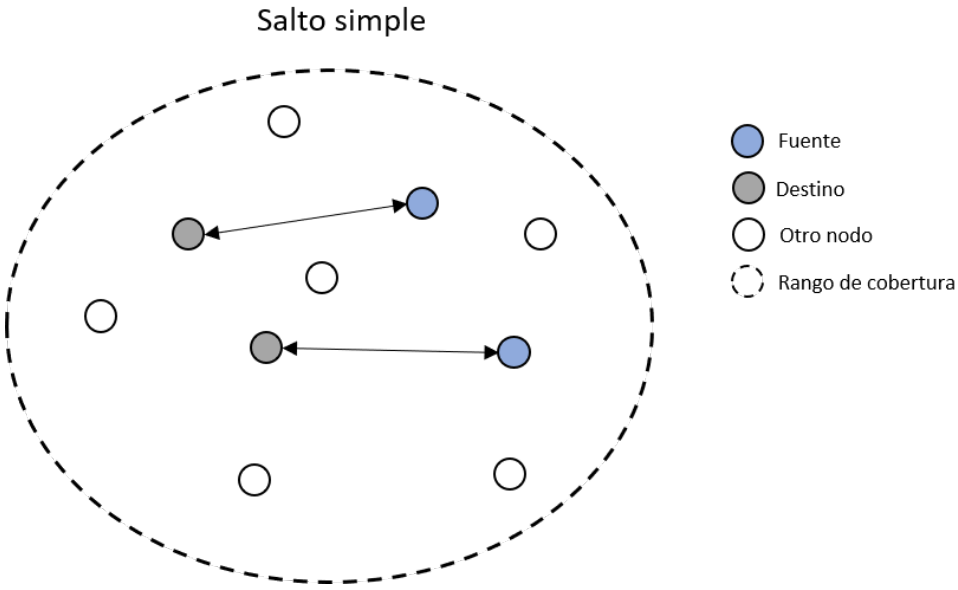

Figura 1-1: Red Ad hoc de salto simple (Loo y cols., 2016)

La ventaja de la topología jerárquica se encuentra en la escalabilidad, sin embargo, el nodo maestro es el responsable de la comunicación con los otros *cluster* lo que significa que, si el nodo falla, esa parte de la red se quedaría sin comunicación. la red híbrida permite la comunicación entre una red jerárquica y plana.

1.1.2. Enrutamiento en MANET

Existen diferentes formas de enrutamiento en las redes Ad hoc y todos buscan satisfacer las necesidades y limitaciones de una topología cambiante, alto consumo de energía, bajo ancho de banda y altas tasas de errores (Vargas, 2016). Los protocolos de enrutamiento se pueden clasificar en tres categorías principales: proactivos o *table-driven*, reactivos o por demanda, e híbridos. Como se ve en la figura **1-3**, cada protocolo difiere en la técnica que se utiliza, conteo de saltos, estado enlace y enrutamiento QoS (*Quality of Service*), en los protocolos basados en el conteo de saltos, cada nodo contiene la información del siguiente salto en su tabla de enrutamiento y el estado del enlace a su destino. Protocolos de estado de enlace mantienen una tabla de enrutamiento con toda la información de la topología la cual es construida encontrado el camino más corto a su destino.

- Protocolo sobre demanda: estos protocolos crean las rutas a su destino cuando es solicitada y es mantenida hasta que no se necesita, de esta forma se reduce el consumo de recursos en las tablas de enrutamiento.

- Protocolo proactivo: los protocolos proactivos mantienen en su tabla de enrutamiento toda la información de rutas hacia los destinos, se actualiza cada cierto tiempo, inunda la red con un

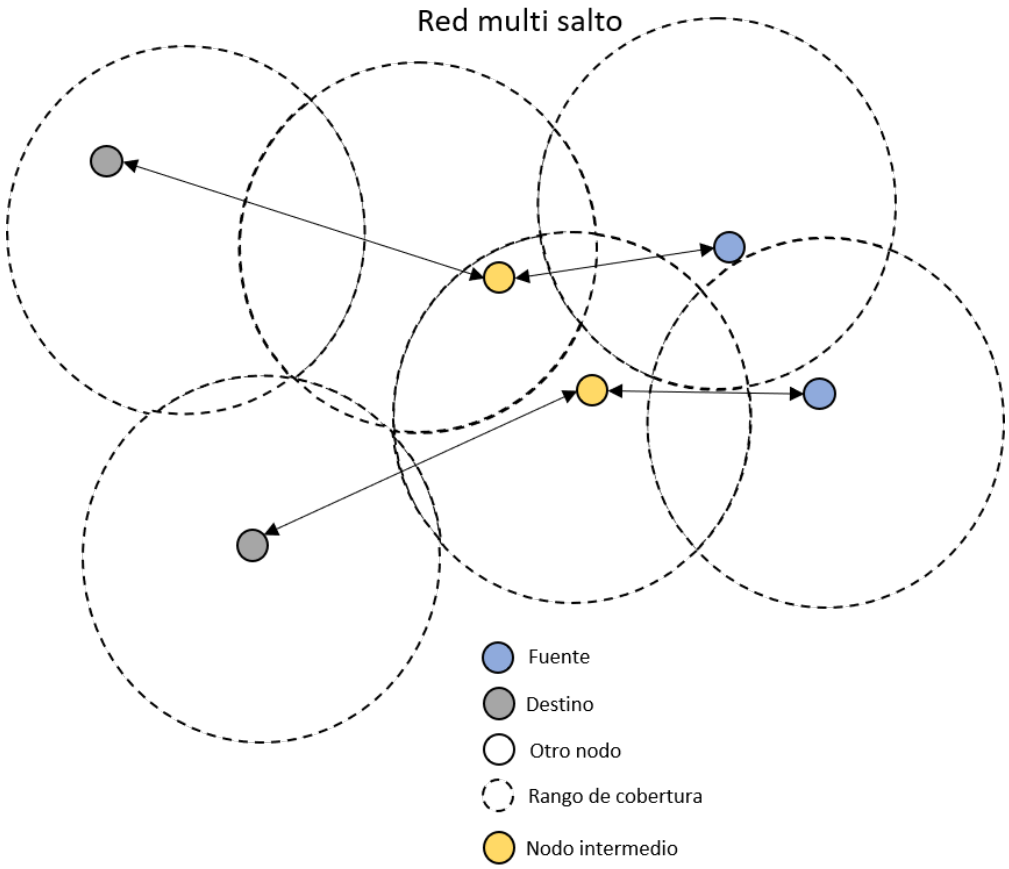

Figura 1-2: Red Ad hoc de multisalto (Loo y cols., 2016)

paquete *broadcast* para conocer su tabla de enrutamiento lo cual reduce el desempeño de la red.

1.1.2.1. Protocolo B.A.T.M.A.N

B.A.T.M.A.N (*Better Approach To Mobile Adhoc Networking*) es un protocolo de enrutamiento proactivo, un solo nodo no mantiene los datos de toda la red eliminando la necesidad de conocer y actualizar todos los cambios en la red. fue desarrollado por el grupo Freifunk (`http://freifunk.net/en/`). El esfuerzo principal con el protocolo B.A.T.M.A.N se enfoca en reemplazar el protocolo OLSR (*Optimized Link State Routing Protocol*).

B.A.T.M.A.N opera en la capa 2 del modelo OSI, esto quiere decir que los datos y la información de enrutamiento son transportados en tramas Ethernet, el algoritmo del protocolo se puede describir de la siguiente forma:

1. Cada nodo transmite mensajes *Broadcast* o también llamados mensajes originadores, de este modo se informa a los nodos vecinos de su existencia. Los mensajes originadores suelen ser

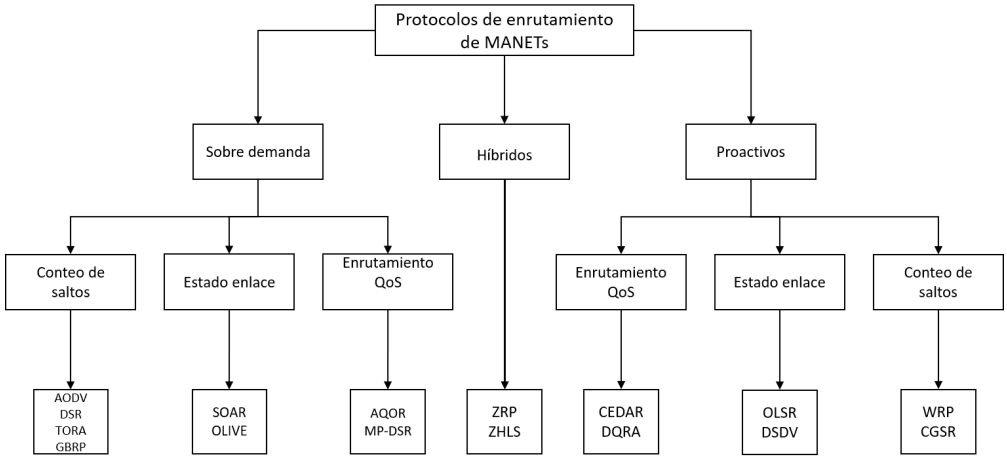

Figura 1-3: Protocolos de enrutamiento en MANET (Vargas, 2016)

livianos 52 bytes, así se intercambia la información de la dirección, la dirección de los nodos que transmiten los paquetes, un TTL (*Time To Live*) y un numero de secuencia.

2. Un nodo X aprenderá de la existencia de un nodo Y recibiendo el mensaje originador, cada nodo almacenará la información de los vecinos en una tabla llamada «tabla de traducción local». si el nodo X tiene más de un camino a Y este escogerá el mejor camino hacia su vecino.

3. Las rutas y la topología de la red sólo es conocida por los nodos que están dentro del alcance directo.

1.1.2.2. Tipos de Paquetes B.A.T.M.A.N

Muchos paquetes *unicast* son enviado por el protocolo B.A.T.M.A.N, todos los paquetes dentro de una clase *unicast* comparten el mismo encabezado: tipo de paquete, versión, TTL, destino. Este tipo de señalización de paquetes sirve para conocer el estado de la red, y son utilizados por servicios como seguridad, *network coding*, entre otros.

- 0x00 - 0x3f Paquetes especiales: son paquetes con reglas especiales que no pueden ser manejados, por ejemplo, mensajes originadores, *broadcast* y *network coding*.

- 0x40 - 0x7f Paquetes *unicast*: son enviados por las rutas establecidas por B.A.T.M.A.N, por ejemplo, *unicast*, *unicast-frag*, *unicast 4addr*, *tvlv unicast*, ICMP.

- 0x80 - 0xff Paquetes reservados.

1.2. Sistemas Distribuidos y Teorema CAP

Los sistemas distribuidos son intrínsecamente concurrentes, lo cual quiere decir que las acciones ocurren en múltiples lugares, por lo tanto, una aplicación puede utilizar múltiples ubicaciones para solucionar una tarea, y sí se replican los datos, existe la posibilidad de un acceso simultaneo en la misma ubicación, lo cual introduce los retos de la sincronización en estos sistemas (Elser, 2012). Un sistema distribuido está compuesto de N procesos, los cuales pueden realizar diferentes tareas de computación, algunas veces funcionando en fracciones muy cortas de tiempo, la suma del conjunto de procesos es lo que constituye el sistema distribuido. Los procesos se comunican intercambiando mensajes, asumiendo que todos los mensajes serán únicos e intercambiados a través de un enlace de comunicación (Cachin y cols., 2014); (van Steen y Tanenbaum, 2017).

Un sistema distribuido comunica sus procesos a través de una red y presenta características como la falta de un reloj físico común, no tienen una memoria compartida, en algunas ocasiones pueden estar separados geográficamente por miles de millas. se puede encontrar una relación entre los sistemas distribuidos y las MANET, sus características pueden ser compartidas para la distribución y asignación de recursos como se ve en la figura **1-4**. De los sistemas distribuidos nacen preguntas de cómo manejar los datos, y cómo controlar la operación de un sistema que posea un comportamiento distribuido, estas preguntas abren la puerta a propuestas como el teorema CAP (*Consistency, Aviability, Partition Tolerance*) que se diseña con el fin de mitigar los inconvenientes presentados en estos sistemas.

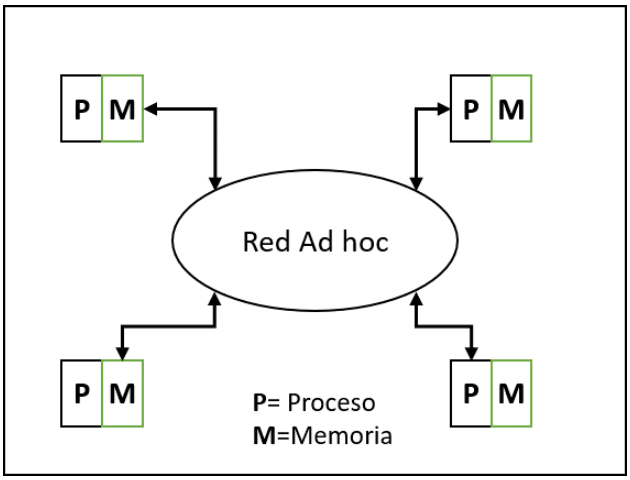

Figura 1-4: Sistema distribuido conectado por red Ad hoc (Cachin y cols., 2014)

La compensación entre consistencia, disponibilidad y tolerancia a particiones es definida en el teorema CAP, el cual, discute la imposibilidad de garantizar aseguramiento (*Safety*) y vitalidad (*Liveness*) en un ambiente no confiable para un sistema distribuido. La consistencia trata con la respuesta adecuada a una solicitud dada, y la disponibilidad busca que toda solicitud tenga una respuesta, es importante mencionar que el teorema CAP en (Gilbert y Lynch, 2012) es diferente a la definición de disponibilidad

en seguridad de la información.

De acuerdo con el teorema CAP ninguna aplicación en un sistema distribuido puede alcanzar estas tres características, una aplicación altamente disponible puede degradar de forma rápida la consistencia. Para (Gilbert y Lynch, 2012) la compensación del teorema CAP debe ser seleccionada dependiendo el tipo de aplicación y sistema a implementar. En la figura **1-5** se muestra la compensación del teorema CAP, por ejemplo, en un sistema que es altamente dinámico y con características de movilidad se requiere alcanzar suficiente disponibilidad y desempeño, por lo tanto, la consistencia debe ser sacrificada.

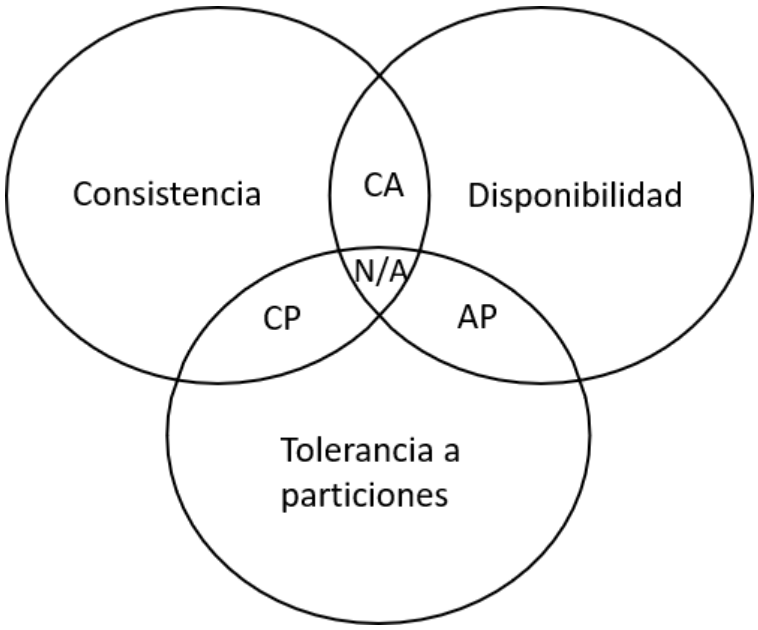

Figura 1-5: Teorema CAP (Gilbert y Lynch, 2012)

- Disponibilidad: La disponibilidad es definida como la capacidad de un sistema de responder a una petición
- Consistencia: la consistencia trata de dar una respuesta correcta a cualquier solicitud dada
- Tolerancia a particiones: el sistema debe seguir funcionando a pesar del fallo de o caídas parciales de un sistema.

1.3. Seguridad en MANET y Sistemas Distribuidos

Como se ha mencionado anteriormente la seguridad no es un tema ajeno en sistemas distribuidos y en las MANETs, incluso trabajos como (Cachin y cols., 2014), (Lin y Lu, 2015) y (Zhong y cols., 2018),

coinciden en resaltar la importancia de la investigación en el campo de la seguridad, la sensibilidad de los datos transportados sobre la red o la disponibilidad de los recursos distribuidos para solucionar una tarea son de vital importancia para conseguir una confiabilidad y consistencia en dichas tecnologías. Muchos modelos se han propuesto para abarcar todas las cuestiones de seguridad, sin embargo, la mayoría de autores coinciden en utilizar el modelo CIA (Confidencialidad, Integridad, Disponibilidad).

1.3.1. Modelo CIA

El modelo CIA (figura **1-6** es un sello distintivo de seguridad para sistemas de cómputo. (*Ceh Cert Ethical Hacker Exam Guide*, 2012); (Andress, 2014); (Whitman y Mattord, 2009) proponen el modelo como una forma de organizar los métodos por los cuales un sistema de seguridad debe funcionar, adicionalmente, existen unos principios extendidos que se pueden relacionar con la triada CIA, por ejemplo, autenticidad, no repudio, trazabilidad, certeza legal, entre otros.

Figura 1-6: Modelo CIA

1.3.1.1. Confidencialidad

Desde hace mucho tiempo, el hombre ha implementado técnicas para que sus mensajes no sean accedidos por entes no autorizados, por ejemplo, la Escítala, utilizada por los antiguos espartanos para ocultar sus mensajes en una tira de cuero o papiro. Actualmente, los sistemas de telecomunicaciones siguen el mismo principio de prevenir el acceso a los datos a usuarios no autorizados, la confidencialidad es proveída a través de mecanismos que previenen a los usuarios de obtener información a la que ellos no están autorizados (Kovacich, 2003). Los sistemas alcanzan cierto nivel de confidencialidad a partir de mecanismos provistos para esto, entre ellos se encuentran, la criptografía, estenografía, controles de acceso, autenticación, *proxys*, *firewalls*, entre otros.

1.3.1.2. Integridad

La integridad busca que la información que viaja a través de la red no sea modificada en ninguna parte de su arquitectura, permitiendo solo a los usuarios autorizados modificar los datos en el ciberespacio. Cuando los datos van de un nodo a otro, a menudo, atraviesan áreas compartidas donde otros actores tienen la habilidad de modificar el dato antes que alcance su destino (Blyth y Kovacich, 2006). La integridad de la información se debe pensar como un rompecabezas, cuyo objetivo es formar una figura combinando correctamente las piezas que se encuentran dispersas, y por ningún motivo ninguna ficha debe faltar, en los sistemas de telecomunicaciones la integridad a menudo se realiza utilizando firmas digitales y algoritmos matemáticos como los que utilizan la función *hash*, la cual es un arreglo de números conocidos como un valor *hash*. Cuando un mensaje es generado, un valor *hash* también se genera, y, se envía al recibidor del mensaje, dependiendo el método de verificación del valor del mensaje, este se aceptará o se rechazará (*Ceh Cert Ethical Hacker Exam Guide*, 2012).

1.3.1.3. Disponibilidad

La computación debe operar en algún nivel para ser útil, por lo tanto, la disponibilidad es un atributo critico de la ciberseguridad que ayuda a balancear las restricciones de los sistemas contra la utilidad. La disponibilidad asegura que el acceso a cualquier componente de la red no pueda ser restringido en una forma no autorizada, la disponibilidad garantiza un acceso pronto y confiable a la información para las entidades autorizadas. Existen diferentes vectores de amenaza que pueden afectar la disponibilidad de una red de sistemas distribuidos, aunque, con frecuencia se enfocan en ataques de denegación de servicio (DoS), los ataques de denegación de servicio distribuidos (DDoS) o los ataques de agujero negro o gris.

En (Yu, 2013) recomiendan diseñar un sistema considerando cuales recursos podrían ser explotados por un atacante y de qué forma podría limitar estos recursos. Un ataque de denegación de servicio es un ataque a un sistema de computadores o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provocan la perdida de conectividad por el consumo de transferencia de la información. Existen tecnologías para contrarrestar los ataques de denegación de servicio, estas incluyen, listas de acceso, filtros, diseños de alta disponibilidad, extra ancho de banda y soluciones en la nube.

1.3.2. Aspectos Generales de Seguridad en MANET

Las redes cableadas o redes inalámbricas usan una infraestructura de gestión y enrutamiento fijo para controlar su tráfico, acceso a la red y envío de paquetes. A diferencia de las MANET, las redes inalámbricas convencionales no son tolerantes a cambios arbitrarios de topología, son vulnerables a diferentes tipos de ataques que puedan afectar su confidencialidad, integridad y disponibilidad, para contrarrestar los diferentes vectores de amenaza comúnmente utilizan soluciones fijas que controlen el tráfico y acceso que entra y sale de la red. En cambio, las MANET deben considerar un cambio

dinámico en los nodos, los cuales también tienen funcionalidad de *routers*, estas redes al utilizar la cooperación entre nodos para su comunicación presentan retos diferentes a las redes con infraestructura fija en cuestiones de seguridad, sin embargo, las MANETs comparten vulnerabilidades como ataques de agujero gris, falsificación y espionaje con las redes de infraestructura fija. Entre las vulnerabilidades de MANET se encuentran: (Pathan, 2016):

- Confianza en los nodos: cuando se construye una red Ad hoc en protocolos como B.A.T.M.A.N, se asume que los nodos no son maliciosos, por lo tanto, es más fácil el acceso a un nodo susceptible a que se corrompa o altera su comportamiento negativamente y afecta a la red.

- Enlaces inalámbricos: como en las redes inalámbricas tradicionales, en las MANET el ancho de banda de la red es susceptible a ataques de espionaje, interferencia o alteración del canal de comunicación, un agente malicioso no necesita acceso a la red para desplegar un ataque.

- Topología dinámica: el cambio dinámico en la topología de red y la entrada y salida aleatoria de los nodos hacen impredecible el comportamiento en la red, los patrones de comportamientos pueden ser alterados fácilmente, lo cual dificulta predecir un ataque o un comportamiento anómalo.

- Información distribuida en la red: debido a su característica de cooperación entre nodos, no es claro el lugar que debería ocupar un sistema de control acceso, un *gateway* o un sistema de monitoreo de red.

- Operación con otro tipo de redes: no son claros los parámetros de seguridad que debe tener una red Ad hoc para operar con redes fijas o cableadas.

Los ataques sobre MANET se pueden clasificar entre activos y pasivos, los ataques pasivos en su mayoría buscan pasar desapercibidos, no tienen un daño directo sobre la red, su objetivo principal es el espionaje, analizar el tráfico y recolectar información sensible. Los ataques activos causan un daño directo sobre la red, son realizados por agentes maliciosos, su principal objetivo es causar el mal funcionamiento de la red, la falta de disponibilidad de las aplicaciones, tiene consecuencias como aislamiento de los nodos, consumo de los recursos e interrupción de las rutas (Chaki y Chaki, 2014).

Los ataques pasivos y activos pueden ser desplegados dentro y fuera de la red, esto quiere decir que pueden ser realizados por nodos que pertenecen a la red, que son legítimos pero que presentan un comportamiento anómalo, debido a los cambios dinámicos en la red son de difícil detección. Los ataques que tienen origen fuera de la red son desplegados por agentes maliciosos ajenos a la red. En la tabla **1-1** se ve un ejemplo de los ataques más comunes utilizados en cada capa del modelo OSI (Tanenbaum, 2003).

Capa	Ataque
Capa de aplicación	Virus, suplantación
Capa de Transporte	Inundación SYN TCP/UDP
Capa de red	Inundación ICMP, análisis de tráfico, agujero negro y gris
Capa de enlace de datos	Monitoreo, análisis de tráfico
Capa física	Interferencia activa, espionaje, consumo de ancho de banda

Tabla 1-1: Ataques comunes en la pila de protocolos

1.3.3. Aspectos Generales de Seguridad en Sistemas Distribuidos

Los sistemas distribuidos por sus características como procesamiento de nodos y medios de transmisión proponen retos en seguridad que son objetivo de investigación. La forma más simple de falla para un proceso es cuando el proceso deja de ejecutarse de forma adecuada, por ejemplo, el proceso ejecuta su algoritmo correctamente, incluido el intercambio de mensajes con otros procesos, hasta algún tiempo t, luego del cual deja de ejecutar cualquier computación y no envía ningún mensaje a otros procesos. En otras palabras, el proceso se bloquea en el momento t y nunca se recupera después de ese momento. Los sistemas deben ser resilientes contra los errores y ataques. Existen diferentes vectores de amenaza que afectan la confidencialidad, integridad y disponibilidad de los sistemas, la mayoría de investigaciones en sistemas distribuidos se enfocan en la confidencialidad e integridad de la red (Anderson, 2008). En (Belapurkar y cols., 2009) se hace un resumen de los ataques generales que explotan las características de una computación distribuida y en donde algunos vectores de amenaza coinciden con las MANET.

Un sistema se considera tolerante a fallas cuando después de haber sufrido un fallo o ataque se recupera y continúa funcionando, en general para los sistemas distribuidos, redundancia y tolerancia a fallas hacen sistemas más resilientes. Una de las principales razones para tener sistemas tolerantes a fallas es hacer los ataques de denegación de servicio menos efectivos o más difíciles de completar, otro posible ataque es forzar a los servidores a utilizar las credenciales almacenadas en su cache.

En los sistemas distribuidos, los sistemas de control de acceso son en su mayoría manejados por la sensibilidad de la información que transportan, por ejemplo, la información que es de alta sensibilidad en un día a los quince días puede ser información sin relevancia. Su función es controlar cuales principios (procesos, maquinas, personas) tienen acceso a cuales recursos del sistema, las abstracciones de cifrado también requieren que las claves se distribuyan de acuerdo con las identidades de todos los procesos, por ejemplo, una MAC (*Message Authentication Code*) requiere un par simétrico de claves compartidas para cada par de procesos, en cambio, para un esquema que funcione con firmas digitales se requiere un par de claves públicas/privadas para cada proceso, de manera que solo el proceso conozca su clave privada y todos los procesos conozcan las claves públicas de todos los demás. En la práctica, se pueden distribuir las claves necesarias durante la configuración del sistema, gene-

ralmente al mismo tiempo cuando se definen las identidades de los procesos en el sistema (Cachin y cols., 2014).

Capítulo 2

Sistema distribuido y dinámico social inspirado TLÖN

2.1. Estado y Normatividad

El Estado es una organización de orden social que busca el control y la organización de un territorio y sus habitantes. Los habitantes de un Estado están sujetos a un conjunto de normas que regulan el comportamiento y disponen las divisiones del poder.

La legitimación del poder se presenta cuando los integrantes de un estado aceptan y acatan las disposiciones y normas que son establecidas para mantener el orden institucional. En (Box, s.f.) el Estado se puede definir como, un instituto político de actividad continuada, cuando y en la medida en que su cuadro administrativo mantenga con éxito la pretensión al monopolio legítimo de la coacción física para el mantenimiento del orden vigente. Dícese de una acción que está políticamente orientada cuando y en la medida en que tiende a influir en la dirección de una asociación política; en especial a la apropiación o expropiación, a la nueva distribución o atribución de los poderes gubernamentales. Éste se caracteriza por ser un orden jurídico y administrativo cuyos preceptos pueden variarse.

El origen de las sociedades políticas se puede relacionar a la capacidad de cooperación entre los humanos para solucionar problemas y crear relaciones que permitan coordinar tareas, crear relaciones íntimas, compartir conocimiento, etc. Maquiavelo (Machiavelli, Bondanella, y Bondanella, 2013) relaciono el Estado con el crecimiento de las poblaciones, ya que en un principio los humanos vivían dispersos y fue el crecimiento de la población lo que permitió el encuentro entre diferentes humanos que llevo al nacimiento de un ordenamiento y organización política que garantizara el mantenimiento de un orden social. La creación del Estado está relacionada con las instituciones, ya que es un instrumento que garantiza la cooperación entre un grupo de gran tamaño, en (Dubreuil, 2010) se menciona que la racionalidad humana está limitada y, por lo tanto, la acción colectiva a gran escala implica la existencia de instituciones que facilitan el procesamiento de la información.

En (Pierson, 2004) se definen las características de los mecanismos de un estado como, monopolio, territorio, soberanía, constitucionalidad, autoría y legitimidad y ciudadanía. El conjunto de estos mecanismos constituye el estado y definen su funcionamiento. En la figura **2-1** se muestran algunos tipos de estado, las definiciones de estado al igual que las normas tienen amplios y variados significados, sin embargo pueden ser utilizados en sistemas multiagentes sociales computacionales, ya que el Estado se puede definir también como un artefacto coordinador de instituciones y pueden ser manejados por el tipo de Estado que se defina, por ejemplo, en un Estado jerárquico, en donde exista solo un agente encargado de crear las políticas, imponer sanciones, distribuir tareas y recompensas.

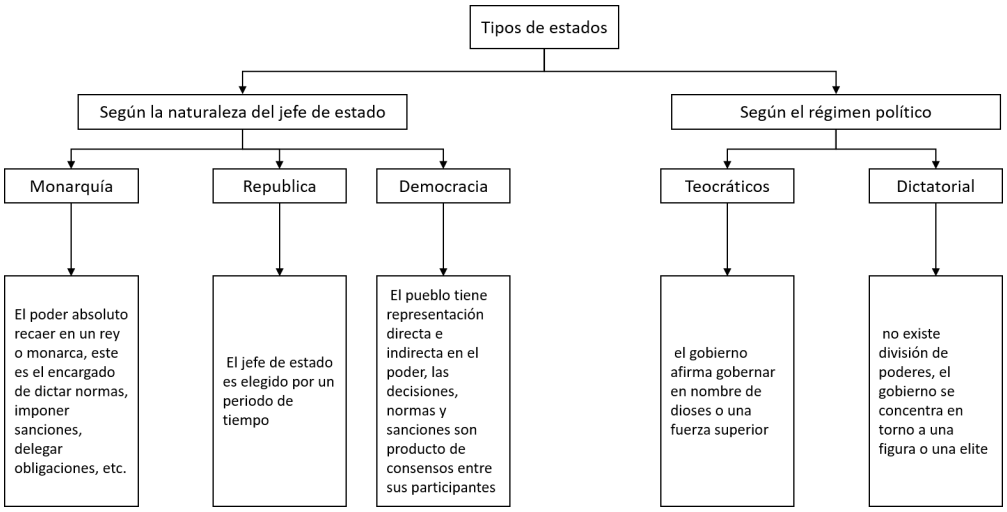

Figura 2-1: Tipos de Estado (Kisak, 2016)

El concepto de norma ha sido definido y aplicado en diferentes ámbitos de investigación, es explorado en la sociología, economía, política, filosofía, e incluso trasladado a la ciencia de la computación. En (Gibbs, 1965) se hace un recuento de las definiciones de norma entre las que se destaca: todas las sociedades tienen reglas o normas, especificando el comportamiento apropiado e inapropiado, las recompensas y castigos para los individuos que están bajo esas normas, son planos para el comportamiento, delimitan el alcance del comportamiento de los individuos, son bases culturales las cuales están justificadas por estándares morales, razonamiento o juicio estético.

Las normas son reglas de conducta, ellas especifican que se debería y que no se debería hacer por varios tipos de actores sociales en varios tipos de situaciones, el termino norma cultural se refiere a una prescripción del curso que una acción debería seguir en una situación dada. De las diferentes definiciones de normas se pueden relacionar aspectos como el comportamiento, ya que se busca delimitar lo aceptable y no aceptable, en algunas definiciones no es claro la consecuencia de no cumplir la norma, en otras se define como una obligación moral para el correcto comportamiento de una sociedad (Gibbs, 1965); (Pitt, Busquets, y Riveret, 2015).

Con el nacimiento de sistemas inteligentes y la evolución en la ciencia de la computación se comenzaron a crear áreas de investigación enfocadas a crear relaciones sociales en los sistemas y definir sistemas normativos para interactuar y funcionar entre Humano-Maquina o Máquina- Máquina como en (Fitoussi y Tennenholtz, 2000).

2.2. Computación Social y Normatividad en un Sistema Multiagente

La computación social busca identificar lo que permite ser un ente social, bajo esta premisa aparecen características como la comunicación, la cual permite generar comportamientos cooperativos o un comportamiento emergente deseado (Nietzsche y Ulapes, 2005). La social inspiración se basa en conceptos filosóficos como los de John Rawls o económicos como los de Elionor Ostrom (Ostrom, 2009). Estos modelos ideales buscan ser implementados o simulados en ambientes computacionales para obtener resultados adecuados con técnicas de optimización y la elección y sistematización adecuada de las reglas dentro del ambiente.

Incluir comportamientos de justicia, ética y confianza crean preguntas como, ¿Es posible modelar comportamientos sociales en sistemas computacionales? Y ¿para qué modelar comportamientos sociales?, bajo estas preguntas aparecen conceptos que pueden ser utilizados para modelar estos comportamientos, por ejemplo, el concepto de agente pude ser tan sencillo como un elemento que recibe entradas (percepciones), las cuales lo llevan a ejecutar acciones (acciones) sobre un ambiente, bajo este contexto se podría pensar en interacciones más complejas que permitan al agente reaccionar bajo diferentes situaciones, ser consiente del espacio que lo rodea, cooperar con otros agentes para solucionar una tarea e incluso crear un sistema normativo que rija todo un sistema multiagente. Estas interacciones entre agentes dan cabida a la aparición de algoritmos sociales que tratan de imitar las relaciones humanas en términos de liderazgo, cultura, asociación o coalición, incluyendo las dinámicas sociales humanas como estrategias de optimización.

En el ambiente de agentes computacionales se pueden presentar ambientes normativos que interactúen con el sistema. Los agentes en el sistema normativo estarán definidos por roles e interactúan entre ellos creando un sistema social basado en normas. Las normas son usadas para mantener el orden social en el sistema multiagente, de esta forma los agentes deben ser capaces de crear y mantener su sistema de normas que puede ir cambiando con respecto a las alteraciones o cambios que sufra el sistema, además, todos los agentes deben ser capaces de adaptarse al cambio en su ambiente adoptando el nuevo orden de normas. En (Boella y cols., 2006) se define el sistema normativo multiagente como el conjunto de agentes cuyas interacciones son normas gobernadas; las normas prescriben como los agentes deberían o no deberían comportarse. En (Carmo y Jones, 2002) se define un sistema normativo multiagente como un sistema de agentes ligados a un sistema normativo en el cual los

agentes pueden decidir si siguen las normas e incluye cómo y cuáles agentes pueden modificar y hacer cumplir las normas.

Por otro lado, las normas tienen diferentes representaciones en los estados del conocimiento, en la ley constitucional las normas son dictadas para controlar el comportamiento de sus habitantes y algunas son de origen restrictivo, lo que quiere decir que pueden causar penalidades, se representan en artículos, códigos de ley, etc. Existen tipos de normas de origen militar, las cuales definen roles y acciones que están permitidas en sus límites y funciones y pueden o no estar relacionadas con normas de Estado, también pueden tener normas independientes o locales, algunas veces son supervisadas por normas superiores, por ejemplo, normas de Estado. Por lo anterior, las normas deben definir un dominio en el cual se representa el Estado en que las normas son aplicadas, y qué sucede cuando una norma es violada. (Boella y cols., 2006) establece los atributos para las normas como:

1. El grado en que ellas son conocidas o reconocidas: Las normas deben ser desplegadas de tal forma que todos los agentes en el sistema tengan la capacidad de asimilar y entender el tipo de norma y su aplicación.

2. El grado en el que las normas son aceptadas como justas: Un agente puede decidir si una norma puede o no puede ser acatada.

3. El grado en el que las normas son aplicadas a todos los grupos o categorías: Pueden existir diferentes grupos o categorías de normas aplicables a un sector particular de agentes, se debe pensar cómo diferenciar o bajo qué criterio se forman los grupos o categorías de agentes.

4. Si las normas son severamente sancionadas o son laxas: Típicamente una norma está asociada a una obligación, el tipo de sanción se aplica dependiendo el tipo violación o la falta a esta obligación.

5. El modo y la consistencia de la ejecución: Las normas deben ser adecuadas a las situaciones del sistema, eventualmente pueden ser cambiadas dependiendo el estado actual del sistema y cómo se comporta en un tiempo determinado.

6. La fuente de autoridad: Las normas necesitan ser monitoreadas, el control, cumplimiento y creación de las normas debe tener un origen, por ejemplo, una institución de agentes creada para la gestión de políticas.

7. El modo en que las normas son transmitidas: Cuando las normas son creadas o actualizadas se deben difundir a todo el sistema.

8. La conformidad de las normas: El grado de conformidad de una norma va ligado a la utilidad de la norma en el sistema y su influencia en los agentes, cuando una norma deja de ser útil debe ser cambiada o actualizada por la fuente de autoridad encargada.

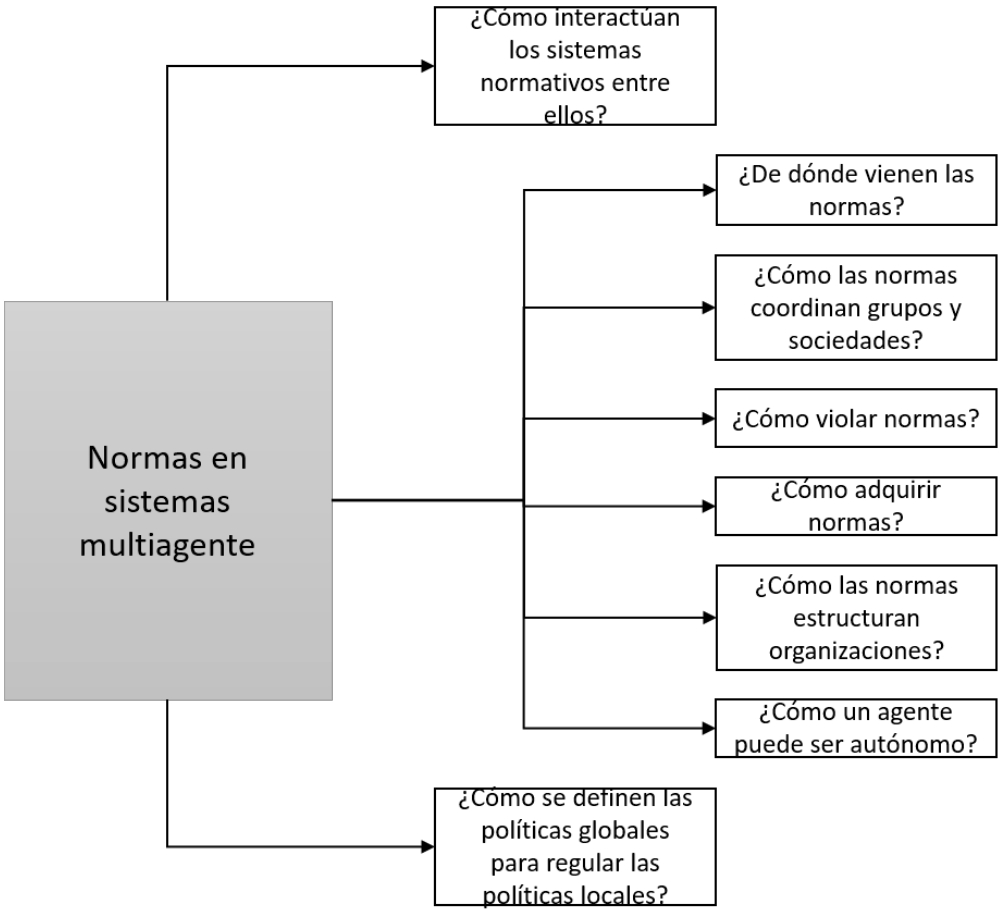

Figura 2-2: Requerimiento normas (Boella y cols., 2006)

En la figura **2-2** se define un marco para definir las normas en un sistema multiagente, se deben representar como un tipo de restricción y representarse en el dominio en el que se producen, por ejemplo, el dominio se puede relacionar a un partido de fútbol donde las normas serian el uso de dos equipos con 11 jugadores en donde cada equipo tiene un arquero y ningún jugador puede coger el balón con las manos excepto el arquero. Como se mencionó anteriormente, las normas pueden ser controladas y creadas por la abstracción de una institución, las cuales pueden enmarcar los comportamientos e interacciones sociales dentro de una sociedad.

2.3. Instituciones en Computación

Como se ve en la figura **2-6** las dimensiones de un agente expresan su entorno de aplicación y propiedades, la autonomía es una base fundamental para crear un sistema de agentes que muestren comportamientos sociales, el agente puede decidir si coopera con otros agentes para resolver una tarea o si convence a otros agentes para alcanzar objetivos propios o colectivos, puede crear nuevas tareas,

ser consciente de sus necesidades y dependencias, ya que de estas genera comportamientos sociales, crea alianzas y relaciones con otros agentes en su entorno. La inteligencia resulta ser un término ambiguo debido a que no se puede establecer una medida de qué tan inteligente debe ser un agente o cuándo un agente es más inteligente que otro, la inteligencia se representa como la capacidad de percibir y actuar a las situaciones que se presentan y cómo resuelve los problemas, y que tan capaz es el agente de moverse a través de su entorno para cumplir un objetivo.

Un artefacto es algo hecho por un agente para ser utilizado por otro agente. Los artefactos pueden coordinar las acciones de los agentes para resolver una tarea con otros agentes, pueden ser de características físicas que representan oportunidades y restricciones las cuales pueden ser utilizadas para coordinar las acciones (Silva y Lima, 2007). Las instituciones son artefactos de coordinación entre agentes, pueden reconocer las restricciones y oportunidades de su ambiente para habilitar una acción coordinada entre ellos. (Swaminathan y Wade, 2016) define las instituciones como artefactos de coordinación los cuales pueden venir en muchas formas: organizaciones, equipos, jerarquías, comportamientos de rutina. Una institución particular puede ser un compuesto de varias instituciones. En (Tomic, Pecora, y Saffiotti, 2018) las instituciones describen situaciones sociales, definen roles en interacciones sociales, y proveen una dimensión normativa, ligando roles a obligaciones, prohibiciones y permisos. así las instituciones codifican el comportamiento social entre agentes.

2.4. Sistema TLÖN

El sistema TLÖN propone un esquema de computación inspirado en modelos Sociales, inviables en la práctica pero muy posibles en entornos artificiales controlados, este sistema basado en los conceptos de Justicia de Jhon Rawls, Inmanencia de Baruch de Spinoza (de Spinoza y Peña, 1999), Paradigma de Thomas Kuhn (Kuhn, 1970), Estado de Thomas Hobbes (Hobbes, 1990) y las concepciones de existencia y esencia de Jean Paul Sartre (Sartre, Franco, y Moreira, 2000), generan una analogía completa de un esquema de virtualización inalámbrica, necesaria para implementar estos modelos sociales en sistemas computacionales, este modelo social inspirado, es una abstracción superior a los modelo bio-inspirados, busca solucionar problemas a través de acciones colectivas en ambientes no estacionarios, para esto, propone un esquema de computación distribuido, dinámico y con una estructura descentralizada. Las dimensiones del sistema TLÖN se preconciben como un modelo por capas como se muestra en la figura **2-3**. En la primera capa está la infraestructura, donde se encuentra la Red Ad Hoc, la segunda capa contiene la virtualización inalámbrica y el Sistema Operativo, en tercer lugar, se cuenta con el sistema multiagente, donde operan comunidades de agentes que proveen servicios a lo largo de la red Ad Hoc en el esquema de virtualización, y finalmente, una capa de aplicaciones específicas del sistema de cómputo. De forma transversal se tiene el lenguaje del sistema y la ontología propia del mismo, generando las interacciones en todas las capas.

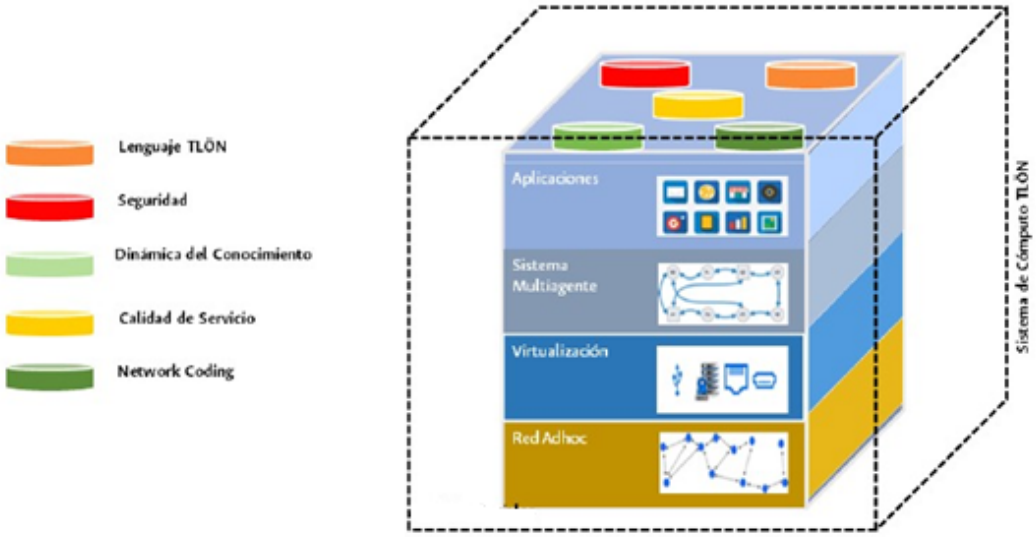

Figura 2-3: Sistema TLÖN

2.4.1. Principios Sociales Sistema TLÖN

La social inspiración como modelo de cómputo propone controlar las interacciones e incluir el componente ético dentro de los sistemas de cómputo, imposible en sociedades reales pero posible en sociedades artificiales, o simplemente comunidades de agentes artificiales o servicios que actúan juntos en busca de entregar una calidad de servicio adecuada y coherente a los miembros de este sistema, es decir, los usuarios finales, propietarios de los dispositivos de cómputo, quienes donan sus recursos para componer este sistema. Este modelo idealiza los principios de Justicia, Inmanencia y da paso a principios derivados, como la equidad, la solidaridad y a la aparición de virtudes como la verdad, dentro de este Sistema Artificial, las interacciones permiten la aparición de nuevas entidades como comportamiento emergente e incluso modelos de Estado, estos dependen de la función, o en términos computacionales, del elemento a optimizar.

En la imagen **2-4** se ve una analogía de un Estado con sus propiedades, territorio, persona, sociedad e instituciones, trasladado al sistema TLÖN, donde la red Ad hoc es el elemento que permite interactuar y moverse dentro del sistema, es capaz de adaptarse y configurarse. El sistema TLÖN es de naturaleza distribuida y está construido bajo el concepto de justicia de John Rawls, este concepto está presente en la asignación de recursos y de esta forma distribuirlos dentro de la red. El sistema crea interacciones entre agentes que se alimentan de estas percepciones y como en cualquier sociedad nace la noción de seguridad, ¿cómo se protege un Estado o quién protege a los agentes dentro de la sociedad?, de esta pregunta nace la necesidad de crear un modelo de seguridad que sea aplicable al sistema y ¿cómo hacer un sistema de seguridad basado en los principios sociales?, la social inspiración puede proporcionar ambientes cooperativos entre agentes para solucionar problemas que afecten de forma

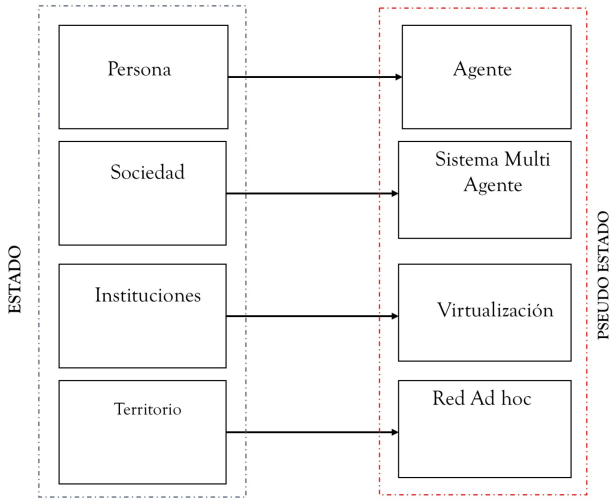

Figura 2-4: Modelo social-inspirado

directa la seguridad e interrumpan el buen funcionamiento de la red, al nacer una comunidad ya sea de agentes o de humanos comienzan a aparecer aspectos como la necesidad de una organización, estas organizaciones están regidas por normas y órganos de control que hacen cumplir las normas, en los agentes computacionales, estas nociones pueden ser creadas y los órganos de control pueden estar presentes, por esta razón nace la idea de crear una institución en la cual se organizan y estructuran los diferentes órganos de seguridad que ayuden al sistema a tener una noción de seguridad.

2.4.2. Capa de Red Ad hoc

Las redes móviles Ad hoc (MANETs) consisten de nodos móviles interconectados por caminos de comunicación inalámbricos multisalto (Mishra, 2008), el objetivo es que un nodo se pueda comunicar directamente con otro nodo siempre que el canal de propagación sea adecuado y esté disponible entre ellos. A diferencia de las redes inalámbricas convencionales, las redes Ad hoc no poseen una infraestructura fija o un soporte administrativo. La topología de una red Ad hoc cambia dinámicamente a medida que los nodos móviles se unen o salen de la red, son auto-creadas, auto-organizables y auto-administradas (Lin y Lu, 2015).

2.4.3. Capa de Virtualización

La capa de virtualización toma recursos de la red Ad hoc, y generan algunas interacciones entre la red y la virtualización, esto crea una abstracción lógica, a partir de un sistema distribuido con un sistema que controla los recursos de hardware. La virtualización es la creación de un conjunto de arquitecturas

lógicas usando un conjunto dado de entidades físicas, pero de una manera transparente para el usuario (Wang, Krishnamurthy, y Tipper, 2013). La virtualización de recursos en este contexto busca, crear un servidor de alta disponibilidad sobre la infraestructura de la red Ad hoc, además se deben tener en cuenta las condiciones de los enlaces dinámicos.
según lo propuesto en el teorema CAP (*Capacity, Avaibility and Partition Tolerance*) el sistema está bajo el modelo de operación AP, disponibilidad y tolerancia a particiones, esto con el fin de concebir un diseño robusto, escalable, y accesible cada vez que sea solicitado.

2.4.3.1. Microservicio

El sistema propone una arquitectura distribuida y descentralizada, por lo cual está basado en una arquitectura de microservicios, ya que su funcionamiento permite la asignación de recursos y ejecución de procesos de forma distribuida. Los microservicios son un estilo de arquitectura ampliamente basado en servicio autónomo desacoplado que puede ser desarrollado, desplegado y operado independientemente de los otros. Como se ve en la figura **2-5**, la arquitectura de microservicio es la creación de un sistema desde la colección de servicios pequeños y aislados, cada uno con su propia data, independiente, aislado, escalable y tolerante a fallas. Cada proceso es realizado de forma aislada y realiza la entrega de la información en esquemas compartidos, sus principales características son: la creación de subsistemas aislados comunicados por protocolos, el requerimiento de comunicación sincrónica y la habilidad de mover servicios alrededor del sistema (S. Newman, 2015).

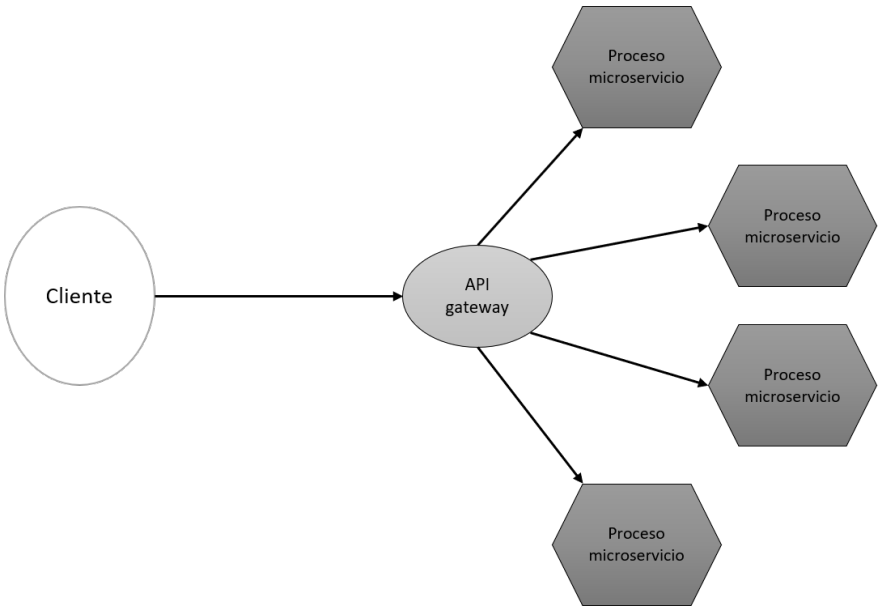

Figura 2-5: Microservicio (S. Newman, 2015)

En las arquitecturas distribuidas los componentes son accedidos remotamente a través de algún protocolo de acceso remoto. Para las arquitecturas basadas en servicios la Modularidad es la práctica

de encapsular porciones de la aplicación dentro de servicios auto contenidos que puedan ser individualmente diseñados, desplegados, desarrollados y probados con poca o ninguna dependencia sobre otros componentes o servicios en la aplicación. Un ejemplo de microservicio es una base de datos distribuida la cual es un conjunto de datos que se encuentran distribuidos en diferentes espacios lógicos y geográficos e interconectados por una red de comunicaciones.

2.4.3.2. S.O.V.O.R.A

Este modelo de Sistema operativo basado en agentes locales, orquestadores y contenedores se encuentra en las dos primeras capas del modelo TLÖN, permite distribuir tareas y componentes dentro del sistema con las características de replicación, migración y monitoreo de forma aislada pero orquestada entre los diversos artefactos computacionales que lo componen.

S.O.V.O.R.A es un modelo de contenedores, el cual funciona sobre el kernel de linux, se usa el módulo B.A.T.M.A.N en el esquema de Comunicación y el servicio de Mensajería A.L.F.R.E.D como módulo de descubrimiento de nodos, al igual que el motor de docker, el modelo de contenedores permite hacer la transición de redes de borde hasta llegar a la computación en la nube.

Existen dos elementos principales en SOVORA, el agente local y el orquestador. El agente local administra, monitorea y gestiona los recursos y aplicaciones usadas dentro del sistema, este elemento gestiona las comunicaciones del nodo con el orquestador, informa de su estado y mantiene un *log* completo de las interacciones con el ambiente base, el agente local es quien nutre la base de información del orquestador y reserva recursos para la ejecución de aplicaciones distribuidas en la red, del mismo modo entrega los resultados de la operación al o a los nodos que requieren la información, este artefacto computacional es base del sistema de detección de fallas presentado en la sección 4.2.2. El orquestador es el agente dotado con las capacidades de toma de decisiones, ejecuta aplicaciones o indica los recursos para la ejecución de una aplicación dentro de un sistema. Los escenarios de pruebas realizados en este trabajo utilizan estos dos elementos para la detección de fallas en la capa de red Ad hoc y virtualización del sistema TLÖN.

2.4.4. Sistema Multiagente

Los sistemas multiagente son parte de la inteligencia artificial distribuida y han ganado considerable aceptación durante los últimos años debido a su capacidad de solucionar problemas complejos. Estos sistemas pueden funcionar de manera descentralizada y permiten incluir propiedades como robustez y adaptación, las cuales son de gran importancia en ambientes no estacionarios. Debido a la ausencia de control centralizado, los agentes pueden competir, cooperar o simplemente coexistir generando la necesidad de construir mecanismos que permitan solucionar problemas a través de acciones colectivas. Estas características los convierten en un modelo promisorio para operar sobre sistemas como las redes Ad hoc.

Un agente es cualquier cosa capaz de percibir su medio ambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores. Cada agente puede percibir sus propias acciones (pero no sus efectos). La percepción es la muestra de que el agente puede recibir alguna entrada. Un agente tomará alguna decisión dependiendo la cantidad de percepciones que reciba en su entrada, La racionalidad, se presenta cuando se obtiene un resultado cada vez mejor, y esto se determina mediante las medidas de rendimiento. Si se sitúa un agente en un medio y realiza acciones de acuerdo con las percepciones que recibe, él debe pasar por una secuencia de estados adecuados para que haya actuado correctamente.

Un agente móvil es un sistema computacional que puede moverse sobre la red y actuar en nombre de un usuario, ellos deben ser capaces de detectar el ambiente y adaptarse dinámicamente a los cambios, debe tener la capacidad de mover su propia identidad sobre la red, por lo cual, un agente puede moverse de una localización a otra, la inteligencia del agente debe ser otro atributo que permita interactuar y aprender de su ambiente, y de esta forma poder tomar decisiones (Genco, 2008). Un agente móvil tiene las características de movilidad, autonomía, adaptabilidad, y colaboración. Un grupo de agentes móviles cooperantes pueden trabajar juntos con el propósito de intercambiar información, resolver problemas y realizar otras tareas, debido a las tareas que realizan, la seguridad es de gran importancia, e involucran todo el sistema de agentes. Uno de los objetivos de realizar investigaciones en el campo de la inteligencia artificial es simular la capacidad de solucionar problemas, tal como, los seres humanos realizan operaciones. Utilizar herramientas como computación distribuida en los agentes permite realizar tareas más complejas y también cooperar con diferentes agentes distribuidos e interconectados.

El entorno sobre el cual los agentes se sitúan representa el conjunto de problemas para los cuales la existencia de estos pretende dar solución. En un nivel abstracto el ambiente puede ser descrito como un conjunto de agentes móviles localizados en un grafo G. Como se ha mencionado un agente puede realizar tareas de cooperación con otros agentes para dar solución a una tarea específica, esto implica tener la capacidad de compartir su memoria y capacidades computacionales, se debe tener claro para los agentes quién realiza una tarea en específico, bajo qué condiciones y cuál será su costo, la interacción en estos ambientes son uno de los más grandes problemas de seguridad

La funcionalidad de las capas anteriores del modelo TLÖN convergen en el sistema multiagente. Un agente del sistema aprovecha la red Ad hoc y virtualización para satisfacer la propiedad de movilidad virtual y física, la cual hace parte de la dimensión de un agente como se ve en la figura **2-6**, por otro lado, la sociabilidad, además de establecer un ambiente de cooperación o coexistencia, busca ir más allá y establecer un modelo de institución, en el cual, se describen situaciones sociales, definen roles en interacciones sociales y proveen una dimensión normativa, vinculando roles a obligaciones, prohibiciones y permisos. Una institución da un estatus a los elementos que domina, por ejemplo, un pedazo de papel que funciona como dinero en una institución, tiene la función de voto en otra

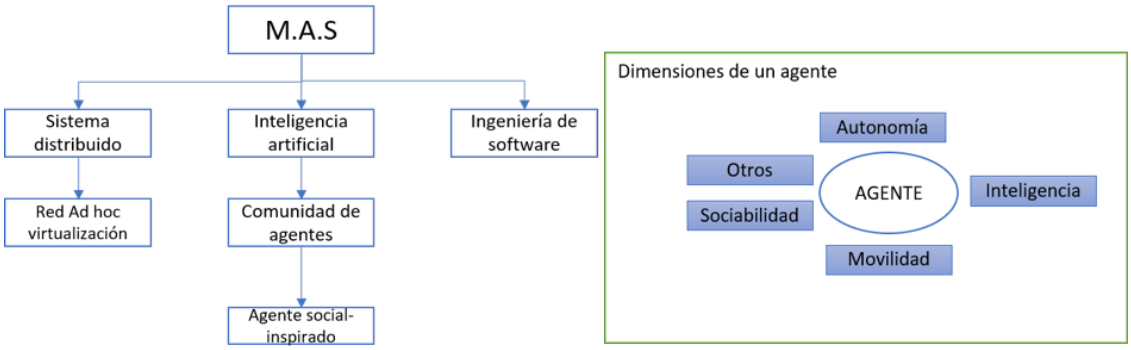

Figura 2-6: Dimensiones de un Agente (Shehory y Sturm, 2014)

institución.

2.4.5. Capas Transversales

Los beneficios de *Network Coding* incluyen un mejoramiento en el rendimiento *(throughput)* de la red, reducción en el consumo de energía, reducción en el costo del ancho de banda, entre otros. El lenguaje TLÖN proporciona una forma de comunicación entre las diferentes capas y el sistema multiagente. La calidad del servicio mide la calidad de los servicios que son considerados en varios aspectos del servicio de red, tales como tasas de errores, ancho de banda, rendimiento, retraso en la transmisión, disponibilidad, *jitter*, etc. Construir sistemas seguros permite proveer una tolerancia a fallas, sistemas robustos y confiables, protección del sistema contra daños e intrusos, etc. Todas son propiedades trasversales del sistema, lo cual quiere decir que están presentes en cada una de las capas del sistema (red Ad hoc, virtualización, sistema multiagente) y deberían funcionar integralmente.

Capítulo 3

Identificación del Problema y Pregunta de Investigación

3.1. Problema

TLÖN es un modelo de computación social que se aborda a través del concepto de estado, esto quiere decir que existen características propias de un Estado social de derecho integrado por personas, por ejemplo, las instituciones, cuyo objetivo es la administración de los recursos del Estado, esta administración es realizada por agentes artificiales que son gobernados por un sistema de normas superior denominado, pseudo constitución. Para poder implementar el modelo de estado propuesto por el sistema TLÖN, es necesario utilizar herramientas que permitan concebir todas las características propias de un Estado, por ejemplo, el sistema está construido sobre una MANET como parte de su analogía de territorio, y mediante la utilización de recursos computacionales físicos permite la comunicación y movilidad de las personas (agentes artificiales).

La computación distribuida es otra característica adoptada por el sistema que permite la creación de tareas y administración de recursos, sin embargo, al ser un sistema computacional, también presenta las vulnerabilidades y amenazas propias de cualquier sistema. Los modelos de computación utilizados para implementar el sistema TLÖN pueden ser útiles para el funcionamiento del sistema, sin embargo, características como movilidad, auto organización y auto administración generan ciertos retos de investigación en el campo de seguridad que no se pueden solucionar con los modelos actuales, además, se debe pensar en la integración de los modelos de seguridad con las características de un modelo social que adopta el concepto de institución. Por lo anterior nace la pregunta ¿Cómo garantizar la seguridad en un sistema distribuido dinámico?, y teniendo en cuenta el enfoque del modelo social inspirado de TLÖN ¿cómo garantizar la seguridad en un sistema social inspirado implementado sobre una MANET que integra elementos de una computación distribuida?

3.2. Justificación

Las MANETs cambian el modo en que se conciben las redes de datos, pero esto trae un gran riesgo para la integridad de los recursos y la información transportada sobre las redes, por ejemplo, cuentas bancarias, GPS en redes vehiculares, contraseñas de acceso, entre otros, es por ello que es necesario proponer una arquitectura que provea un mecanismo de seguridad es indispensable para los desarrollos futuros de una aplicación. En (Sahoo, Sahoo, y Panda, 2015) se propone una arquitectura para redes dinámicas tales como las MANETs que trabajan en una pequeña zona y transportan datos en topologías de salto simple, sin embargo, las redes deben ser escalables, y proporcionar mecanismos de seguridad para redes de gran tamaño se convierte en una tarea difícil. Mientras se comparte la carga entre los principales controladores de la red, es deseable tener una comunicación segura en el tránsito de la información, en (Flauzac, González, Hachani, y Nolot, 2015), los autores describen una arquitectura para un ambiente IoT (*Internet of the Things*), aunque no dan un mecanismo de seguridad para implementarla. El internet de las cosas involucra algunos dispositivos inteligentes a internet para comunicarse e intercambiar información, es una nueva tecnología que puesta en el lugar correcto puede lograr mayor eficiencia en la automatización industrial y la seguridad en los procesos de producción, entre otras cosas (Bin, 2012).

Una red que realiza un intercambio de información muy sensible, debe usar algunos modelos para controlar los problemas de ataques. Diferente a otras comunicaciones tradicionales, donde los nodos debían tener acceso físico a la red o comunicarse a través de muchos perímetros de defensa como *firewalls* y *Gateway*, los atacantes pueden usar el medio inalámbrico para atacar sobre una red inalámbrica, los ataques pueden venir de cualquier dirección y atacar cualquier nodo, esto da como resultado una gama de ataques disponibles que pueden ser utilizados en la red (*Ceh Cert Ethical Hacker Exam Guide*, 2012); (*Wireless Security*, 2006).

Las redes inalámbricas por su alta movilidad y dinamismo son una excelente herramienta para aplicaciones que requieran movilidad y una topología dinámica, desde vehículos conectados con redes vehiculares VANET (Hartenstein y Laberteaux, 2009), hasta redes de sensores WBAN (*Wireless Body Area Network*) (El-Bendary, 2014) para el monitoreo de la salud de los pacientes. El creciente número de aplicaciones para las redes MANET ha llevado a un despliegue de diferentes tecnologías y técnicas para cumplir con los requerimientos de movilidad y dinamismo, por ejemplo, la virtualización inalámbrica, ya que reduce la necesidad de una infraestructura física y permiten crear ambientes que favorezcan al dinamismo de una topología cambiante. La seguridad surge como un atributo clave en cualquier red por la sensibilidad de datos que maneja, aunque se han realizado varios esfuerzos aun es un campo carente de investigación por la reciente creación y despliegue de aplicaciones.

3.3. Objetivos

3.3.1. Objetivo General

Construir un módulo de seguridad para el sistema TLÖN que actúe sobre la capa de red Ad hoc y la capa de virtualización y proporcione capacidades de disponibilidad.

3.3.2. Objetivos Específicos

1. Diseñar un modelo con las principales características de la capa de seguridad para el sistema TLÖN.

2. Diseñar las políticas de seguridad para sistemas distribuidos y redes inalámbricas sobre las cuales se implementará el sistema TLÖN.

3. Construir un módulo que proporcione requerimientos de disponibilidad y mitigue los posibles ataques de negación de servicio sobre la red.

4. Validar la implementación del sistema de seguridad sobre la infraestructura TLÖN.

3.4. Producción Académica

- Dos artículos de investigación.

    1. Molina Sanchez S, Ortiz Triviño J, Zarate Ceballos H. Aspectos de seguridad para sistemas de cómputo social-inspirados construidos sobre manets. ingeniare [Internet]. 12sep.2017 [citado 26feb.2019];(23):69-2. Available from: `https://revistas.unilibre.edu.co/index.php/ingeniare/article/view/2884`
    2. Molina Sanchez S, Ortiz Triviño J, Zarate Ceballos H. Modelo de disponibilidad para sistemas distribuidos dinámicos. (en revisión)

- Presentación poster: Molina Sanchez S, Ortiz Triviño J, Zarate Ceballos H. Construcción de un módulo de seguridad informática en el sistema TLÖN que permita tener características de disponibilidad. Primera Jornada de ciberseguridad, Universidad Nacional de Colombia. Available-from: `https://www.ticketcode.co/organizador/uqbarun`

- El desarrollo de Software que permite realizar la detección de fallas sobre redes inalámbricas Ad hoc.

3.5. Aportes al Conocimiento

- El desarrollo de un modelo de seguridad para el sistema social inspirado TLÖN en sistemas distribuidos inalámbricos, mediante la creación de una institución de seguridad que maneje e implemente políticas locales y globales que regulan los comportamientos en el sistema.
- El desarrollo de un artefacto de seguridad que permite detectar fallas en la calidad del enlace en un sistema distribuido implementado sobre redes Ad hoc.
- La integración del módulo de seguridad con el sistema social inspirado TLÖN en sus capas de red Ad hoc y virtualización.
- La implementación de un modelo detector de fallas en dispositivos móviles descentralizados.

Capítulo 4

Análisis y Diseño de la Institución de Seguridad TLÖN

4.1. Institución de Seguridad TLÖN

Como se mencionó en la sección 2.1, un Estado puede ser un ente coordinador de instituciones. Como parte de su naturaleza social inspirada, el sistema TLÖN es un generador de instituciones basado en un Estado que es el generador de políticas y distribuidor de tareas. Existen diferentes tipos de estado, para este trabajo la institución de seguridad fue creada bajo el modelo de Estado república, sus políticas están bajo un poder constituyente, así nace el concepto de pseudo constitución TLÖN, la cual tiene el propósito de constituir la separación de poderes, definiendo y creando los poderes constituidos (legislativo, ejecutivo y judicial) estos pueden ser representados por comunidades de agentes y su rol jugara el mismo papel que en una república de Estado político, por ejemplo, los agentes a cargo del poder judicial serán los encargados de regular las normas sancionatorias a comportamientos que violen las normas que están escritas bajo la pseudo constitución. La pseudo constitución tendrá las políticas globales y dicta el modo en que cambian y se crean las instituciones, define sus respectivos controles y equilibrios, además es la ley fundamental del Estado, con rango superior al resto de normas, incluye el régimen de los derechos y libertades de los agentes y delimita los poderes e instituciones de la organización.

Las instituciones en el Estado TLÖN describen las interacciones sociales, definen roles, y proveen una dimensión normativa ligando roles a obligaciones, prohibiciones y permisos. Los agentes deberán ser capaz de razonar sobre los roles que juegan en una institución en particular, las obligaciones que ellos deben cumplir, cómo utilizar los artefactos que componen el sistema e interactuar con los diferentes roles de agentes. Las instituciones coordinan los comportamientos dentro de la sociedad, así, ellos son algunas veces llamados como «artefactos de coordinación».

La institución de seguridad TLÖN se crea bajo la pseudo constitución y se define como una fuerza

permanente con atributos de confidencialidad, integridad y disponibilidad. Estos atributos tienen como finalidad la defensa de los diferentes recursos y la integridad del territorio TLÖN, velando por el cumplimiento de las normas establecidas en la pseudo-constitución. De (Tomic y cols., 2018) se toma el *framework* para definir la institución. Los ingredientes que definen una institución en su nivel más abstracto son los mostrados en la ecuación 4-4, artefactos, roles y acciones.

$$Art = \{art_1, art_2, art_3, ..., art_n\} \tag{4-1}$$

$$Roles = \{rol_1, rol_2, rol_3, ..., rol_n\} \tag{4-2}$$

$$Acc = \{acc_1, acc_2, acc_3, ..., acc_n\} \tag{4-3}$$

Los artefactos (ecuación 4-1) son elementos creados por los agentes o las instituciones y son utilizados por los agentes en sus diferentes roles, las acciones (ecuación 4-3) definen lo que puede o no puede realizar un agente dentro del sistema, por ejemplo, un agente encargado del control de acceso, su rol es gestionar el acceso a la red o alguna aplicación, sus artefactos pueden ser algoritmos de verificación de identidad o una base de datos de los agentes con permiso de ingresar al sistema, y las acciones son permitir o rechazar el acceso a la red teniendo en cuenta lo dictado por las normas. Por lo anterior, una institución se define como:

$$I = \{Art, Roles, acc\} \tag{4-4}$$

Otro elemento fundamental de la institución son las normas, ellas guían el comportamiento de los agentes, ayudan a coordinar sus actividades en varias situaciones sociales y tomar decisiones, las normas convierten el comportamiento en algo predecible. como en (Tomic y cols., 2018) se definen tres tipos de normas, norma de obligación, norma modal y norma cardinal.

- Norma de obligación: Son normas restrictivas tienen calificadores únicos como debe o no debe realizar alguna acción.
- Norma modal: Puede tener aspectos como dónde y cómo las acciones deben realizarse, una norma modal puede tener n-ario calificadores, el cual puede ser usado para especificar una relación entre dos o más declaraciones.
- Norma caridinal: Están presentes en los roles e indican el mínimo y máximo número de agentes que pueden tener un rol.

La organización de agentes para crear una institución se debe al propósito o la necesidad que esta busca subsanar, su objetivo puede ser dictado por una entidad superior, en este caso por una pseudo-constitución, la cual dicta la forma en la que se crean las instituciones, este objetivo puede cambiar dependiendo las necesidades que sean deducidas por las entidades superiores. Para la institución de seguridad se define:

- Objetivo: La institución de seguridad TLÖN tiene como objetivo primordial la formulación y adopción de las políticas, planes generales y proyectos de seguridad, para garantizar la disponibilidad, integridad y confidencialidad del sistema de cómputo TLÖN.

Las funciones de la institución definen cómo se van a cumplir los objetivos, son dispuestas por la pseudo-constitución, para la institución de seguridad se consideró una estructura jerárquica ya que la complejidad del sistema TLÖN y sus diferentes características en cada capa del modelo facilitan la descripción y modelamiento de soluciones para proporcionar características de seguridad. La institución de seguridad TLÖN tiene las siguientes características:

- Participar en la formulación de la política del Estado TLÖN en los temas que les correspondan y adelantar su ejecución: la institución debe ser capaz de interactuar con diferentes instituciones del sistema, creando y proponiendo normas.
- La dirección de la institución corresponde al Estado superior de seguridad y defensa: la dirección de la institución estará a cargo de un agente encargado de direccionar las políticas dispuestas por el Estado.
- Coordinar con el gobierno y las demás instituciones los temas relativos al cumplimiento de la misión del Sector Defensa: realizar tareas con diferentes instituciones para cumplir un objetivo general del sistema

4.2. Funcionamiento de la Institución de Seguridad TLÖN

La institución tiene una estructura jerárquica para su funcionamiento, como parte de sus interacciones sociales contiene consejos de seguridad que funcionan cuando se requiere tomar alguna acción, coordinar tareas y disponer decisiones que afecten el sistema. El Estado TLÖN contiene todos los recursos del sistema, los coordina y controla, por esta razón el estado es el encargado de disponer los recursos que deban ser protegidos por la institución de seguridad. Los recursos son cualquier aplicación, herramienta o componente físico o lógico presentes en el estado.

4.2.1. Estado Superior de Seguridad y Defensa

Es la más alta autoridad de la institución, es el enlace de comunicación entre el Estado TLÖN y los componentes de la institución, esto quiere decir que en este nivel se tiene la comunicación directa con las otras instituciones y el estado. Existen diferentes elementos que son manejados por el Estado superior:

- Base recursos del sistema por controlar: El estado TLÖN define cuáles son los elementos que deben ser cuidados y en qué nivel. En computación se podría representar como una base de datos distribuida con elementos como espectro de difusión, memoria, CPU, anchos de bandas y cualquier elemento que sea un activo del Estado.

- Recursos de la institución: La institución debe tener una base con los mecanismos o artefactos con los que cuenta la institución para hacer cumplir las políticas de seguridad. son creados por el Estado TLÖN, manejados y administrados por el comando de las fuerzas de seguridad y desplegados por los gestores de seguridad, los artefactos se construyen en el marco de la integridad, confidencialidad y disponibilidad.

- Políticas de seguridad: Las políticas de seguridad definen los controles que se deben implementar en sistemas computacionales para lograr un objetivo, estos controles definen la mejor forma en la que se puede asegurar un sistema. El estado TLÖN en su separación de poderes construye y definen todas las políticas de seguridad para el sistema, estas se representan como documentos que deben ser interpretados por el estado superior.

Estos elementos son las principales fuentes de comunicación entre la institución de seguridad y el Estado TLÖN, debe existir un mecanismo o algoritmo capaz de recolectar esta información, además, se necesita de una organización entre el Estado y la institución para poder discutir y relacionar las políticas de seguridad, los recursos que se deben proteger y con qué artefactos, a estos consensos se les llama acuerdos, Los acuerdos son decisiones, que toma el consejo (el Estado TLÖN y la institución), referidas a asuntos específicos de interés institucional, que expresan la voluntad del órgano de gobierno para practicar un determinado acto o sujetarse a una conducta o norma institucional. Al finalizar, el comando deberá tener una base de conocimiento con los acuerdos que será transferida y discutida con las tres dependencias de la institución (integridad, disponibilidad, confidencialidad).

El Estado TLÖN es el encargado de designar al agente representante del estado superior, este agente debe ser capaz de asistir a los consejos con el estado y tener comunicación con todas las dependencias de la institución. El Estado superior puede tener otros agentes de apoyo que asisten las actividades del Estado superior, sin embargo, no pueden influir en toma de decisiones ni tienen poder sobre las dependencias de la institución.

4.2.2. Comando General de las Fuerzas de Seguridad

El Comando General de las fuerzas de seguridad es un órgano colegiado con el más alto nivel de planeamiento y dirección estratégica para las fuerzas de seguridad TLÖN. Bajo su mando están las fuerzas de disponibilidad, integridad y confidencialidad del sistema, dictan las directrices y políticas que sean dispuestas por el Estado TLÖN y el Estado superior de seguridad y defensa de las fuerzas. El cumplimiento de los acuerdos puede necesitar la implementación de operaciones que requieran la combinación de dos o más fuerzas de seguridad, por ejemplo, disponibilidad e integridad. En el consejo se toman decisiones de cómo se implementarán las políticas de seguridad, qué mecanismos son los necesarios para hacer cumplir las políticas y coordinar de manera conjunta las operaciones de seguridad. este organismo está formado por:

1. Estado superior de seguridad y defensa: Debe hacer presencia el agente a cargo del Estado superior, es el que presenta las directrices y políticas que fueron dispuestas por el estado TLÖN y el Estado superior de seguridad y defensa.

2. El comando de Disponibilidad: Debe haber un representante de la fuerza de disponibilidad que tenga la capacidad de tomar las decisiones y tener el conocimiento total del Estado de su dependencia.

3. El comando de integridad: Debe haber un representante de la fuerza de integridad que tenga la capacidad de tomar las decisiones y tener el conocimiento total del Estado de su dependencia

4. El comando de confidencialidad: Debe haber un representante de la fuerza de confidencialidad que tenga la capacidad de tomar las decisiones y tener el conocimiento total del Estado de su dependencia

5. Un asesor externo: Este puesto puede ser ocupado por un agente de alto nivel en el Estado TLÖN y que haya participado en la creación de las políticas de seguridad.

Todos los consejos que hacen parte de la institución deben tener ciertas características para funcionar, en la figura **4-1**, se hace la representación de un consejo, los agentes deben tener un espacio físico o lógico donde reunirse para desarrollar los consejos, además, deben tener una unidad de procesamiento conjunta que permita organizarse y tomar las decisiones que lleven a los consensos que serán almacenados en una memoria conjunta, se debe contar con un protocolo de comunicación para que los agentes se puedan comunicar entre ellos y un reloj que marque el inicio y final del consejo, finalmente, un módulo de salida con las decisiones y directrices del consejo. El comando general decidirá con qué frecuencia se realizan los consejos y que duración deben tener.

4.2.3. Consejo de Disponibilidad, Integridad y Confidencialidad

Cada dependencia de las fuerzas tiene su propio consejo, en él se coordinan las tareas y se dictan las directrices que tienen que ver únicamente con su área, también se realiza para escoger al representante y líder de la dependencia, quien participará en el consejo del comando general de las fuerzas de

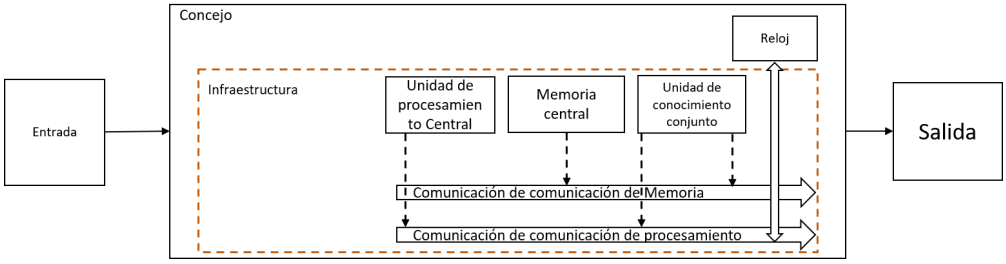

Figura 4-1: Funcionamiento Institución de Seguridad TLÖN

seguridad. las tareas deben ser distribuidas, discutidas y organizadas para llegar a un consenso y de esta forma distribuirlas a los gestores de seguridad. En los consejos participarán:

1. Líder de dependencia.
2. Gestor de seguridad de aplicaciones.
3. Gestor de seguridad de agentes.
4. Gestor de seguridad virtualización.
5. Gestor de seguridad infraestructura.

En estos consejos se evalúa el cumplimiento de las políticas de seguridad y el estado de los gestores de seguridad, los consejos evaluarán el cumplimiento de las políticas, es posible ajustar las políticas basados en la base de conocimiento obtenida por los gestores, si los consejos no pueden solucionar los problemas para ajustar las políticas deben ser llevadas a él comando general de las fuerzas.

4.2.4. Gestores de Seguridad

Son los agentes dispuestos por el Estado TLÖN para hacer cumplir las políticas de seguridad y proteger los recursos del territorio, están administrados y coordinados por el consejo de dependencia (disponibilidad, integridad y disponibilidad), y su principal función es proteger los recursos del sistema. Los gestores de seguridad pueden manejar más de un artefacto, por ejemplo, un detector de fallas y un enlace redundante para la disponibilidad, administran una base de conocimiento con los resultados de utilizar los artefactos para cumplir las políticas, ese conocimiento es presentado en los consejos para evaluar el desempeño del artefacto, definen la frecuencia con la que se ejecutan los artefactos de seguridad y cómo se controlan. Existen 4 tipos de gestores de seguridad cada uno por dependencia (confidencialidad, integridad, disponibilidad):

1. Gestor de seguridad de aplicaciones: Las aplicaciones son los servicios que el sistema provee, son controladas y dirigidas por agentes. El gestor debe tener conocimiento de qué aplicaciones están en el sistema y qué función cumplen.

2. Gestor de seguridad de agentes: Los agentes y las comunidades de agentes están expuestos a diferentes vectores de amenaza, son parte de la inteligencia artificial del sistema, tienen arquitectura distribuida y están soportados por la virtualización y la infraestructura como entorno de movilidad, por esta razón los gestores de seguridad deben ser capaces de interactuar entre agentes, tener bajo su mira todos los agentes que hacen parte del sistema además de sus interacciones.

3. Gestor de seguridad virtualización: El gestor debe ser capaz de percibir las abstracciones lógicas creadas a partir de la infraestructura del sistema.

4. Gestor de seguridad infraestructura: El territorio define el dominio donde el sistema es implementado, el gestor debe ser capaz de garantizar la seguridad en todo el territorio.

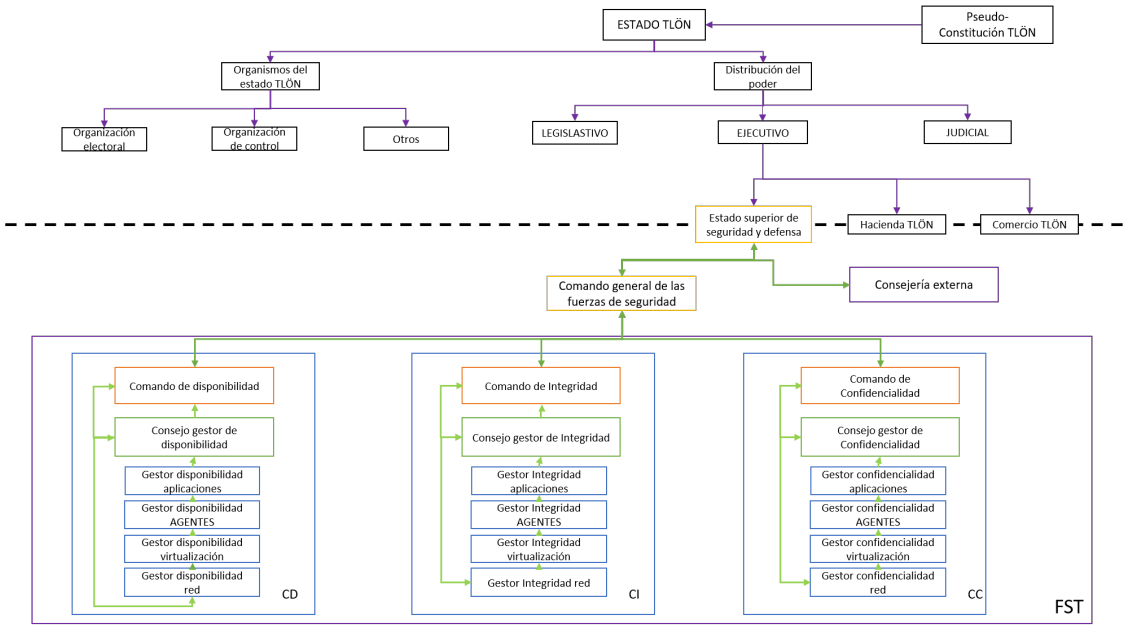

Figura 4-2: Modelo de Seguridad TLÖN

En la figura **4-2** se ve la estructura general de la institución de seguridad TLÖN, la función principal es proveer seguridad en todo el sistema en el marco de la confidencialidad, integridad y disponibilidad. En primera instancia la institución toma elementos como la base de recursos del sistema y las políticas de seguridad. Estas medidas provienen de consensos e intercambio de información entre las instituciones del Estado y alimentan la base de la institución de seguridad a través del Estado superior de seguridad y defensa. El Estado superior se encarga de recibir y mantener una base con los recursos que controla la institución, además se alinea con las políticas institucionales, procesa la información que luego es enviada al comando general de las fuerzas de seguridad. El comando general por ser un cuerpo colegiado requiere hacer un consejo entre los diferentes representantes de las fuerzas (confidencialidad, integridad y disponibilidad), ellos se reúnen para tomar decisiones sobre

la información que es entregada por el Estado superior, al final del consejo se obtendrán las tareas que deberá realizar cada uno de los representantes de las fuerzas, estas tareas se enfocaran en los recursos del sistema, los cuales se protegerán de acuerdo a la base de recursos de la institución de seguridad. Cada representante de las fuerzas tiene su propio cuerpo colegiado, el cual se reúne para discutir lo relacionado con su dependencia, por ejemplo, el consejo de disponibilidad se reunirá cada cierto tiempo para revisar y tomar decisiones que puedan mejorar o mantener la seguridad de su dependencia. Las dependencias cuentan con los gestores de seguridad, los cuales son agentes diseñados para hacer cumplir las políticas de seguridad y garantizar el bienestar de los recursos del Estado.

Una vez que se han desplegado los gestores de seguridad con las políticas y asignados los recursos que van a proteger, la institución comienza a funcionar. Los gestores de seguridad son la base del conocimiento de la institución, ellos son los que recopilan información, vigilan el cumplimiento de las políticas de seguridad, y despliegan los mecanismos que son utilizados para ejecutar las políticas. El conocimiento que es recopilado, es utilizado en los consejos de cada dependencia (Confidencialidad, Integridad, Disponibilidad) para revisar el estado y cumplimiento de las políticas de seguridad, de acuerdo a lo establecido en el consejo, si se encuentra algún problema con la política de seguridad, la información será enviada al comando general de las fuerzas. El comando general reunirá su consejo con un representante de cada dependencia de las fuerzas y deliberará si el problema con la política de seguridad puede ser resuelto con la ayuda de otra dependencia, por ejemplo, alguna aplicación que necesite cierto nivel de disponibilidad y confidencialidad, si se logra un consenso sobre el problema, se ajusta la política de seguridad y continua su funcionamiento, si no se llegó a un consenso con la política de seguridad, el problema será escalado con el Estado superior de seguridad y defensa quien presentará el problema de la política al Estado TLÖN.

4.3. Generalidades de la Disponibilidad

La disponibilidad se refiere a la aplicación o servicio accesado por un agente final realizando un trabajo y este depende del recurso de los dispositivos como redes, sistemas de *software y hardware* o cualquier componente cuyo daño podría afectar un servicio o aplicación, se expresa comúnmente como la relación entre el periodo de actividad aceptable del sistema y el tiempo total en un periodo determinado (Edgar y Manz, 2017). La disponibilidad en el sistema se puede ver afectada por ataques activos y pasivos, los cuales pueden exponer los componentes de hardware, software, energía, ambiente, enlace, configuración (dinámica de cambio), utilización del recurso y diseño, sin embargo, se puede ver comprometida por actividades no relacionadas a un ente malicioso, por ejemplo, un fallo de energía (Jones, 2001). Se dice que un nodo es malicioso cuando exhiba características como perdida de paquetes, agotamiento anormal de batería, consumo del ancho de banda, entre otros (Teufel, Min, You, y Weippl, 2014).

En redes convencionales (redes cableadas e inalámbricas con estructura fija) la disponibilidad suele

ser medida como una variable relacionada con el tiempo como se ve en la ecuación 4-5. Se representa como una probabilidad de que el sistema sea capaz de realizar su función cuando sea solicitada, como se ve en la figura **4-3**, la disponibilidad se mide en función del tiempo, esto es útil para deducir tiempos de recuperación y diseño de las soluciones para contrarrestar la falta de disponibilidad.

$$A = \frac{Tiempotrabajando}{Tiempotrabajando + Tiempodenotrabajo} \tag{4-5}$$

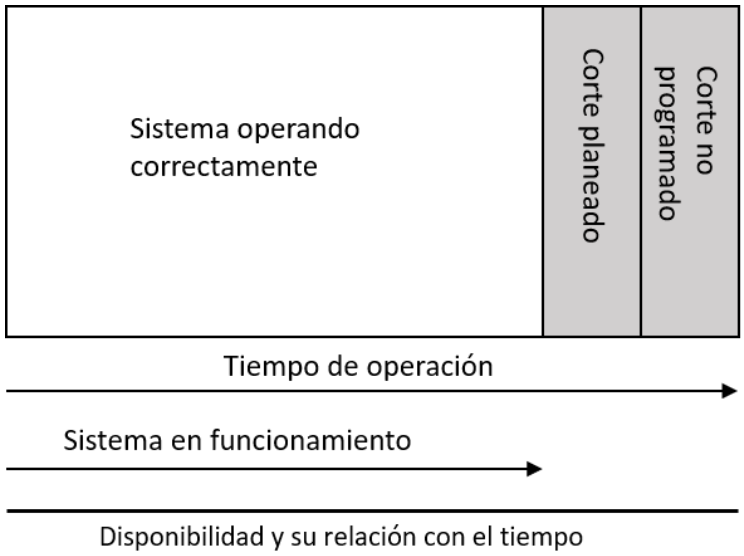

Figura 4-3: Disponibilidad y su Relación con el Tiempo

una forma de incrementar la disponibilidad de un sistema es duplicar los componentes de tal forma que si uno falla el otro pueda tomar su lugar, en este caso el componente de reemplazo debe ser capaz de soportar la carga de trabajo y transferir el control. La redundancia puede implicar tomar algunas acciones para hacer uso de los componentes redundantes en la recuperación de un sistema, la tolerancia a fallas la definen en (Critchley, 2016) como redundancia inteligente.

4.4. Prototipo de la Dependencia de Disponibilidad

La capa transversal de seguridad del modelo TLÖN propone un reto de investigación debido a las características presentes en un sistema distribuido dinámico, por esta razón, se buscó desarrollar un módulo de disponibilidad para sistemas distribuidos inalámbricos, que funcione a nivel de la capa de red Ad hoc y la capa de virtualización del modelo TLÖN. Como se vio anteriormente la seguridad para un sistema puede ser contenida en tres principios, disponibilidad, integridad y confidencialidad, el

sistema TLÖN se rige bajo estos tres principios para todas sus capas, sin embargo, en este trabajo se hace énfasis en la disponibilidad.

En las MANET la arquitectura de la disponibilidad depende de la infraestructura de red, puede ser plana o jerárquica de acuerdo a las aplicaciones. Cuando un diseño de MANET tiene una estructura plana todos los nodos son considerados del mismo nivel, por lo tanto, la red debe considerar cuáles nodos son los más aptos para asumir el rol de seguridad o si será una estructura cooperativa. Para una estructura jerárquica los nodos tienen un nivel diferente, forman agrupaciones y son gestionados por un nodo líder. Las gestiones de disponibilidad en las redes convencionales (redes cableadas o inalámbricas con infraestructura fija) no se pueden trasladar a las MANET, ya que su comportamiento dinámico dificulta el uso de las mismas medidas, por lo tanto, se debe tener en cuenta el uso de nuevas herramientas que permitan tener una visión del comportamiento y disponibilidad de los nodos en la red.

4.4.1. La calidad Transmitida B.A.T.M.A.N.

B.A.T.M.A.N por ser un protocolo proactivo mantiene información sobre todos los nodos en la red que son accesibles vía salto único o multi salto, la estrategia de B.A.T.M.A.N es buscar el mejor camino para comunicarse con el nodo de destino, para esto aprende las rutas que tienen camino hacia su destino y selecciona la mejor. La red se inundará con mensajes OGM (mensajes originadores), el número de OGM recibidos desde un originador dado a través de cada vecino local es usado para calcular la calidad de una ruta (ruta de un solo salto o múltiple salto), para calcular la mejor ruta, B.A.T.M.A.N contará los mensajes recibidos y los logs, los cuales han sido enviados por el vecino de enlace local.
B.A.T.M.A.N divide la calidad de un enlace en dos partes: calidad del enlace receptor y calidad del enlace de transmisión, la calidad de recepción describe la probabilidad de éxito de transmisión hacia el nodo, la calidad de enlace de transmisión describe la probabilidad de éxito en la transmisión hacia un nodo vecino (batman.adv foundation, 2018). Como se ve en la figura **4-4** el protocolo puede conocer la calidad de recepción contando paquetes de los nodos vecinos, también se conoce la calidad de enlace de echo contando las retransmisiones de sus propios vecinos, finalmente como se ven en la ecuación 4-6 se puede calcular la calidad de transmisión dividiendo la calidad de echo (EQ) y la calidad de recepción (RQ).

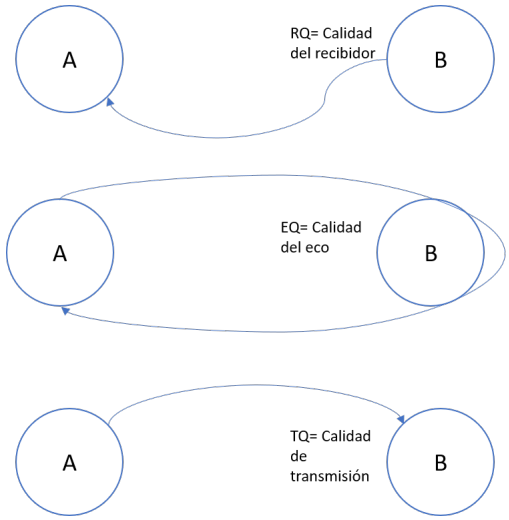

Figura 4-4: Calidad de Transmisión

$$TQ = EQ/RQ \tag{4-6}$$

La calidad de enlace local debe ser propagada a través de la red para informar a los otros nodos sobre la calidad de transmisión, como se ve en la figura **4-5** cuando un mensaje originador es generado se asume una calidad de enlace local máxima (255), el vecino que reciba el mensaje calculará su propia calidad de enlace local por medio de la ecuación 4-7 y enviará el mensaje, así cada nodo que lo reciba sabrá la calidad de transmisión hacia el nodo originador.

$$TQ = TQ(entrante) * TQ(local) \tag{4-7}$$

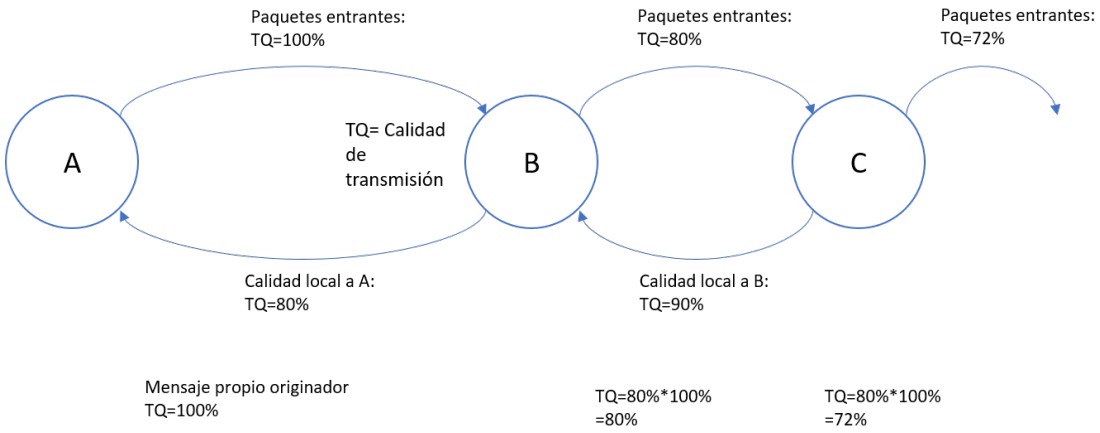

Figura 4-5: Propagación de la Calidad de Transmisión

B.A.T.M.A.N realiza el seguimiento de dos diferentes valores TQ:

1. El TQ local que representa la calidad de transmisión hacia cada salto único (calculado por el recuento de paquetes más el TQ).

2. La calidad del enlace global la cual describe la calidad del enlace hacia cada vecino multi salto.

3. Calidad de enlace de recepción.

El valor de TQ está definido en una escala de 0 y 255 (donde 0 indica que no hay conexión y 255 indica una excelente calidad). Cuando llega un paquete de datos para la transmisión, el nodo se refiere a la tabla de originadores para determinar la dirección en la que se enviarán los paquetes. Específicamente, comprueba su tabla de originador y reenvía el paquete hacia el destino con la entrada de rango más alto: el mejor vecino siguiente. Si un nodo o un enlace es degradado o falla, se verá reflejado en la métrica TQ y el nodo o el enlace será evitado, y sí existe, se buscará un mejor camino disponible.

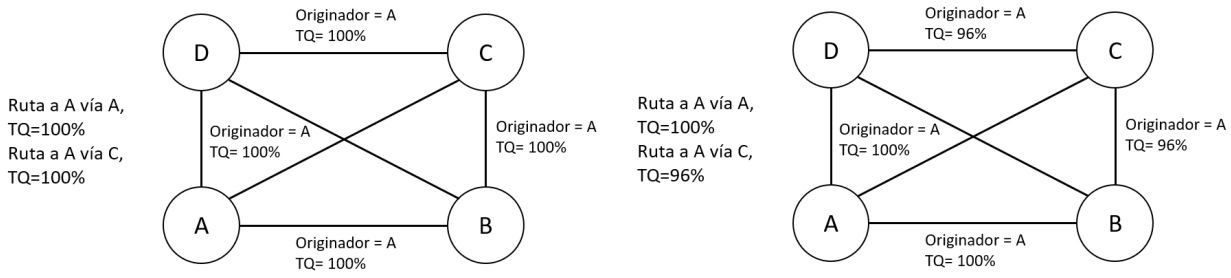

Figura 4-6: Creación de Rutas

B.A.T.M.A.N no solo utiliza la calidad del enlace para calcular la mejor ruta a su destino, en ciertas configuraciones de redes la calidad de enlace puede ser muy similar pero el número de saltos puede ser mayor, en esos escenarios es deseable escoger el camino más corto para reducir el ancho de banda y reducir la latencia. En la figura **4-6** existen cuatro nodos con calidad de enlace similar, en este escenario el nodo D puede asumir que el mejor camino para llegar al nodo A es a través del nodo C, sin embargo, tienen una comunicación directa que sería el escenario ideal. B.A.T.M.A.N utiliza una penalización de salto para favorecer a las rutas directas, cada vez que un mensaje atraviese un nodo se resta cierto porcentaje a la calidad del enlace.

B.A.T.M.A.N acepta paquetes arbitrarios desde cualquier fuente y construye su tabla de enrutamiento analizando las estadísticas de los mensajes originadores recibidos. Existen diferentes escenarios que limitan el correcto funcionamiento y disponibilidad de la red, si un nodo A alcanza un nodo C por medio de un nodo B en una cierta localización geográfica como se ve en la figura **4-7**, y dado el caso donde el nodo B deje de funcionar o el enlace de transición TQ se degrade, la comunicación entre el nodo A y el nodo C se vería afectada(batman.adv foundation, 2018). Puede existir el escenario de la figura **4-8** en el que los nodos se comuniquen con una calidad de enlace TQ de 255 y después de cierto tiempo o un ataque de denegación de servicio se degrade la calidad del enlace, o en el peor de

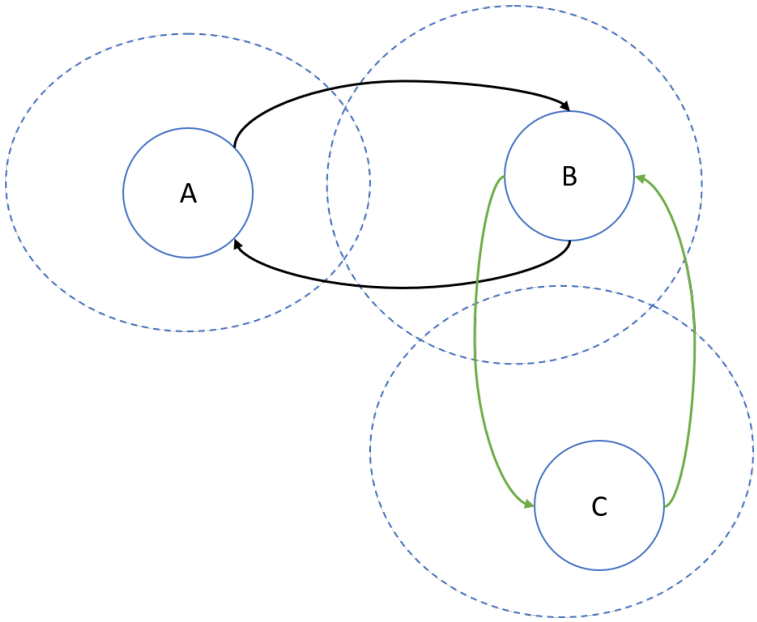

Figura 4-7: Mensaje hacia el nodo C a través de B

los casos se pierda la comunicación.

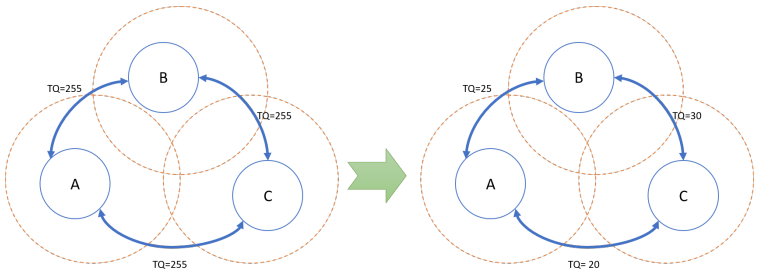

Figura 4-8: Degradación de la Calidad del Enlace

como se ve en la tabla **4-1** la calidad del enlace proporciona información sobre cómo B.A.T.M.A.N evalúa este enlace. 1.00 significa 100 % calidad de enlace, 2.00 significa 50 %, 3.00 es 33.3 % y 4.00 es 25 %, etc. El número dice cuántos paquetes necesita enviar para obtener una sola transmisión exitosa.

4.4.2. Modelo Propuesto

se podría pensar en un escenario distribuido como el que se muestra en la figura **4-9**, el cual contiene en una aplicación distribuida que funciona por medio de un nodo maestro que es el encargado de repartir tareas a los nodos trabajadores a través de la red Ad hoc y de esta forma solucionar un problema.

TQ	Porcentaje de paquetes
255	100 %
235	92 %
205	80 %
205	80 %
175	69 %
145	57 %
125	49 %
95	37 %
35	14 %
0	0 %

Tabla 4-1: Porcentaje de pérdida de paquetes

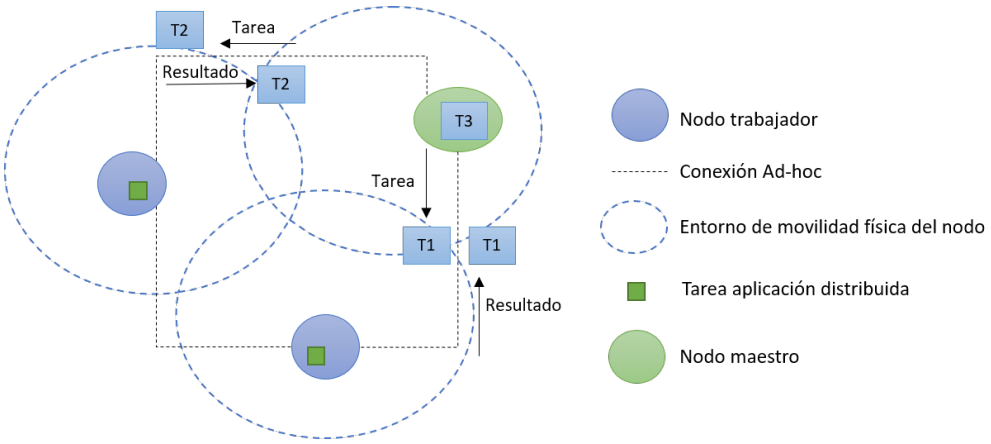

Figura 4-9: Entorno Ad hoc Distribuido

Una combinación de, una abstracción de procesos, abstracción del enlace y una abstracción de un detector de fallas, definen un modelo de sistema distribuido para alcanzar una alta confiabilidad y seguridad del sistema (Cachin y cols., 2014). En la Figura **4-10** se muestran los diferentes modos de falla con los que se deberían desarrollar los procesos en los sistemas distribuidos. Una falla ocurre cuando el proceso no se comporta de acuerdo al algoritmo, cuando el proceso falla, todos los componentes fallan al mismo tiempo (Elser, 2012). En un sistema distribuido, es posible que los mensajes se pierdan al transitar a través de la red y de esta forma comprometer la disponibilidad, por esta razón, es importante su consideración en el diseño de un sistema seguro y confiable. Cuando un conjunto de acciones trata de comprometer los atributos de seguridad como confidencialidad, integridad y disponibilidad, esas acciones se definen como maliciosas y la detección de tales acciones se definen como sistemas de detección de ataques. Las funciones básicas de un sistema de detección de ataques dependen de componentes como recolección de datos, detección y respuesta. la recolección de los

datos puede venir desde varias fuentes, tráfico de red, calidad de comunicación, etc. El módulo de detección es responsable de analizar la información obtenida y tomar decisiones sobre un posible ataque al sistema. La detección de ataques puede ser dividida en detección de Host y detección de red. entre las formas de detección de ataques se destacan:

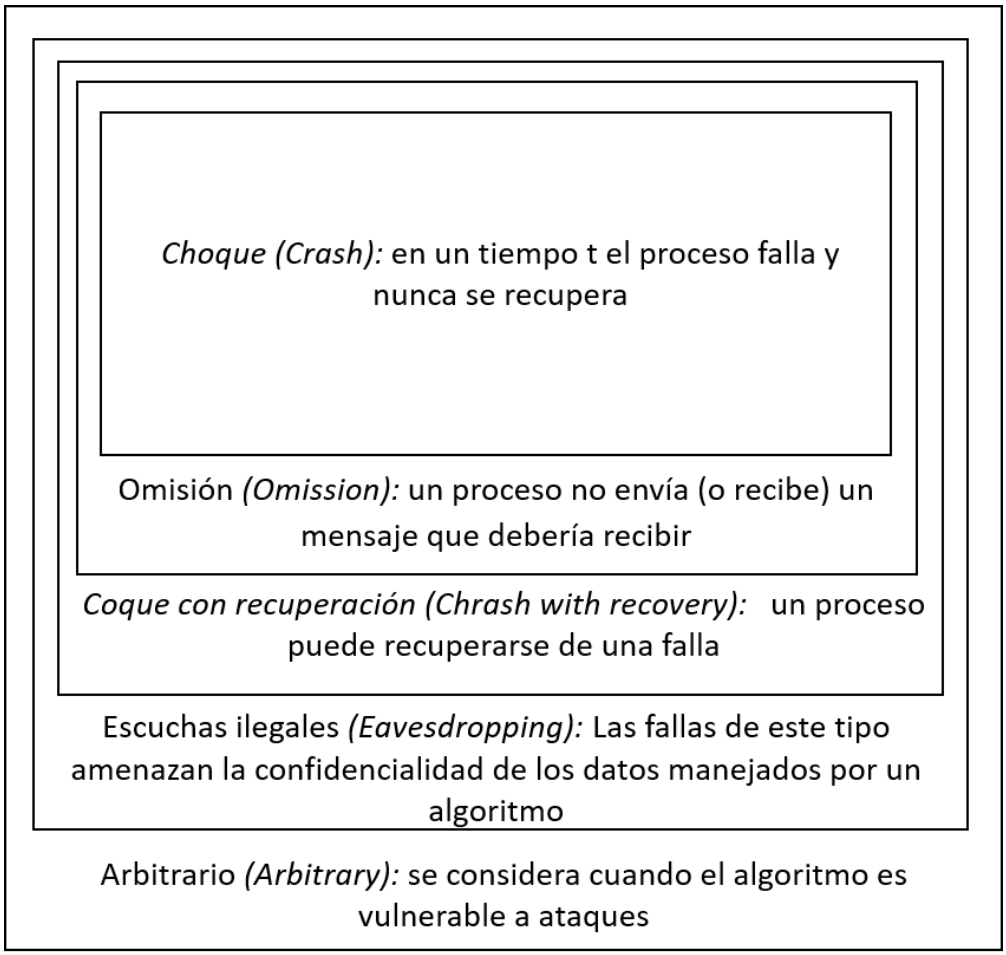

Figura 4-10: Tipos de Fallas de Procesos

- Arquitectura local: Cada nodo es independiente y debe detectar sus ataques, recolecta su propia información, no tienen relación con los otros nodos, por lo tanto, es una técnica poco efectiva para sistemas dinámicos.
- Arquitectura distribuida o cooperativa: En esta arquitectura todos los nodos colaboran para detectar un posible ataque, un nodo maestro es el encargado de recolectar toda la información de los nodos y de forma cooperativa toman decisiones y realizan una detección global.
- Arquitectura jerárquica: En esta arquitectura los nodos maestro o *cluster head* son los encargados de realizar la detección y control de los ataques, pueden cooperar con nodos de otros *clusters*.

Diferentes modelos para detectar intrusos o fallas han sido propuestos como (Kazi y Adhoni, 2016) en el cual se implementa un sistema de detección de intrusos utilizando el protocolo de enrutamiento DSR (*Dynamic Source Routing*) y hace uso de paquetes de reconocimiento desde la fuente hasta el destino para detectar el comportamiento del nodo al que envía el mensaje, si el nodo destino reenvía el mensaje de reconocimiento la comunicación puede ser establecida. En (Puttini, Percher, Me, y de Sousa, 2004) cada nodo corre un detector de intruso local que coopera con los otros nodos, se utiliza una técnica de redundancia que compensa la movilidad y dinámica de los nodos. En (Chadli, Emharraf, Saber, y Ziyyat, 2014) proponen una arquitectura de detección de intrusos para MANET, esta arquitectura es un modelo de combinación jerárquico basado en clusters y un modelo de cooperación basado en un sistema multiagente. En esta arquitectura, los agentes utilizan un conocimiento relacionado con una ontología de seguridad global, que se puede usar para inferir nuevas reglas de detección.

Un detector de fallas puede ser un elemento necesario en el diseño de un modelo de sistema distribuido, provee información sobre cuales procesos están en falla y cuales están funcionando. En el modelo propuesto de la Figura **4-11**, se busca mitigar la posible falta de disponibilidad que puedan surgir de los procesos que fallen o los enlaces que no sean óptimos para una comunicación, para esto, se propone diseñar un nodo que sea un detector de fallas y tenga un monitoreo constante sobre la red, las métricas usadas para esto serán la dirección IP, la dirección física MAC y la calidad del enlace asociada. El agente gestor de seguridad puede decidir incluir diferentes funciones para proteger la red, su cooperación con el agente líder u orquestador puede proporcionar una visión más amplia del comportamiento de los nodos y aplicaciones que en algún momento puedan sufrir un ataque, por la dinámica de la red se consideró que el papel del detector de fallas pueda ser desempeñado por cualquier nodo excepto el nodo orquestador, a continuación, se describen los componentes propuestos del detector de fallas.

- Módulo de comunicación: Este módulo se encarga de entablar una comunicación con los diferentes nodos en la red, y de esta forma, intercambiar las métricas de medición que serán utilizadas por el nodo gestor de seguridad para medir la disponibilidad de los demás nodos en la red, las métricas serán solicitadas por el nodo gestor de seguridad cada cierto tiempo. Cada nodo en la red enviará su dirección IP, MAC, información de calidad del enlace (TQ), y podría incluir alguna información asociada con alguna aplicación que esté utilizando.

- Estados del nodo/sistema: Este módulo se encarga de procesar la información obtenida por el módulo de comunicación, su tarea principal es organizar las diferentes métricas para su estudio. Debe mantener una actualización constante sobre la red, ya que algunos nodos podrían entrar o salir de la red constantemente.

- Políticas de estado y módulo comparador: Cada nodo en la red debe mantener una base con las principales políticas de estado. Las políticas de estado son condiciones y reglas que se establecen para un correcto funcionamiento de la red, por ejemplo, si un enlace de comunicación

cae por debajo del 50 % podría estar faltando a una política de óptimo desempeño de la red. El módulo de estados del sistema entregará la información de los nodos al módulo comparador, y de esta forma, los nodos serán verificados basándose en las políticas de Estado que hayan sido establecidas para el sistema. Adicionalmente, las políticas de estado pueden ser alimentadas por un nodo orquestador, el cual es una figura creada para asignar recursos y distribuir tareas (nodo maestro).

- Detección de fallas y módulo estadístico: Finalmente, si el módulo comparador encuentra algún nodo que este fallando se detecta una falla y podría ser enviado el reporte al módulo estadístico y al nodo orquestador donde se tendrán estadísticas de fallas, *logs* y medidas de concurrencia.

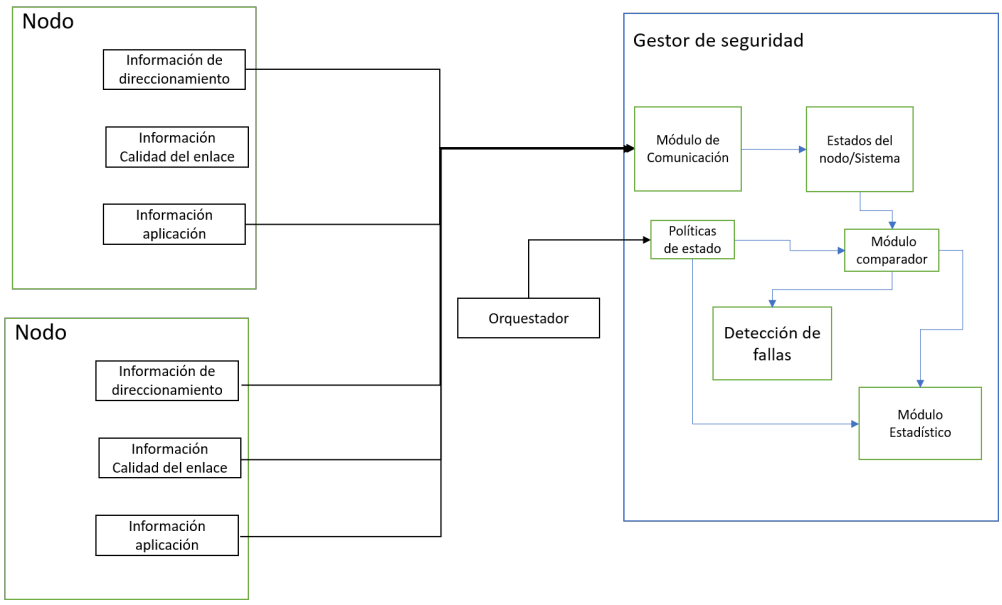

Figura 4-11: Modelo de Disponibilidad Para el Sistema TLÖN

4.4.3. Mecanismo de Detección de Fallas

Para la validación del detector de fallas se utilizaron cuatro dispositivos raspberry pi modelo 3 para la construcción de un cluster físico y lógico Ad hoc con todas sus características dinámicas y estocásticas. En el modelo de pruebas se utilizaron herramientas como Batman.adv, python, Docker, entre otras, sin embargo, existen elementos restrictivos que limitan el desarrollo del sistema deseado.

El esquema aprovecha la interfaz inalámbrica de los dispositivos embebidos para activar el módulo batman-adv que viene incorporado dentro del kernel de linux, este módulo permite utilizar las características del protocolo batman.adv el cual permite auto configurar direcciones IPs mediante el servicio avahi, las direcciones IPs son diseminadas por la red mediante el uso del servicio de mensajería ALFRED, esto permite que los elementos como el orquestador y el gestor de seguridad puedan

anunciarse en la red.

Como se vio en la sección 2.4.3.2 el agente local es la base para el mecanismo detector de fallas, por lo tanto, utiliza un modelo de comunicación sobre la red con un conjunto de servidores TCP y UDP multi hilo, esta comunicación sirve para intercambiar información con el orquestador sobre el estado actual y contiene un servidor activo para recibir tareas o mensajes asignados por el orquestador. El gestor de seguridad pertenece a la institución de seguridad del sistema TLÖN, por esta razón, él intercambia información y recibe tareas de los agentes pertenecientes a la institución.

En el algoritmo 1 está la rutina base del módulo detector de fallas y las tareas que realiza para mantener un monitoreo y comunicación constante con los miembros del sistema, el módulo funciona sobre un agente local, esto quiere decir que el nodo también realiza funciones como el despliegue de aplicaciones en contenedores, los cuales se despliegan teniendo en cuenta el número de núcleos del dispositivo, finalmente, se guarda el histórico de consumo de recursos en el nodo, estas funciones y la interacción con el orquestador representan las dos capas del modelo TLÖN, y por lo tanto se toman como base para el desarrollo del módulo de seguridad propuesto. En las líneas 1-4 se inician los servicios de comunicación MANET, estos servicios son importantes porque le permiten al detector de fallas descubrir el entorno del sistema en el que se encuentra y comienza a descubrir elementos como el orquestador y otros representantes de la institución de seguridad. En la linea 5-10 está el proceso principal donde se tienen arreglos que guardan un histórico de la medida de calidad de enlace y las direcciones de los nodos en la red, esta tarea se realiza de manera constante y mediante la ejecución de las políticas de seguridad se decide enviar los estados almacenados al representante superior de la institución. En la figura **4-12** se ven las tablas generadas por el detector de fallas en tiempo de diseño y ejecución, estas se encuentran en el agente local y son base de medición y toma de decisiones para el gestor de seguridad.

Tiempo de Ejecución

- NODE: Esta tabla contiene toda la información de los nodos en la red, esta información es recolectada mediante el servicio de reconocimiento ALFRED, los campos de esta tabla son: la dirección lógica IP y la dirección MAC.
- MONITOR: Esta tabla recibe la información del estado de la red está compuesta por: la dirección MAC, Last seen (última vez que fue visto el nodo en la red), TQ.
- MONITOR NETWORK: Esta tabla es formada por las dos tablas anteriores y es la base principal del detector de fallas, está compuesta por: MAC, IP, TQ, LAST SEEN, NEXT HOPE, TIME.

Tiempo de Diseño

- Node_satate: Esta tabla contiene la política de seguridad que utiliza el nodo gestor de seguridad para detectar una falla.

Algorithm 1 Failure Detector

[H]

Require: Ad hoc mode (B.A.T.M.A.N) enable
Require: Message service (A.L.F.R.E.D) enable
Ensure: Communication Network TQ status

```
RUN ADHOC MODE
RUN ALFRED SERVICE
RUN ThreadDFServer(STATES)
RUN Log files
t ← 0
while N ≠ 0 do
    RUN Discovery TQ
    if TQ ∧ ips =!0 then
        mergeddf
        save out
        client send(m,port,mergeddf)
        t + +
    end if
end while
```

El módulo detector de fallas realiza dos tareas fundamentales, mediante el protocolo batman.adv y el servicio de descubrimiento ALFRED se comunica con los diferentes nodos en la red para mantener una base actualizada del estado del enlace y las direcciones físicas y lógicas asociadas a cada nodo. esta rutina es desarrollada en un While, y se toman las medidas cada cierto tiempo, la segunda tarea es la revisión de la política de seguridad que está asociada al recurso del enlace y toma de decisión para enviar la información hacia el nodo superior en la institución de seguridad.

Como se mencionó en el la sección 4.1 el módulo detector de fallas funciona como mecanismo de seguridad que es consultado por el gestor de disponibilidad, por esta razón, es el nivel más bajo dentro de la institución y funciona bajo el modelo Observar-Decidir-Actuar, el cual recibe datos sin procesar, en este caso el recurso de calidad del enlace, luego pasa a la fase de decisión, en la que utiliza las políticas de seguridad que previamente han sido entregadas por la institución, finalmente, actúa con base en la información que ha recolectado y la decisión que haya tomado. Para el detector de fallas, la distribución de la institución se ve en la ecuación 4-8 y 4-9, donde se definen los artefactos de la institución que en este caso es el detector de fallas y los roles que deben estar en la institución para que funcione. Las acciones que debe tener cada rol están en la imagen **4-13**, allí se definen las acciones que desempeña cada rol, el orquestador es la entidad con más alta jerarquía en el sistema y el nodo gestor de seguridad es el nodo el cual implementa el artefacto dispuesto por el Estado TLÖN para desempeñar labores de seguridad. El detector de fallas se construye como la idealización de un mecanismo de seguridad el cual está gobernado por unas políticas de seguridad que sirven para miti-

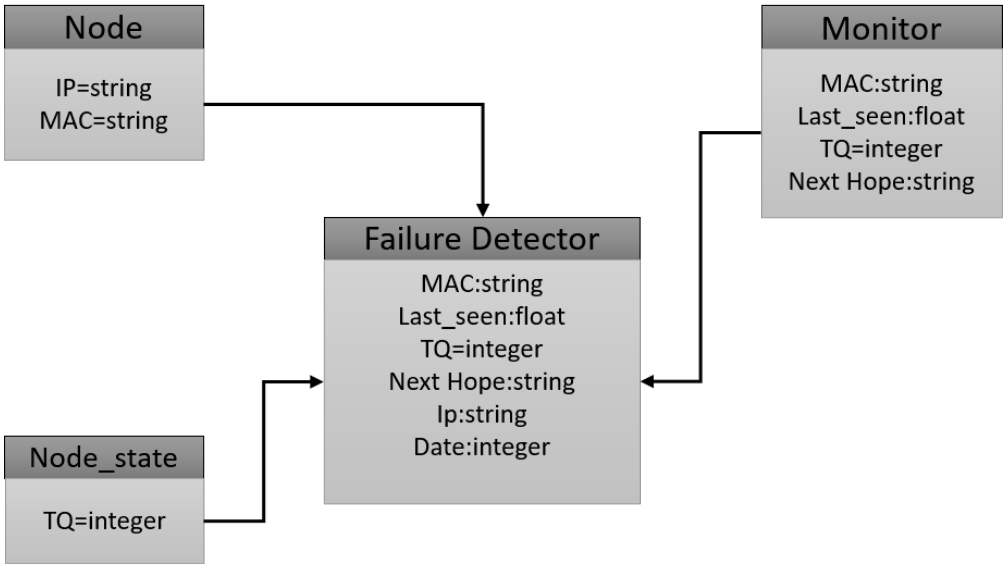

Figura 4-12: Tablas Tiempo de Ejecución Detector de Fallas

gar posibles ataques de denegación de servicio.

$$Art = \{Detector\_de\_fallas\} \tag{4-8}$$

$$Roles = \{Gestor\_de\_disponibilidad, Consejo\_de\_disponibilidad, Comando\_Gnal, ORQ\} \tag{4-9}$$

4.4.4. Políticas de Seguridad y Política Desarrollada

Una política de seguridad define cómo debería comportarse un sistema y utiliza un conjunto de normas o reglas que buscan diferenciar el estado autorizado y no autorizado en un sistema. En (Bishop, 2018) una política de seguridad se define como una declaración que divide los estados de un sistema dentro de un conjunto de comportamientos autorizados, o seguros, no autorizados o inseguros, las políticas describen conductas, acciones y autorizaciones, definiendo, usuarios autorizados y uso autorizado. Las políticas de seguridad por si solas no garantizan un sistema seguro, ellas solo son declaraciones de qué o quién puede hacer o no algo, sin embargo, las políticas pueden ser violadas por entes no autorizados en un sistema, por ejemplo, un *malware*, por lo tanto, los sistemas deben utilizar mecanismos de seguridad que hagan cumplir las políticas. Un mecanismo de seguridad es una herramienta o técnica que se utiliza para implementar los servicios de seguridad, puede funcionar por sí solo, o con otros, para proporcionar un servicio determinado. Los servicios de seguridad especifican

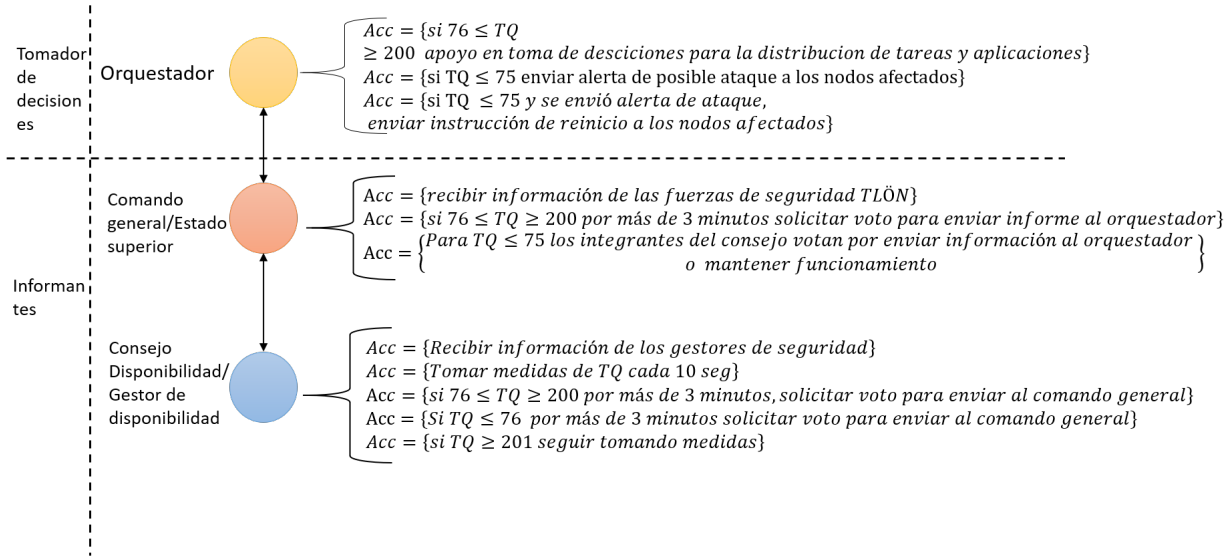

Figura 4-13: Acciones Institución de Seguridad TLÖN

qué controles son requeridos y los mecanismos especifican cómo deben ser ejecutados los controles, por ejemplo, un servicio de seguridad es un control de acceso y el mecanismo es un sistema biométrico para el reconocimiento dactilar de un usuario.

Para que una política de seguridad pueda definir los comportamientos aceptables en un sistema se requiere un contexto que incluya bajo qué escenarios los sistemas pasan de estar en un estado seguro a un estado no autorizado, por esta razón, las políticas se pueden agrupar bajo la disponibilidad, confidencialidad e integridad, lo cual permite descomponer los sistemas en sus diferentes características para manejar de una manera más eficiente las políticas de seguridad, las políticas se pueden agrupar en:

- Políticas de confidencialidad: Estas políticas buscan proteger la confidencialidad de un sistema y prevenir el acceso no autorizado a la información, pueden incluir autorizaciones temporales a localizaciones específicas de un sistema, identificar cuándo un sistema está presentando fugas de información, etc. Existen algunas ocasiones en las que el uso de mecanismos de seguridad pueda dificultar la implementación de las políticas de seguridad, por ejemplo, un mecanismo de redundancia de red para proveer disponibilidad puede ser utilizado por un atacante para espiar los datos que viajan a través de ella, por esta razón, en los sistemas se deben desarrollar las políticas de seguridad teniendo en cuenta el tipo de aplicación que se pretenda proteger.

- Políticas de integridad: Estas políticas identifican cuáles entidades están autorizadas para alterar o realizar cambios en la información, las políticas también describen las condiciones y la forma en la que los datos pueden ser alterados.

- Políticas de disponibilidad: Estas políticas describen cuándo un recurso o elemento del sistema se está comportando de una forma inadecuada, también puede delimitar la forma en la que se puede ingresar o utilizar un recurso y por cuánto tiempo. En algún momento, si no se puede acceder a un recurso se considera una denegación de servicio, esto quiere decir que el recurso no está realizando la acción que debería hacer, existen diferentes formas de denegación de servicio, por ejemplo, una asignación inadecuada de los recursos, factores ambientales, agentes maliciosos, etc.

La creación de las políticas de seguridad se basan en el modelo institucional propuesto en la figura **4-2**. Como se mencionó, las políticas pueden ser creadas por un elemento superior como la pseudo constitución TLÖN, la cual decide que comportamientos son válidos en el sistema y diseña las políticas para proteger uno o varios recursos en el sistema, para el caso de este trabajo, se utilizó el recurso de calidad del enlace como métrica para evaluar la disponibilidad en el sistema, ya que este recurso es el que permite la comunicación entre los agentes locales y el orquestador. Las políticas fueron creadas para utilizar el método Observar-Decidir-Actuar, por el cual se le pide al mecanismo de seguridad recibir datos como entradas, organizarlos y decidir sobre ellos. en la figura **4-14** el mecanismo recibe los datos mostrados en la figura **4-12**, estos datos una vez son organizados se empiezan a procesar por parte del detector de fallas, esta información pasa a una segunda fase de decisión donde se validan las tablas generadas en el tiempo de diseño y ejecución, los campos claves que son utilizados son el TQ y la dirección IP, finalmente, dependiendo la política de seguridad se toma la decisión para actuar frente al estado que se ha observado. El estado de validación de la política le permite al gestor de seguridad enviar la información de los problemas encontrados a la entidad superior en la institución de seguridad.

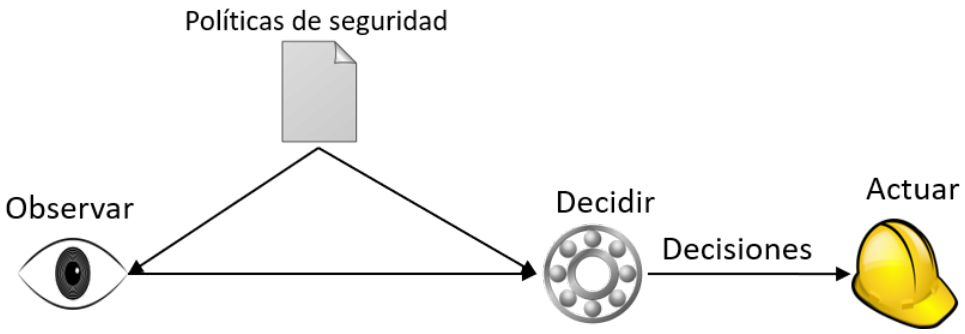

Figura 4-14: Política Observar, Decidir, Actuar

Capítulo 5

Pruebas y Resultados

Para el desarrollo de este trabajo se utilizaron sistemas embebidos (Raspberry pi modelo 3) y las interfaces inalámbricas 802.11 /g/b/c para activar el modo Ad hoc. Sobre estos dispositivos se utilizó distribuciones de sistemas operativos libres trabajando con una versión de kernel 4.14.71-v7+. La aplicación del detector de fallas se desarrolló en Python 2.7, utilizando el protocolo B.A.T.M.A.N.ADV y el servicio de mensajería ALFRED para obtener las métricas que permiten identificar posibles negaciones de servicio en el sistema TLÖN.

5.1. Descripción de las Pruebas

Para la validación del modelo de operación del módulo detector de fallas se seleccionó el ataque de denegación de servicio que se ve en la figura **5-1**, inundación TCP. El ataque de sincronización (*SYN flood*) es el mecanismo más común de inundación, está basado en la iniciación de comunicación TCP, *three way handshake*, como se ve en la figura **5-1** el atacante envía varias solicitudes a la víctima para que falle en recibir el paquete ACK, por lo tanto, mantiene una conexión abierta por un periodo de tiempo. El atacante puede falsificar las direcciones IP fuentes para que la víctima no pueda establecer una conexión, esto evitará que las solicitudes de conexión legitimas no puedan ser establecidas por falta de disponibilidad en el recurso. El ataque se verá reflejado en el consumo de memoria y ancho de banda. Al realizar un tipo de ataque como la inundación TCP, recursos del sistema TLÖN como la calidad del enlace y la memoria de los nodos puede verse afectada. Como se mencionó el detector de fallas utiliza como base el agente local como se ve en la figura **5-2** desde allí toma las medidas que sirven para la toma de decisiones basado en las políticas de seguridad. Para validar el funcionamiento del detector de fallas en las dos primeras capas del sistema TLÖN se utilizaron los escenarios de pruebas mostrados en la figura **5-2**, las iteraciones se hacen de manera directa sobre el canal inalámbrico con apoyo del servicio de mensajería ALFRED y B.A.T.M.A.N, haciendo un poco más ligera la carga de información sobre la red. Para el ultimo escenario se propone el funcionamiento de la institución de seguridad TLÖN. Las siguientes son las operaciones realizadas por cada integrante del sistema:

- Agente local: Sus operaciones son la medición del consumo cpu, medición del consumo de me-

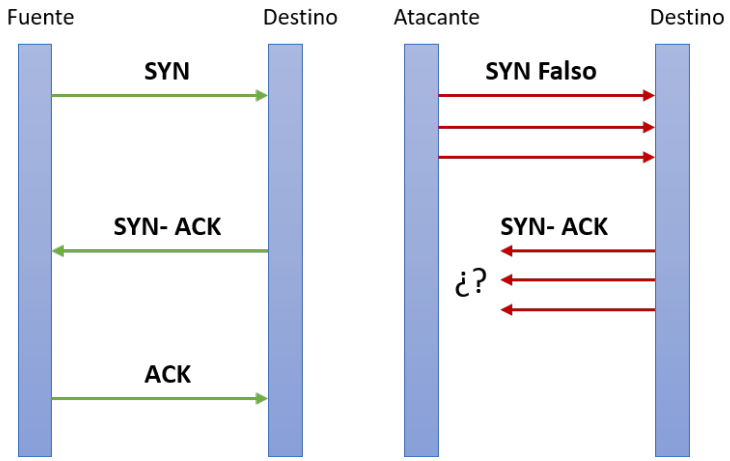

Figura 5-1: Ataque *SYN flood*

moria RAM, medición del *throughput* de las aplicaciones en los contenedores, gestión del espacio de almacenamiento asignado y mapeo de dispositivos de entrada y salida (bajo este esquema otras tareas pueden ser añadidas como parte de las rutinas del Agente Local)

- Orquestador: Este artefacto computacional es otro tipo de agente sobre el sistema con mayor jerarquía, tiene la capacidad de indicar los comandos a realizar por los Agentes Locales de acuerdo con las políticas del sistema, contiene toda la información del sistema en tablas que le permiten conocer la arquitectura del dispositivo, el estado actual del dispositivo, el máximo *throughput* de las aplicaciones en un dispositivo y el estado actual de la aplicación.

- Detector de fallas: El agente local es la base del detector de fallas, desde ahí se realizan todas las actividades de monitoreo en la red manteniendo una base de todos los dispositivos conectados en la red y almacenando las métricas de calidad del enlace y direcciones IP que son utilizadas por las políticas de seguridad para tomar decisiones en la institución de seguridad TLÖN, este es el aporte de este trabajo al sistema TLÖN.

Como ejemplo, el despliegue de un detector de fallas se puede ver en la figura **5-3**, en ella se evidencia el intercambio de información entre los nodos de la red. El agente local es la base del detector de fallas, por esta razón se toman medidas del estado del nodo, sus contenedores y aplicaciones. El detector de fallas utiliza la información que está dentro de sus tablas para tomar decisiones considerando la política de seguridad.

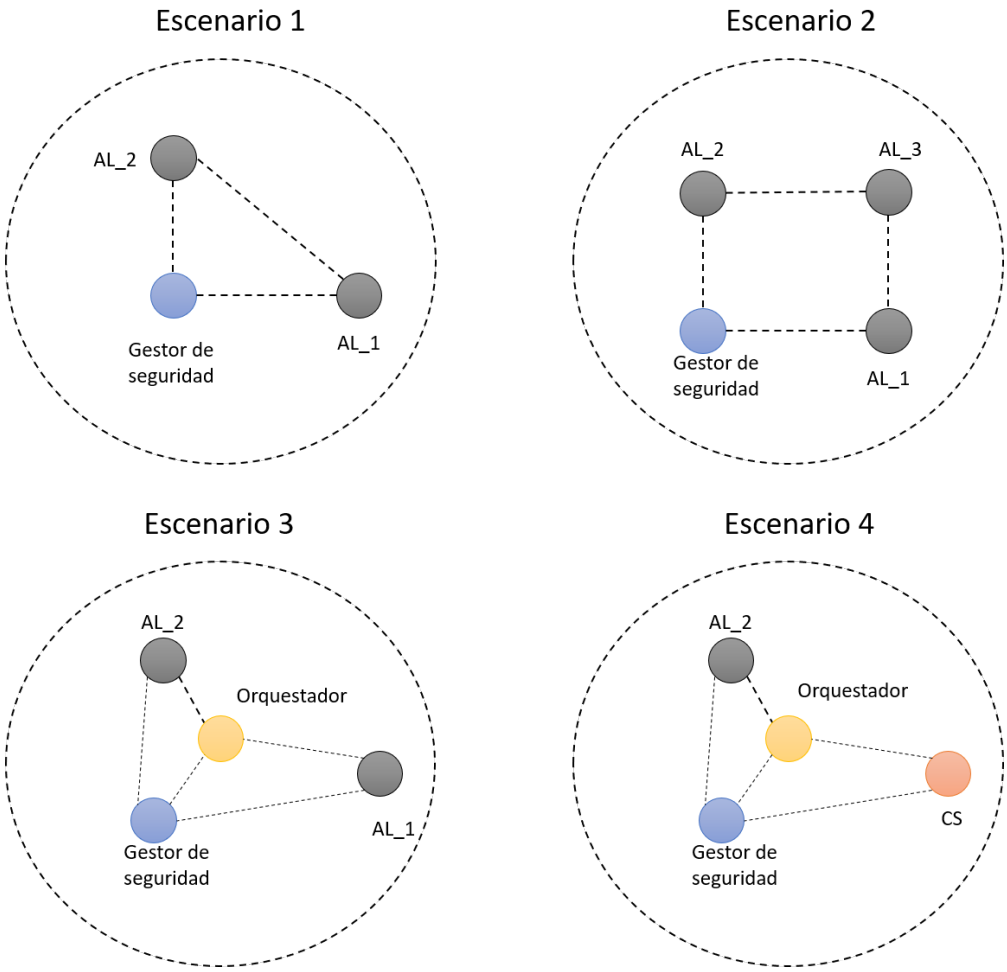

Figura 5-2: Escenarios de Pruebas

5.2. Diseño Experimental

5.2.1. Escenario 1: Ataque de Denegación de Servicio (DoS)

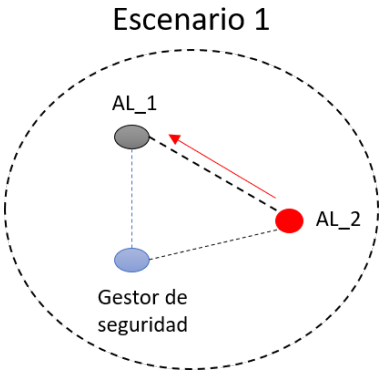

Figura 5-4: Escenario 1

```
------------------------------------------------------------
                    FAILURE DETECTOR
------------------------------------------------------------
                mac     LS   TQ                 NH             ip  date
0  b8:27:eb:d6:e3:10  0.170  188  b8:27:eb:6a:22:34  169.254.6.157    10
1  b8:27:eb:d6:e3:10  0.170  247  b8:27:eb:d6:e3:10  169.254.6.157    10
2  b8:27:eb:6a:22:34  0.000  185  b8:27:eb:d6:e3:10   169.254.7.19    10
3  b8:27:eb:6a:22:34  0.000  255  b8:27:eb:6a:22:34   169.254.7.19    10
read unix socket
--------------------------
       Iteration:  2
--------------------------
apps "0"
throughput : 0.0
[1317.86, 0, 0.0, [0, 0, 0, 0]]
--------------------------
       NODE STATE
--------------------------
pow:  1322.1028
% CPU:  2
thr_node:  0.0
BW used MB/s:  0.005568
BW avaliable MB/s:  9.994432
--------------------------
--------------------------
       Container   0
--------------------------
thr_cont_ 0 :  0.0
mean_cont_ 0 :  0.0
thr min :  0.0
Cont-Controller 50000
BW used Mb/s:  0.005568
BW avaliable Mb/s:  9.994432
--------------------------
Valor THR global 0.0
 Delay _ Curr 0
[0.0, 0.0, 0.0, 0.0]
4
[1073.104580168] announce master ...
apps "0"
throughput : 0.0
[1317.86, 0, 0.0, [0, 0, 0, 0]]
--------------------------
       NODE STATE
--------------------------
pow:  1322.1028
% CPU:  2
```

Figura 5-3: Detector de Fallas Sobre el Agente Local

Para el primer escenario de validación se tuvieron en cuenta dos agentes locales y un gestor de seguridad que es el encargado de realizar las tareas de seguridad, en este escenario no existe un nodo orquestador, por lo tanto, cada nodo es autónomo de su comportamiento y aplicaciones. Como se ve en la figura **5-4** el nodo gestor de seguridad consulta el detector de fallas, el cual, tiene un monitoreo constante de los nodos en la red. En este escenario se utilizó un modelo de movilidad *Random Walk Model* que se basa en direcciones y velocidades aleatorias, en este modelo un nodo móvil se mueve de su dirección actual a una nueva localización escogiendo una dirección y velocidad aleatoria.

Para la creación de la política de seguridad es importante conocer los valores en los que trabaja el sistema, para este tipo de escenario el comportamiento normal se muestra en la figura **5-5** y **5-6**, el comportamiento normal quiere decir que no existe una carga maliciosa o de operación en el sistema que perjudique los recursos de red, los parámetros generales de la prueba están en la tabla **5-1**, los

agentes locales son desplegados y el detector de fallas comienza a tomar medidas cada 5 segundos para tener una muestra del estado actual de la red. Como se vio en el capítulo 4 calidad de enlace (TQ) está definido en una escala de 0 y 255 (donde 0 indica que no hay conexión y 255 indica una excelente calidad), entre menor sea el valor se necesitarán enviar más paquetes para obtener una comunicación exitosa. Cuando llega un paquete de datos para la transmisión, el nodo se refiere a la tabla de originadores para determinar la dirección en la que se enviarán los paquetes. Como se ve en la tabla **5-2** en promedio la calidad de enlace es 192 esto le permite al sistema mantener comunicaciones con los diferentes nodos en la red e intercambiar mensajes sin gastar muchos recursos.

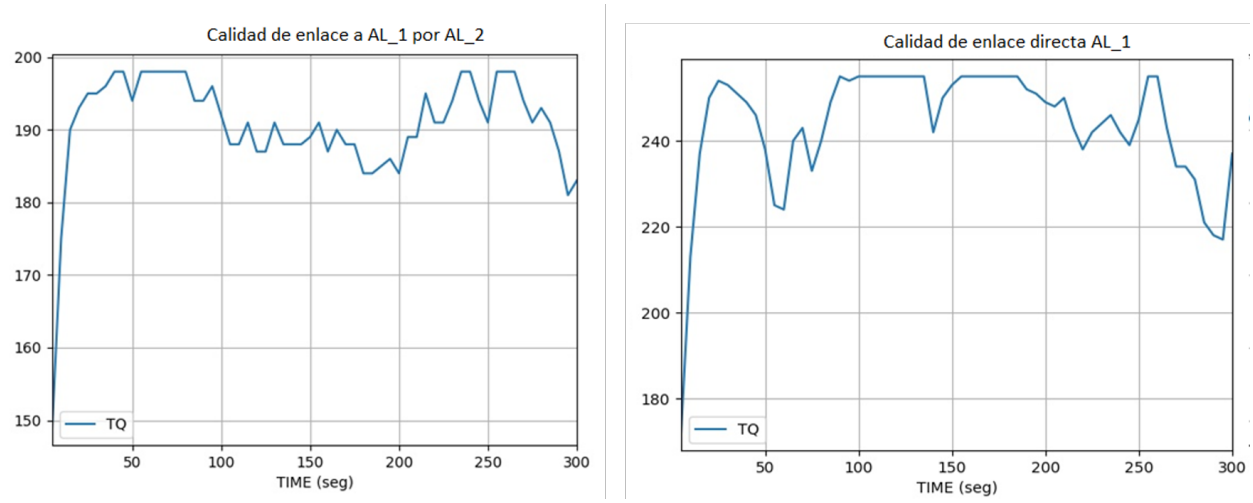

Figura 5-5: Calidad de enlace a AL_1

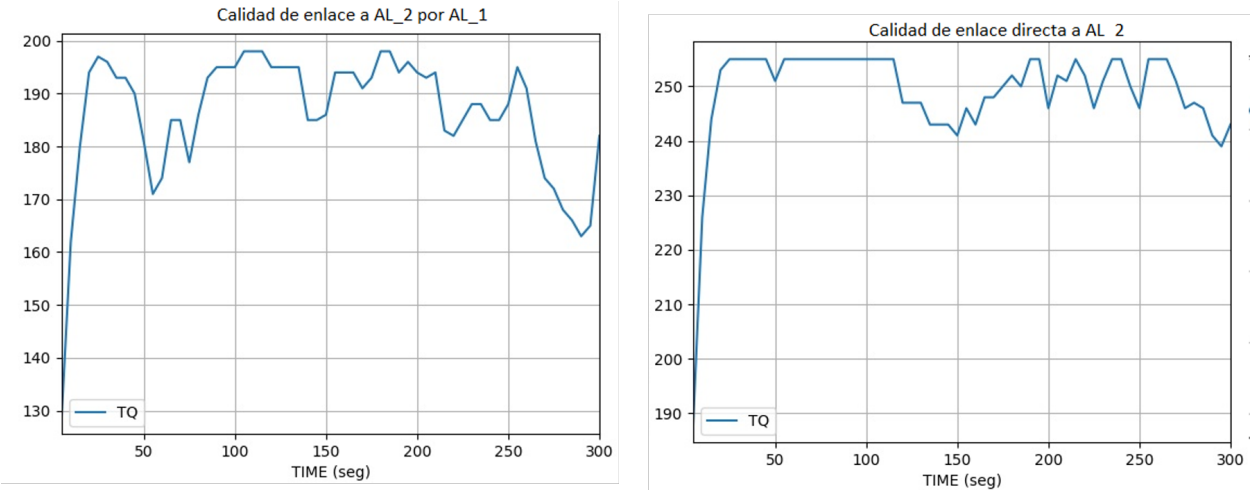

Figura 5-6: Calidad de Enlace a AL_2

Parámetro	Característica
Duración de la prueba	300 segundos
Tiempo de muestreo detector de fallas	5 segundos

Tabla 5-1: Parámetros Generales de la Prueba

Medida	AL_1	AL_2
Media	191,7068966	250,6842105
Desviación estándar	4,742523648	4,848261451
varianza	22,49153055	23,5056391

Tabla 5-2: Análisis Estadístico TQ, Camino Directo, Comportamiento Normal Escenario 1

Para este escenario, se ha propuesto un ataque de inundación TCP, para esto, existe un nodo malicioso que hace parte de la red y es el encargado de desplegar el ataque, este ataque envía solicitudes a un puerto en el nodo AL_1, el cual, tiene una aplicación de servidor apache, este ataque afectó el recurso de la calidad del enlace como se ve en la figura **5-7** y **5-8**, los parámetros generales de la prueba son mostrados en la tabla **5-3**, se realizan dos ataques para demostrar la afectación que sufre el canal del enlace y mirar la recuperación que el nodo tiene luego del final del ataque. Los resultados obtenidos en la tabla **5-4** demuestran la disminución en la calidad del enlace y la afectación que presentó el sistema para comunicarse entre los nodos.

Parámetro	Característica
Duración de la prueba	400 segundos
Tiempo de muestreo detector de fallas	10 segundos
t inicial ataque 1	90 segundos
t final ataque 1	230 segundos
t inicial ataque 2	280 segundos
t final ataque 2	340 segundos
puerto objetivo	80

Tabla 5-3: Parámetros Generales de la Prueba Escenario 1 ataque

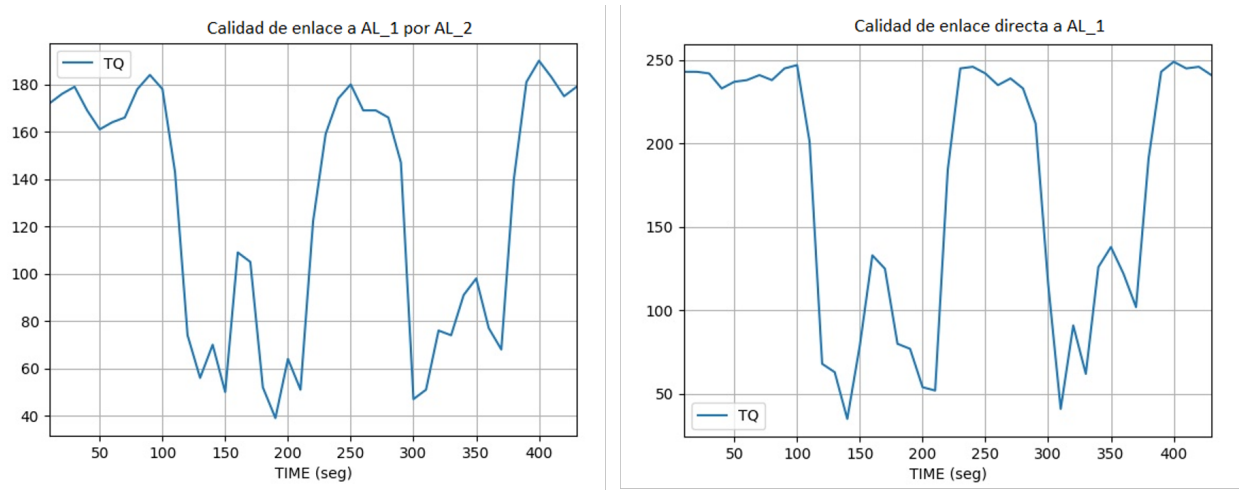

Figura 5-7: Calidad de enlace a AL_1 Ataque Escenario 1

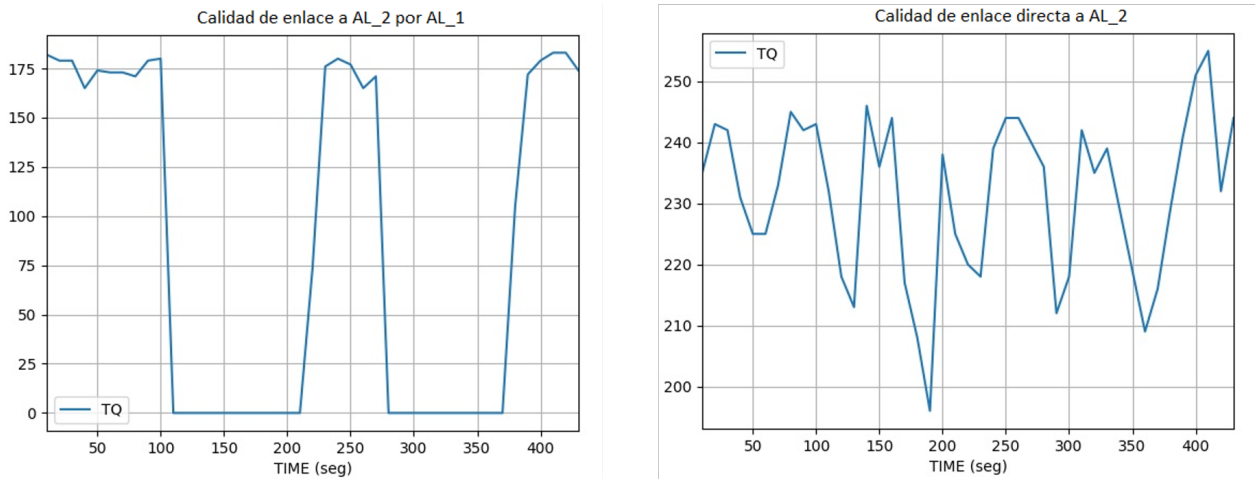

Figura 5-8: Calidad de Enlace a AL_1 Ataque Escenario 1

Medida	AL_1	AL_2
Media	97,06666667	40,6
Desviación estándar	49,7600911	73,76003564
varianza	2476,066667	5440,542857

Tabla 5-4: Análisis Estadístico TQ, Ataque 1 Camino Directo, Escenario 1

5.2.2. Escenario 2: Ataque de Denegación de Servicio Distribuido (DDoS)

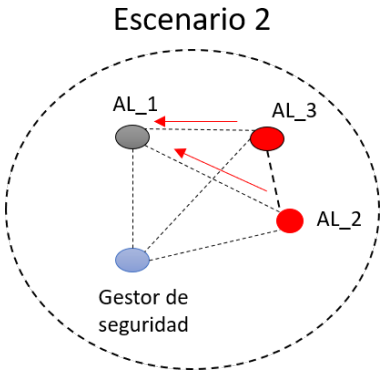

Figura 5-9: Escenario 2

Para el segundo escenario de validación se tuvieron en cuenta tres agentes locales y un gestor de seguridad que es el encargado de realizar las tareas de seguridad, al igual que en el escenario 1, en este escenario no existe un nodo orquestador, por lo tanto, cada nodo es autónomo de su comportamiento y aplicaciones. Como se ve en la figura **5-9** el detector de fallas tiene un monitoreo constante de los nodos en la red. En este escenario se utilizó un modelo de movilidad *Random Walk Model*, el cual se se basa en direcciones y velocidades aleatorias, en este modelo, un nodo móvil se mueve de su dirección actual a una nueva localización escogiendo una dirección y velocidad aleatoria.

Para el segundo escenario se realizó un ataque de denegación de servicio distribuido en el cual se toma el control de dos nodos por parte de un atacante buscando deshabilitar o reducir la calidad del enlace en la red. Los dos nodos maliciosos actúan como esclavos generando tráfico TCP y falsificando direcciones IP, la víctima intentó responder a las solicitudes provenientes de las fuentes maliciosas, sin embargo, no se obtuvo ninguna respuesta y después de un tiempo se saturó la capacidad de memoria del nodo.

Debido a que el agente local tiene la capacidad de desplegar contenedores, migrar contenedores, detener aplicaciones desplegadas en contenedores y mediante un sistema de comunicación por puertos recibe las órdenes del orquestador para ajustar el *throughput* de las aplicaciones de acuerdo a las políticas desplegadas en el sistema, un ataque de denegación de servicio afecta seriamente la comunicación y funcionamiento del sistema TLÖN. en la tabla **5-5** se ven los parámetros generales de la prueba. Durante la prueba se desplegaron dos ataques hacia el nodo AL_1 en dos instantes diferentes de tiempo para observar la recuperación de la calidad del enlace una vez termine el ataque.

Parámetro	Característica
Duración de la prueba	350 segundos
Tiempo de muestreo detector de fallas	5 segundos
t inicial ataque 1	60 segundos
t final ataque 1	180 segundos
t inicial ataque 2	230 segundos
t final ataque 2	320 segundos
puerto objetivo	80

Tabla 5-5: Parámetros generales de la prueba escenario 2

En las figuras **5-10**, **5-11**, **5-12**, **5-13**, **5-14** y **5-15** se observa el comportamiento de los dos ataques en los tiempos 60 segundos y 230 segundos, este ataque afecta de manera directa los caminos hacia el AL_1 ya que en fracciones de tiempo el camino por medio de AL_2 y AL_3 es cero, incluso la comunicación directa con el nodo se ve afectada y esto afecta la toma de decisiones que el protocolo B.A.T.M.A.N debe realizar para seleccionar el mejor camino hacia su destino.

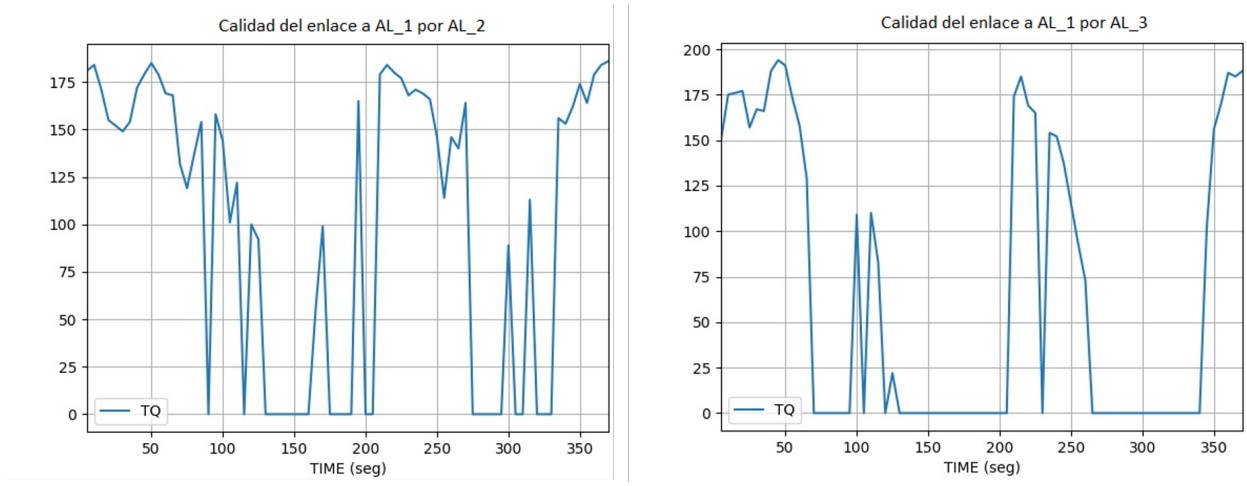

Figura 5-10: Calidad del Enlace a AL_1 por AL_2 y AL_3 Escenario 2

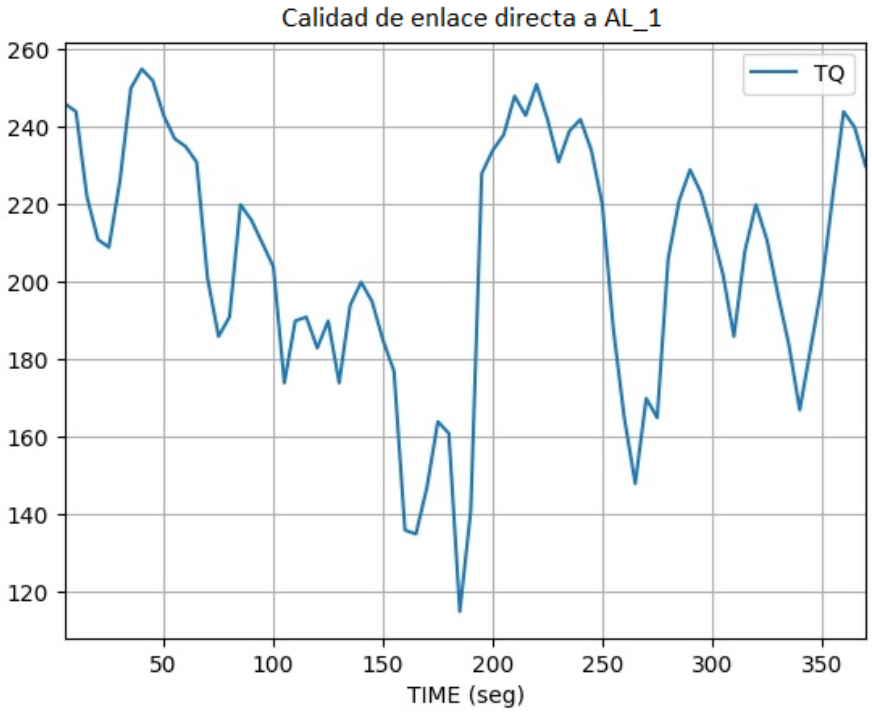

Figura 5-11: Calidad del Enlace Directa a AL_1 Escenario 2

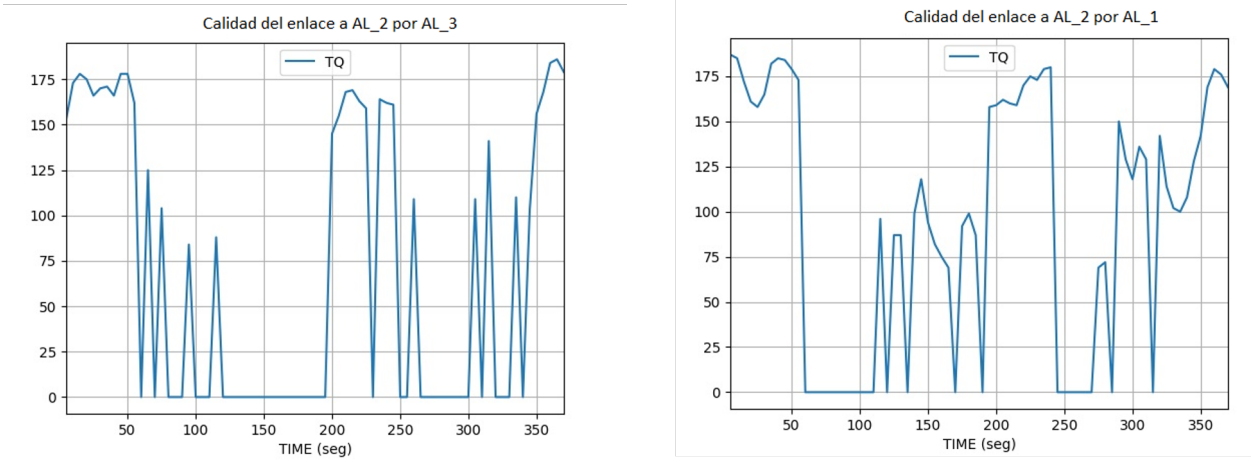

Figura 5-12: Calidad del Enlace a AL_2 por AL_1 y AL_3 Escenario 2

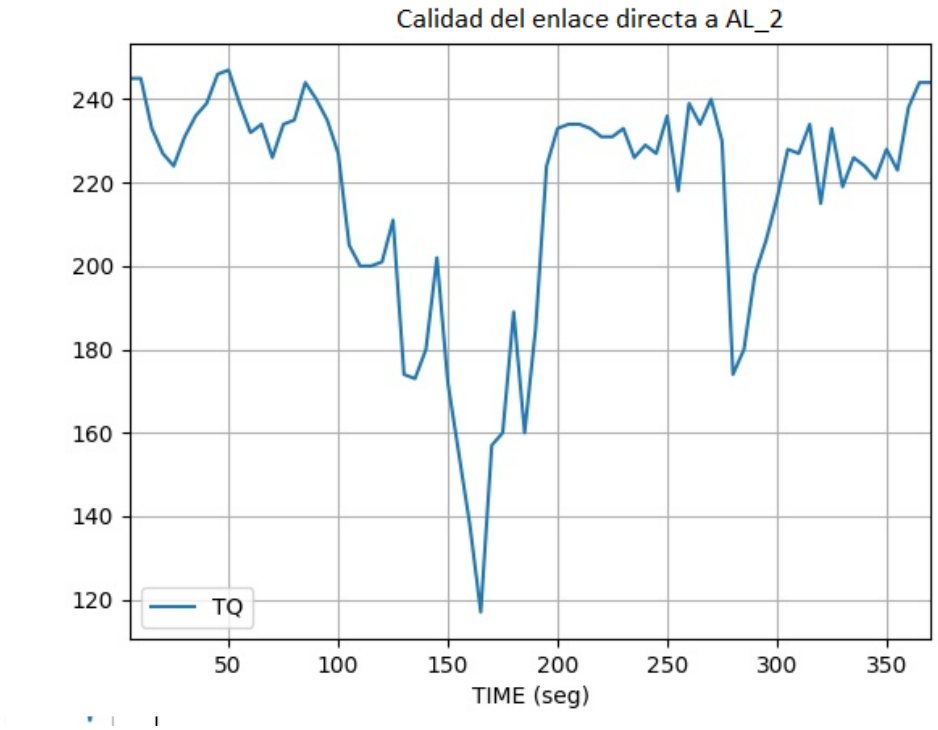

Figura 5-13: Calidad del enlace directa a AL_2 escenario 2

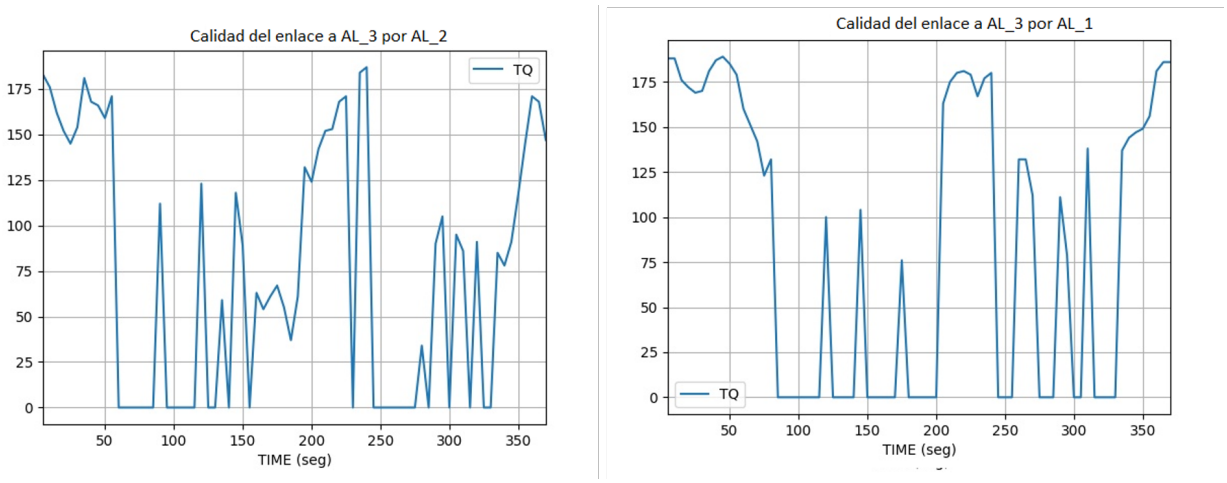

Figura 5-14: Calidad del Enlace a AL_3 por AL_1 y AL_2 Escenario 2

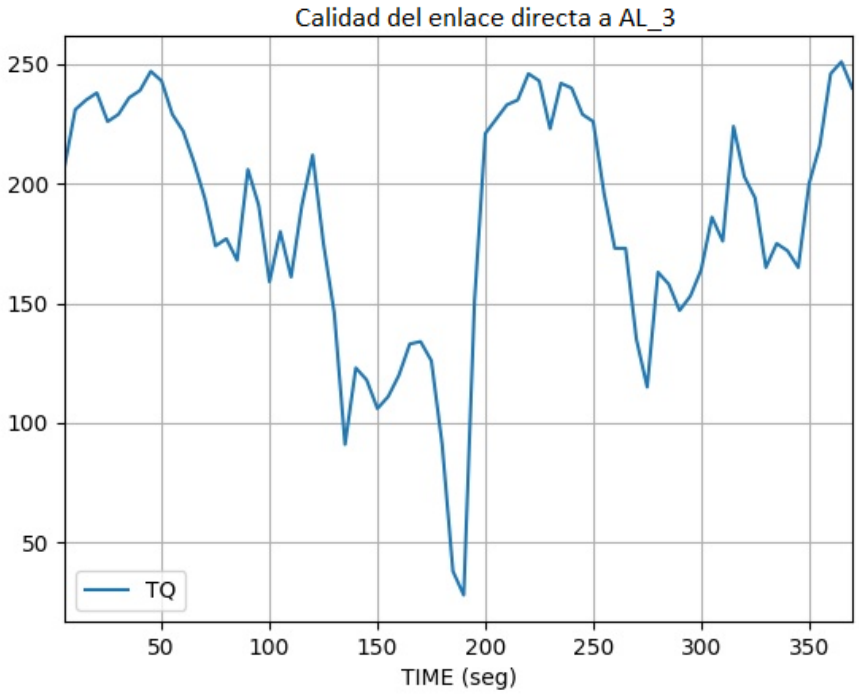

Figura 5-15: Calidad del Enlace Directa a AL_3 Escenario 2

para la tabla **5-6** las medidas fueron tomadas teniendo en cuenta el instante de inicio y final del ataque, se puede deducir que el promedio de calidad del enlace tiene grandes variaciones con respecto al comportamiento normal, las medidas de varianza y desviación estándar muestran una dispersión más grande entre los datos de un comportamiento normal y el ataque. La consecuencia de estas medidas se ven reflejadas en la eliminación de caminos hacia algún destino, lo cual dificulta la decisión de envío de paquetes que debe realizar el protocolo B.A.T.M.A.N.

Medida	AL_1	AL_2	AL_3
Media	70	21,65384615	30,80769231
Desviación estándar	47,02930347	13,27012982	43,1008299
varianza	4601,25	2283,593333	1893,956667

Tabla 5-6: Análisis Estadístico Calidad del Enlace Escenario 2

5.2.3. Escenario 3: Ataque de Negación de Servicio con Orquestador

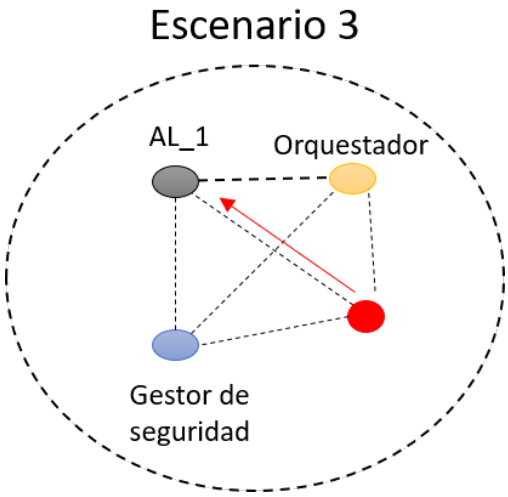

Figura 5-16: Escenario 3

A diferencia de los escenarios 1 y 2 para el escenario número 3 se utilizó un nodo orquestador, el cual es el agente con mayor jerarquía sobre el sistema, tiene la capacidad de indicar comandos en los agentes locales de acuerdo con las políticas del sistema, conoce la arquitectura de todos los dispositivos y aplicaciones en la red. Según el modelo propuesto en la figura **4-2** y la computación social propuesta, el nodo orquestador actúa como la presidencia del sistema o estado TLÖN, lo cual quiere decir que todos los nodos e instituciones presentes en el sistema deben responden a el nodo que tenga este rol.

En este escenario el gestor de seguridad debe hacer monitoreo de la política de seguridad que sea entregada por el sistema, la política debe hacer seguimiento a un recurso en el sistema, para este trabajo se hace monitoreo de la calidad del enlace (TQ). Para este escenario y teniendo en cuenta las medidas estadísticas obtenidas en la tabla **5-2**, **5-4** y **5-6** se asume que la política de seguridad fue violada cuando la calidad del enlace caiga por debajo de 70 hacia cualquier nodo, el gestor de seguridad detecta el cambio brusco en la calidad del enlace e inicia una comunicación con el orquestador indicándole la anomalía en el sistema. El esquema del escenario opera sobre redes inalámbricas utilizando el protocolo B.A.T.M.A.N y el servicio de mensajería ALFRED, en todos los escenarios el detector de fallas tiene una etiqueta que lo diferencia como nodo detector de fallas. Como parte de la solución de este escenario se desarrolló un modelo de comunicación sobre la red con un conjunto de servidores y clientes TCP multihilo en cada uno de los nodos, en sus diferentes roles, ya sea como Agente Local, detector de fallas y Orquestador, para llevar la información de seguridad del sistema. Una vez que el agente local ha identificado y decidido que existe una violación a la política de calidad del enlace se comunica con el orquestador utilizando un esquema de comunicación como el que se muestra en la figura **5-17**, en ella se pueden observar los clientes en amarillo y los servidores en azul, este esquema de comunicación funciona de manera constante para que se pueda enviar la informa-

ción hacia el orquestador.

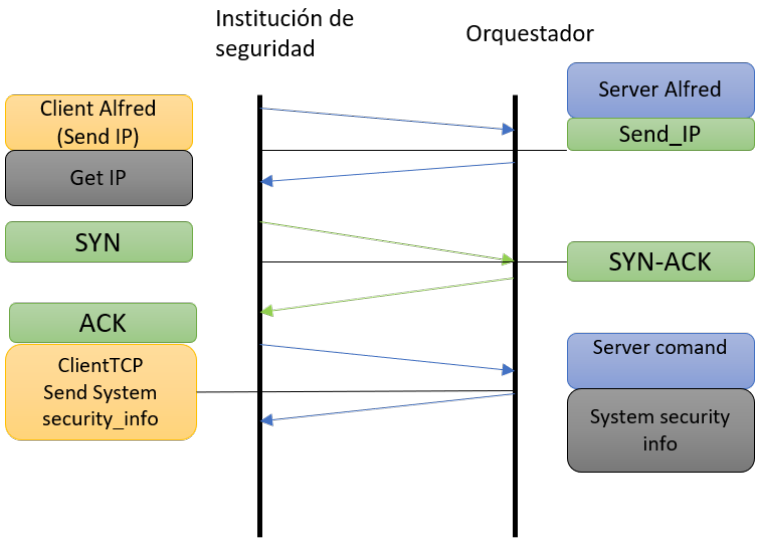

Figura 5-17: Modelo de Comunicación

El nodo orquestador mantiene una tabla con la información general de todo los nodos en la red y mediante el modelo de comunicación mostrado en la figura **5-18** puede ejecutar acciones que permitan mitigar el ataque de denegación de servicio, un ejemplo puede ser ejecutar la acción de reinicio en los nodos que presenten una anomalía en la calidad del enlace como se ve en la figura **5-19**, para esto, previamente se ha enviado al orquestador los nodos que presenten anomalías en la calidad del enlace, una vez son recibidas por el orquestador el espera un tiempo de 3 minutos para tomar la acción de reinicio en el agente local afectado.

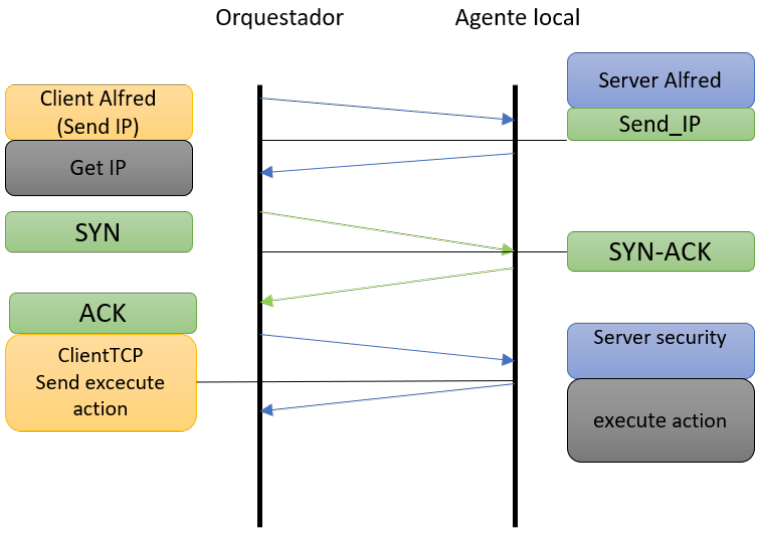

Figura 5-18: Modelo de Comunicación Orquestador- Agente local

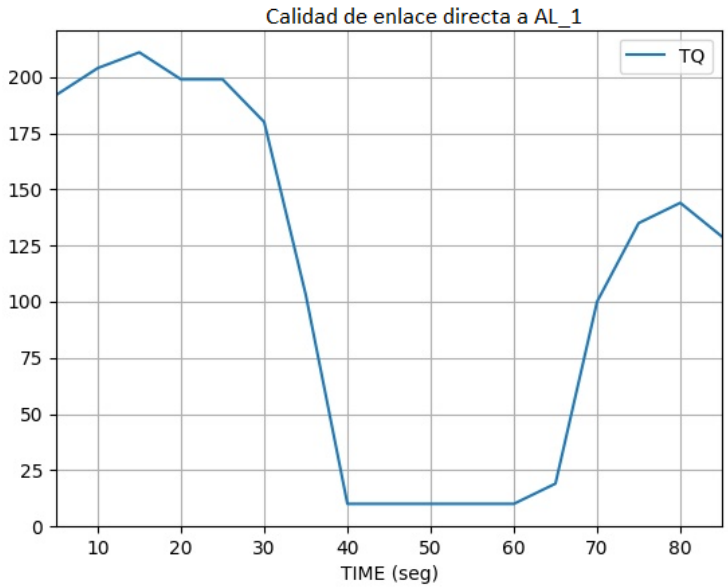

Figura 5-19: Calidad de Enlace a AL1 Escenario 3

5.2.4. Escenario 4: Funcionamiento de la Institución de Seguridad TLÖN

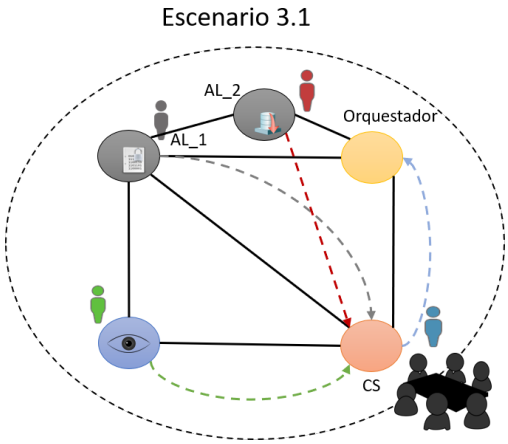

Figura 5-20: Escenario institución

Este escenario (figura **5-20**) busca tener una visión del funcionamiento de la institución de seguridad TLÖN, en un sistema que tiene como base la social inspiración, existen diferentes roles dentro de la institución que deben ser cubiertos para lograr cumplir con el objetivo de seguridad del sistema, estos roles deben ser asignados por el sistema a agentes que deben realizar las tareas que son asignadas por la institución, por lo tanto, los agentes en cada rol son seleccionados así:

- Estado superior de seguridad y defensa: Es el puesto con mayor rango y jerarquía dentro de la institución, por lo tanto, su puesto es asignado por el estado TLÖN, teniendo en cuenta los

requerimientos y exigencias del rol, por ejemplo, debe tener suficiente autonomía para tomar decisiones y realizar tareas sin que sea supervisado.

- Gestor de seguridad infraestructura: su puesto es asignado por el estado superior de seguridad y defensa, consulta los mecanismos de seguridad dispuestos por la institución para garantizar la seguridad en la red

- Gestor de seguridad virtualización: su puesto es asignado por el estado superior de seguridad y defensa, consulta los mecanismos de seguridad dispuestos por la institución para garantizar la seguridad en el sistema de virtualización.

- Gestor de seguridad de agentes: Su puesto es asignado por el estado superior de seguridad y defensa, realiza interacciones con otros agentes e integrantes del sistema TLÖN, por lo tanto, debe tener características de comunicación entre agentes y utiliza los mecanismos de seguridad dispuestos por la institución para garantizar la seguridad de la comunidad de agentes del sistema.

- Gestor de seguridad de aplicaciones: Su puesto es asignado por el estado superior de seguridad y defensa, utiliza los mecanismos de seguridad dispuestos por la institución para garantizar la seguridad de las aplicaciones, servicios y demás instituciones del sistema.

Algorithm 2 Agente gestor de seguridad

Función: Agente gestor de seguridad
Entradas: Percepción, mecanismos de seguridad
$estado \leftarrow actualizar(estado, perce)$
if $Secuencia$ está vacía **then**
$Objetivo \leftarrow Formula - Objetivo(politicaseg)$
$Problema \leftarrow Formula - Problema(politicaseg, objetivo)$
$Secuencia \leftarrow Bsqueda(problema\_Entrada)$
end if
$Acc \leftarrow Primero(Secuencia)$
devolver Acc

El algoritmo 2 es un agente gestor de seguridad. Su primera tarea es crear un objetivo y problema basado en las políticas de seguridad, busca las acciones que resuelven el problema y ejecuta las acciones para resolver ese problema, cuando ha finalizado comienza de nuevo.

Según el modelo de seguridad propuesto, los gestores de seguridad son el nivel más bajo en la institución, ellos manejan los artefactos de seguridad para hacer cumplir las políticas de seguridad, en la imagen **5-21** se puede ver el nivel de escalamiento que tiene la institución, para este escenario el detector de fallas es manejado y consultado por el gestor de seguridad de red, el cual recolecta la

información que sirve de base para velar por el cumplimiento de las políticas, en el siguiente nivel está el comando general, el cual reúne a las tres fuerzas de seguridad TLÖN en un consejo para discutir y tener una visión general de la seguridad en el sistema, estas reuniones son precedidas por el estado superior de seguridad quien es el encargado de manejar el consejo, realizar los consensos y de ser necesario comunicar al estado TLÖN los resultados finales a los que se llegaron. El algoritmo 3 es un consejo de seguridad, su primera tarea es establecer un protocolo de comunicación con los otros agentes integrantes del consejo, luego se crean los objetivos y problemas basados en las políticas de seguridad, finalmente se ejecutan las acciones para resolver el problema, cuando ha finalizado comienza de nuevo

Figura 5-21: Funcionamiento Institución de Seguridad

Como parte de su funcionamiento, los gestores de seguridad no son estáticos, ya que son agente con características de movilidad, esto les permite utilizar los recursos del territorio TLÖN para moverse y consultar elementos como el detector de fallas, el cual es un elemento fijo que se encuentra en un agente local, de esta forma los agentes pueden reunirse para realizar los consejos de seguridad y tomar decisiones respecto a su funcionamiento, en la imagen **5-20** se puede ver la interacción que tiene cada agente gestor de seguridad (integridad,confidencialidad, integridad) con los artefactos de seguridad que son dispuestos por la institución de seguridad TLÖN para recolectar la información del

Algorithm 3 Consejo de seguridad

Función: Consejo de seguridad
Entradas: Percepción, Agentes Integridad, Confidencialidad, Disponibilidad
$estado \leftarrow actualizar(estado, perce)$
$comm \leftarrow Comunicacion(Protocolo)$
if $comm$ está establecida **then**
$Objetivo \leftarrow Formula - Objetivo(politicaseg)$
$Problema \leftarrow Formula - Problema(politicaseg, objetivo)$
$Secuencia \leftarrow Bsqueda(problema\_Entrada)$
end if
$Acc \leftarrow Primero(Secuencia)$
devolver Acc

comportamiento y cualquier cosa que tenga relación al cumplimiento de las políticas de seguridad. Los agentes forman el comando de seguridad el cual es precedido por el estado superior de seguridad, en este consejo se realizan acciones como en la imagen **4-13** y se decide enviar la información al orquestador el cual representa el estado TLÖN para que tome acciones de enviar alertas a los nodos afectados por ataques o instrucciones de reinicio a los procesos.

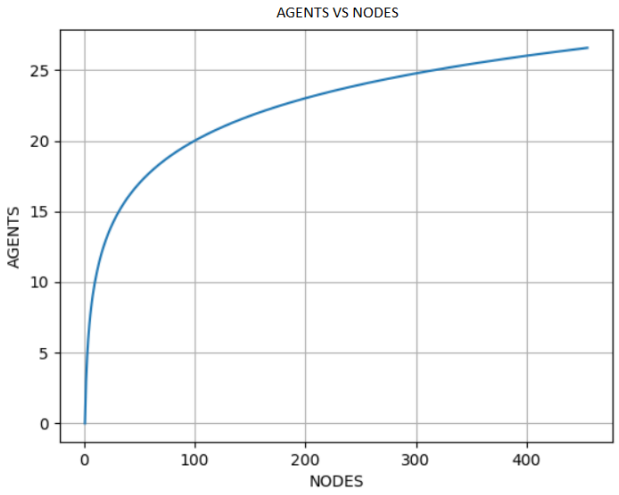

Figura 5-22: Relación Nodos vs Agentes

La institución de seguridad posee varios elementos que en una red de tamaño pequeño no son de gran utilidad, por ejemplo, para este escenario no se considero la presencia de los gestores de seguridad de virtualización, agentes y aplicaciones. debido a esto, se han considerado los elementos que están en la gráfica **5-22**, donde se muestra un estimado de la cantidad de nodos que son necesarios para el funcionamiento de la institución de seguridad TLÖN con respecto a la cantidad de nodos que existen en el sistema, ya que manejar todos los recursos que se encuentran en una red de gran tamaño

resulta difícil de gestionar por una cantidad pequeña de agentes pertenecientes a la institución de seguridad. No se puede dar un número exacto de cuánto agentes se necesitan para manejar la institución de seguridad, sin embargo, existen parámetros que pueden ayudar a tener un estimado de esto, por ejemplo:

- Capacidad de la red inalámbrica: Entre más nodos estén en la red menor será el *throughput* ocasionando retardos en la red y la comunicación en los nodos.
- Memoria: Elementos como el detector de fallas utilizan la memoria para almacenar la información de los estados del enlace, por lo tanto, si existe un gran número de nodos en la red, es posible que el detector de fallas no pueda manejar el flujo de información.

por lo anterior, se podría pensar que la institución de seguridad debe considerar el tamaño del sistema para poder calcular el número de agentes y artefactos necesarios para cumplir con las funciones de seguridad, por ejemplo, para el detector de fallas se puede limitar el tamaño de la red a máximo 200 nodos.

Capítulo 6

Conclusiones y Recomendaciones

6.1. Conclusiones

El desarrollo de este trabajo está presente en las dos primeras capas del sistema TLÖN (redes Ad hoc y virtualización), sin embargo, La seguridad es un aspecto presente en todas las capas del sistema y el desarrollo de la capa de seguridad continua con lo propuesto en este trabajo, Las abstracciones propuestas del modelo de seguridad permitirán escalar e integrar lo propuesto con elementos del sistema TLÖN tales como, el sistema multiagente, el lenguaje TLÖN, dinámica del conocimiento, Network coding, entre otros.

Las conclusiones más relevantes de veste trabajo son:

A Los escenarios propuestos, han sido evaluados e integrados con los componentes de red Ad hoc y virtualización (agente local y orquestador) del sistema TLÖN, demostrando la viabilidad de integrar un sistema de seguridad para estos componentes que funcionan de forma distribuida y descentralizada.

B Elegir un modelo de computación que tenga modelos pseudosociales permite generar estructuras de agentes que estén regidos bajo reglas que controlan su comportamiento y permite crear ambientes de cooperación para solucionar una tarea.

C El tipo de institución adoptado para la capa de seguridad TLÖN permitirá crear interacciones sociales entre los agentes dentro y fuera de la institución, estos comportamientos pueden evolucionar y permitir a los agentes crear relaciones más complejas y aplicar mecanismos de seguridad más robustos que garanticen una confianza en el sistema.

D El mecanismo detector de fallas permite monitorear redes con un comportamiento dinámico, abriendo la puerta a mecanismos de seguridad más complejos que no requieran de un control centralizado, y sean adaptativos a una topología dinámica.

E Para sistemas como MANET donde sus dispositivos son heterogéneos y las capacidades computacionales puedan ser reducidas, la seguridad no puede ser tarea de un solo nodo o agente, por lo tanto, proponer un modelo de seguridad que permita crear interacciones sociales entre los agentes de una institución puede reducir los consumos de recursos distribuyendo las tareas entre agentes para buscar una solución.

F El modelo propuesto para la selección del siguiente salto para enviar un mensaje en el protocolo B.A.T.M.A.N está expuesto a caer en situaciones en las que no se pueda escoger un camino de envío para los mensajes a consecuencia de un ente malicioso que afecte la red.

6.2. Recomendaciones

Del modelo de seguridad propuesto se pueden generar ideas que son posibles con las capacidades computacionales actuales, a lo largo de esta investigación se han resuelto algunas dudas y definido algunos parámetros que pueden variar en el tiempo. Las siguientes son las recomendaciones y trabajo futuro generado de esta investigación que están en su mayoría fuera del alcance de este trabajo:

A Fortalecer la solución propuesta en este trabajo para que no dependa de parámetros como el sistema operativo donde se despliega.

B El desarrollo más a profundidad de temas relacionados con la integridad y confidencialidad del sistema TLÖN, ya que en este trabajo solo se generaron las directrices generales de su funcionamiento y se tuvo un enfoque en la disponibilidad.

C Validar el funcionamiento del sistema propuesto en este trabajo con otros orquestadores en la red.

D Agregar un componente de network coding para mejorar las condiciones de despliegue del sistema y permita reducir la tasa de falsos positivos en la detección de fallas.

E Validar la operación del sistema con dispositivos ajenos al sistema TLÖN.

F Validar la operación de la institución de seguridad TLÖN con otras instituciones pertenecientes al sistema y crear un modelo de pseudo constitución que funcione para todas las instituciones.

G Estudiar otro tipo de medidas que puedan ser utilizadas por los mecanismos de seguridad de la institución TLÖN para detectar posibles ataques o violaciones a las políticas de seguridad.

Apéndice A

Anexo: Ataque de denegación de servicio (DoS/DDoS)

Los modelos de denegación de servicio se extienden para incluir los ataques a un sistema, un ataque de denegación de servicio ocurre cuando un usuario no autorizado hace que un servicio esté deshabilitado o intermitente por medio de herramientas que puedan afectar un sistema (Bhattacharyya y Kalita, 2016); (Amiri y Soltanian, 2015). Un requerimiento base para el uso de un sistema y una red es el acceso, por lo tanto, cuando sucede un ataque que afecte la disponibilidad, también se afecta la confidencialidad. En (Yu, 2013) se mencionan dos formas de realizar un ataque de denegación de servicio, el primero es deshabilitar un sistema enviando paquetes, los cuales son diseñados con las vulnerabilidades de la víctima, la segunda forma es utilizar un aumento de paquetes que agote los recursos de una víctima, tales como, ancho de banda, capacidad de computo, estructuras de datos del sistema operativo, etc. En (*Ceh Cert Ethical Hacker Exam Guide*, 2012) y (Gupta, 2011) se mencionan diferentes mecanismos para desplegar un ataque de denegación de servicio que aprovechan el envío de paquetes y el consumo de recursos, entre ellos se encuentran.

- Ping de la muerte: envía un paquete ping de gran tamaño que termina deshabilitando un sistema.
- Inundación ICMP: Un atacante envía paquetes ICMP desde direcciones fuente falsas.
- Ataque de fragmentación: Estos ataques toman ventaja de los sistemas que reconstruyen los paquetes fragmentados.
- Ataques volumétricos: Consumen todo el ancho de banda disponible para un sistema o servicio.
- Ataque de sincronización (*SYN flood*): Aprovecha el método *three way handshake* para enviar miles de paquetes de sincronización TCP desde direcciones fuente falsas para agotar los recursos de un sistema.

En la figura **A-1** se ve un ejemplo de ataque de denegación de servicio distribuido, este tipo de ataque utiliza varios integrantes de la red que convierte en *bots* para desplegar un ataque y agotar los recursos de la víctima.

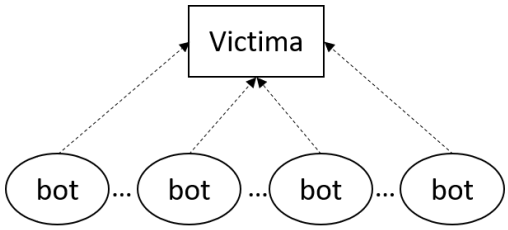

Figura A-1: Ataque de Denegación de Servicio Distribuido

Referencias

Amiri, I., y Soltanian, M. (2015). *Theoretical and experimental methods for defending against ddos attacks*. Elsevier Science. Descargado de https://books.google.com.co/books?id=75-4CgAAQBAJ

Anderson, R. (2008). *Security engineering: A guide to building dependable distributed systems*. Wiley. Descargado de https://books.google.com.co/books?id=ILaY4jBWXfcC

Andress, J. (2014). *The basics of information security: Understanding the fundamentals of infosec in theory and practice*. Elsevier Science. Descargado de https://books.google.com.co/books?id=9NI0AwAAQBAJ

batman.adv foundation. (2018). *B.A.T.M.A.N. IV - enhanced TQ algorithm*. https://www.open-mesh.org/projects/batman-adv/wiki/BATMAN_IV#Wikis/. ([Online; accessed 25-april-2018])

Belapurkar, A., Chakrabarti, A., Ponnapalli, H., Varadarajan, N., Padmanabhuni, S., y Sundarrajan, S. (2009). *Distributed systems security: Issues, processes and solutions*. Wiley. Descargado de https://books.google.com.co/books?id=7mx1zhZqfmEC

Bhattacharyya, D., y Kalita, J. (2016). *Ddos attacks: Evolution, detection, prevention, reaction, and tolerance*. CRC Press. Descargado de https://books.google.com.co/books?id=DewbDAAAQBAJ

Bin, G. (2012, Dec). Research on low consumption ad hoc network method of mine the internet of things. En *Proceedings of 2012 2nd international conference on computer science and network technology* (p. 699-702). doi: 10.1109/ICCSNT.2012.6526030

Bishop, M. (2018). *Computer security: Art and science*. Pearson Education. Descargado de https://books.google.com.co/books?id=bz58DwAAQBAJ

Blyth, A., y Kovacich, G. (2006). *Information assurance: Security in the information environment*. Springer London. Descargado de https://books.google.com.co/books?id=Cz2m7N121IsC

Boella, G., van der Torre, L., y Verhagen, H. (2006, 01 de Oct). Introduction to normative multiagent systems. *Computational & Mathematical Organization Theory*, *12*(2), 71–79. Descargado de https://doi.org/10.1007/s10588-006-9537-7 doi: 10.1007/s10588-006-9537-7

Box, M. C. G. (s.f.). 1) fundamentando la elección.

Cachin, C., Guerraoui, R., y Rodrigues, L. (2014). *Introduction to Reliable and Secure Distributed Programming*. Springer Berlin Heidelberg. Descargado de https://books.google.com.co/books?id=rNHcoQEACAAJ

Carmo, J., y Jones, A. J. I. (2002). Deontic logic and contrary-to-duties. En D. M. Gabbay y F. Guenthner (Eds.), *Handbook of philosophical logic: Volume 8* (pp. 265–343). Dordrecht: Springer Netherlands. Descargado de https://doi.org/10.1007/978-94-010-0387-2_4 doi: 10.1007/978-94-010-0387-2_4

Ceh cert ethical hacker exam guide. (2012). McGraw-Hill Education (India) Pvt Limited. Descargado de https://books.google.com.co/books?id=sdE3AwAAQBAJ

Chadli, S., Emharraf, M., Saber, M., y Ziyyat, A. (2014, Oct). The design of an ids architecture for

manet based on multi-agent. , 122-128. doi: 10.1109/CIST.2014.7016605

Chaki, N., y Chaki, R. (2014). *Intrusion detection in wireless ad-hoc networks*. CRC Press. Descargado de https://books.google.com.co/books?id=SELSBQAAQBAJ

Critchley, T. (2016). *High-performance it services*. CRC Press. Descargado de https://books.google.com.co/books?id=k4iKDQAAQBAJ

de Spinoza, B., y Peña, V. (1999). *Ética demostrada según el orden geométrico*. Alianza Editorial. Descargado de https://books.google.com.co/books?id=jR_LeHsFWIMC

Dubreuil, B. (2010). *Human evolution and the origins of hierarchies: The state of nature*. Cambridge University Press. Descargado de https://books.google.com.co/books?id=qBXvK0EkTcwC

Edgar, T., y Manz, D. (2017). *Research methods for cyber security*. Elsevier Science. Descargado de https://books.google.com.co/books?id=aRl2DQAAQBAJ

El-Bendary, M. (2014). *Developing security tools of wsn and wban networks applications*. Springer Japan. Descargado de https://books.google.com.co/books?id=dzBgBQAAQBAJ

Elser, A. (2012). *Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services*. Springer London. Descargado de https://books.google.com.co/books?id=yG9y-VMluwYC

Fitoussi, D., y Tennenholtz, M. (2000, mayo). Choosing social laws for multi-agent systems: Minimality and simplicity. *Artif. Intell.*, *119*(1-2), 61–101. Descargado de https://doi.org/10.1016/S0004-3702(00)00006-0 doi: 10.1016/S0004-3702(00)00006-0

Flauzac, O., González, C., Hachani, A., y Nolot, F. (2015, March). Sdn based architecture for iot and improvement of the security. En *2015 ieee 29th international conference on advanced information networking and applications workshops* (p. 688-693). doi: 10.1109/WAINA.2015.110

Genco, A. (2008). *Mobile agents: Principles of operation and applications*. WIT. Descargado de https://books.google.com.co/books?id=IZOVwjeamLIC

Gibbs, J. P. (1965). Norms: The problem of definition and classification. *American Journal of Sociology*, *70*(5), 586–594. Descargado de http://www.jstor.org/stable/2774978

Gilbert, S., y Lynch, N. (2012, feb). Perspectives on the CAP Theorem. *Computer*, *45*(2), 30–36. doi: 10.1109/MC.2011.389

Gupta, B. (2011). *An introduction to ddos attacks and defense mechanisms*. Lap Lambert Academic Publishing GmbH KG. Descargado de https://books.google.com.co/books?id=bwOppwAACAAJ

Hartenstein, H., y Laberteaux, K. (2009). *Vanet: Vehicular applications and inter-networking technologies*. Wiley. Descargado de https://books.google.com.co/books?id=VNbkpbIg1EoC

Hobbes, T. (1990). *Leviatán o la materia, forma y poder de una república eclesiástica y civil*. Universidad de València Servicio de Publicaciones. Descargado de https://books.google.com.co/books?id=6M1lLvv4kXEC

Jones, V. (2001). *High availability networking with cisco*. Addison-Wesley. Descargado de https://books.google.com.co/books?id=T9NFAQAAIAAJ

Kalinin, M., Zegzhda, P., Zegzhda, D., Vasiliev, Y., y Belenko, V. (2016). *Software defined security for vehicular ad hoc networks*. doi: 10.1109/ICTC.2016.7763528

Kazi, S. B., y Adhoni, M. A. (2016, March). Secure ids to detect malevolent node in manets. , 1363-1368. doi: 10.1109/ICEEOT.2016.7754906

Kisak, P. (2016). *The many types of government: From chaos to control*. CreateSpace Independent Publishing Platform. Descargado de https://books.google.com.co/books?id=QWVljwEACAAJ

Kovacich, G. (2003). *The information systems security officer's guide: Establishing and managing an information protection program*. Elsevier Science. Descargado de https://books.google.com.co/books?id=WGEYrDj68xsC

Kuhn, T. (1970). *The structure of scientific revolutions*. University of Chicago Press. Descargado de https://books.google.com.co/books?id=cpDuAAAAMAAJ

Lehto, M., y Neittaanmäki, P. (2018). *Cyber security: Power and technology*. Springer International Publishing. Descargado de https://books.google.com.co/books?id=vfRZDwAAQBAJ

Lin, X., y Lu, R. (2015). *Vehicular Ad Hoc Network Security and Privacy*. Wiley. Descargado de https://books.google.com.co/books?id=BCzWCQAAQBAJ

Loo, J., Mauri, J., y Ortiz, J. (2016). *Mobile ad hoc networks: Current status and future trends*. CRC Press. Descargado de https://books.google.com.co/books?id=k-zRBQAAQBAJ

Machiavelli, N., Bondanella, J., y Bondanella, P. (2013). *Discourses on livy*. W. Ross MacDonald School Resource Services Library. Descargado de https://books.google.com.co/books?id=MIZjDwAAQBAJ

Mishra, A. (2008). *Security and Quality of Service in Ad Hoc Wireless Networks* (1st ed.). New York, NY, USA: Cambridge University Press.

Murthy, C., y Manoj, B. (2004). *Ad hoc wireless networks: Architectures and protocols, portable documents*. Pearson Education. Descargado de https://books.google.com.co/books?id=U-yLb-9nXyYC

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, *45*(2), 167–256.

Newman, S. (2015). *Building microservices: Designing fine-grained systems*. O'Reilly Media. Descargado de https://books.google.com.co/books?id=jjl4BgAAQBAJ

Nietzsche, F., y Ulapes, A. (2005). *Ecce homo*. Longseller S.A. Descargado de https://books.google.com.co/books?id=xVYpipCR2CQC

Ostrom, E. (2009). *Understanding institutional diversity*. Princeton University Press. Descargado de https://books.google.com.co/books?id=LbeJaji_AfEC

Pathan, A. (2016). *Security of self-organizing networks: Manet, wsn, wmn, vanet*. CRC Press. Descargado de https://books.google.com.co/books?id=ZtBnZoijaDcC

Pierson, C. (2004). *The modern state*. Routledge. Descargado de https://books.google.com.co/books?id=Maai7H0fR6AC

Pitt, J., Busquets, D., y Riveret, R. (2015, agosto). The pursuit of computational justice in open systems. *AI Soc.*, *30*(3), 359–378. Descargado de http://dx.doi.org/10.1007/s00146-013-0531-6 doi: 10.1007/s00146-013-0531-6

Puttini, R., Percher, J. ., Me, L., y de Sousa, R. (2004, July). A fully distributed ids for manet. , 1,

331-338 Vol.1. doi: 10.1109/ISCC.2004.1358426

Reddy, G., y M, K. (2016). *Mobile ad hoc networks: Bio-inspired quality of service aware routing protocols*. CRC Press. Descargado de https://books.google.com.co/books?id=yCkNDgAAQBAJ

Regalado, D., Harris, S., Harper, A., Eagle, C., Ness, J., Branko, S., ... Sims, S. (2015). *Gray Hat Hacking* (four ed.). United States: McGraw-Hill Education.

Sahoo, K. S., Sahoo, B., y Panda, A. (2015, Dec). A secured sdn framework for iot. En *2015 international conference on man and machine interfacing (mami)* (p. 1-4). doi: 10.1109/MAMI.2015.7456584

Santi, P. (2005). *Topology control in wireless ad hoc and sensor networks*. Wiley. Descargado de https://books.google.com.co/books?id=uH_lCkPKsP0C

Sarkar, S., Basavaraju, T., y Puttamadappa, C. (2016). *Ad hoc mobile wireless networks: Principles, protocols, and applications, second edition*. CRC Press. Descargado de https://books.google.com.co/books?id=5lLOBQAAQBAJ

Sartre, J., Franco, C., y Moreira, M. (2000). *O ser e o nada: ensaio de ontologia fenomenológica*. Vozes. Descargado de https://books.google.com.co/books?id=FoYztgEACAAJ

Shehory, O., y Sturm, A. (2014). *Agent-oriented software engineering: Reflections on architectures, methodologies, languages, and frameworks*. Springer Berlin Heidelberg. Descargado de https://books.google.com.co/books?id=2pjIAwAAQBAJ

Silva, P., y Lima, P. U. (2007). Institutional robotics. En F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, y A. Coutinho (Eds.), *Advances in artificial life* (pp. 595–604). Berlin, Heidelberg: Springer Berlin Heidelberg.

Swaminathan, A., y Wade, J. B. (2016). Institutional environment. En M. Augier y D. J. Teece (Eds.), *The palgrave encyclopedia of strategic management* (pp. 1–7). London: Palgrave Macmillan UK. Descargado de https://doi.org/10.1057/978-1-349-94848-2_608-1 doi: 10.1057/978-1-349-94848-2_608-1

Tanenbaum, A. (2003). *Redes de computadoras*. Editorial Alhambra S. A. (SP). Descargado de https://books.google.com.co/books?id=WWD-4oF9hjEC

Teufel, S., Min, T., You, I., y Weippl, E. (2014). *Availability, reliability, and security in information systems: Ifip wg 8.4, 8.9, tc 5 international cross-domain conference, cd-ares 2014 and 4th internationalworkshop on security and cognitive informatics for homeland defense, secihd 2014, fribourg, switzerland, september 8-12, 2014. proceedings*. Springer International Publishing. Descargado de https://books.google.com.co/books?id=BctsBAAAQBAJ

Tomic, S., Pecora, F., y Saffiotti, A. (2018). Norms, institutions, and robots. *CoRR*, *abs/1807.11456*.

van Steen, M., y Tanenbaum, A. (2017). *Distributed systems*. CreateSpace Independent Publishing Platform. Descargado de https://books.google.com.co/books?id=c77GAQAACAAJ

Vargas, R. (2016). *Implementación y análisis de algoritmos de confianza en una red manet*. Uniandes. Descargado de https://books.google.com.co/books?id=yzYitAEACAAJ

Wang, X., Krishnamurthy, P., y Tipper, D. (2013, Jan). Wireless network virtualization. , 818-822. doi: 10.1109/ICCNC.2013.6504194

Whitman, M., y Mattord, H. (2009). *Principles of information security*. Thomson Course Technology. Descargado de https://books.google.com.co/books?id=gPonBssSm0kC

Wireless security. (2006). McGraw-Hill Education (India) Pvt Limited. Descargado de https://books.google.com.co/books?id=B-3Bmh8ZV9sC

Yu, S. (2013). *Distributed Denial of Service Attack and Defense*. Springer New York. Descargado de https://books.google.com.co/books?id=ryi{_}BAAAQBAJ

Zhong, S., Huang, X., Yang, P., Shi, J., Xie, L., y Wang, K. (2018). *Security and privacy for next-generation wireless networks*. Springer Nature. Descargado de https://books.google.com.co/books?id=R4F7DwAAQBAJ