



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Un sistema multi-agente para la auto- configuración de las operaciones de red en la subcapa MAC del modelo OSI

Juan Carlos Rivera Zabala

Universidad Nacional de Colombia
Ingeniería, Telecomunicaciones (Bogotá, Colombia)

2017

Un sistema multi-agente para la auto-configuración de las operaciones de red en la subcapa MAC del modelo OSI

Juan Carlos Rivera Zabala

Tesis de investigación presentada como requisito parcial para optar al título de:
Magister en Ingeniería de Telecomunicaciones

Director (a):

Luis Fernando Niño Vásquez, Ph.D.

Línea de Investigación:

Inteligencia artificial

Grupo de Investigación:

LISI

Universidad Nacional de Colombia

Ingeniería de Telecomunicaciones

Bogotá, Colombia

2017

No pienso nunca en el futuro porque llega muy pronto.

Albert Einstein

Agradecimientos

Por el valioso apoyo a Luis Fernando Niño Vásquez, Ph.D. Profesor Asociado de la Universidad Nacional de Colombia y su importante enfoque que me mantuvo en la dirección correcta en el desarrollo de la metodología científica, en los sistemas de inteligencia artificial y en especial en los sistemas multi-agente. Al Grupo de Investigación LISI por guiarme sobre el método científico, sus críticas y todas las reflexiones para mejorar este trabajo.

Además, todo el apoyo del Sebastián Eslava, Ph.D. Jefe de la Oficina de Tecnología de la Información OTIC Bogotá por el apoyo con recursos tecnológicos y laboratorios de prueba, la tutoría de los Profesores e Ingenieros de Proyectos Erick Ardila, MsC, Oscar Agudelo, MsC y Henry Zárate, MsC.

Resumen

En el presente trabajo se propone una arquitectura auto-configurable empleando un sistema multi-agente para el diagnóstico y optimización de una red de telecomunicaciones por medio de la implementación de un sistema de recolección de información en estaciones, servidores y equipos activos de la red, la información capturada se analiza, estandariza, filtra y procesa para el logro del objetivo seleccionado. Como soporte, se cuenta con un modelo de datos, el cual define y permite guardar la información necesaria para la reconfiguración de la red utilizando un modelo de optimización mediante el cual se realizan la reconfiguración de equipos y estaciones de red. El proceso de optimización es el mecanismo para el logro del objetivo que nos lleva a la toma de decisión. El sistema multi-agente interviene en el proceso de resolver muchos problemas de manera simultánea y desplegar la solución que consta de tareas aplicadas de una manera distribuida. Los resultados indican la factibilidad de una alternativa y permiten lograr la capacidad de ejecutar una solución automática de problemas. Una vez los agentes son comunicados de una tarea actúan sin la necesidad de un control centralizado.

Palabras clave: (Sistema multi agente, redes de computación, gestión de red, auto-configuración de redes, computación adaptativa, Capa MAC / OSI).

Abstract

In this work, a self-configuration computer network framework over a multi agent system for diagnostic, optimization and management was developed. The system follows several steps such as collecting information from workstations, servers and other active equipment in network. This information is tuned through parsing, standardization, and thus a data model takes scheme which who allows to keep the relevant information for the decision process. An optimization model is a key mechanism for achieving decision process. The multi agent system involves a process of solving simultaneously several problems and uses its distributed capacity for deploying a solutions. As a conclusion, it was possible to reach the goal, providing and deploying a self-configuration system on a network in a distributed way, when each agent does a job, it does not need a central control. When a computer network come into a partition state (two o many separate pieces), each piece must resolve their interconnection problems such as well-functioning, stability or self-diagnostic.

Keywords: (Multi agent system, computer networks, data network management, self-configuration computer network, adaptive computer, Layer MAC / OSI model)

Contenido

	Pág.
Resumen	IX
Lista de figuras	XIV
Lista de tablas	XVI
Lista de abreviaturas	XVII
Introducción	1
1. Los sistemas multi-agentes y la autoconfiguración de operaciones de red	7
1.1 Los sistemas complejos y la modelación con sistemas multi-agente	8
1.1.1 Auto-similitud en el tráfico y estacionalidad	11
1.2 Los sistemas distribuidos y la modelación con sistemas multi-agentes	14
1.3 Necesidad de las redes con auto-organizables	15
1.3.1 Modelos de redes auto-organizables	17
1.3.2 Características de las redes autónomas	19
1.4 Condiciones de red que llevan a un aumento de la complejidad	21
1.5 Necesidad de la auto-configuración de redes	24
1.6 El modelo OSI, la subcapa MAC y el sistema multi-agente	25
1.7 Conceptualización del sistema multi-agente	32
1.7.1 Definición de Agente	32
1.7.2 Arquitecturas de agentes	32
1.7.3 Características de los agentes	33
1.7.4 Capacidades del sistema multi-agente	36
1.7.5 Ubicación de los agentes	37
1.8 Jade como herramienta de sistema multi-agente	39
1.9 Modelo del agente recolector, actuador y de consolidación	41
2. Proceso de decisión en el modelo de agentes y redes de datos	43
2.1 Que es una decisión	43
2.2 La optimización	43
2.3 Escenarios de Auto-configuración	47
3. Descripción del experimento	49
3.1 Infraestructura de software	51
3.1.1 Agente principal de Jade	51
3.1.2 Capa media	52
3.1.3 Consola de agentes	52
3.1.4 Consola del software de gestión	54

3.1.5	Capa del cliente final.....	56
3.2	Infraestructura de hardware.....	56
3.2.1	Modelo de switches y routers.....	57
4.	Desarrollo del modelo e implementación del producto de software	61
4.1	Recolección de los datos.....	63
4.1.1	Granularidad, escala e intervalo de muestreo.....	63
4.1.2	SNMP.....	66
4.1.3	Analizadores de paquetes.....	66
4.1.4	RMON.....	68
4.1.5	Captura de flujos.....	68
4.2	Abstracción de datos y relaciones	70
4.2.1	Modelo de base de datos.....	72
4.2.2	Datos estacionales.....	77
4.3	Implementación.....	78
4.3.1	Plataforma base.....	78
4.3.2	Un servidor de base de datos para guardar los datos de tendencias, muestreos de información, según la ontología de redes (Ver 4.1.5 Captura de flujos	79
4.3.3	Diagrama de secuencia.....	82
4.3.4	Agente recolector y de análisis sintáctico.....	83
4.3.5	Agente de Consolidación de datos	85
4.3.6	Agente actuador.....	85
4.3.7	Probador de ancho de banda y aplicación	87
4.3.8	Integración de productos de software	88
4.4	Validación del modelo de auto-configuración	88
4.5	Estadísticas y análisis	89
5.	Conclusiones.....	95
5.1	Recomendaciones	96
5.2	Alcances y limitaciones	97
5.3	Trabajo futuro	97
A.	Anexo: Código del producto de software.....	101
B.	Anexo: Manual de instalación del software.....	102
	Bibliografía.....	CV

Lista de figuras

	Pág.
Ilustración 1. Ciencia de sistemas complejo dividido por áreas de conocimiento.....	9
Ilustración 2. Tráfico de red con intervalo de 5 minutos.	13
Ilustración 3. Tráfico de red con intervalo de 30 minutos.....	13
Ilustración 4. Estacionalidad y temporada de tráfico.	14
Ilustración 5. Modelo de capas OSI.....	26
Ilustración 6. Elementos involucrados en la captura de datos.	27
Ilustración 7. Modelo borde centro en topología.	30
Ilustración 8. Modelo capas borde-centro en modelo OSI.....	30
Ilustración 9. Ambiente multi-agente.	33
Ilustración 10. Coordinación y Control.....	36
Ilustración 11. Implementación del sistema multi-agente.	38
Ilustración 12. Diagrama general para el software.....	42
Ilustración 13. Definición de capacidad.	44
Ilustración 14. Función de transferencia entre estaciones.....	45
Ilustración 15. Agente principal Jade.....	51
Ilustración 16. Comunicación entre agentes.....	51
Ilustración 17. Persistencia de información en la base de datos.....	52
Ilustración 18. Consola de agentes.....	53
Ilustración 19. Portal principal de gestión.	54
Ilustración 20. Selección de opciones estadísticas.	55
Ilustración 21. Laboratorio de pruebas.	59
Ilustración 22. Diagrama de estados de TCP.....	62
Ilustración 23. Detalle de un datagrama.	65
Ilustración 24. Captura binaria de enrutador.....	67
Ilustración 25. Captura de paquetes con Wireshark.....	68
Ilustración 26. Conversaciones de red.	69
Ilustración 27. Ejemplo de múltiples conversaciones de red.....	70
Ilustración 28. Elementos tangibles de una red.	72
Ilustración 29. Relación MAC, IP, Interfaz.	74
Ilustración 30. Relaciones entre tráfico de red y la tabla ARP.	75
Ilustración 31. Modelo de la base de datos.....	77
Ilustración 32. Plataforma física para el sistema multi agente.	81
Ilustración 33. Diagrama de secuencia.....	83
Ilustración 34. Ejecución de comando en el sistema operativo.....	84

Ilustración 35. Máquina de estado del software.....	86
Ilustración 36. Ejemplo de funcionamiento de <i>iperf</i>	87

Lista de tablas

	Pág.
Tabla 1. Uso de abstracción OSI para sistema MAS.	31
Tabla 2. Equipos de laboratorio.	58
Tabla 3. Analogía de Química y Biología con Redes de datos.	64
Tabla 4. Productos de software.....	82
Tabla 5. Integración de productos de software.	88

Lista de abreviaturas

Abreviatura	Término
<i>3G</i>	Tercera generación de tecnología celular
<i>4G</i>	Cuarta generación de tecnología celular
<i>ABM</i>	Agen Base Model
<i>ACE</i>	Autonomic Communications Elements from CASCADAS
<i>ACL</i>	Agent Communication Language Specifications
<i>AD</i>	Active Directory
<i>Ad-hoc</i>	Wireless network ad-hoc
<i>AMS</i>	Agent Management System
<i>ANA</i>	Autonomic Network Architecture
<i>ANEMA</i>	Autonomic Network Management Principles
<i>ANM</i>	Autonomic Network Management
<i>AP</i>	Access Point Wireless
<i>ARP</i>	Address Resolution Protocol
<i>ASCII</i>	American Standard Code for Information Interchange
<i>ATM</i>	Asynchronous Transfer Mode
<i>BDI</i>	Belief, Desire and Intention.
<i>BGP</i>	Border Gateway Protocol
<i>BPDU</i>	Bridge Protocol Data Unit
<i>CDP</i>	Cisco Discovery Protocol
<i>COBIT</i>	Control Objectives for Information and related Technology
<i>CPU</i>	Central Processing Unit
<i>DAO</i>	Data Access Object
<i>DDOS</i>	Distributed Denial Of Service
<i>DF</i>	Directory Facilities
<i>DHCP</i>	Dynamic Host Configuration Protocol
<i>DNS</i>	Domain Name Service
<i>DSCP</i>	Differentiated Service Code Point

<i>EIGRP</i>	Enhanced interior gateway router protocol
<i>ETB</i>	Empresa de Telecomunicaciones de Bogotá
<i>FIPA</i>	Foundation for Intelligent Physical Agents
<i>FOCALE</i>	A novel Autonomic Networking Architecture
<i>FSF</i>	Free Software Foundation
<i>GNS-3</i>	Real time network simulator
<i>GNU</i>	G Not Unix
<i>GNU GPL</i>	GNU General Public License
<i>GSM</i>	Global System for mobile communications
<i>HA</i>	High Availability
<i>IDS</i>	Intrusion Detection System
<i>IEC</i>	International Electrotechnical Commission
<i>IEEE</i>	Institute of Electrical and Electronics Engineers
<i>IPS</i>	Intrusion Prevention System
<i>IPv4</i>	Internet Protocol Versión 4
<i>IPv6</i>	Internet Protocol Versión 6
<i>ISO</i>	International Organization for Standardization
<i>ISP</i>	Internet Service Provider
<i>ITIL</i>	Information Technology Infrastructure Library
<i>ITU</i>	Telecommunication Standardization Sector
<i>JADE</i>	Java Agent Development Framework
<i>JRE</i>	Java Runtime Environment
<i>L2</i>	Layer 2 del modelo OSI
<i>L3</i>	Layer 3 del modelo OSI
<i>LACP</i>	Link Aggregation Control Protocol
<i>LAN</i>	Local Area Network
<i>LDAP</i>	Lightweight Directory Access Protocol
<i>LGPL</i>	Lesser General Public License Versión 2
<i>LLC</i>	Logical Link Control
<i>LLDP</i>	Link Layer Discovery Protocol
<i>LTE</i>	Long Term Evolution

<i>MAC</i>	Media Access Control
<i>MAS</i>	Multi Agent System
<i>MPLS</i>	Multi Protocol Label Switching
<i>MRTG</i>	Multi Router Traffic Grapher
<i>MSS</i>	Maximum Segment Size
<i>MST</i>	Multiple Spanning Tree
<i>MTU</i>	Maximum Transmission Unit
<i>NAT</i>	Network Address Translation
<i>NMS</i>	Network Management System
<i>NS-3</i>	Discrete network event simulator
<i>ODBC</i>	Open Database Connectivity
<i>OSI</i>	Open System Interconnection
<i>OSPF</i>	Open short path first
<i>PC</i>	Personal computer
<i>PRS</i>	Procedure Reasoning System
<i>QoS</i>	Quality Of Service
<i>RAM</i>	Random Access Memory
<i>RFC</i>	Request For Comments
<i>SDN</i>	Software Define Network
<i>SFTP</i>	Secure File Transfer Protocol
<i>SNMP</i>	Simple Network management protocol
<i>SOA</i>	Service Oriented Architecture
<i>SON</i>	Self organizing network
<i>STP</i>	Spanning Tree Protocol
<i>TCP</i>	Transmission Control Protocol
<i>TILAB</i>	Telecom Italy Laboratories
<i>UDP</i>	User Datagram Protocol
<i>UUID</i>	Universal Unique Identifier
<i>VLAN</i>	Virtual LAN
<i>VPN</i>	Virtual Private Network
<i>WAF</i>	Web Application Firewall
<i>WCCP</i>	Web Cache Control Protocol

Introducción

Una forma de abordar los procesos de diagnóstico, operación y optimización de una red de datos es considerarlos como un sistema complejo; revisando sus características se encuentran que están compuestos por múltiples elementos, muchas interacciones entre elementos, cambios en el sistema que se propagan generando otras interacciones, y adicionalmente el comportamiento del sistema es difícil de deducir por sus elementos (Gershenson, 2010). Las propiedades del sistema como producto de las interacciones muestran comportamientos emergentes, en lo concerniente a las redes de datos, se puede encontrar que involucran muchos elementos entre los cuales están hardware, software, servicios y personas. La cantidad de interrelaciones es numerosa y se presenta entre cualquier combinación de componentes. Como consecuencia de las interrelaciones entre los elementos emergen comportamientos y fenómenos tales como la concurrencia en las transacciones de red o de una base de datos, los cambios en la calidad del servicio, la competencia por un recurso escaso, fallas por la sobrecarga en el uso de un recurso de manera intensa o el alcance del límite no conocido de cualquier recurso o variable, los cambios en los caminos para transportar información y restricciones de velocidad.

Por otro lado, una red de datos tiene una estructura denominada topología que se construye con elementos, equipos y canales de comunicaciones. Es usual que la topología de una red cambie en cada instante del tiempo debido a daños físicos, interferencias electromagnéticas, fallos humanos, cortes de energía entre otros tipos de incidentes. Estos cambios afectan a todos los demás elementos que componen una red y desencadenan una serie de interrelaciones entre todos los componentes.

Los usuarios de una red de datos la emplean para lograr propósitos en sus trabajos académicos e incluso para entretenimiento; en este momento el comportamiento de la red es influenciado sus decisiones, gustos, preferencias y modas. Estas necesidades se

transforman en intercambios de mensajes entre múltiples orígenes y destinos de una forma no predecible. Los elementos constitutivos de una red de datos también intercambian mensajes dentro de su proceso natural de administración y gestión. Esta interacción genera información nueva considerada otro elemento de los sistemas complejos.

En el uso de la red se considera que no hay un punto central de control. Cada usuario usa los recursos en el momento que lo requiere y de la forma que considera adecuada. No se debe confundir la administración de una red con el punto de control por cuanto esta función solo intenta mantener la red en operación. Pero no hay un elemento interactúe e indique cuando deben ocurrir las transacciones de red, por ello se considera un comportamiento aleatorio.

Respecto al aumento constante de la cantidad de elementos de interacción en un sistema complejo, se observa que sucede similar en las redes de datos. Las redes empresariales, industriales, caseras e Internet siguen creciendo en cantidad de usuarios, en multiplicidad de equipos de interconexión, enlaces de comunicaciones y cantidad de servicios ofrecidos (Sandvine, 2016).

No existe una arquitectura que permita entender o aproximarse a mostrar las dinámicas del sistema completo denominado "red de datos". No hay una solución o alternativa que satisfaga a diseñadores, administradores, fabricantes y usuarios. Las soluciones se enfocan en resolver de manera específica retos de administración, configuración, diagnóstico y optimización. Esto lleva a que se utilicen o se combinen muchas herramientas para lograr entender un problema e intentar optimizar. En general, no existe cooperación entre herramientas.

Por lo expuesto anteriormente, en este trabajo se propone un modelo y arquitectura que apoye la solución. Se requiere de la conformación de una plataforma de recolección, depuración y normalización de los estados de la red, un modelo de optimización y la ejecución de cambios de configuración o estructura en muchos componentes y dispositivos de red ubicados en lugares geográficos diferentes de manera integrada. También existe un reto adicional con el dinamismo de cambio de las condiciones de una red en cada instante. La solución óptima debe ser aplicada en lapsos de tiempo entre 5 a 300 segundos y de una forma coordinada para lograr efectividad antes que las condiciones para las que

fue planeado ya no apliquen. Las soluciones deben optimizar la utilización de recursos y brindar servicios de calidad (Mearns & Leaney, 2013).

La cantidad de información que se genera como consecuencia de nuestras acciones como humanos, crece de manera exponencial (Yang, Wakamiya, Murata, Iwai, & Yamano, 2016). Desde el punto de vista de la administración y gestión de las redes de computadores se cuenta con muchos mecanismos para tomar la información en diferentes tipos de granularidad y detalle. La forma como se debe actuar para garantizar los servicios está cambiando y cada vez implican retos más altos que deben ser satisfechos en menores tiempos.

Se ha podido observar el crecimiento de Internet en múltiples servicios, como el video por demanda domiciliario, las redes sociales, la masificación de las redes inalámbricas (WIRELESS), Internet de las cosas (Giordano, Spezzano, & Vinci, 2016), sistemas embebidos que llegaría a 6.000 millones para el 2016 (Flach et al., 2016).

Como consecuencia, la cantidad de profesionales y las herramientas actuales para administrar los sistemas de redes de computadores no serán suficientes para realizar todas las acciones que se requieran (Al-fares, Loukissas, & Vahdat, 2008).

Una alternativa que se tiene para enfrenar estos retos es la delegación de tareas en sistemas autónomos e inteligentes que con el empleo de sistemas distribuidos y cooperativos logren realizar la auto-configuración de las redes de datos (Ardila Triana, 2013), (Wooldridge, 2009). Los sistemas multi-agente (MAS) o modelado con agentes (ABM) permite la realización de las tareas con la delegación de recolección de información, toma de acciones correctivas mediante actuadores y toma de decisión por medio de agentes de consolidación (Uri Wilensky, 2015).

En este trabajo se diseñó, implementó y se verificó un sistema multi-agente para la autoconfiguración de las operaciones de una red de área local específicamente en la subcapa MAC del modelo OSI como son el ancho de banda latencia, retardo y pérdida de paquetes (Sánchez Cifuentes, 2012); en consecuencia se determinaron los componentes de una arquitectura necesaria para el diseño y la implementación del sistema de recolección y extracción en línea de los datos complementario al sistema tradicional que

emplea modelos de gestión como *snmp* o *netflow* (protocolo desarrollado por Cisco Systems para recolectar información sobre tráfico IP).

Por su característica de granularidad y diferencias en el proceso de recolección dentro de varios sistemas operativos, se hace necesario utilizar procesos de limpieza y estandarización de los datos para ser revisados, procesados y analizados.

Para obtener una mejor solución se empleó una estrategia con procesos de decisión; por medio de una metodología de optimización para una función objetivo de una variable; los resultados de la optimización entregaron el punto y la variable que debe ser modificada mediante un módulo de software que logra la ejecución de los cambios dentro de la configuración de los equipos activos de una red y en cada uno de los objetos que contengan los agentes como son PCs o servidores. En este mismo punto, se puede observar el funcionamiento del modelo de agentes los cuales mediante su proceso social de comunicación recibe y cambia el ambiente.

Se enfocó el trabajo en lograr la recolección de datos y variables de una red. Las variables están enfocadas en la subcapa MAC de la capa 2 del modelo OSI y a su dependencia con variables de IPv4. El manejo de IPv6, redes Wireless o Ad-hoc o manejo de la internacionalización por sistemas operativos en múltiples idiomas no se encuentra dentro del alcance de este proyecto.

Se desarrolló el trabajo como investigación aplicada y cuantitativa. Mediante trabajo en laboratorio y de forma experimental se utilizó un grupo de *switches*, enrutadores, PCs y servidores donde se simula la congestión de la red, problemas físicos de conectividad, fenómenos de retransmisión de paquetes, desvío de información para alimentar una base de datos. Se caracterizó el comportamiento del tráfico por muestreo de datos a un intervalo regular mediante los agentes autónomos. Formulando una arquitectura de software y hardware se logra realizar varias pruebas como medir el ancho de banda desde múltiples lugares y envío de pruebas a un servidor web en transacciones habituales de una topología de red tradicionalmente en forma de árbol. La arquitectura de laboratorio incluye la ubicación de servicios de servidor para someter a estrés la red con "*nping*" ("Nmap," 2017), servidor web y medidor de ancho de banda con el software IPERF ("lperf," 2017) (*open source FSF GNU GPL* ("GNU," 2017) (Stallman, 2017).

El lector podrá conocer y entender un conjunto de componentes que deben emplearse para modelar sistemas multi-agente con los objetos aplicados a problemas de redes de datos, contar con una herramienta marco de trabajo para analizar de manera holística una red de datos con el sistema distribuido, capacidad de enviar un cambio de configuración en múltiples elementos participantes en una red como son los objetos clientes finales.

También encontrarán algunas bases para dotar de inteligencia y autonomía a una red de datos, y desarrollar una herramienta para el administrador de red para ofrecer un similar valor de servicio en cada punto de la red.

1. Los sistemas multi-agentes y la autoconfiguración de operaciones de red

Los sistemas multi-agentes son considerados por algunos pensadores como una bifurcación de los sistemas de inteligencia artificial (Shehory & Sturm, 2014), mientras otros pensadores insisten que es otra forma de abordar problemas que tienen la característica de contar con alta riqueza de información contenida dentro de muchos objetos o entidades. Las redes de telecomunicaciones cuentan con la propiedad de contener información en sus componentes estructurales y además cada nodo (PC, servidor, teléfono inteligente) tiene contacto con la identidad de los humanos y consecuentemente sus comportamientos, hábitos y deseos. Otros pensadores también aseguran que los sistemas multi-agentes son una nueva forma de pensar y se consideran que son la siguiente evolución de la programación orientada por objeto (Wooldridge, 2009).

La riqueza de las redes de telecomunicaciones para los humanos radica en el uso de esta herramienta como elemento para trabajar cooperativamente, compartir recursos, agilizar los procesos de comunicación, incluso llegando a modelos estándares de intercambio de información.

En el momento en el cual converge la información, máquinas y humanos se inician todo tipo de interacciones, transformación de datos. Las interacciones que se generan afectan de manera positiva y negativa a todos los objetos, a la forma en que se comunican entre ellos y sus decisiones futuras. Un ejemplo, es notar las múltiples formas de comunicar punto a punto, punto a multi-punto o punto al universo. Las interacciones se forman en una malla infinita de combinaciones. A pesar de conocer los principios de diseño de las redes de datos, es imposible predecir el siguiente estado de interacción entre los componentes o los individuos. Este comportamiento nos lleva a la necesidad de utilizar sistemas de modelado que entiendan y analicen estos sistemas conocidos como sistemas complejos

(Wooldridge(eds.), 2012). En los sistemas de gestión de redes tradicionales se han utilizado métodos de recolección de información desde un punto central que han llevado a limitaciones como dificultad en la toma la información con granularidad; con este método se pierde la captura de las interacciones, dado implica cantidades considerables de esfuerzo computacional en el punto central para el logro de la recolección y procesamiento de los datos. Los procesos de cálculo alcanzan el nivel de complejidad computación n^n por las m ecuaciones que se intentan optimizar para obtener una mejor satisfacción en el sistema de redes. Para el momento en el cual los resultados de optimización se encuentran procesados, en muchas ocasiones la aplicación del modelo de configuración sobre los elementos ocurre muy tarde dado que la red para este momento cambio a su siguiente estado. Además, la ecuación de cambio puede en una red puede estar constituida por una complicada lista de pasos y configuraciones en diferentes elementos de la red en ubicaciones geográficas diferentes.

Los sistemas multi-agente, por el contrario, permiten modelar un sistema complejo porque permite contar muchos elementos denominados agentes (“en computación se denomina agente de software a una porción de código de un lenguaje de programación que funciona de manera autónoma”) ubicados en múltiples lugares geográficos y lógicos. A través de los agentes es posible realizar tareas distribuidas para recolección de información al estilo de las sondas de los modelos centralizados. Pero los agentes pueden contener procesos para transformación del ambiente cercano de forma autónoma. Los agentes también poseen la forma de comunicar a los demás, la información de su ambiente permitiendo a los demás mejores decisiones.

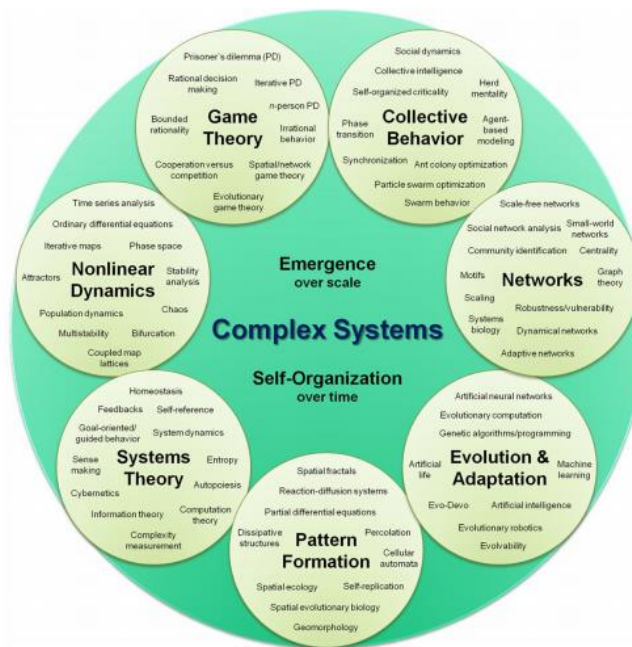
1.1 Los sistemas complejos y la modelación con sistemas multi-agente

Los sistemas complejos están hechos de un número de componentes que interactúan entre sí de una forma no lineal. Los sistemas complejos pueden surgir y evolucionar a través de la auto-organización de tal forma que no son ni completamente regulares ni totalmente aleatorios, permitiendo la emergencia de comportamiento a escala macroscópica (Sayama, 2015).

Los procesos emergentes están relacionados con las propiedades del sistema vistos a diferentes escalas microscópicas y macroscópicas. La emergencia es una relación no trivial entre el sistema y la escala cuando no se puede explicar en lo microscópico. Esta definición es común en la mayoría de autores y pensadores de los sistemas complejos.

Es importante anotar que no se debe confundir los fenómenos emergentes con auto-organización la cual es un proceso dinámico por el cual el sistema espontáneamente forma estructuras macroscópicas y comportamiento en el tiempo. Muchas áreas del conocimiento estudian los sistemas complejos. Se muestra una clasificación por áreas de conocimiento que emplean modelado de sistemas complejos en una división en siete áreas (propuesta por Sayama).

Ilustración 1. Ciencia de sistemas complejo dividido por áreas de conocimiento.



Tomado de "Introduction to the Modeling and Analysis of Complex Systems, Hiroki Sayama, 2015"

La utilización de sistemas multi-agentes (también modelo por agentes ABM) es útil para enfrentar los retos de los sistemas complejos como el abordaje al diseño del entretreído de relaciones. Es difícil separar o definir las interrelaciones relevantes entre objetos o individuos. La interacción entre individuos contiene información del sistema que no se

puede despreciar y afecta el funcionamiento futuro de todos los individuos del sistema. Este enfoque es diferente del pensamiento clásico donde se aborda el análisis de fenómenos aislando cada uno de los componentes hasta sus componentes atómicos. Para algunos fenómenos muy simples puede aplicarse el modelo de reduccionismo dado que somos conscientes de la información que se ignora. Pero no es el caso para los sistemas complejo (Gershenson, 2010).

Los sistemas multi-agentes se utilizan para modelar sistemas complejos como enjambres de aves, colonias de hormigas y termitas, cadenas industriales de suministro, redes sociales, decisiones en los mercados de valores. La metodología para resolver sistemas complejos es universal utilizando modelos con agentes mediante una arquitectura modular compuesta por comportamientos asignados a los agentes, espacios de tiempo, un programador de tareas (*scheduler*), bitácoras y una interfaz (North, 2014). El formalismo se construye basado en el cálculo- λ .

El cálculo λ puede ser llamado el lenguaje de programación universal más pequeño del mundo. El cálculo consta de una única regla de transformación llamada conversión β y un único esquema definición de función. Este cálculo fue presentado en la década de 1930 como una herramienta para formalizar el concepto de que algo es computable efectiva (Rojas, 2015). El cálculo es universal en el sentido que cualquier función computable puede ser expresada y evaluada utilizando este formalismo. En este sentido es equivalente a las máquinas de Turing. Sin embargo el cálculo enfatiza el uso de reglas de transformación simbólica y no tiene en cuenta la implementación de la máquina. Este es un enfoque más relacionado con el software que con el hardware (Varela, 2013).

Se encuentran por tanto particularidades del sistema complejo útil para el sistema multi-agente MAS

- Interacción entre elementos.
- Comportamientos emergentes.
- Aprender de la experiencia.
- Se toman decisiones de manera autónoma.
- Los elementos mediante comportamientos modifican el ambiente.
- Adaptabilidad.
- Heterogeneidad de los objetos.

- Altamente dinámico. No se puede formalizar con ecuaciones lineales.

Como consecuencia es necesario trabajar en los problemas de decisión de manera específica e independiente. Se encuentra probado de manera formal (cálculo- λ) que el desempeño de modelo MAS es de forma asintótico en el tiempo y espacio para que sea computacionalmente óptimo. En este contexto el término "óptimo" significa que no existe otra técnica que pueda solucionar el problema de manera computacional (North, 2014).

La cantidad de CPU (tiempo) y almacenamiento (espacio) pueden ser muy grandes para modelar un sistema complejo. Si el trabajo se divide y se clasifica la experiencia en el sistema entregando a los agentes tareas entonces se evitan cuellos de botellas o fallas del sistema por falta de robustez. La determinación de nivel de descripción o escala nos permite determinar la aplicabilidad de los sistemas complejos o reduccionistas. Las escalas de muestreo de los datos hace la diferencia en la interpretación de comportamientos (Agudelo Rojas, Oscar; Hernández Pérez, 2006).

Para el caso del modelo en este trabajo, se revisó el impacto del comportamiento de estacionalidad y auto similitud en el tráfico de las redes de computación. Se ha estudiado y comprobado mediante caracterización estadística que existe un comportamiento de auto-similitud como en los fractales a diferentes escalas (Agudelo Rojas, Oscar; Hernández Pérez, 2006).

1.1.1 Auto-similitud en el tráfico y estacionalidad

En las redes de datos se pueden observar dos fenómenos que son importantes para el análisis, diseño, control e implementación de sistemas de gestión, administración de redes que afectan el tráfico de datos; son la auto-similitud y la estacionalidad del tráfico.

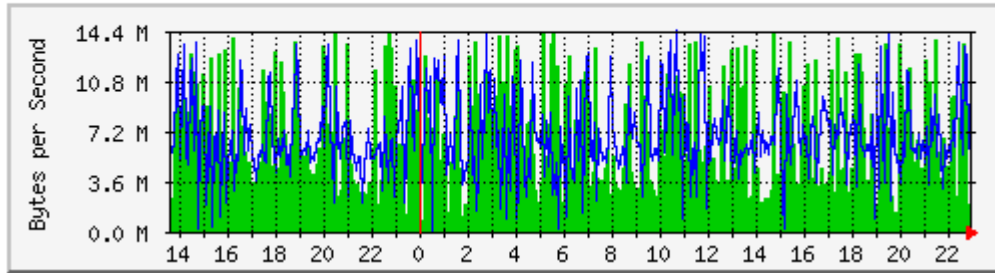
Los autores (Park & Willinger, 2000), (Paxson & Floyd, 1995) demostraron que el tráfico de datos está compuesto de ráfagas de transmisión de datos que llevan a picos de utilización de los recursos de ancho de banda y cantidad de procesamiento en la infraestructura de red (término en inglés *burst* o *burstiness*) que son impredecibles. Además, si se cambia la escala de tiempo en el intervalo de muestreo en las muestras de

tráfico desde milisegundos, minutos, horas se observa el mismo comportamiento en las gráficas de tráfico. Se observó que este comportamiento no se puede moldear a ninguna ecuación y es totalmente estadístico. Se ha considerado que el conjunto de variables en el tráfico de redes es independiente, en especial la cantidad de conexiones de red y la duración de la transmisión. La correlación entre estas variables no decae tan rápidamente y puede persistir a través de muchas escalas de tiempo. Este fenómeno, que afecta significativamente el desempeño de las redes de comunicaciones, se puede representar adecuadamente mediante modelos de tráfico fractal o auto-similitud.

El estudio del componente de Hurst (Hurst, Black, & Simaika, 1965) nos permite medir la rugosidad de las gráficas de tráfico y su proceso de comportamiento aleatorio. El exponente de Hurst ocurre en varias áreas de las matemáticas aplicadas, incluyendo los fractales y la teoría del caos, procesos de larga memoria y análisis espectral. La estimación del exponente de Hurst se ha aplicado en áreas que van desde la biofísica a las redes de computadoras. La estimación del exponente de Hurst fue desarrollada originalmente en la hidrología. Sin embargo, las modernas técnicas para estimar el exponente de Hurst vienen de las matemáticas fractales (Hurst et al., 1965).

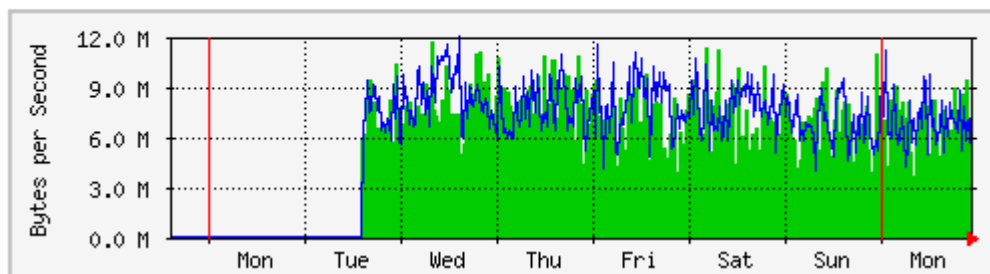
Por otra parte la estacionalidad de tráfico (en inglés *seasonal*) ha demostrado importantes aproximaciones al modelamiento del tráfico y la sobrecarga en la red. La predicción del tráfico mediante la utilización de series de tiempo ha permitido mejorar los sistemas de gestión preparándolos para las ráfagas. Para estos modelos se agrega el monitoreo de las ráfagas de tráfico según la forma mediante la cual se utiliza la red LAN (Kim, 2011). Se recogen datos históricos en períodos largos (años si es posible), y las fechas de períodos críticos. Las temporadas de uso de ráfagas de red coinciden con los eventos humanos de fechas o hitos importantes de proyectos o de la vida cotidiana entre otros podemos mencionar los cierres financieros, cierres de notas en las universidades, períodos de vacaciones, fines de semana, eventos de audiencia mundial como los Juegos Olímpicos o eventos de noticias, los períodos nocturnos, días festivos (Paxson & Floyd, 1995).

Ilustración 2. Tráfico de red con intervalo de 5 minutos.

'Daily' Graph (5 Minute Average)

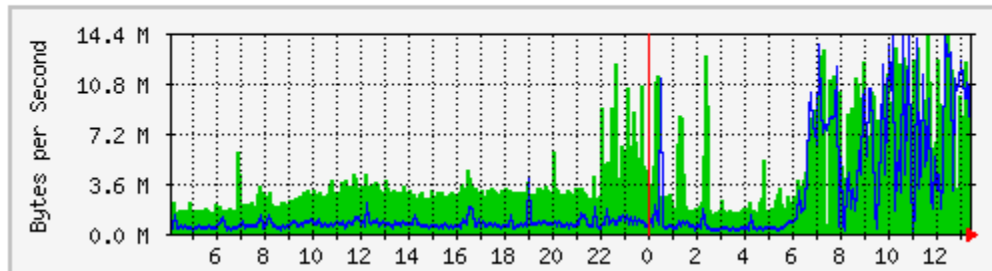
Para evidenciar este comportamiento, se realizó un muestreo a intervalo de 5 minutos en un equipo de red (*switch Cisco Nexus 5000*) ubicado en el corazón de la red, en un puerto que transporta la información de servidores en el Centro de cómputo de la Universidad Nacional de Colombia sede Bogotá. En las gráficas se encuentra la cantidad de bits de entrada y salida del puerto (en la gráfica se encuentra representado en dos colores verde y azul). El tráfico presentado muestra las ráfagas y picos (*burstiness*) mencionados por Hurst.

Ilustración 3. Tráfico de red con intervalo de 30 minutos.

'Weekly' Graph (30 Minute Average)

En el cambio de escala de muestreo a un intervalo de 30 minutos, se observa un comportamiento similar. Se puede suponer un comportamiento de auto-similitud en este ejercicio.

Ilustración 4. Estacionalidad y temporada de tráfico.

'Daily' Graph (5 Minute Average)

Para el mismo ejercicio de muestreo de tráfico cada 5 minutos, se puede observar el comportamiento de estacionalidad. Para un día domingo o festivo se observa un tráfico por debajo de los 3.6 Megabits/seg. (En la gráfica se observa desde la izquierda a derecha, las 6:00am del día anterior, hasta las 6:00 del día normal laboral). En los momentos de día laboral normal se comprueba una utilización entre 10.8 a 14.4 Megabits/segundo.

1.2 Los sistemas distribuidos y la modelación con sistemas multi-agentes.

La forma de construir sistemas de gestión óptimos para la red nos permite adoptar filosofías como un modelado modular y funcional de las capas del modelo OSI para realizar una solución distribuida. Incluso se puede determinar un observador por cada capa y manejarlo con un módulo de computación distribuida independiente. De este forma se ve a la red como un optimizador para maximizar la utilidad como medida de satisfacción (Chiang, Low, Calderbank, & Doyle, 2007). Los sistemas distribuidos son una extensión de los sistemas concurrentes que nos permiten hacer uso de los hilos de ejecución (programación concurrente) para de ejecutar tareas de manera paralela. No olvidar las limitaciones o problemas que existen si se utilizan sistemas de memoria compartida en la cual dos procesos llegan a modificar el mismo sector de memoria RAM.

Los sistemas distribuidos poseen retos adicionales. De acuerdo con el teorema CAP (*Capacity, Availability and Partition Tolerance*), postulado por Erick Brewer (Gilbert & Lynch, 2002), se debe tener en cuenta en un sistema distribuido sus propiedades de consistencia (la entrega de una respuesta correcta por parte de un servidor a una petición realizada), disponibilidad (cada petición eventualmente recibe una respuesta) y la tolerancia a

particiones (son las fluctuaciones de las comunicaciones entre servidores, las cuales pueden llevar a segmentos de red en donde múltiples grupos de elementos no se pueden comunicar entre sí). Adicionalmente se deben tener en cuenta los resultados del modelo FLP (Fischer, Lynch y Patterson), donde se presenta el teorema del consenso para garantizar en un sistema asíncrono la respuesta a una solicitud hecha. Es importante anotar que no es posible en un sistema de cómputo distribuido tener los componentes de Completos del teorema CAP. Por lo cual en el sistema propuesto, se tendrá el modelo de operación bajo A-P (por su sigla en inglés de *Availability and Partition Tolerance*), donde se persigue la disponibilidad y tolerancia a particiones, con el fin de priorizar la conectividad en la capa física y ser tolerante a particiones. Se deben implementar los modelos de replicación y alta disponibilidad.

1.3 Necesidad de las redes con auto-organizables

Cuando se habla de sistemas auto-organizables se debe enfocar la atención en el paradigma de control en los sistemas complejos. Se descubrió que los sistemas que están compuestos de una gran cantidad de subsistemas necesitan alguna clase de control autónomo que permite el adecuado funcionamiento del sistema con capacidad de escalamiento. Además, estos sistemas puede enfrentar cambios en el ambiente y adaptarse a condiciones desconocidas (Dressler, 2008). Así entonces, las redes auto-organizables (término en inglés *adaptive o self organizing systems*) centran su atención en las interacciones entre individuos de un sistema para obtener una gran cantidad de información para mejorar la satisfacción mediante el proceso social (Gershenson, 2010). En las redes auto-organizables se estudian los sistemas adaptativos y la capacidad de escalabilidad.

Computación adaptativa.

En las redes de datos se ha observado una tendencia mundial del crecimiento de servicios, aplicaciones y objetos disponibles para una variedad de perfiles o tipos de usuarios. Los procesos de gestión y administración de red deben construirse para ser más autónomos e inteligentes como requisito para poder atender las necesidades de los usuarios y sus retos de encontrar información o computación (Chowdhury & Boutaba, 2009). Se sugiere construir modelos para procesar grandes cantidades de datos de gestión de red, tomar

decisiones descentralizadas aplicables a una parte del universo pero manteniendo un equilibrio entre la satisfacción de todos, en un símil al sistema nervioso del cuerpo humano donde se delegan en varios subsistemas la toma de decisiones o acciones.

En Colombia¹, desde el año 2013 se han iniciado los trabajos de implantación de redes celulares con la llamada 4G LTE (Viering, Dattling, & Lobinger, 2009), que es la implantación de la norma LTE desarrollado por el grupo de investigación 3GPP (<http://www.3gpp.org/specifications/specifications> Versión 10) conformada por los fabricantes mundiales Cisco, Ericcson, Nokia, Siemens. Dado que la tecnología de red celular avanzó a la siguiente generación 4G LTE, los estudiosos, fabricantes e investigadores plantean la necesidad de investigar, evolucionar e implementar sistemas de gestión con las facilidades de auto-configuración, auto-optimización (Agoulmine, 2011) y auto-remediación (Whittle, Sawyer, Bencomo, Cheng, & Bruel, 2009).

Por otra parte, además de los modelos de implementación de tecnología, se encuentra el reto de propender por la disminución o control de los costos de operación y mantenimiento de una red (den Berg et al., 2008). Se han enfocado recursos y atención en el diseño de redes con auto-organización. Por tanto se hace necesario estudiar y resolver los mecanismos para lograr la auto-configuración automatizada (Derbel, Agoulmine, & Salaün, 2009).

Mark Weiser (Weiser D., Mark, 1987) considerado el padre de la computación ubicua sentó las bases para las redes autónomas (en inglés, *pervasive networks*), y planteó que la tecnología de comunicaciones y computación avanzaría de manera rápida hasta estar embebida y presente en todos los procesos humanos (Shambhu Upadhyaya Abhijit Chaudhury, 2002).

En los años recientes se han realizado múltiples avances y aportes para la autonomía en las redes. Un primer avance se observó con el esfuerzo de científicos de IBM en redes autónomas y el concepto de descentralización de funciones de red que comenzó en el 2001 (Berrayana, Youssef, & Pujolle, 2012).

¹ Ministerio de Comunicaciones y Tecnología abrió licitación para 4G LTE en Marzo de 2013. <http://www.mintic.gov.co/portal/604/w3-article-1716.html>

En los años posteriores, se han planteado varios modelos de auto-gestión de red como son ANEMA (Derbel et al., 2009) impulsado por IBM logrando auto-optimización asistida según los objetivos que elige un operador humano. El modelo ANA (*Autonomic Network Architecture*) agrega la importancia de la parametrización de los nodos de red actuando de manera continua sobre el ambiente. De igual manera se avanzó en formas de auto-organizar redes con autonomía federada (Bouabene et al., 2010). Los modelos ACE (CASCADAS) y FOCALÉ, *HiMang* (Choi et al., 2011), ANM (*Autonomic Network Management*) presentan en común la necesidad de administración, autonomía, escalabilidad, robustez adaptándose a las tendencias mundiales. También se promueve el uso de las redes autónomas aplicadas a la explotación de red móviles de LTE (Mortier & Kiciman, 2006).

Más recientemente, los fabricantes Nokia-Siemens trabajaron en las redes del futuro con su proyecto SON (*Self-Organizing Network*). Encontraron que la automatización de las redes y cómputo es una forma para lograr alcanzar niveles más avanzados en la calidad de los servicios entregados a los usuarios en las modernas redes de 4G LTE. Proveedores de tecnología como ERICSSON y HUAWEY enfocan todos los nuevos desarrollos en software de gestión como redes inteligentes. El fabricante Cisco por su parte ha desarrollado un enfoque para la auto configuración de redes enfocado a la defensa contra ataques informáticos (Hu, Zhang, Zheng, Yang, & Wu, 2010). Los fabricantes y los investigadores asociados a universidades como Harvard, CALTECH, MIT presentaron varios modelos de autonomía en redes como son ANEMA de IBM, DAVINCI (He et al., 2008), ANM e incluso un modelo gráfico (guiado o asistido por gráficos de color) (Bugenhagen, Morrill, & Edwards, 2008). Del año 2012 hasta el presente, se ha planteado el modelo de redes definidas por software SDN (*Software-Defined Networking*) en donde se intenta centralizar el plano de control de los equipos activos y centralizar la administración de red (Astuto, Mendonça, Nguyen, Obraczka, & Turletti, 2014).

1.3.1 Modelos de redes auto-organizables.

Se encontraron varios modelos para realizar una implementación de sistema de autonomía para gestión de red mediante los cuales se confirma la necesidad de trabajar en este segmento.

Modelo de Gestión dirigido por una interfaz gráfica. Patente US8279874b1. Método para presentar la información de las comunicaciones entre varias estaciones de una red de modo gráfico. Se presenta el diseño de una base de datos de datos de gestión de red para visualizar las interacciones entre aplicaciones y estaciones. De igual forma, se hace énfasis en que no se pueden observar todas las comunicaciones entre pares de estaciones de manera directa, si no que siempre se necesita una tercera estación que captura la traza de la conversación (Lu et al., 2012). Se hace evidente la necesidad de una estación de la red que capture información entre pares de estaciones, mediante una pre-selección de las características de comportamiento que se desea observar enfocado al comportamiento del usuario (Bugenhagen, Morrill, & Edwards, 2008).

Modelo ANEMA. Modelo que confirma la percepción de los administradores de red, respecto a que el crecimiento de escala de las redes llegará a tal punto que será muy difícil de administrar y monitorear. IBM propuso como alternativa la computación autónoma (ANEMA) aplicada a los equipos de red según la utilización de una función $f(x)$ de utilidad que satisface los objetivos de los usuarios. Se propone crear una política de comportamiento que será acatada y adoptada por los equipos de red. Se plantea la forma de encontrar una política de comportamiento de necesidades de usuario (Derbel et al., 2009).

Modelo de red Davinci. Modelo que propone compartir el substrato de la red (todos los elementos que componen la estructura del corazón del funcionamiento de una red) para ofrecer los servicios de redes virtuales y reales, a los clientes mediante asignación de recursos dinámicos. Se encontraron riesgos altos de estabilidad. Se propone una utilización del modelo de asignación de recursos de manera adaptativa y se prueba que es un modelo estable y que maximiza el desempeño. La aplicación del modelo de red Davinci demuestra la usabilidad y la adaptabilidad de las redes (He et al., 2008).

Modelo ANM (Autonomic Network Management). Modelo de red que auto-detecta, diagnostica y repara fallas, adapta su configuración y optimiza su desempeño y calidad de servicio. Desde hace tiempo han existido varios métodos de trabajo autónomo para resolver problemas de red, como los algoritmos de control de enrutamiento de estado de los canales de comunicaciones (*link-state*) conocido como *OSPF* y *BGP*. En otro escenario se puede ver el control de congestión de ventana deslizante de las transmisiones de TCP.

Se promueve el uso de las redes autónomas aplicada a la explotación de las redes móviles LTE (Champrasert & Suzuki, 2007). El modelo plantea que se deben tener acciones y decisiones para cada fenómeno que se pueda presentar. Se sugiere el conocimiento del ámbito o entorno, de las variables del medio ambiente y datos de entrada permitiendo tomar acciones automáticas. Es importante tener presente que la cantidad y calidad de los datos lleva a una buena toma de decisión. Muchos modelos se construyen sobre supuestos o condiciones ideales. Los cambios en el entorno invalidan todas las suposiciones que se tengan. Por esto, se debe dar manejo a las fallas comunes. Además de conocer el ambiente, es importante conocer los escenarios de prueba y comprobar que los comportamientos automatizados controlan de manera adecuada el problema (Mortier & Kiciman, 2006) .

1.3.2 Características de las redes autónomas.

Dado que existen muchos problemas y retos en la redes desde sus componentes de físicos (hardware), su lógica (software y algoritmos) para estabilidad de su topología y manejo de tráfico, se presenta una clasificación de las redes autónomas; se presentan las clases de soluciones aplicables en el entorno de cada problema. Se muestra como las redes auto-configurables son un caso particular de las redes autónomas (*Autonomic Networks*).

Red Auto-Configurable. Corresponde al aumento de la confiabilidad y desempeño reduciendo costos con técnicas automatizadas (Mortier & Kiciman, 2006) . En las redes de cobertura mundial y con alta complejidad sus operaciones de red son difíciles de controlar como un todo. Se emplea la descentralización, capacidad de toma de decisiones autónomas con metas y objetivos muy precisos para lograr resultados exactos. Una red auto configurable se compone de elementos que automáticamente son provistos: los recursos de red están preparados para dar satisfacer el servicio ante la ocurrencia de un evento. Para cumplir con un servicio se modela un perfil que contiene uno o más comandos que configuran una estrategia sobre uno o varios dispositivos de red que conforman una conectividad de extremo a extremo. Esta estrategia puede estar aplicada a nivel de un puerto físico. Lo que implica que se pueden desplegar un grupo de eventos en secuencia sobre otros dispositivos (Lu et al., 2012).

Red Adaptativa. Sistema de red que permite la toma de decisiones para adaptar servicios y recursos de acuerdo a los cambios del entorno y las necesidades del usuario. Se debe entender y ver más allá de una simple comprensión del uso plano de las máquinas y mejor verlo como un grupo de sensores capaz de adaptarse y entregar soluciones más cercanas al sentir humano (Cheng et al., 2010).

Auto-Gestión. En las redes inalámbricas se aplica la adaptabilidad de la red para poder brindar un canal de transporte a pesar de la cantidad de cambios en los canales o por la movilidad de los usuarios. Se debe contar con la función de red de auto-organización. Se presentan cuatro paradigmas reflejados en los protocolos de diseño de interacciones locales, coordinación explícita, estados y diseño de protocolos para adaptar cambios (Perlman, 2009).

Auto-análisis. El plano de control de mantenimiento y operación de una red es una capa media entre los objetos y la red. Se puede lograr conformar que una capa media (conjunto de equipos, software, interfaces orientadas a SOA) entregue información de diagnóstico elaborada a las estaciones de gestión (*NMS* por su sigla del inglés *Network Management System*) con acciones correctivas y con resultados predictivos. Esto daría mejor efectividad en problemas recurrentes. De esta forma la capa media automáticamente le provee a la red la capacidad de auto-análisis de ingeniería (Glitho & Svensson, 2001).

Auto-optimización. La tendencia mundial de las comunicaciones las está imponiendo las redes de 3G y LTE. Para reducir costos operacionales estas redes están provistas de sistemas avanzados de auto optimización. Se presenta un método de traspaso (movimiento de un servicio o usuario a otra antena por efecto de congestión *handover*) basado en la cantidad de celdas sobrepuestas. Se ajustan de manera automática varios parámetros como tiempo de disparo, intervalos de medida, histéresis mejorando los resultados tradicionales del traspaso. Es interesante observar y comprobar en el laboratorio que existen mejores forma de optimizar una red (Zhang, Wen, Wang, Zheng, & Lu, 2009).

Sistema de auto-estabilización de red. Son decisiones que generan actividades reactivas de auto-configuración. Por ejemplo tener la opción de elegir manejar de manera dinámica el enrutamiento según varias parámetros de decisión. Para interactuar con el

protocolo de auto-configuración se debe tener un módulo de resolución de conflictos de configuración (Forde, Doyle, & Mahony, 2005) y (Konstantinou, Florissi, & Yemini, 2002).

Sistema de auto-diagnóstico de falla. Sistema que permita encontrar la falla. Es el proceso de lograr inferir la falla exacta en una red partiendo de un conjunto de síntomas observados. Las fallas en la red son inevitables pero el diagnóstico y detección sin claves para la estabilidad, consistencia y desempeño (Dusia & Sethi, 2016).

1.4 Condiciones de red que llevan a un aumento de la complejidad

Los sistemas de redes de datos se ven sometidos a condiciones adversas para la óptima transmisión de información. Suponiendo una red perfecta en su capa física en donde no existan pérdida de paquetes o retransmisión, aun así existen asuntos que deben resolverse en el diagnóstico para lograr altas tasas de eficiencia de transmisión (término en inglés *throughput*). Existen varias clases de retos que debe afrontar una conversación de red para lograr su transmisión.

Problemas perceptibles al usuario. Cuando se utiliza una red de datos se perciben problemas como la demora en la descarga de información de una página WWW dado que se compara el comportamiento de un día anterior o por su uso rutinario. De igual forma cuando un computador presenta alguna falla. A continuación se presenta los problemas perceptibles:

- Congestión por alto tráfico. Comportamiento de temporada, picos, alta tasa de transmisión.
- Agotamiento de la cantidad de sesiones, memoria en la estación origen, destino o en equipos de red en el camino.
- Agotamiento de los recursos de computación en las estaciones del origen y destino.
- Problemas en alguno de los equipos de comunicaciones o en sus interfaces de físicas de red en equipos de transmisión.

- Fallas en los medios de transmisión o fallas de energía.

Problemas imperceptibles al usuario. Existen otras situaciones que ocurren en los planos de datos y control de los dispositivos de red. Los administradores de red y software de apoyo realizan configuraciones para cumplir algún objetivo de seguridad informática, optimización de uso de ancho de banda o necesidades de auditoría. A continuación se presenta un conjunto de problemas imperceptibles.

1. Las múltiples versiones de sistemas operativos cuentan con implementaciones diferentes de los protocolos. Se encuentra interacciones entre versiones de TCP y algoritmos de manejo de ancho de banda y control de congestión.
2. Compresión de la transmisión de datos. Compresión entre cliente y servidor web.
3. Encaminadores de paquetes. Se desvían servicios de red a otros servidores en el medio que procesan alguna función (término en inglés conocido como *route map*).
4. Servicios intermediarios explícitos o transparentes. Servicios que realizan funciones a nombre de otro servicio o usuario (*service proxy*). Utilizados para control u optimización de tráfico.
5. Desvío selectivo de protocolos de navegación. Método mediante el cual se desvían grupos de protocolo como navegación a internet (http), descarga de archivos (ftp) mediante servidores intermedios. (*Web Cache Control Protocol WCCP*).
6. Sistemas de antivirus. Sistema que interviene el tráfico mediante agentes que implementan funciones de control, filtrado de código malicioso. También puede realizar eliminación o cierre de sesiones TCP con el fin de dar manejo en línea a una epidemia de virus. Estos sistemas tienen problemas de falsos-positivos.
7. Protectores de intrusos y Firewalls (*IDS Intrusion Detection System, IPS Intrusion Prevention System*). Sistemas que permiten realizar la inspección de las conversaciones de red y de los paquetes. Utilizados para realizar control de perímetro para dar acceso a la información solo a las personas correctas. También tienen consecuencias de falsos positivos.
8. Sistemas de control de contenido y limpieza de código malicioso (*malware, rootkits*). Sistemas que permiten que los usuarios utilicen los servicios de red que cumplan con el

objetivo misional de la organización. También permite clasificar los horarios de navegación por categorías o cumplir con una cuota de navegación por usuario.

9. Administradores de ancho de banda. (*Quality of Service QoS, Differentiated Service Code Point DSCP*). Modelan el tráfico de red asignando espacios virtuales en los canales de comunicaciones. Para los tráficos considerados por la organización como no misionales se encuentran sometidos a limitaciones de ancho de banda.

10. Algoritmos de enrutamiento (*BGP, OSPF* entre otros). Se utilizan para manejar múltiples caminos entre origen y destino. Los algoritmos de enrutamiento permiten redundancia en la comunicación. Pero cuando existen problemas físicos en los enlaces como transmisión en un solo sentido de un enlace afectarán la entrega de la información. Estos fenómenos se hacen complejos en la interconexión de los proveedores de Internet (ISP).

11. Servicios de redes virtuales privadas (*VPN* sitio a sitio, Redes *VPN* y enrutamiento de *VPN*). Esto supone un camino con utilización de sistemas de cifrado. Sobre una red y un enlace de comunicaciones se transporta información cifrada y en texto claro. Cuando no se establecen los dominios de alcance del cifrado, en las estaciones que intervienen se logra crear bucles de conversaciones de red entre el túnel cifrado y la red normal.

12. Sistemas militares de transcripción a texto de conversaciones o filtros por palabras clave. Sistemas intermedios que realizan funciones de auditoría que aumentan el uso de los procesadores de los equipos de red afectando el desempeño.

13. Agregación de ancho de banda (*Link Aggregation Control Protocol LACP*). Es una forma de combinar de manera lógica varios canales físicos para aumentar la velocidad de transmisión. Pero esto realiza un balanceo de uso de las sesiones asociada a algoritmos basados en las direcciones físicas de red (*mac-address*) para la entrega o conexión con el destino. Si la estación que recibe no entiende de manera correcta el protocolo entonces una considerable cantidad de estaciones no pueden entregar la información. Sus problemas son la congestión e implementación particular de los fabricantes de los componentes electrónicos.

14. Utilización de balanceadores de carga y la persistencia de las transacciones. Los balanceadores nos permiten aumentar disponibilidad en los servicios. Emplean grupos de

servidores en paralelo para atender un servicio. Se utilizan servicios de difusión de red (*multicast*) para ofrecer el servicio mediante direccionamiento IP virtual. La falla de uno de los servidores implica la asignación de otro para que lo reemplace. En ocasiones las sesiones de conexión se pierden o no es posible retomar una conexión porque algunos paquetes o variables de sesión se pierden. También se utilizan métodos de persistencia de sesión (mediante marcadores llamados *cookies*).

15. Cambios de topología de red tanto local, como el proveedor de servicio (convergencia de *Spanning Tree Protocol STP*, *Multi Protocol Label Switching MPLS*). Los algoritmos de control de la red física determinan la existencia de un mejor camino para la entrega de paquetes, Durante el proceso de transición a la nueva forma es posible el corte de sesiones o pérdida de algunos paquetes.

16. Traducción de direcciones IP y su efecto en la sobrecarga de los puertos y la concurrencia de conexiones y uso de memoria de los equipos de red de capa 3 encaminadores (*routers*) y filtros (*Firewalls*).

Todos estos problemas deben ser considerados en un modelo de auto-configuración de red. La recolección de información necesita determinar las variables adecuadas y los lugares de la red donde se encuentra la información que define el problema.

1.5 Necesidad de la auto-configuración de redes

Se puede observar según los trabajos realizados por muchos académicos y empresas fabricantes de tecnología en telecomunicaciones que es necesario implementar redes autónomas que permitan dar soluciones a los problemas futuros de escalabilidad en la red como:

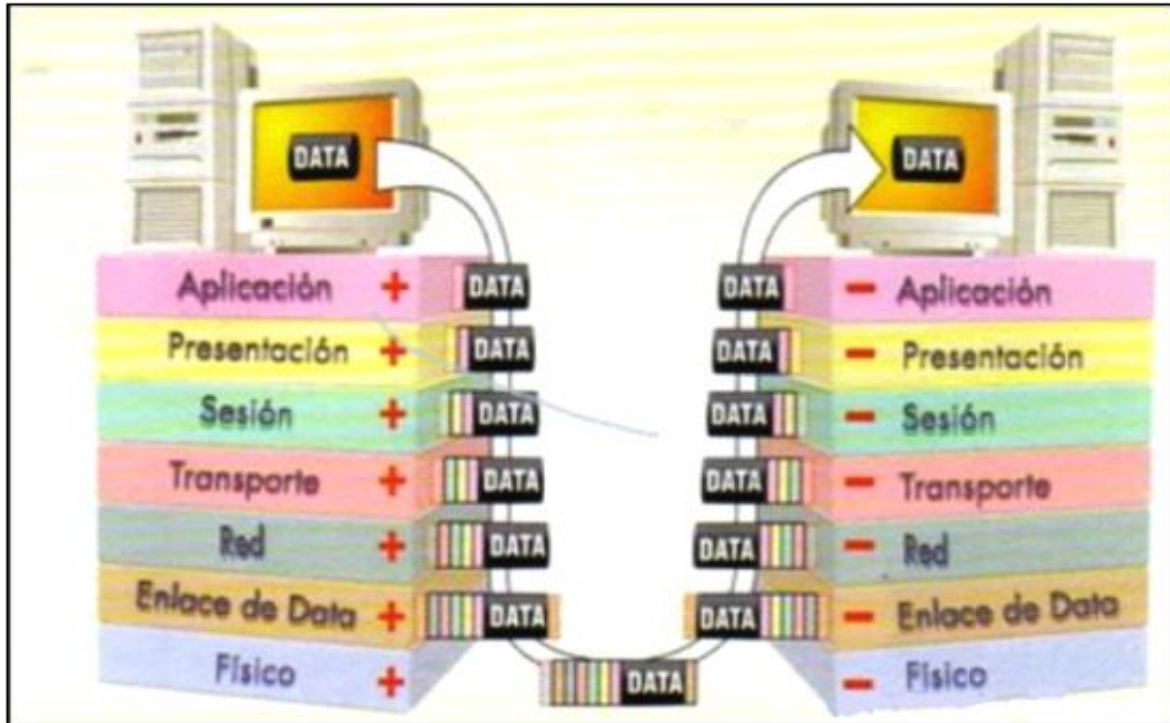
- Dificultad en el análisis de los datos en línea de la red y el tráfico por su volumen de la información
- Comportamientos de red que no han sido estudiados o no resultan obvios de inferir con consultas tradicionales.
- Dificultad de configurar todos los equipos de red para lograr un cambio con un objetivo específico.

- Imposibilidad de determinar cuáles equipos activos o cuales servicios se deben modificar para mejorar la calidad de un servicio.
- El tiempo necesario para diagnosticar un fenómeno de red es muy alto (semanas) comparado con el tiempo de respuesta de solución que requiere el usuario (horas) y el tiempo de reconfiguración de la red y sus servicios (minutos).
- La gran cantidad de servicios que se agrupan para ofrecer un producto de red. El aporte de cada uno, la evaluación de sus estados, la falta de intercambio de información de estado entre procesos, la falta de un ente inteligente que coordine y la influencia de cada sub-sistema aumenta la complejidad de diagnóstico, dificultad de entendimiento del problema por parte de los administradores de red y, en consecuencia, dificultad de aplicar un cambio que mejore la calidad del servicio.

1.6 El modelo OSI, la subcapa MAC y el sistema multi-agente

El estándar de siete capas OSI es un modelo conceptual que caracteriza y estandariza la comunicación de las redes de datos y telecomunicaciones para lograr interoperabilidad. Después de los años 60-70, en donde nacieron muchos protocolos de comunicaciones propietarios, incompatibles y heterogéneos patrocinados por diferentes fabricantes surgió la necesidad de realizar una estandarización en los modelos, hacer un diseño, fijar criterios de normalización. Después de varios años de trabajo y esfuerzo de varios organismos internacionales como la ISO, ITU-T, IEEE (“IEEE,” 2017) se logró publicar y acoger un documento marco denominado “Modelo de referencia OSI (Association, 1998) (en inglés *Open Systems Interconnection Reference Model*)” mediante los códigos ISO/IEC 7498-1:1994 ó ITU-T X.200.

Ilustración 5. Modelo de capas OSI.



Rojas, J. (2017) Ilustración de MODELOS OSI Y TCP/IP. Recuperado de <http://www.jesusrojas.es/informatica>.

La arquitectura del modelo se dividió en capas la comunicación de extremo a extremo para dividir de manera diferenciada y definida las funciones. El objetivo del modelo es normalizar la comunicación entre capas, dar simplicidad para que sea manejable, y minimización de flujo entre capas. Cada capa realiza un conjunto de funciones resolviendo un problema diferente. Cada capa sustenta a la inferior y proporciona servicios a la superior y los cambios en una capa no implican cambios en las otras capas.

El sistema multi-agente adopta el modelo OSI para utilizar y tomar información de las funciones de varias capas o subcapas del modelo OSI específicamente la MAC. El desarrollo de software en este trabajo, utiliza en mismo principio de utilización ordenada de las funciones capas 2 a 3 del modelo OSI (*Data Link Layer* y *Network Layer*). Se utilizan varios agentes y múltiples comportamientos para tomar información de manera estructurada.

La capa de enlace de datos (término en inglés *Layer 2*) está encargada del direccionamiento físico, del acceso al medio, la detección de errores, control de flujo y la distribución ordenada de tramas (término en inglés *frames*). La capa se encuentra dividida

como Libpcap que permite contar con una interfaz de alto nivel para realizar la captura de paquetes (“TCPDump,” 2017).

Del lado de los equipos de redes, se pueden encontrar un conjunto de protocolos para la entrega de tramas (*frames*). La asociación mundial de ingenieros IEEE trabaja para diseñar y compartir los avances en tecnología. Se puede referenciar los grupos de trabajo denominado 802 que han definido los estándares de la industria para redes Ethernet. En especial el grupo 802.3 trabaja todo el estándar Ethernet. Incluye el desarrollo del protocolo para manejo de las redes virtuales (VLANS 802.1Q-2014), el transporte de redes virtuales por un enlace de comunicaciones mediante la marcación de las tramas (puertos troncales), protocolo de árbol expandido (STP y MST), sistemas de control de conmutación (BPDU), la capacidad de entrega de tramas entre equipos de diferente tecnología física (*bridging*). El estándar 802 debe considerarse como un conjunto de estándares relacionado con la capa MAC y la conmutación de tramas (*bridging*) (*802.1Q-2014 IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks*, n.d.).

Es importante el análisis de redes de interés mediante el cual se unen varios enlaces paralelos de comunicación entre dos equipos adyacentes como un solo ente lógico para aumentar velocidad y tolerancia a fallos conocido como 802.1ad. Además se puede utilizar para balancear tráfico según las direcciones físicas (*mac-adress*) de origen o destino (*IEEE Std 802.1ad-2005 (Amendment to IEEE Std 802.1Q-2005): IEEE Standard for Local and Metropolitan Area Networks---Virtual Bridged Local Area Networks---Amendment 4: Provider Bridges*, 2006).

Las redes Ethernet no soportan bucles dado que las tramas entran en un proceso de difusión y crea un proceso de retransmisión infinita. Las tramas Ethernet no poseen en su cabecera un campo para contar los saltos (*Time to Live o hub count*) como lo tiene IP. La importancia de STP (*spanning-tree*) como algoritmo es mantener el transporte de tramas al interior cada VLAN Ethernet libre de bucles mediante un árbol expansivo que rompe los bucles de manera automática (*802.1D-2004 IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Bridges*, n.d.).

De igual forma, muy útil conocer cuales equipos son vecinos de otros. Nos permite tomar una idea de la topología de una red en su región o partición. Con LLDP se tiene la capacidad de descubrimiento de vecinos como protocolo abierto mediante el estándar

(802.1AB-2009 IEEE Standard for Local and Metropolitan Area Networks -- Station and Media Access Control Connectivity Discovery, n.d.).

La dirección física *mac-address* fue definida en el estándar ISO/IEC 10039. Se definió como un conjunto de 48 bits (6 bytes). Todos los fabricantes de tarjetas de red deben registrar los rangos de *mac-address* fijados en la electrónica de sus componentes mediante los 24 bits más representativos. El grupo de la IEEE denominado “*The Standard Group MAC*” administra la asignación y registro de las *mac-address* de todos los fabricantes. Esto nos permite identificar a un fabricante por la dirección *mac-address* (“ISO/IEC 10039:1991,” 1991).

En el funcionamiento normal de la pila de protocolo, las direcciones *mac-address* se relacionan con una dirección IP. Esta unión permanece según el tiempo de préstamo de la IP que le asignó (*lease time*) el sistema dinámico de configuración de la red (*DHCP*). Dado que las redes son dinámicas esta relación cambiará. La dirección física *mac-address* define una interfaz de red de una máquina (no a la máquina en sí). Esto tiene consecuencias para los análisis de red donde será necesario identificar de manera única una máquina y los instantes en los cuales transmitió información con una tripleta *mac-address /IP/Fecha*.

La implementación del software desarrollado en este trabajo utiliza de manera intensiva la información del protocolo ARP que nos actualiza la relación *mac-address -IP*, además de observar las direcciones de envío de información a individuos, grupos conocidas como difusión y multidifusión (*unicast y multicast*) (Wallace, 2015).

Para que una red alcance un estado de estabilidad existen dos algoritmos que debe estar en un punto de convergencia. Se le dice convergencia cuando el algoritmo no debe volver a ejecutar para resolver alguna configuración o problema. Los algoritmos son:

- El árbol expansivo *Spanning Tree* (STP) en la capa 2 de enlace.
- Enrutamiento (*routing*) (Academy, 2014) en la capa 3 de red.

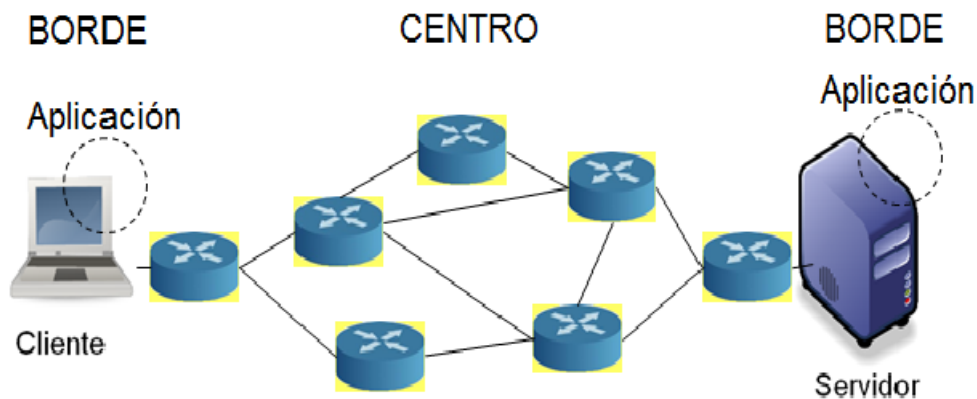
En general, todos los conceptos aquí mencionados permiten el correcto entendimiento del comportamiento de una red y la toma de información de manera correcta.

Modelo borde-centro de una red.

Para lograr la transmisión de información entre dos puntos A y B se deben cruzar las siete capas del modelo OSI. A las estaciones de cada extremo de la comunicación se les

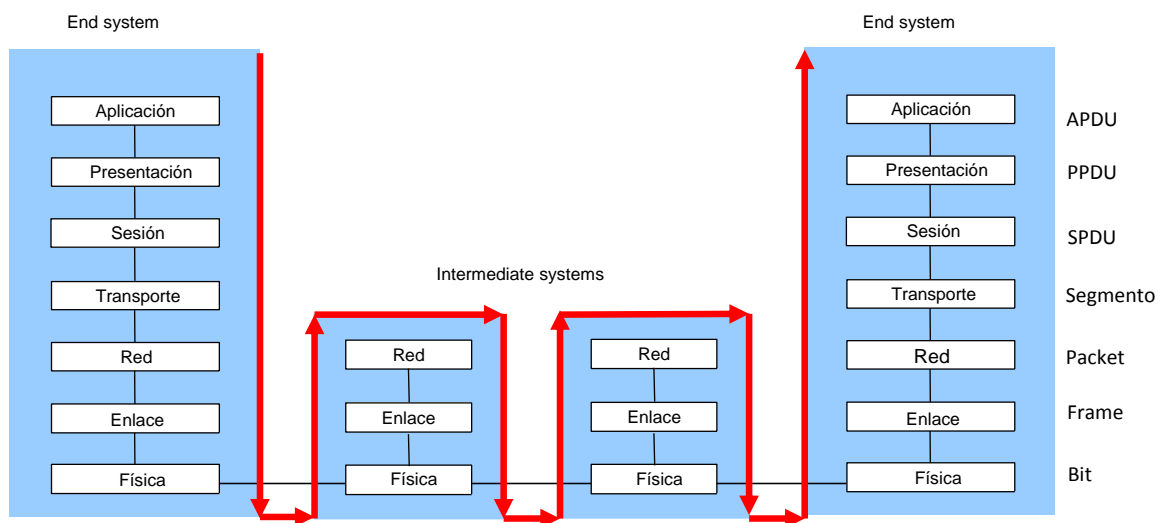
denomina “borde”. Pero es importante anotar que estas siete capas existen en las estaciones. En medio de las estaciones existe un grupo de equipos de redes que tiene la función de pasar la información hasta llegar al otro extremo.

Ilustración 7. Modelo borde centro en topología.



Estos equipos en el medio se les denominan el “centro de la red. El centro de la red, se diferencia por implementar las tres primeras capas del modelo OSI (físico, enlace y red) (Kurose James F.; Ross, 2013).

Ilustración 8. Modelo capas borde-centro en modelo OSI.



Toda la información anterior se requiere para alimentar la base de datos (modelo de datos) para realizar los procesos de toma de decisión de auto-configuración. Los agentes instalados de manera distribuida recogerán, verificarán y contrastarán la información. Esto permitirá verificar fallas, errores o problemas de configuración.

En la arquitectura adoptada en este trabajo, el sistema multi-agente requiere utilización de información de todas las capas y subcapas. En particular de la subcapa MAC de donde se extrae las direcciones físicas (*mac address table*) y las subredes físicas VLAN (*vlan-id*) a las que pertenecen las estaciones.

Modelo OSI	Descripción	Modelo Multi-agente (comportamientos)
Capa 7	Aplicación	Persistencia de los datos en mysql. Capa de presentación de resultados con un portal Web. Sistema de registro central de los agentes en la consola de JADE.
capa 6	Presentación	Análisis sintáctico de los comandos <i>arp</i> , <i>netstat</i> con las particularidades por versiones de sistemas operativos y sus idiomas.
Capa 5	Sesión	
Capa 4	Transporte	Inspeccionar o tomar información de los puertos <i>udp</i> y <i>tcp</i> .
Capa 3	Red	Recolectar direcciones IP y relacionarlas con las <i>mac-address</i> .
Capa 2	Enlace	Recolectar <i>mac-address</i> de estaciones, <i>switches</i> , <i>routers</i> y su correspondencia con las redes virtuales.
Capa 1	Física	No requerido.

Tabla 1. Uso de abstracción OSI para sistema MAS.

Las funciones de cada capa son emparejadas con un comportamiento de un agente para tres tipos de objetos de red: switches/routers, estaciones como PCs y servidores.

De esta forma, cuando se requiere la tabla de direcciones físicas (*mac-address*) de un conmutador de paquetes (*switch*) entonces se utilizará un comportamiento (*behavior*) del agente "*switch*". De esta forma, se puede realizar un desarrollo de software en donde cada clase está orientada a una función de la capa OSI.

Se utilizan funciones de la capa 3 y 7 (*IP, TCP, HTTP, SSH*) y el sistema de mensajería de agentes para que el comportamiento se comuniquen y se realice el acceso a las bases de datos para persistir la información, envío de tramas de comunicación, registro de agentes en el contenedor principal de JADE para el modelo social.

Los comportamientos que dan paso a la auto-configuración de red vienen de un proceso de cálculo y computación, comunicado desde la capa 7. Descenderá por las capas del modelo OSI y después mediante agentes que entienden la capa 2 realizarán los cambios en la subcapa MAC, entre otros como el cambio de red virtual (*VLAN*) o un cambio de direcciones IP en las estaciones.

1.7 Conceptualización del sistema multi-agente

Un sistema multi-agente es un conjunto de elementos de computación que interactúan entre de una forma social. Los agentes deben poseer dos capacidades básicas; ser capaces de realizar al menos una acción de manera autónoma y ser desarrollar la habilidad de comunicarse con los otros agentes de una manera social como lo hacen los humanos: cooperación, coordinación y negociación (Wooldridge, 2009).

1.7.1 Definición de Agente

Un agente es un sistema de computación que se encuentra ubicado en un ambiente y posee la capacidad de actuar de manera autónoma sobre su ambiente para alcanzar los objetivos para los que fue diseñado (Wooldridge, 2009). Es importante anotar que un agente puede decidir cooperar para lograr beneficio mutuo o competir por un recurso (Fabio Luigi Bellifemine Giovanni Caire, 2007). Esto significa que el agente tiene control sobre sus acciones.

1.7.2 Arquitecturas de agentes

Se define arquitectura como los mecanismos que se utilizan para manejar la autonomía de los agentes. Existe un alto rango de arquitecturas que están en un rango entre el agente reactivo y deliberativo. El agente reactivo opera sobre el ambiente en una forma simple mediante un estímulo y respuesta. Los agentes deliberativos están basados en patrones de lógica mediante reacción y deliberación. El modelo más extremo se conoce como BDI

o modelo de creencias, deseos e intenciones (sigla en inglés *Belief, Desire, Intention*) (Rao & Georgeff, 1995).

Se conocen cuatro grandes clases:

- Basados en lógica.
- Reactivos.
- Deliberativos BDI. Un caso especial es el Razonamiento procedimental PRS (*Procedural Reasoning System*).
- Por capas. Combinación flexible de todas las anteriores.

1.7.3 Características de los agentes

Las tendencias actuales de la masificación de la tecnología permiten que todos los usuarios cuenten con mejores y menos costosos equipos electrónicos, dotados de muchos sensores y se observa la entrada de la era de la ubicuidad (Giordano et al., 2016). Además, con mayor poder de cómputo a costos competitivos y contando con la interconexión de los sistemas se está logrando el desarrollo del área de los sistemas distribuidos e inteligentes. En este sentido, la inteligencia implica darle mayor control a las máquinas.

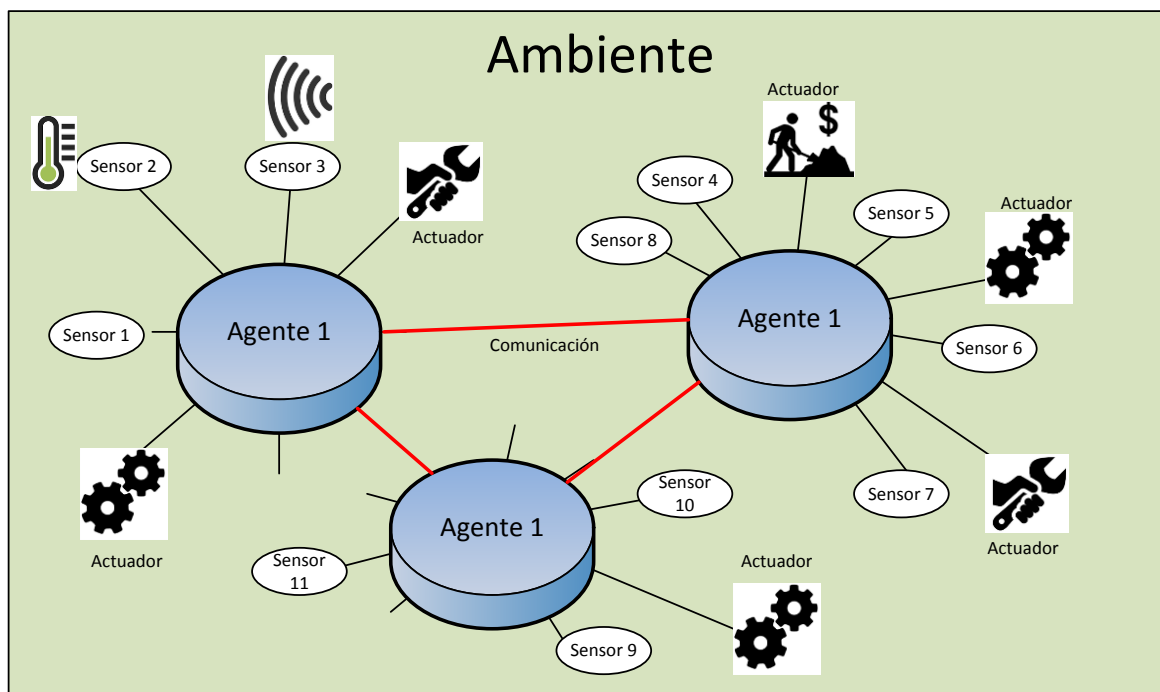


Ilustración 9. Ambiente multi-agente.

De acuerdo con lo anterior, se definen:

Autonomía - Un agente puede actuar por su cuenta sin la intervención humana directa, controla sus propias acciones y estado interno. Un agente es capaz de percibir y responder a un entorno cambiante sin demora o en un plazo corto de tiempo según la información que recibe de su interacción social.

Proactivo (iniciativa) - Un agente es orientado hacia los objetivos y puede lograr los objetivos sin preguntar al usuario u otro agente. También es capaz de adaptarse a los cambios en el medio ambiente.

Habilidad Social - Un agente es capaz de comunicarse con los seres humanos u otros agentes usando un lenguaje de comunicación entre agentes.

- Cooperación. Comportamientos emergentes en las sociedades.
- Coordinación. Lenguajes para comunicarse.
- Negociación. Creencias y aspiraciones, Reconocer sus creencias, objetivos, metas, acciones, resolver los conflictos.

Aplicación de las características en las redes de computación.

Para el caso de las redes de computadores el “agente” nos aporta en alcanzar las soluciones:

- Cuando un elemento de red está interconectado informa a todos sus vecinos. Informa mediante una función mide el desempeño de la red y recursos en el ambiente.
- Cuando un elemento se encuentra aislado almacena información y estados de su partición. Intentar con su proceso social encontrar a sus vecinos y definir las fronteras de la partición. Si la partición se mantiene aislada colabora para elegir a un nuevo controlador central de registro de agentes.
- Buscar forma alterna de comunicarse al coordinador si encuentra interfaces inalámbricas.

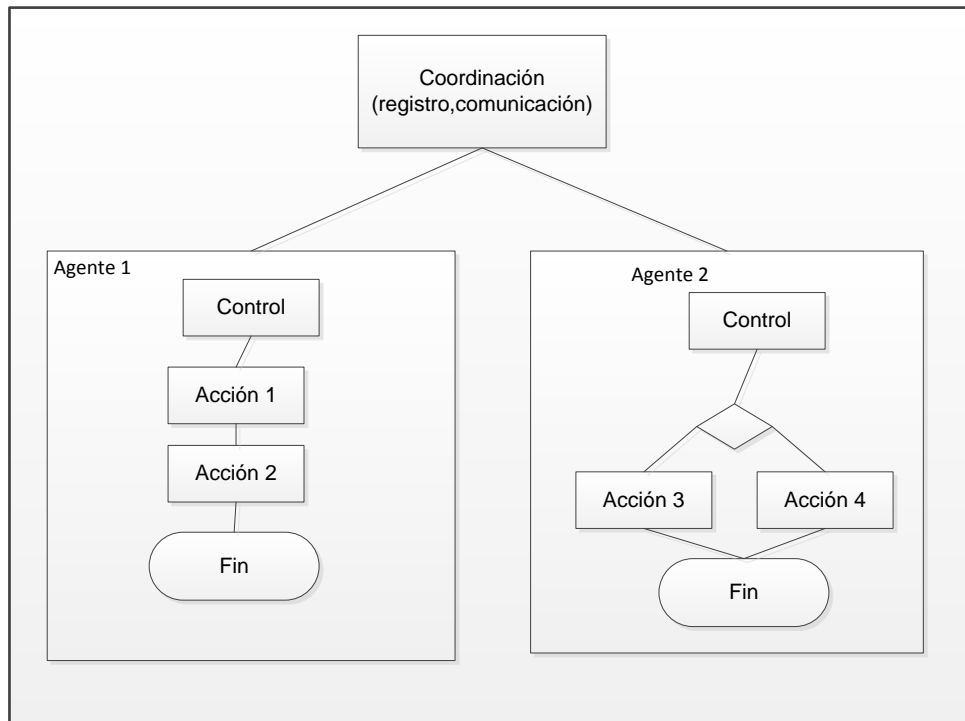
- Realiza diagnóstico del problema. Puede realizar procedimientos para intentar encontrar el punto de falla que llevo a la partición de la red.

Interacciones y Social

- Comunicar alteraciones en el ambiente para quien las necesite.
- Si encuentra método alternativo de comunicación informa. Puede cambiar su comportamiento para convertirse en un nodo de interconexión y resolver una partición de red.
- Establece interacciones con vecinos para establecer el alcance de la falla. (Establece una región). De cada lado de la falla.
- Comunica a los vecinos de región los estados que mejoraron.

Es importante anotar, que los sistemas multi-agente coordinan sus actividades como una fortaleza de los sistemas distribuidos (Varela, 2013). La coordinación no supone dependencia. Al interior de cada agente se encuentra el desarrollo de las acciones y tareas. Estas se realizan bajo el control normal del programa del sistema operativo. La coordinación esta fuera de las fronteras de una máquina (Dressler, 2008).

Ilustración 10. Coordinación y Control.



1.7.4 Capacidades del sistema multi-agente.

Los sistemas multi-agente ofrecen una forma natural de ver y caracterizar los sistemas de inteligencia artificial. Las interacciones se encuentran profundamente unidas y acopladas y no pueden funcionar de manera aislada. Existen varias capacidades deseables para su implementación (Weiss, 1999).

Los agentes pueden trabajar de manera asincrónica y en paralelo mejorando su velocidad. Se espera que la coordinación no afecte el desempeño.

Robustez y confiabilidad. La falla de uno o varios agentes no produce una falla total del sistema dado que otros tomarán su lugar para relevarlos.

Escalabilidad y flexibilidad. Se pueden aumentar la cantidad de agentes sin comprometer la operación.

Costos. Posee una relación costo-beneficio mejor que los sistemas centralizados.

Despliegue y reutilización. Cada uno de los agentes puede ser fabricado por especialistas diferentes. Las pruebas y el mantenimiento se realizaran de manera más fácil. De igual forma se pueden reutilizar agentes en diferentes aplicaciones.

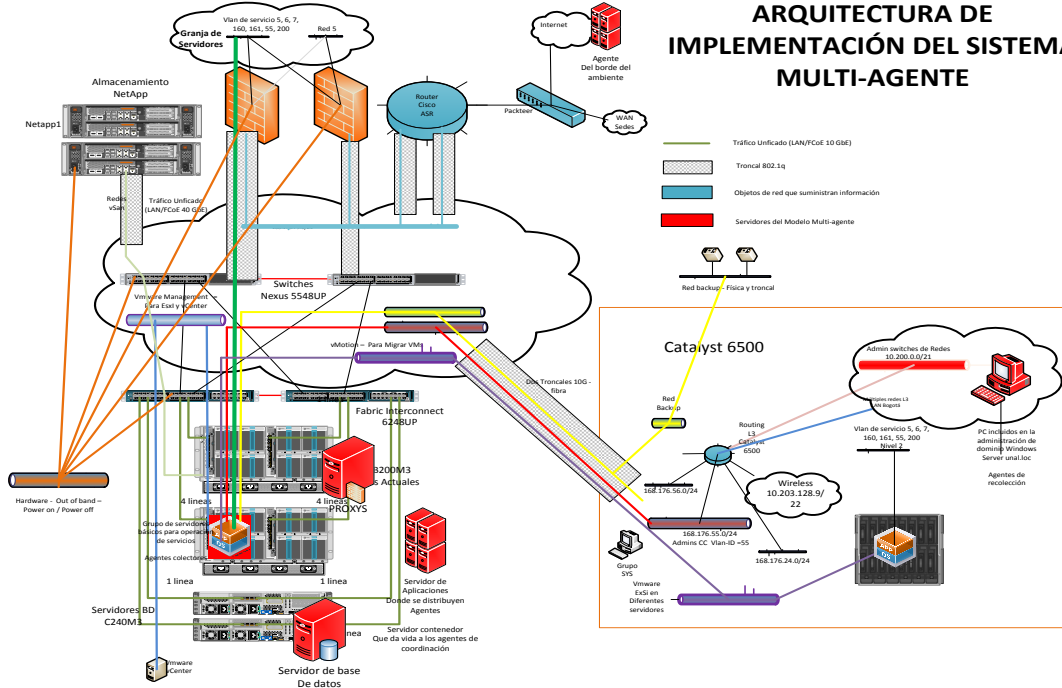
1.7.5 Ubicación de los agentes

La ubicación de los agentes está relacionada de manera directa al lugar donde se encuentra la información valiosa o un fenómeno que se desea medir con alguna sonda. En general, deben encontrarse en contacto con el ambiente y el universo de variables que pueda sentir. De igual forma, para realizar alteraciones en el ambiente se requiere contar con agentes con la capacidad de realizar actuaciones.

- “Los agentes” viven donde pueda tomar información del ambiente “**sense**”
- “Agente debe ser útil”. Capacidad de actuar sobre el ambiente.
- “Agente” requiere recursos de computación para poder existir.

Para manejar los equipos de red. Los agentes se deben colocar en una estación que este adyacente al dispositivo de red en una conexión de alta velocidad. Se espera que no existan filtros de seguridad que evite actuar sobre el elemento. La proximidad debe ser física y lógica. Si es posible en el mismo bastidor y circuito eléctrico donde se encuentre el equipo de red. A nivel lógico en la misma subred física y con direccionamiento IP similar. Esta ubicación se debe a que generalmente los dispositivos de red poseen sistemas operativos cerrados o propietarios que no permiten adición de software interno. Cada fabricante de elementos de red como equipos de conmutación, enrutadores, filtrado de paquetes diseñan el hardware para optimizar las funciones de su sistema operativo.

Ilustración 11. Implementación del sistema multi-agente.



Se presenta aquí una topología de una red empresarial donde están los componentes de red centrales, los filtros de seguridad y las conexiones para los centros de cómputo. En color rojo se observa la ubicación de los agentes dentro de la computación tradicional pero de manera cercana a los equipos de red que pretenden influenciar para modificar el ambiente.

Para las estaciones. Las estaciones son equipos de cómputo con sistemas operativos tradicionales como Microsoft Windows, Android, Linux o Unix. Los agentes utilizarán recursos de procesador, memoria y disco de su huésped (*host*). Para observar y modificar el ambiente requieren de todas las interfaces que pueda tener el hardware base con conectividad a alguna red. De igual forma, se ejecuta el agente principalmente en las estaciones que estén en el centro de varias conexiones de red, que contenga sistemas de virtualización o acceso a redes privadas. Por ser elementos de paso o interconexión la cantidad de información es valiosa. El agente además puede utilizar su capacidad de transporte de información para comunicar agentes que puedan estar aislados.

1.8 Jade como herramienta de sistema multi-agente

La plataforma para el desarrollo de sistemas multi-agente JADE fue desarrollada por el grupo TILAB en el laboratorio *Research Labs of Telecom Italia* (Fabio Luigi Bellifemine Giovanni Caire, 2007). El software se consideró pertinente para el trabajo y para el desarrollo del sistema multi-agente por las siguientes razones:

- Es software libre y se desarrolla con un equipo de *Open Source GNU*.
- Puede crecer hasta 100.000 agentes. (Jorg P. Muller Wolf Ketter, 2015).
- No es comercial y no requiere la adquisición de una licencia costosa.
- Se encontró documentación suficiente para el desarrollo del software.
- Desarrollado en Java que es un lenguaje líder, arquitectura neutra y puede estar presente en todas las estaciones con diferentes sistemas operativos.
- Posee ambiente de desarrollo para sistemas Android.
- Cumple con el estándar internacional FIPA para interoperabilidad entre agentes de diferente fabricante (*Foundation for Intelligent Physical Agents*).
- Se mantiene activo con lanzamiento de nuevas versiones.
- Tiene varios grupos de trabajo cooperativo que contribuyen con nuevas facilidades de interfaces.
- Se lanzó la versión de JADE 4.4.0 el 23/12/2015.
- Permite agregar las bibliotecas de Java existentes el mundo *Open Source*. Los recursos de bibliotecas en Java son numerosas.
- Para este trabajo, no se requería utilizar procesadores de las GPU como FLAME. Estas plataformas usan el lenguaje C para realizar tareas de paralelismo sobre los procesadores CUDA.

Mediante la herramienta de sistemas multi-agente JADE se obtiene las características que se requieren para la implementación.

Utilización de Java.

Java es un lenguaje simple, orientado a objetos, tipado sintácticamente, distribuido, interpretado, robusto, seguro, de arquitectura neutral, multi-hilo, con recolector de basura

(*Garbage Collector*), portable, de alto rendimiento, sobre todo con la aparición de hardware especializado, dinámico, para desarrollar aplicaciones empresariales con un alta disponibilidad en ambientes. Utilizados por gran volumen de usuarios, es *open source*, a diferencia de aplicaciones desarrolladas en tecnologías como PHP que tienen una carencia en la arquitectura, manejo de hilos y que presentan graves problemas de disponibilidad. La versión de Java *jdk* versión 1.8 permite desarrollar al igual que .NET con lambda expresión pero con una arquitectura bien definida en Java.

Autonomía. En redes de datos, existe el reto de mantener conectividad constante. Cuando un elemento, grupo de elementos, o varios grupos se aíslan del resto deben realizar actividades autónomas. Un comportamiento de diagnóstico, establecer los vecinos que si alcanza conectividad, calcular un grupo de estado, intentar otros métodos de comunicación es una de las bases de MAS (autonomía y acciones independientes).

Sociabilidad. Se requiere intercambio de mensajes para comunicar estados de la red y comportamiento de estaciones.

Reactividad. Permite tomar datos del ambiente, anotar la fecha ocurrencia y duración, repetición de ocurrencia. Para realizar la reconexión a la red, se puede activar pruebas de enlace y eficiencia de transmisión, buscar otros caminos de conectividad y enviar datos para emitir un diagnóstico y mediante sociabilidad comunicar la solución.

Proactividad. Permite cambiar los parámetros de algún equipo de red cercano o enviar tráfico a otro agente intermediario para comunicar mensajes de estado.

Creencias "Belief". Se plantea comparar las configuraciones de los equipos de redes que intervienen entre el origen y destino de una conversación de red, mediante las sucesivas versiones de configuración que se aplican y que reflejan las decisiones humanas comparando contra los estados que perciben los agentes distribuidos por toda la red LAN de una compañía. Se cree que una configuración no puede ser mejorada si funciona. Pero no se conoce el punto o el indicador para establecer si mejoró o retrocedió. Pueden existir puntos más óptimos que solo serán encontrados por procesos de decisión.

Deseo "Desire". Las redes de datos tanto fijas como inalámbricas poseen una topología definida por los diseñadores de la red. La red transporta muchos tipos de tráfico, muchas combinaciones de origen destino, según los deseos y necesidades de los usuarios. Se desea que el tráfico que cumpla con perfiles similares permite reagrupar estaciones en una misma subred o que se configure una VLAN especial para que estas transacciones fluyan de manera eficiente sin necesidad de intervención humana.

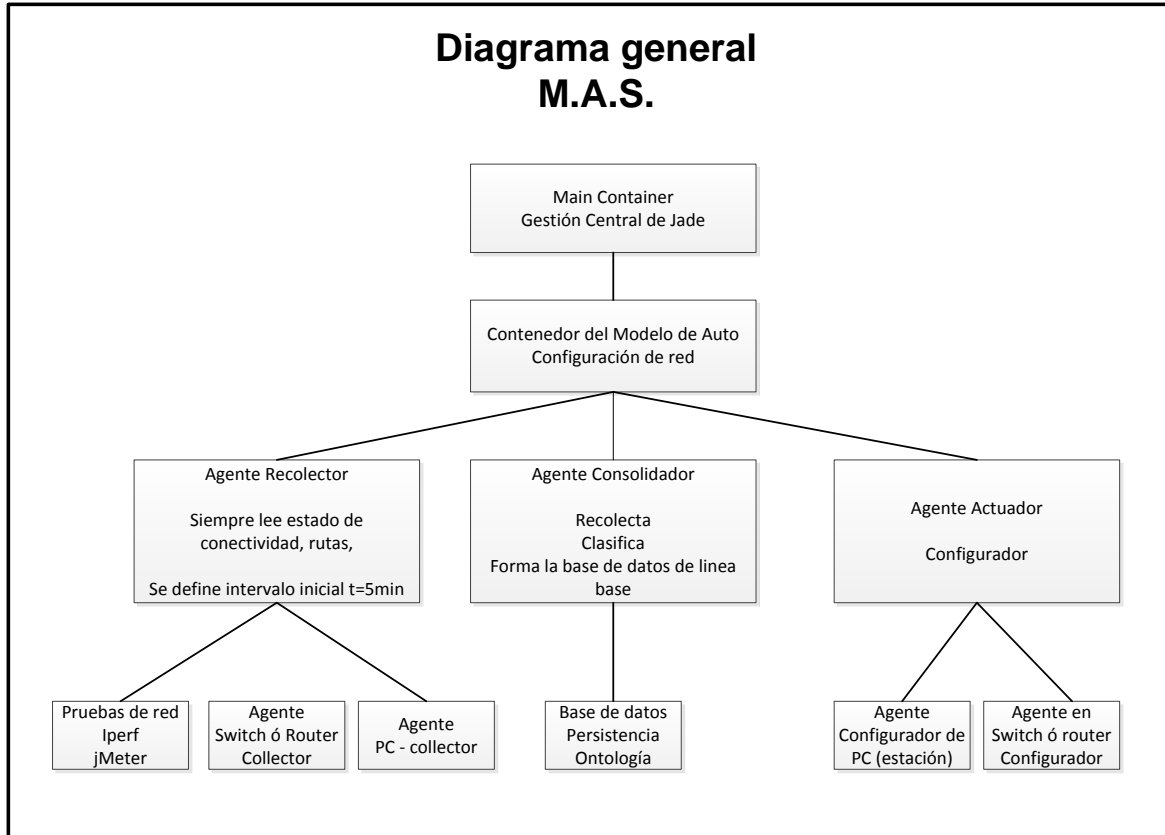
1.9 Modelo del agente recolector, actuador y de consolidación

Se presenta el modelo general para la implementación del sistema multi-agente. Se utiliza una primera capa que nos proporciona Jade para la administración, registro y comunicación de los agentes.

La metodología (Nikraz1a, Magid; Caireb, Giovanni; Bahria, 2006) sugerida por los desarrolladores nos sugiere organizar todas las aplicaciones que se diseñen en cajas (termino en inglés *containers*) con las respectivas instancias de agentes que se desplieguen. Para el modelo desarrollado en este trabajo se simplificaron los casos de uso en tres tipos: Agente Recolector, de Consolidación y Actuador.

El agente recolector permite conseguir la información de un objeto directo de la fuente (PC, *router*, *switch*) y nos ayuda a filtrar y procesar la información. El agente de consolidación permite guardar información para generar histogramas, series de tiempo o líneas base del comportamiento del tráfico o una estación. El actuador permite realizar los cambios en el ambiente. Para que los cambios no sean cambios arbitrarios, se emplea un modelo de optimización para seleccionar los objetivos apropiados. En este punto, se seleccionan los agentes actuadores que deben aplicar el cambio deseado.

Ilustración 12. Diagrama general para el software.



2. Proceso de decisión en el modelo de agentes y redes de datos

2.1 Que es una decisión.

El proceso de auto-configuración de red debe cumplir con un objetivo y una consecuente decisión de cambiar el comportamiento de uno o varios elementos de una red. El cumplimiento del objetivo se traduce en un proceso de optimización. Tomar la mejor decisión entre múltiples soluciones.

Otro enfoque para mejorar el desempeño o la calidad de un servicio se logra como el acuerdo entre cliente y servicio sobre el manejo de los recursos del sistema que se debe traducir en reglas aplicables en la red y en las especificaciones del lenguaje de programación utilizado (Larus, Rajamani, & Rehof, 2004). De igual forma, se debe formar un modelo matemático que explique el fenómeno, realice la optimización para elegir el tipo de configuración en la red.

2.2 La optimización

Se define problema de optimización mediante variables y funciones que se involucran en el tráfico de redes.

Definiciones:

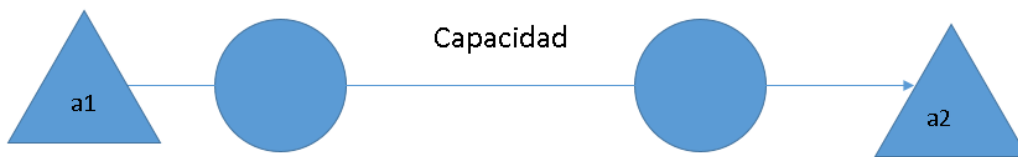
T : tiempo de registro de la muestra. Se realizan muestreos del comportamiento de una variable como ancho de banda en intervalos regulares. El tiempo T nos permite controlar la secuencia en el tiempo.

D : delta de tiempo entre muestreo. Se estableció $D=5$ minutos. El intervalo de muestreo permite contar con el nivel de detalle para capturar el comportamiento. No se debe utilizar un intervalo demasiado pequeño dado que la cantidad de información que se guarda para

realizar las series de tiempo puede llegar a ser gigante (un intervalo de muestreo de 100 milisegundos puede generar un archivo de información de 10 Giga en 15 minutos).

C: Capacidad. Es la cantidad máxima de bytes que se pueden transportar por esta conexión en un segundo. La unidad de tiempo en redes habitualmente son los segundos.

Ilustración 13. Definición de capacidad.



BW: Ancho de banda: Cantidad máxima de información que se puede transmitir entre a_1 y b_1 en un instante t .

Analogía entre teoría de conjuntos, funciones y las redes de computación.

Para facilidad de modelar la ecuación de optimización, se utilizan conceptos básicos de las matemáticas como son la teoría de conjuntos. De igual forma, se utilizó una función que represente la cantidad de información transportada entre dos elementos.

Asimilaremos las subredes virtuales VLAN (grupos de equipos) como conjuntos con alguna característica en común. Las estaciones de la red son los elementos de un dominio. Cuando un equipo de computación desea usar la red debe ser conectado. Esta acción se puede modelar mediante una relación de pertenencia con el conjunto.

Se definen las relaciones entre conjuntos como los enlaces de comunicación, la capacidad máxima de transmisión y el ancho de banda.

Sea a_1 , a_2 , b_1 , b_2 : Elementos (estaciones de computación conectadas a una red).

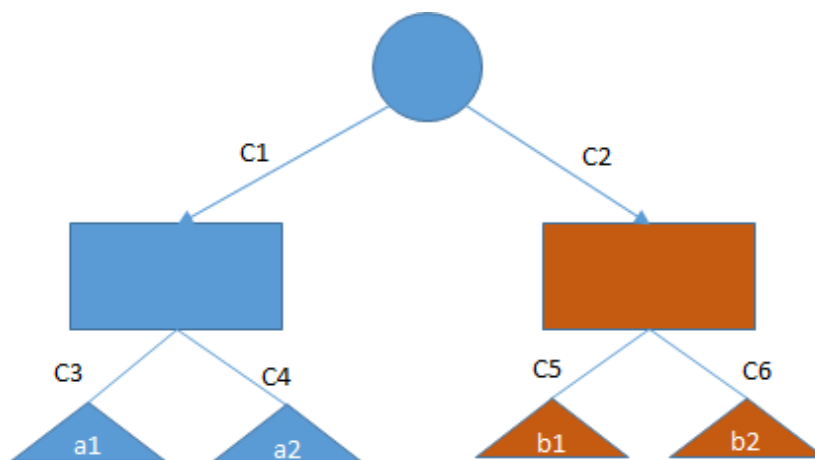
VLAN: Conjunto de elementos similares. En color azul pueden estar representados las máquinas del área de contabilidad y en color rojo se observan las estaciones del área de ventas. Los conjuntos corresponden a grupos de estaciones con tráfico en común.

Los elementos a1 y a2 pertenecen al conjunto de color azul (VLAN contabilidad).

Los elementos b1 y b2 pertenecen al conjunto de color rojo (VLAN ventas).

Sea la función $G(x,y)$ que mide la transferencia de información entre un par de estaciones sin importar su ubicación. En la gráfica se presenta una red en forma de árbol donde solo existe un camino para transmitir información entre cualquier x , y estaciones.

Ilustración 14. Función de transferencia entre estaciones.



$t=1, \dots, n$: tiempo en el cual se toma la información

$C1, C2, C3, C4, C5, C6$: capacidades de los enlaces.

En la redes de computación la suma de las capacidades $C3 + C4 = K * C1$. En donde K es un factor multiplicador 10, 100 o 1000. Esto quiere decir que cuando las estaciones a1 y a2 desean transmitir al máximo de su capacidad hacia la VLAN de color rojo se excederá la máxima capacidad $C1$.

Ahora bien, la capacidad de transmisión entre dos estaciones que están en VLANs diferentes se define como la mínima capacidad del camino de interconexión (suponiendo que las demás estaciones no transmiten en ese momento).

Se define la capacidad de transmisión máxima entre las estaciones a1 y b1 como el mínimo de las C_i en el camino. Así, $\min \{C3, C1, C2, C5\}$

Se define la capacidad máxima $CM(x, y) = \min \{C_i \text{ en el camino entre } x, y \}$

Los equipos de conmutación de redes (representado por el rectángulo y círculo rojo y azul) poseen límites en su capacidad de entrega de datos. Estos límites son conocidos como velocidad del bus (*backplane*).

Se define la capacidad de bus $CB = (1 / m1) * (C3 + C4)$. Donde $1/m1$ representa el porcentaje total de tráfico agregado que puede transportar. Normalmente este valor oscila entre 20 a 50%.

$M1, M2$: Representa los límites de los buses cada uno de los equipos de red.

Formalización matemática:

En redes de computación, se prefiere que la mayor cantidad de información transferida se quede dentro de su VLAN para aumentar las velocidades de transmisión. Esto significa que para el caso de la VLAN azul

$G(a_i, b_i)$: Función que mide la transferencia de información entre un par de estaciones
 “>>” : mucho mayor

$$\text{Minimizar } \sum G(a_i, b_j) = K * \sum(a_i, a_j) + K * \sum(b_i, b_j)$$

Para el tiempo $t=1$

Sujeto a:

$$\sum G(a_i, b_j) \leq CM(i, j)$$

$$\sum G(a_i, a_j) \leq K * C1$$

$$\sum G(b_i, b_j) \leq K * C2$$

$$CB = (1 / m1) * (C3 + C4).$$

$$CB = (1 / m2) * (C5 + C6).$$

Donde:

C1: Capacidad de alta velocidad en el corazón de red.

C2: Capacidad de alta velocidad en el corazón de red.

C3, C4, C5, C6: Capacidad de transmisión de cada estación con su conmutador (*switch*).

En el proceso de optimización se desea lograr encontrar:

- Las estaciones que necesitan estar más cerca del servidor de archivos habitual (mayor utilización) o del servidor de parches (volumen considerable de información).
- Se da uso a las temporadas de picos de tráfico reconfigurando la red para entregarla a los servicios que más consumen. Esto es definir una constante K que sea un multiplicador muy alto comparado con el tráfico de las demás estaciones.
- Mover gran cantidad de información entre un servidor central y local para aprovechar los momentos de desocupación.

El proceso de decisión se convierte en la posibilidad de auto-configurar así:

- Mover un elemento de un conjunto a otro.
- Unir dos conjuntos.
- Dividir o especializar un conjunto.

2.3 Escenarios de Auto-configuración

La auto-configuración de red se puede encontrar de manera intuitiva en varios escenarios de red, pero muy difíciles de aplicar por actuaciones manuales de los administradores de red. El objetivo del sistema permitirá aplicarla de manera automatizada en muchos puntos de la red de manera concurrente y distribuida.

Ejemplos de escenarios de auto-configuración de red.

Se presentan a continuación una serie de problemáticas que enfrentan los administradores de red y las empresas operadores de Internet (ISP).

- Escenario 1: Dos estaciones transmiten volúmenes altos de información y están en VLAN diferentes.
- Escenario 2: Varias estaciones interactúan con un mismo servidor o recurso de manera concurrente (servidor de archivos).

- Escenario 3. Muchas estaciones en la misma subred son inundadas de tráfico (*multicast, broadcast, echo-reply*) desde una estación.
- Escenario 4. La expectativa de una final de un gran evento deportivo hace que un gran número de equipos se conecten a servidores de internet a un servicio de *streaming* de video.
- Escenario 5: Varias estaciones hablan con muchas para generar un ataque informático DDOS y bloquean los *switches* agotando su capacidad de *backplane, buffers* de los puertos de red o capacidad del *uplink*.
- Escenario 6: En una red de datos, se anuncia una *mac-address* en uno o múltiples puertos de varios *switch* agotando la capacidad del bus, memoria de los puertos de red.
- Escenario 7. Múltiples tipos de tráfico (servicios) en una misma subred. Se encuentra mezcla de grupos de usuarios con diferentes intereses. Un ejemplo es una red de video o edición en línea mezclada en la misma red de transacciones de bases del sistema de contabilidad.

3. Descripción del experimento

El software desarrollado supone que existe una red en funcionamiento, posee usuarios que utilizan la red para navegar a Internet y se conectan a sistemas de información de la organización. Dada esta suposición, deben encontrarse en operación los servicios que apoyan el funcionamiento de la red de telecomunicaciones.

Pre-requisitos para los experimentos.

- Una red de telecomunicaciones formada por *switches* y enrutadores.
- Servidor de asignación de direcciones IP.
- Conjunto de direcciones IP disponibles.
- Un servidor de nomenclatura de nombres de dominio DNS.
- Un servidor directorio y políticas de seguridad.
- Un servidor y servicio que contenga los usuarios y un servicio de autenticación.
- Un sistema de distribución de software.
- Conexión a Internet.
- Servicio de portales web.
- Contenedores de aplicaciones.
- Sistemas de bases de datos.
- Una conexión o enlace de Internet.
- Sistema de control perimetral.

Experimento elemental.

Mediante un laboratorio elemental se conformó una red compuesta por tres (3) *switches* y dos (2) enrutadores en forma de árbol sin bucles mediante equipos reales (no se utilizaron simuladores como GNS3).

- Se simuló tráfico con un grupo de 5 PCs y 4 servidores.
- Se enviaron ráfagas de transferencia de archivos mediante el protocolo sftp (*Secure File Transfer Protocol* en el puerto TCP 22).
- Simulador de tráfico.
- Intervalo de muestreo cada 5 minutos.

Dado que la red está aislada de un sistema en producción

- Se puede controlar y cambiar el ancho de banda en los puertos de un *switch*.
- Se puede controlar el ancho de banda.
- Se puede simular paquetes perdidos, recibidos, *jitter*, y modificar los parámetros de calidad de servicio (QoS).
- Se puede controlar los caminos de conexión con STP.
- Se puede controlar los enrutadores, y aplicar desvío de paquetes (*route map*).

Objetivo: Resolver el escenario 1 y 2 (Ver 2.3 Escenarios de Auto-configuración) respecto a la especialización de las VLAN para aumentar eficiencia.

Experimento en una red de computación en producción.

La Universidad Nacional de Colombia en la Sede Bogotá cuenta con una red de computación que conecta 120 edificios, una red inalámbrica que posee 12.000 usuarios conectados y un canal de Internet de 1.8Gigabits/segundo. Posee dos centros de cómputo interconectados por enlaces de 20Gigabits/segundo. Los servidores y la nube de computación están conectada al corazón de la red mediante enlaces de 80 Gigabits/segundo (“Licitación CON-BOG-007-2016 Universidad Nacional de Colombia,” 2016).

Se elige un ambiente controlado para realizar las pruebas del sistema multi-agente. Mediante 3 estaciones ubicadas en 5 subredes virtuales se puede probar el sistema distribuido de agentes en donde cada subred representa un ambiente diferente. Se solicitó a la Universidad contar con acceso a algunos *switches* de red en modo de solo lectura, para probar los agentes recolectores. En este tipo de redes empresariales, cuentan con servidores que controlan los perfiles de los usuarios mediante un directorio. Se solicitó contar con una unidad organizacional (OU) del sistema controlador de dominio (*Microsoft Active Directory*).

Dado que no se desean resultados inesperados que puedan afectar el servicio y por solicitud de los administradores de la red se inhiben los cambios automáticos en los *switches*. En su reemplazo, se presenta un informe o planeación de modificaciones que sugiere el modelo de optimización con su respectiva configuración para aplicación manual por parte de los administradores de la red.

Objetivo: Resolver el escenario 1 y 2 (Ver 2.3 Escenarios de Auto-configuración) respecto a la especialización de las VLAN para aumentar eficiencia mediante un “Plan de cambios de aplicación manual”.

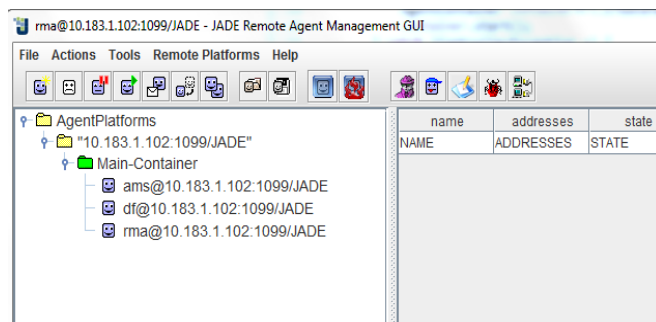
3.1 Infraestructura de software

La infraestructura de software permite mostrar el conjunto de componentes de hardware y software mínimos para desplegar un sistema distribuido de multi-agente que pueda recoger información de la red con un enfoque holístico.

3.1.1 Agente principal de Jade

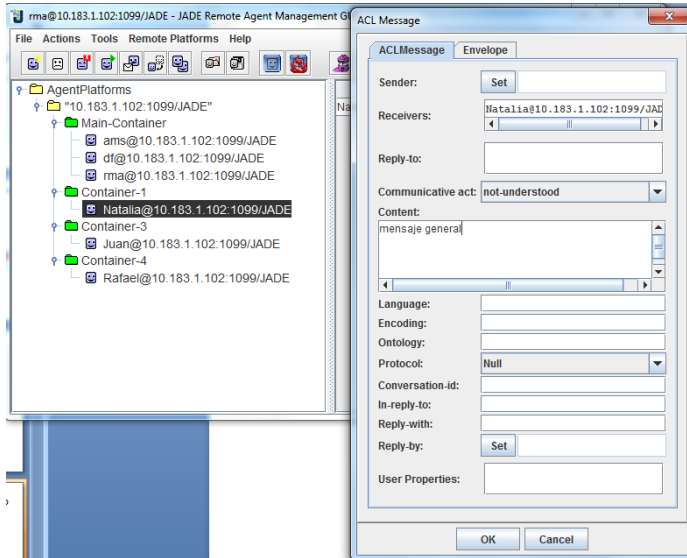
Administra el sistema central de registro de los agentes. Cada agente se registra mediante un nombre y se incluye en un contenedor. Si cada agente cuenta con múltiples comportamientos, debe registrarlos para ofrecerlos a su comunidad social. Cada agente

Ilustración 15. Agente principal Jade.



se registrará ante ente central mediante el agente DF (*Directory Facility*). El agente central contiene el agente denominado AMS (*Agent management System*) que permite manejar la comunicación entre agentes (“no se confunda con un sistema de chat o email) para pasar estados, información, código, reglas lógicas, entender el ambiente. Desde el punto de vista de cada agente, cada uno tiene su propia visión del ambiente. Un agente entre más cerca se encuentra de un fenómeno mejor información suministra. Los mensajes entre agentes permiten que cada uno reacciones sobre el ambiente para cambiarlo o adaptarlo. El sistema central mantiene colas de comunicación entre agentes para el procesamiento de los hilos de trabajo.

Ilustración 16. Comunicación entre agentes.



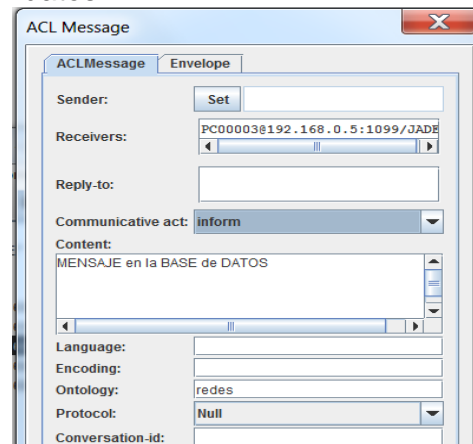
El sistema de mensajes permite aplicar una ontología de manera automática. Si se conocen las relaciones entre los objetos, las jerarquías y el lenguaje. En la comunicación se hace necesario cumplir con el estándar internacional FIPA y el lenguaje para comunicar entre agentes ACL (*Agent Communication Language*).

3.1.2 Capa media

Los agentes pueden hacer uso de herramientas externas o del sistema operativo. Para el caso del agente de consolidación de información se utiliza un acceso a base de datos mediante una conexión ODBC.

Se integró al sistema de capa media, un sistema de portal web mediante un servidor en Linux con el sistema Apache y un contenedor de aplicaciones para desplegar la interfaz de administración. En el contenedor de portal web se desplegará un desarrollo realizado en Java JSP y metodología J2EE.

Ilustración 17. Persistencia de información en la base de datos.

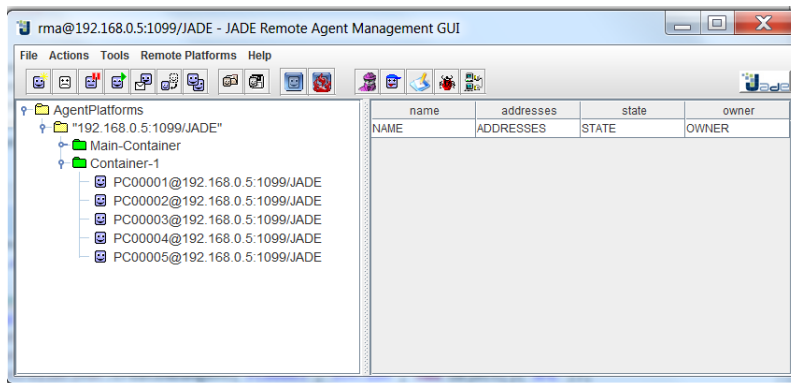


3.1.3 Consola de agentes

Selección e Instalación de plataforma computacional. El servidor de agentes se espera que tenga recursos de RAM. Se requiere un (1) mega de memoria por cada 100 agentes. JADE funcionó en Windows y en Linux.

La consola de agentes muestra el nombre de todos los agentes agrupados mediante una estructura denominada contenedor. Sin importar el contenedor los agentes mantienen su capacidad de comunicarse. El nombre con el cual se registra el agente es clave para enviarle estímulos. En el caso de las redes de datos, es posible tener problemas de homónimos. En ocasiones se utilizan las direcciones IP para identificar los agentes.

Ilustración 18. Consola de agentes.

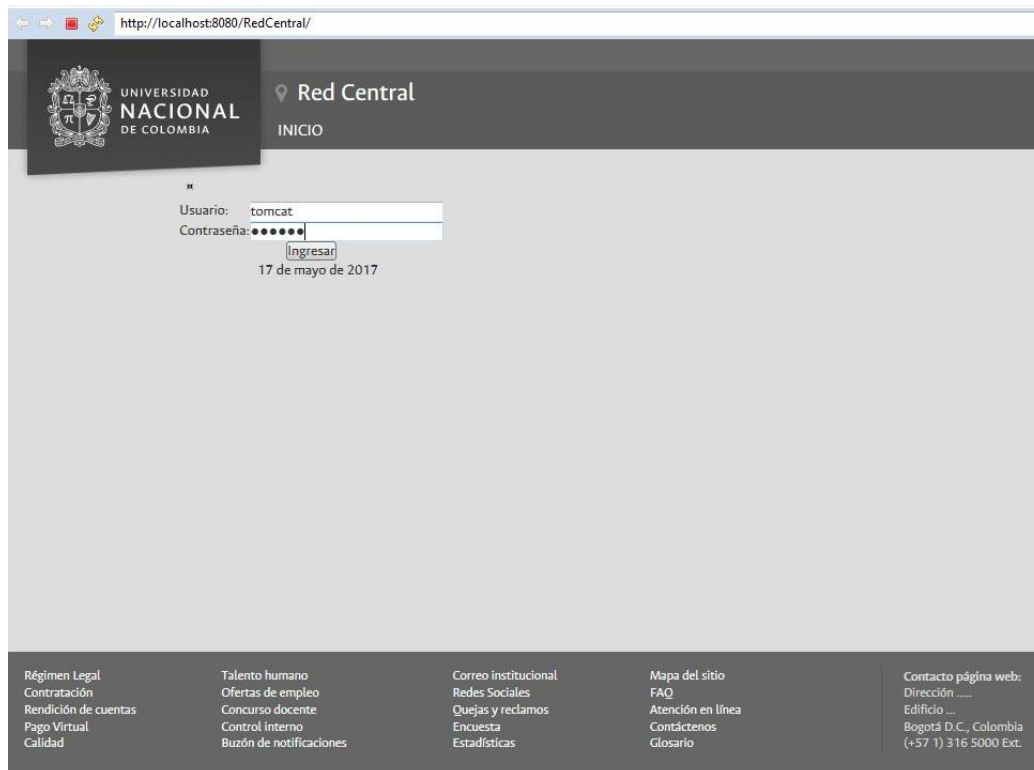


Este método no es confiable dado que las direcciones IP son cambiadas o rotadas por las políticas de asignación del DHCP o de las políticas de seguridad de la organización. En el caso de los equipos portátiles o aquellos que poseen varias interfaces de red tanto fijas como inalámbricas, los agentes se registrarían múltiples veces con diferente información. Se invita a que el registro se realice mediante un número único del máquina con es el UUID (*Universal Unique Identifier*) RFC4122 (Leach, Mealling, & Salz, 2005).

3.1.4 Consola del software de gestión.

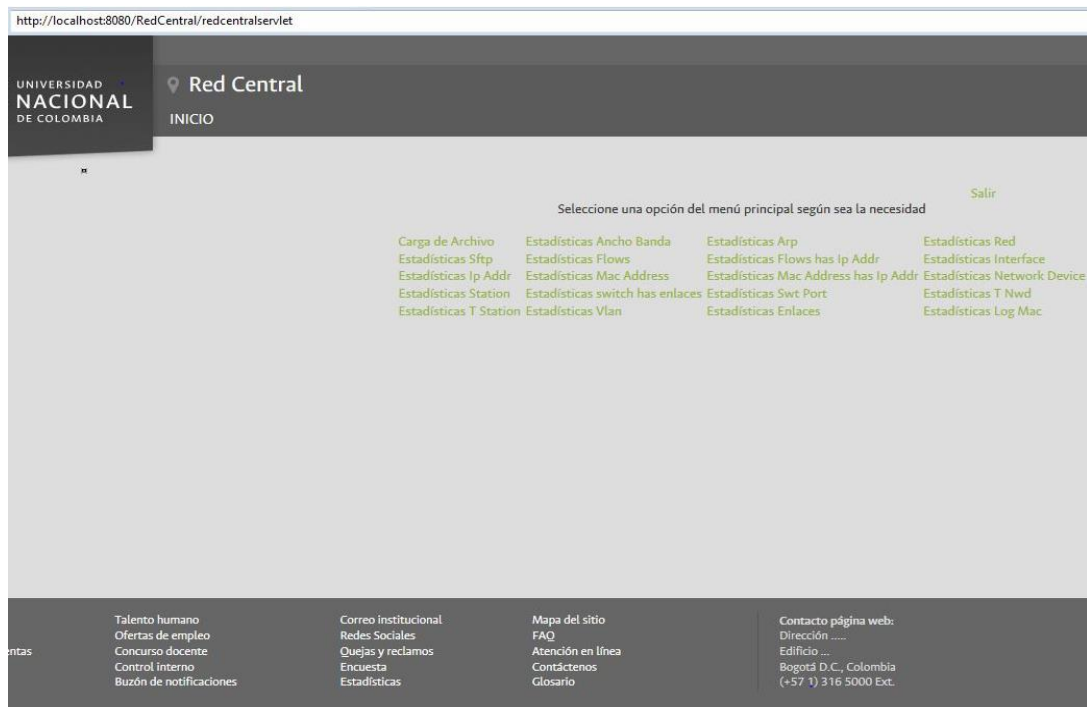
Se diseñó e implementó un portal web para uso del administrador de la red. Se definió un sistema de control de roles para diferentes tipos de perfiles de usuario que se conecte.

Ilustración 19. Portal principal de gestión.



Con toda la información recolectada por los agentes, el administrador puede optar por seleccionar varios tipos de análisis sobre los datos. El sistema le presenta la lista de informes disponibles para la consulta.

Ilustración 20. Selección de opciones estadísticas.



Tecnología.

La aplicación Web RedCentral se desarrolló en J2EE 1.4 (*Servlet* 2.4, *Jsp* 2.0), desplegado sobre un Servidor *Web Apache Tomcat* 5.5.28 con las bibliotecas *JFreechart*, *Jsch* y *jdk* 1.8.

Arquitectura.

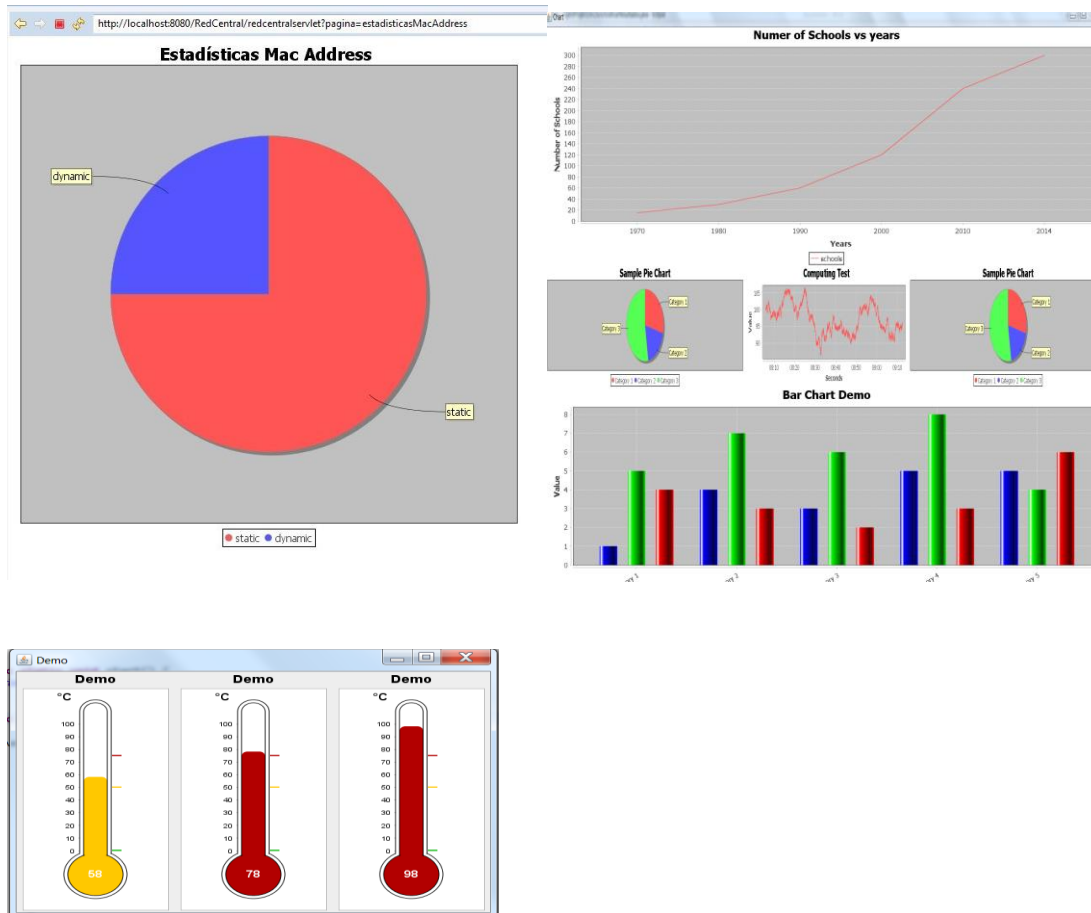
Se utilizaron patrones de desarrollo como MVC2 (Modelo Vista Controlador con énfasis en la capa de negocio), DAO (*Data Access Object*), *Singleton*, *Service Locator*, *Front Controller*, *Bean*, *Factory* (fábrica de objetos en memoria o base de datos), *Facade*.

Descripción.

Red Central Web tiene implementado el módulo de autorización, autenticación de usuarios por medio del servidor de aplicaciones *Apache Tomcat*. Dependiendo del rol que tenga el usuario la aplicación permite personalizar las opciones de menú por usuario, sobre el menú puede verificar por medio de la opción "Carga de Archivo" el tiempo que tarda un archivo en ser enviado desde el cliente al servidor *sftp*, permitiendo realizar análisis sobre la red, visualizar la información de las estadísticas generadas por la aplicación multi-agente para el ancho de banda, *arp*, estadísticas *sftp*, estadísticas *flows*, mac-address de forma gráfica.

3.1.5 Capa del cliente final

Módulo que permite la visualización en tiempo real del comportamiento de la red y la actuación del sistema multi-agente. La información se presenta en tableros de control para el administrador de la red o para el gobierno de tecnología.



3.2 Infraestructura de hardware

Se realizaron las pruebas con equipos reales de marca Cisco. No se utilizaron simuladores de *switches* o enrutadores como NS3 o GNS3. Para el laboratorio se requiere la simulación de tráfico de usuarios desde un grupo de estaciones como PC y servidores. Además, de la simulación del tráfico se generan ráfagas de tráfico para probar el ancho de banda de las conexiones.

Realizar el laboratorio sobre tres simulaciones anidadas puede conducir a resultados no exactos, no reales o sesgados por la computación donde se corre.

3.2.1 Modelo de switches y routers

El modelo de laboratorio se fabricó con equipos físicos de marca Cisco en un esquema que nos permita probar los casos cotidianos de un administrador de red.

Modelo para *switch*: “*core-distribution-access*”

- *Switch Ethernet* capa 2 del corazón de red. Los *switch* se interconectan en forma de árbol en donde es habitual colocar el *switch* con mejor desempeño en la raíz (mayor capacidad en la matriz de la conmutación (*switching*) interno denominado bus o *backplane*). A esta capa se le denomina corazón (*core*). Estos equipos generalmente ubicados en los centros de cómputo principales de una organización. En esta capa no se conectan estaciones de usuario final.
- Los *switch* de distribución: los de segundo nivel de capacidad de backplane, se conectan como hijos del primer *switch* y se denomina a esta capa “distribución”. Se ubican en el centro de un edificio. Si el edificio es muy grande (distancias mayores a 90 metros) hasta el *switch*, entonces se divide el edificio en partes por cuadrados hasta 90 metros. El *switch* de distribución entonces se encargará solo de conectar a todos los *switch* del edificio.
- Los *switch* de tercer nivel del árbol se utilizan para conectar todos los equipos finales de usuarios como son PC, impresoras, dispositivos embebidos, cámaras IP, Smart-TV, teléfonos IP.
-

Para el laboratorio se configuran dos capas corazón y acceso (*core-access*) conformado por los equipos cuyos nombres son: *sonia*, *monica* y *yamile*.

No es usual encontrar en una red un bucle cerrado. Los diseñadores de red hacen diseños físicos donde evitan al máximo cerrar bucles. El estándar Ethernet no permite tener bucles dado que genera problemas de inundación de paquetes y los *switch* tienden a quedar en ciclos infinitos. El estándar 802.3-2010 ya incluye algoritmos que fabrican el grafo de la red

y resuelven los bucles mediante una negociación entre *switch* para cerrar uno o varios puertos y dejar un árbol.

En este laboratorio se deja un bucle cerrado con el propósito de revisar la problemática de bucle (“*loop*”) y de la utilización de este enlace como redundante en procesos de particionamiento de la red para volver a contar con la conectividad a un sector de la red. De igual forma, se opta por la utilización del algoritmo de MST (*multiple spanning tree*) que permite el uso de la conexión de redundancia para transporte de tráfico de VLANS diferentes.

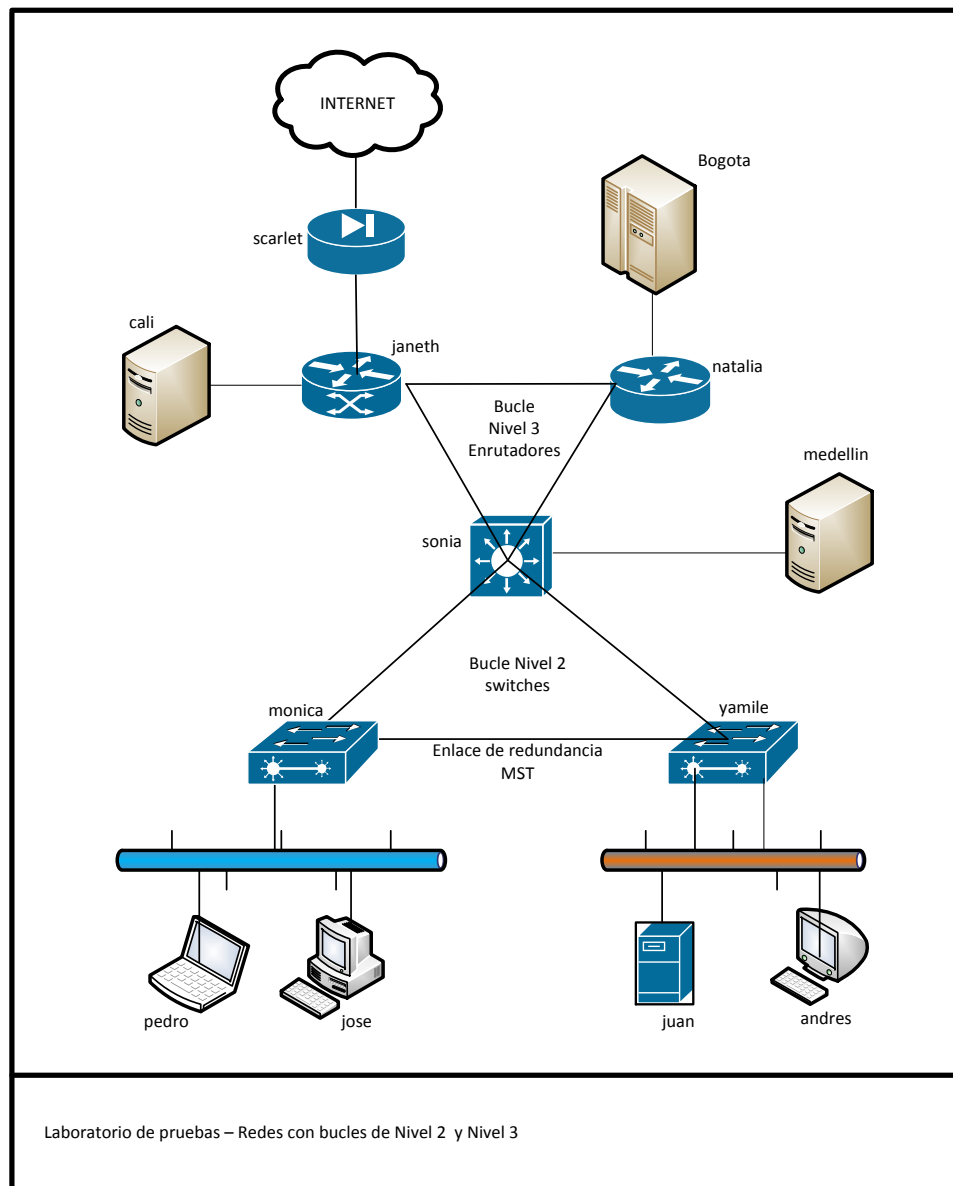
Enrutadores. Se construyó otro bucle con dos equipos enrutadores de marca Cisco y con un *switch*. Se activó la instancia de enrutamiento capa 3 en el *switch* de corazón de red. Este *switch* entonces actuó con doble propósito, capa 2 y 3. Para el laboratorio se eligió en algoritmo de enrutamiento L3 “*eigrp*”.

Tabla 2. Equipos de laboratorio.

Equipo	Nombre	Referencia
<i>Switch</i>	Sonia	Cisco Catalyst 3750
<i>Switch</i>	Yamile	Cisco Catalyst 2950
<i>Switch</i>	Monica	Cisco Catalyst 2960
enrutador	Janeth	Cisco 1800
enrutador	Natalia	Cisco 2600
Enrutador banda ancha	Scarlet	Huawei

Para realizar pruebas a redes conexas, se realizó una conexión a Internet mediante una conexión de banda ancha con el proveedor de Internet ISP ETB Empresa de Telecomunicaciones de Bogotá.

Ilustración 21. Laboratorio de pruebas.



Los enlaces entre los *switch* se realizaron mediante la agregación de varios puertos (4 de 100 Mbps) Ethernet para contar con mayor ancho de banda. Se utilizó el protocolo LACP (*Link Aggregation Control Protocol 802.1ad*). De igual forma se configuró varias subredes virtuales de la capa 2 (VLAN) para colocar estaciones en diferentes redes y hacer transitar la información por toda la red. Estos enlaces agregados se configuraron como puertos troncales 802.1q.

4. Desarrollo del modelo e implementación del producto de software

El modelo implementado tiene en cuenta que se tienen variables que representan:

- Estados. Variables que tiene un conjunto de valores finito de valores. Un ejemplo Un puerto de un equipo de red en el algoritmo de STP puede estar en el conjunto *{Blocking, Listening, Learning, Forwarding, Disabled}*.

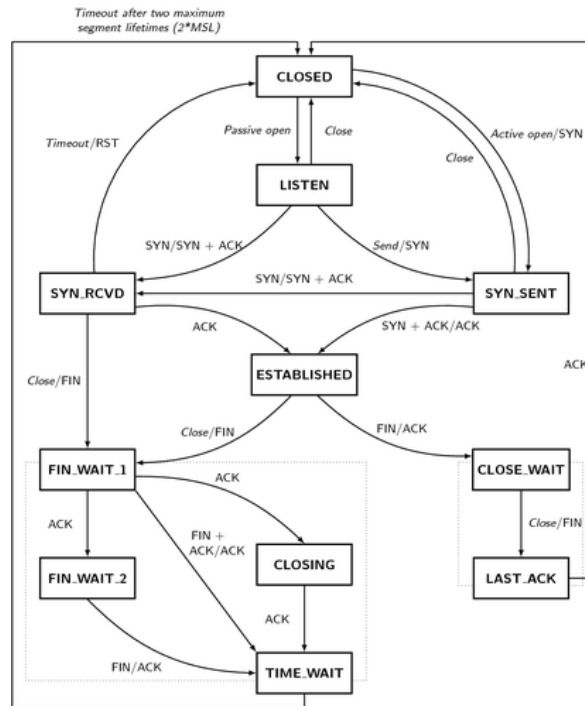
En la transmisión de paquetes en la capa 3 se puede observar varios estados y transiciones. Conocer el estado en el cual se encuentra el datagrama permite entender el siguiente estado al que pasará. Si no sucede ese cambio o pasa a otro estado se considera un mal funcionamiento del sistema de red.

Agregar la metodología (según Sustentación)

Modelo de desarrollo en espiral

(Boehm & Turner, 2015)

Ilustración 22. Diagrama de estados de TCP.



- Estadísticas. Una variable puede tomar valores en el espacio de los números enteros en el intervalo $[0, \text{Máxima_velocidad_puerto}]$. Los valores cambian en el tiempo y representan el uso de la red por parte de una estación. Los equipos de red no almacenan la información histórica de los valores. Por esta razón, se utilizan sistemas externos que capturen el valor de manera repetida con un intervalo de muestreo para lograr contar con una estadística de horas o semanas.
- Configuración. Los elementos de red poseen archivos de configuración que definen un comportamiento global de la red, como los caminos de los paquetes, los grupos de estaciones en una VLAN, la calidad de servicio de los paquetes que deben ser diferenciados como Voz IP, los algoritmos de enrutamiento, traducciones de direcciones. Es importante realizar un seguimiento a las configuraciones de estos elementos para mantener un diagnóstico correcto de las situaciones.
- Límites físicos o lógicos. Existe valores que tienen los equipos de red que no cambian. La capacidad máxima de transmisión de un puerto es fijo porque está asociada al diseño electrónico de los componentes. De igual forma a las velocidades de transmisión de los cableados de cobre y fibra. Estas variables deben ser recuperadas y mantenidas en el software para realizar las ecuaciones

de optimización. La diferencia es que estas variables no es necesario consultarlas de manera repetida.

4.1 Recolección de los datos

Se pueden encontrar múltiples mecanismos utilizados para la recolección de información en las redes de datos:

- Software de captura de paquetes.
- Hardware para realizar copia de los paquetes de un puerto físico de red.
- Bibliotecas de software para intervenir la pila de protocolo del núcleo del sistema del sistema operativo.
- Funciones embebidas en los equipos de red para administración de red.

La estructura y forma de realizar la extracción de información tiene su base en la estructura y definición de la pila de protocolos de red (Ver 1.6 El modelo OSI, la subcapa MAC y el sistema multi-agente), los modelos de gestión (NMS de TCP-IP), y otros protocolos propietarios incluidos diferentes fabricantes de equipos de redes.

El sistema multi-agente, permite la construcción de una sonda que desde origen a destino presente valores efectivos de una conversación de red con parámetros como envío y pérdida de paquetes, mediante la interrelación de los agentes. Además, es posible realizar la toma de información entre varias *n-tuplas* de objetos de red.

4.1.1 Granularidad, escala e intervalo de muestreo.

La recolección de información se puede realizar a diferentes niveles de detalle. Iniciando en la capa 1 como son los bits, pasando a la capa 2 donde se observan las tramas (*frames*), en el nivel 3 se encontrarán los paquetes (*packets*) y de esta forma se asciende hacia las capas 6 y 7 donde se encuentra con los servicios y su presentación como por ejemplo el protocolo http o una base de datos o un servicios de DNS. Es importante anotar que los modelos de operación, gestión y gobierno de tecnología (COBIT – ITIL) agrupan varios servicios para elaborar otros indicadores de disponibilidad del servicio ofrecido a los usuarios.

Se utiliza una analogía con los sistemas químicos y biológicos para que no se olvide la implicación de la escala en la toma de la información.

Tabla 3. Analogía de Química y Biología con Redes de datos.

Biológico / Químico	Redes de Computación
Electrón: giro, energía.	Secuencias de bits (<i>Streams</i>).
Atómico: Relaciones entre átomos	Segmentos (<i>frames</i>)
Moléculas	Datagrama
Compuestos	Flujos
Proteínas	Servicios
Tejidos	Agregación de servicios

La importancia de la escala en la toma de información radica en los tipos de fenómenos que se pueden observar. Las interacciones entre átomos permiten observar fenómenos de atracción entre electrones que no se pueden encontrar a nivel de una molécula o de un tejido. El fenómeno en a nivel atómico ocurren en femto-segundos (1×10^{-15} seg).

En las redes de datos, los fenómenos que se observan en la capa 1 como puede ser las interferencias electromagnéticas, problemas en el cableado no son fáciles de observar a nivel de un paquete o de un servicio.

Para realizar la explicación a un problema o fenómeno de red, en ocasiones se hace necesario realizar toma de información a diferentes niveles de escala. La combinación de los resultados a diferente escala puede apoyar mejores resultados en el proceso de decisión para la gestión de red.

En la recolección de información no se posible tomar la información del universo dado que sería necesario contar con recursos de computación y almacenamiento muy grandes (miles de CPUs, miles de Terabytes de almacenamiento). La realidad es que se crean modelos y se ejecutan simulaciones realizando un muestreo a un intervalo regular. No se debe olvidar que los fenómenos de auto-similitud en el tráfico de internet estudiado por Hurst (ver 1.1.1 Auto-similitud en el tráfico y estacionalidad) que nos permiten estar seguros que al modificar la escala los análisis sobre el tráfico son válidos.

La elección del intervalo de muestreo permite elegir la medida del intervalo tan pequeño como sea necesario para lograr atrapar la información que permite evidenciar un fenómeno. Para capturar una contraseña de un usuario posiblemente sea necesario realizar captura de paquetes continua por espacio los 20 segundos, misma cantidad de tiempo que le toma al usuario digitarlo en su estación y justo cuando está realizando una autenticación. Las demás muestras de información ya no serían útiles para ese tipo de trabajo. En la gestión de tráfico de red generalmente los intervalos se establecen entre 10 segundos, 30 segundos, 1 minuto y 5 minutos.

La granularidad de la recolección de la información hace referencia a la cantidad de variables o datos son útiles para explicar un fenómeno. Si se realiza un análisis a la calidad

de servicios (QoS) de un flujo de voz, la información relevante se encuentra entre las direcciones IP de origen, destino, los campos de los paquetes *udp* y *tcp* y los campos de *DSCP*, *DIFFSERV*, el tiempo medio entre llegada de paquetes (*jitter*), y la pérdida de paquetes. No es necesario capturar toda una llamada de voz. Como consecuencia se escogen las variables y se toman los datos de los encabezados de los paquetes o tramas para alimentar la información que el modelo necesita. No es necesario obtener todo el paquete con su carga (*payload*), recordando además que se podría estar infringiendo leyes de privacidad.

Ilustración 23. Detalle de un datagrama.

Octeto	0							1							2							3										
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Puerto de origen														Puerto de destino																	
32	Número de secuencia																															
64	Número de acuse de recibo (si ACK es establecido)																															
96	Longitud de Cabecera	Reserv.	NS	CS	WE	UR	AK	PH	TS	FIN	Tamaño de Ventana																					
128	Suma de verificación														Puntero urgente (si URG es establecido)																	
160	Opciones (Si la Longitud de Cabecera > 5, relleno al final con "0" bytes si es necesario)																															
...	...																															

Además, la granularidad también nos permite aplicar filtros para tomar la información necesaria dado el considerable volumen de flujos de información que cruza por la red.

Retos en cada escala. Se anota que manejo de las variables y su interpretación siempre debe estar sometida a:

- Las variables tiene su interpretación en su capa.
- Se puede abstraer el funcionamiento de cada capa de red por un conjunto de variables.
- Pueden existir varias soluciones por la interpretación de variables. Se acompañan de variables de las capas adyacentes del modelo OSI para mejorar la precisión de las conjeturas.

A continuación se presentan el conjunto de métodos que se utilizan para obtener información en la redes de datos. En este trabajo se implementó agentes que actúan como sondas directamente en el ambiente. Los siguientes métodos podría implementarse como agentes de recolección de información complementarios o para equipos de red que solo cuentan con un método para presentar la información de gestión.

4.1.2 **SNMP**

SIMPLE NETWORK MANAGEMENT PROTOCOL. Esquema de extracción de datos en esquema cliente y servidor estandarizado en todos los software comerciales. Estandarizado en el RFC 1155, 1156, 1157 (McCloghrie & Rose, 1990) y (Case, Fedor, Schoffstall, & Davin, 1990). Se requiere de una estación central para recolección de los datos, análisis y procesamiento denominada NMS (*Network Management System*). Los datos obtenidos siempre son solicitados desde el NMS a cada una de las estaciones mediante un mensaje de consulta (*SNMP Get*). Cada una de las estaciones debe poseer un agente de SNMP y una palabra clave para que NMS lo pueda consultar y el agente le responda (Rose & McCloghrie, 1990).

Los agentes SNMP en las estaciones por iniciativa propia pueden enviar alertas o alarmas de emergencia al NMS cuando se presenten fallas mayores. Se realiza mediante el mensaje de disparo (*SNMP Trap*). Todo el sistema de SNMP es considerado otro protocolo de control dentro de la pila de protocolos TCP-IP.

Una estación cliente cuenta con mucha información para ofrecen del sistema operativo, de la red o cualquier software que puede estar corriendo. Los diseñadores de SNMP y los fabricantes de organizaron toda la información en un árbol jerárquico para presentar la información y facilitar la consulta. Esta estructura se conoce como MIB (*Management Information Base*).

4.1.3 **Analizadores de paquetes**

Esquema para interceptar paquetes y generar un archivo de bitácora de la transmisión. La mayoría de los analizadores utilizados sobre Linux y Windows emplean las bibliotecas para copiar la pila de protocolos a un área de análisis (*libpcap*) (“TCPDump,” 2017) como lo hace el software Wireshark (Combs, 2017). Todos se basan en el núcleo de Linux mediante las estructuras de la pila de protocolos mediante las estructuras conocidas como “*skbuff*” (Mauerer, 2008) .

Los analizadores de paquetes requieren poder observar todos los paquetes que cruzan por la red de la estación que se desea observar. Generalmente, se utilizan configuraciones especiales en los equipos de red y en las tarjetas de los analizadores conocidas como

puertos espejo y puerto promiscuo (en referencia a tomar los paquetes y tramas que cruzan aunque no sea el remitente).

Ilustración 24. Captura binaria de enrutador.

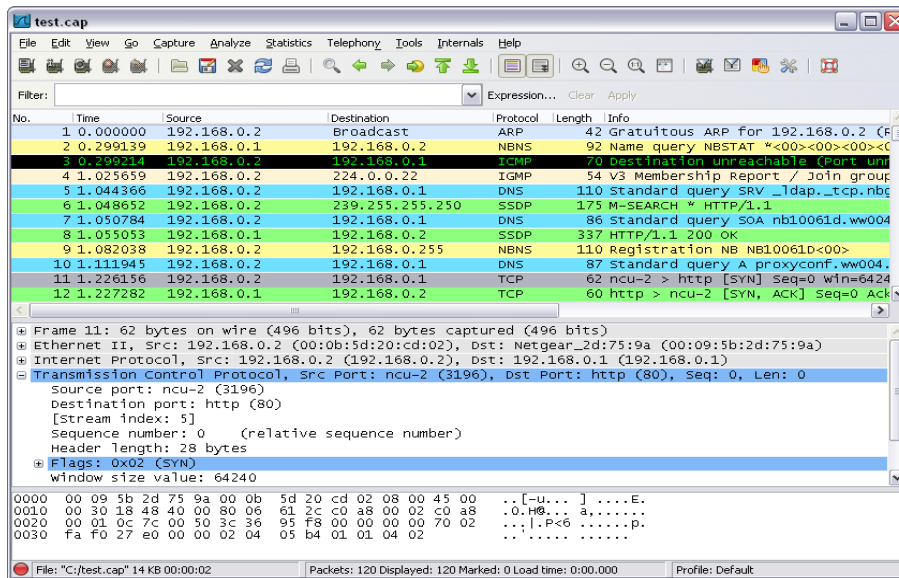
```
Streams binarios
Router# show monitor capture buffer pktrace1 dump

11:13:00.593 EDT Mar 21 2007 : IPv4 Turbo : Fa2/1 Fa0/1 65B6F500:
080020A2 44D90009 E94F8406 08004500 .."DY..iQ....E. 65B6F510:
00400F00 0000FE01 92AF5801 13025801 .@....~/X...X. 65B6F520:
58090800 4D1A1169 00000000 0005326C X...M.i.....2l 65B6F530:
01CCABCD ABCDABCD ABCDABCD ABCDABCD .L+M+M+M+M+M+M+M
65B6F540: ABCDABCD ABCDABCD ABCDABCD ABCD00
+M+M+M+M+M+M+M. 11:13:20.593 EDT Mar 21 2007 : IPv4 Turbo :
Fa2/1 Fa0/1 65B6F500: 080020A2 44D90009 E94F8406 08004500 ..
"DY..iQ....E. 65B6F510: 00400F02 0000FE01 92AD5801 13025801 .@....~/
X...X. 65B6F520: 58090800 FEF91169 00000000 0005326C X...^y.i.....2l
65B6F530: 4FECABCD ABCDABCD ABCDABCD ABCDABCD
Ol+M+M+M+M+M+M+M. 65B6F540: ABCDABCD ABCDABCD ABCDABCD
ABCDFF +M+M+M+M+M+M+M
```

En los equipos centrales de la red con capacidades limitadas, se puede activar los puertos de monitoreo pero la información debe desviarse a un servidor alternativo para capturar la información en una bitácora. Por la velocidad en la que cruzan las conversaciones (miles por segundo) no es posible realizar un análisis manual. Esta es la razón por la cual se usan sistemas que apoyen estos análisis.

Una vez se cuenta con la bitácora de las transacciones se puede utilizar herramientas que mediante utilitarios especializados por protocolo, apoye el análisis. Cada una de las capas del modelo OSI posee su formato al igual que cada protocolo. Se presenta un *Packet Analyzer* conocido como Wireshark (de la capa 2 a la 7).

Ilustración 25. Captura de paquetes con Wireshark.



4.1.4 RMON

Modelo de administración, análisis de protocolos y monitoreo diseñado por el IETF. RFC 2819, 4502. Se apoya en el sistema de SNMP. Mantiene el esquema de cliente y servidor para almacenar la información de gestión. Como diferencia con SNMP se encuentra que las estaciones mantienen los estados de las variables y no requiere un sondeo periódico para conocer los cambios. Cuando ocurren novedades las comunica a la estación de gestión. Su objetivo es optimizar el tráfico generado por la gestión (Waldbusser, 2000) y (Waldbusser, 2006).

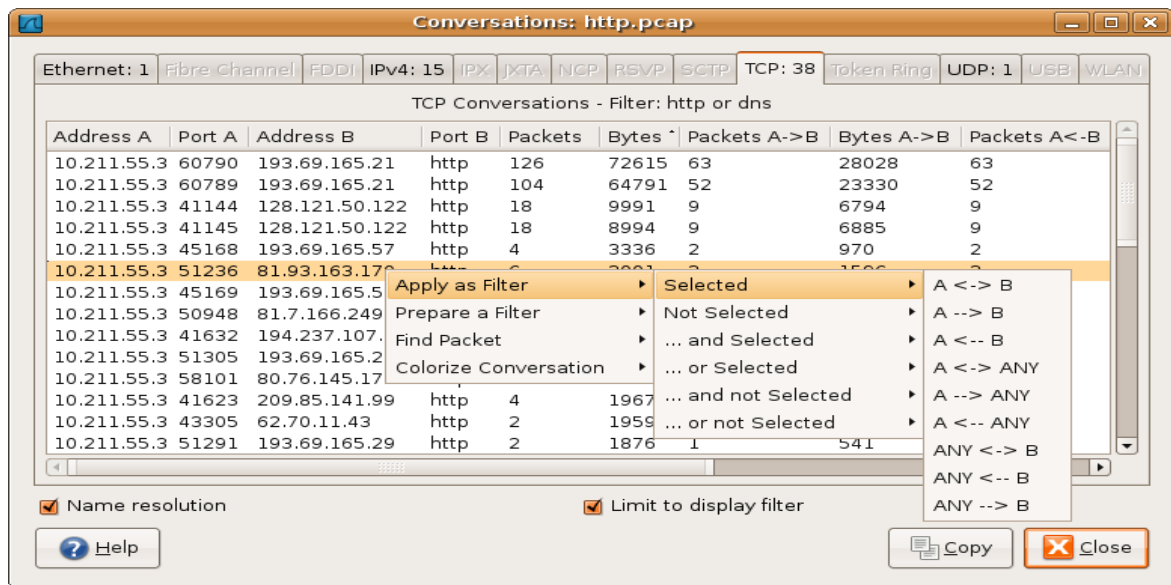
4.1.5 Captura de flujos

Cuando se establece una interacción en una red, es importante entender todo el diálogo entre las partes para establecer comportamientos de la capa 6 y 7 del modelo OSI. Se define conversación como el tráfico intercambiado entre dos puntos específicos de forma bi-direccional.

Netflow. Captura de información de trazas de información mediante inspección rápida de cabeceras IP utilizado y patentado por Cisco RFC3954 (Claise, 2004).

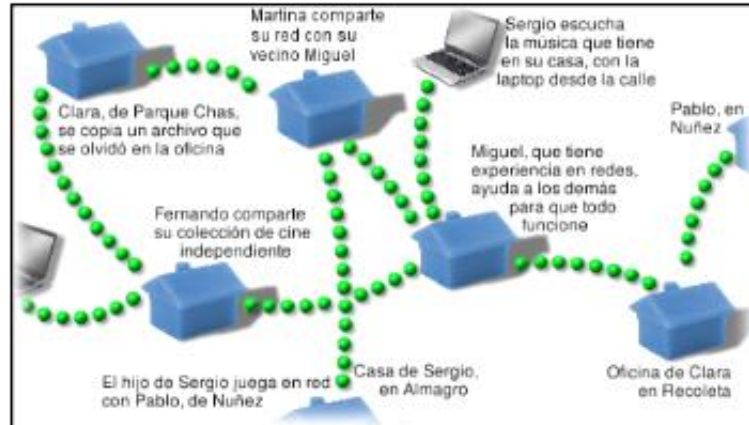
Sflow. Captura de trazas de información con muestreo. No necesariamente analiza todo un flujo de datos. Fue impulsado por los fabricantes de redes como alternativa a *Netflow* de Cisco RFC3176 (Phaal, Panchen, & McKee, 2001).

Ilustración 26. Conversaciones de red.



Mediante un sistema de captura de la información de las conversaciones de red se logran establecer análisis de alto nivel. Si se requiere conocer cual protocolo o servicio consume mayor ancho de banda se requieren sistemas como *netflow* o *sflow*. Los análisis de tráfico de alto nivel nos permiten observar la multiplicidad de intereses o necesidades de los usuarios.

Ilustración 27. Ejemplo de múltiples conversaciones de red.



Es importante anotar, que las capturas de las conversaciones (*flows*) están en la capa 7 del modelo OSI. Una conversación involucra miles de tramas y paquetes IP. Las conversaciones comienzan con el inicio de sesión hasta la respuesta con la información solicitada.

4.2 Abstracción de datos y relaciones

La descripción de los conceptos y relaciones entre ellos relacionado con redes de datos, son parte importante del conocimiento de un agente o una sociedad de agentes.

Se definió el uso de una ontología como la especificación de una conceptualización y de un recurso artificial que se crea. Permite formular el modelo abstracto de algún fenómeno del mundo en el que se identifican los conceptos que son relevantes

Resuelve:

- Abundancia de comunicación entre agentes. Permite homogeneidad de idioma, vocabulario y protocolos.
- Interoperabilidad de sistemas y plataformas.
- Problemas entendidos en el mismo entorno.
- Semánticos. Se da similar sentido a los términos por el contexto.

Para “redes de computación” la ontología definiría objetos como:

- Switch router AP- controller Firewall

- Cable utp fiber
- Address ip mac_address
- Vlan2 vlan3 routing table mac_table
- Ethernet tcpip protocol
- Tcp udp icmp
- Interfaz ethernet, fastEthernet, gigabit, tenGiga
- Mtu delay jitter speed bandwidth
- Equipment active pasivo.

Y en las relaciones entre objetos, se pueden ver:

- Estación se conecta a dispositivo de red.
- Asociación entre direcciones *mac-address* e IP.
- Caminos de transporte de paquetes.
- Algoritmos de enrutamiento.
- Traducción de direcciones IP.

Base para la base de conocimiento de la inteligencia artificial.

Los modelos anteriores con enfoque reduccionista, olvidan las interacciones entre los objetos. La clasificación de la información – base Ontológica - compuesta por los datos y relaciones se aplicó a esta implementación mediante un modelo de base de datos, mediante modelo entidad-relación (tablas y la integridad referencia).

El modelo permite tomar de información de forma holística porque el observador es el mismo que envía la información. Permite tomar información de la afectación de otros usuarios en línea por las acciones de otros. La afectación de su grado de “satisfacción”.

En otros modelos, cuando el punto que observador está centralizado y lejos de fenómeno pierde mucha información y no muchos comportamientos. El modelo distribuido permite interpretar en el origen, analizar e informar en el momento en el que ocurre la demora de un procesamiento central.

4.2.1 Modelo de base de datos

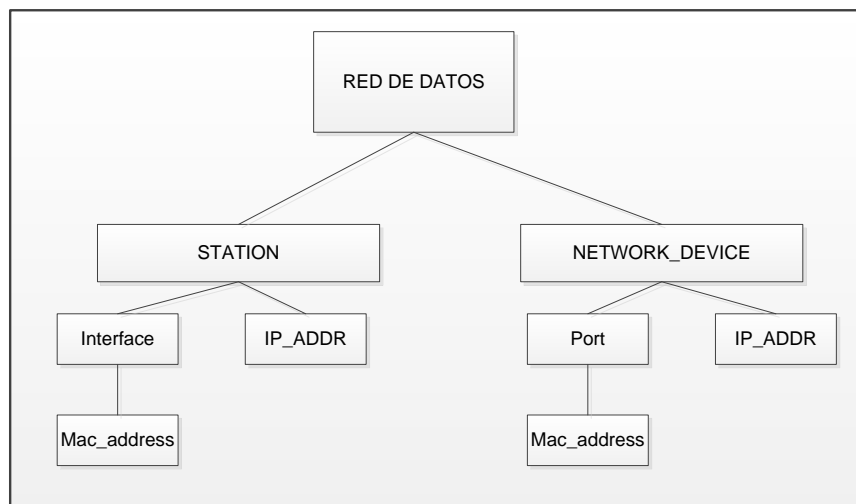
Jerarquía física

El modelo de base de datos diseñado se enfoca a dar una representación a relaciones entre objetos de red y sus estaciones. Se pueden ubicar las variables y fenómenos en su relación con el universo. El modelo permite almacenar información de un conjunto de observaciones (datos tomados por los agentes sensores). Se logra contar con secuencias de uso de tráfico y observación en el tiempo (histogramas, líneas bases).

El diseño de este modelo comienza por los elementos tangibles (objetos) de una red de computación.

- Estaciones STATION
- Equipos de red NETWORK_DEVICE

Ilustración 28. Elementos tangibles de una red.



En una red existen muchos tipos de estaciones como son computadores de escritorio, computadores portátiles, impresoras, *scanners*, servidores, máquinas virtuales, celulares. Es útil conocer las clasificaciones y cantidades de estos dispositivos. Las agrupaciones de los dispositivos pueden ser consultados a través de la tabla T_STATION.

Por otro lado, se tiene varios tipos de equipos de red según la función en la que participan. Se pueden encontrar *switch*, *router*, Balanceadores de carga (*load balancers*), equipos de seguridad como FIREWALL, *IDS*, *IPS* (*intrusión prevention system*), *WAF* (*Web application Firewall*), *APs*, *APs controller*. Dentro de las agrupaciones que son importantes tener

presente se encuentran las que aportan a la capa 2, capa 3, seguridad. Se pueden observar esta relación mediante la consulta a la tabla T_NWD.

Por este camino de análisis se continuó desde lo general a lo particular. El siguiente nivel de detalle, es el modelado para contar de los detalles internos de cada estación y dispositivo de red.

Del lado de las estaciones, se observó los múltiples tipos de interfaces disponibles entre las cuales están las tarjetas de redes fijas, inalámbricas, *bluetooth*, celulares como GSM, 3G, 4G.

- Tipo de interfaz. Ethernet Mbps, ATM, Token ring.
- Velocidad: 10,100, 1000, 10000, 56, 112, 480, 155 Mbps.
- Medio físico.

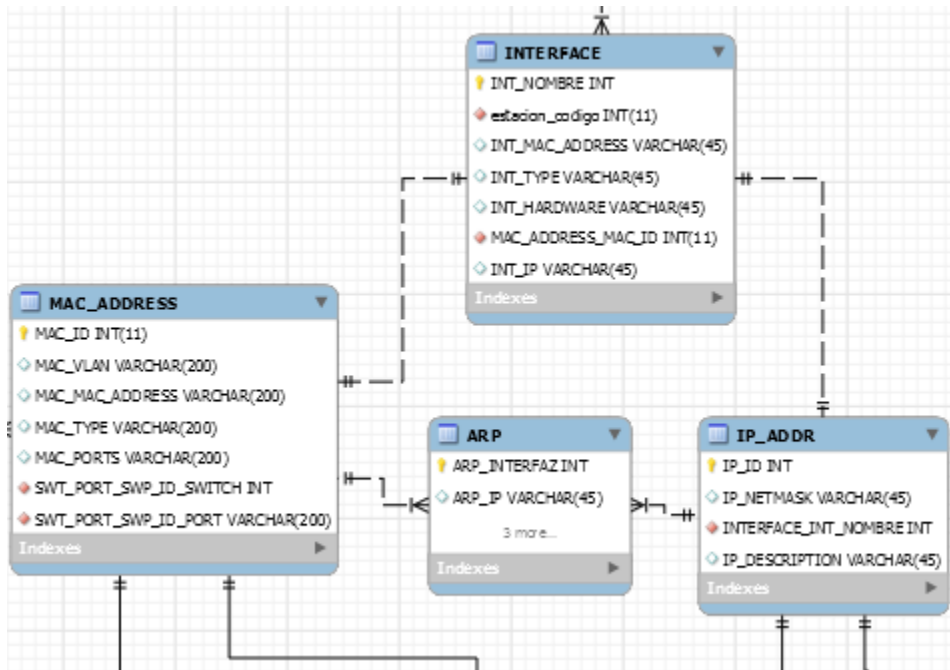
Estas características pueden ser útiles para agrupar fenómenos que aplican al tipo de tecnología de acceso al medio. Se agrupan mediante la tabla INTERFACE. Por el lado de los dispositivos de red se encontraron los puertos físicos e inalámbricos que nos permite llevar al nivel de acceso a la red (capa 2). Se agrupan mediante la tabla SWT_PORT. En este punto, se encontró la relación de conectividad de red.

Con la llegada de los sistemas de virtualización tanto en servidores como en estaciones se modeló la relación entre los puertos e interfaces, las direcciones físicas *mac-address* y sus direcciones IP. Entonces se ven las siguientes relaciones:

- Una interfaz tiene una o más direcciones IP. En la tabla IP_ADDR se registran las direcciones encontradas en el descubrimiento de los sensores.
- Una dirección IP existe en una sola interfaz a la vez de una red. Es un identificador de unicidad.
- No se permiten direcciones IP duplicadas para que existan transmisión de información.
- Es posible encontrar muchas direcciones iguales en muchas interfaces de estaciones con redes privadas.
- El direccionamiento privado también impone un reto para la conexión sobre redes públicas con el uso del NAT (*network address translation*).

- Una interfaz física tiene una dirección *mac-address* fijada por el fabricante de la tarjeta de red.

Ilustración 29. Relación MAC, IP, Interfaz.



- El sistema de virtualización coloca grupos dinámicos de direcciones *mac-address* (pool) para asignarlos a cada interfaz virtual.
- En un puerto de dispositivo de red, se presentan múltiples IP-ADDRESS y múltiples *mac-address*.
- La relación entre interfaz, *mac-address* e IP-ADDRESS tiene sentido solo en intervalos de tiempo. Las direcciones IP son asignadas de manera dinámica por los sistemas de DHCP. De igual forma las direcciones *mac-address* también pueden estar cambiando. Como consecuencia, se registran las relaciones IP- *mac-address* en la hora que estuvieron unidas. Esta relación se pudo observar en la tabla MAC_ADDRESS_has_IP_ADDDR.
- El protocolo de ARP de los enrutadores (nivel 3) tiene que resolver las relaciones entre *mac-address* e IP constantemente. Las estaciones también poseen su tabla de ARP que intercambian entre vecinos. En la tabla denominada ARP se encontró esta relación que permitió aportar en la solución de problemas como inundación de paquetes "flooding", IP duplicadas o bucles de red (*loop*).
- Cualquier pareja de estaciones puede establecer una conexión para transportar información y se establece una conversación bidireccional. Esta conversación tiene

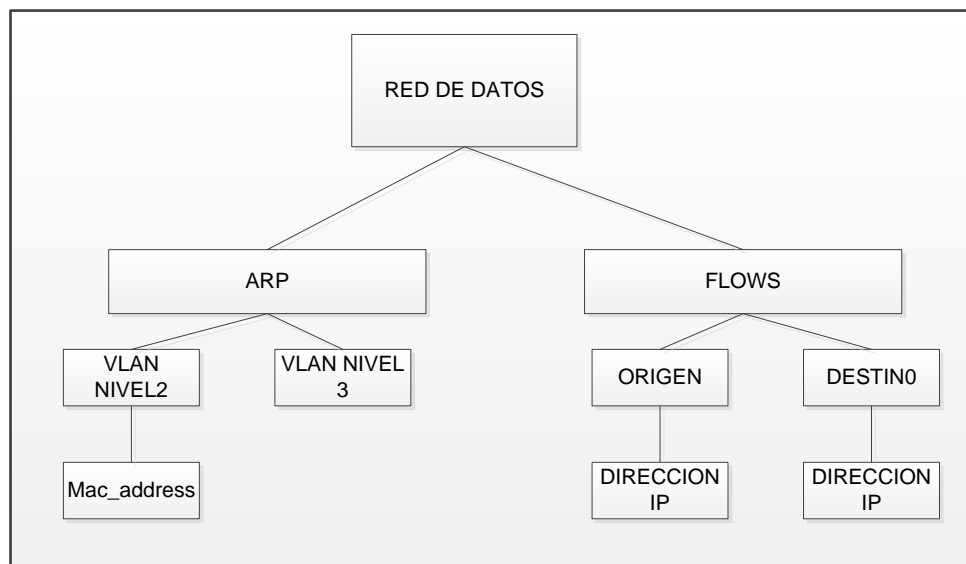
variables interesantes como la cantidad de información transportada, paquetes perdidos, retransmitidos, o demorados. Estos datos se colocan en la tabla FLOWS.

Con la presencia en la actualidad, de los DATACENTER DEFINED BY SOFTWARE, se observó que las interfaces son de dos tipos: Virtuales y reales.

Jerarquía desde la funciones de la red

Conversaciones (Flows). Conversaciones de red entre un origen y un destino. Son creadas de manera dinámica. Existen y desaparecen según las necesidades del usuario. Es útil contar con los *flows* para realizar diagnósticos de tendencias de red cuando hay uso que sale de la línea base, como consecuencia de una moda, un virus o un ataque informático. Esta información relevante de algún fenómeno está en la tabla FLOWS.

Ilustración 30. Relaciones entre tráfico de red y la tabla ARP.



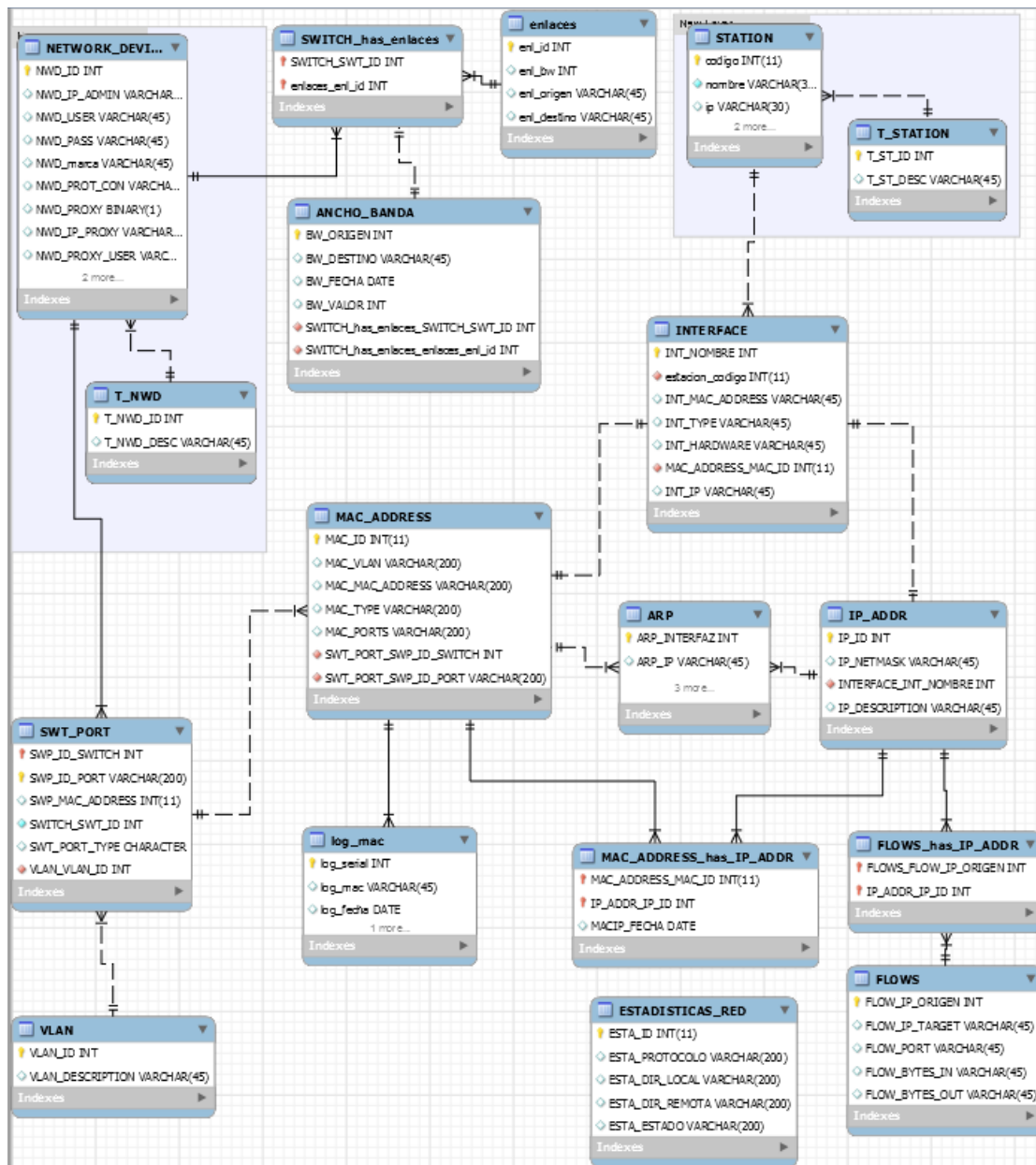
VLAN. Son las divisiones lógicas de la red en conjuntos de estaciones para permitir agrupar los usuarios por grupos funcionales. Es importante que la mayoría del tráfico masivo se quede en estos grupos. Se encuentra esta información en la tabla VLAN y nos permite asociar las estaciones, IP y *mac-address*. Los dispositivos de red que permiten realizar esta función son los *switch* y los correspondientes puertos.

Relaciones en Tablas para el modelo de optimización.

Pruebas de ancho de banda entre A y B. Dado que los agentes sensores inician pruebas entre ellos para medir el ancho de banda a intervalos regulares o según algún estímulo recibido desde otro agente, se coloca la información recogida y la fecha en la tabla ANCHO_BANDA.

A continuación, se presenta el modelo completo de todas las relaciones entre los objetos de la red y estaciones. La estructura presentada en el modelo completo refleja una taxonomía de red y sus relaciones. Es una aproximación a la ontología de una red de datos (Orozco, 2012).

Ilustración 31. Modelo de la base de datos.



4.2.2 Datos estacionales

La estacionalidad del comportamiento del tráfico se enfoca en la implementación con las siguientes reglas:

- Los humanos trabajan de 8 am a 5pm.
- Los días laborables son de lunes a viernes
- Sábado se realiza trabajo de manera ocasional o medio día.

- Los días domingo y festivos poco u rara vez se realizan trabajos. Puede aplicar un tráfico alto en temas de ocio en los ambientes domiciliarios (*Netflix*, juegos electrónicos en línea).
- Las aplicaciones de sistemas de información se usan al máximo en fechas límites o cuando ocurren ciertos cortes para procesos masivos.
- Ocurre alta concurrencia de usuarios por un recurso similar por modas o interés público.
- Cada sistema de información tiene sus periodos de máxima utilización.

4.3 Implementación

La implementación es el resultado de la integración de un conjunto de servicios y productos de software. Está conformado por una plataforma base, una plataforma para gestión, pruebas, generadores de tráfico, un desarrollo de software para crear el sistema multi-agente y los procesos de decisión automáticos. Para el desarrollo de este proyecto se utilizó la arquitectura de Agente reactivo (Ver 1.7.3 Características de los agentes).

La implementación del producto de software requiere varios conjuntos de servidores para alojar los componentes aquí presentados.

4.3.1 Plataforma base

Es un grupo de servidores ubicados en los sitios centrales (proximidad al corazón) de la red, y con conexiones de alta velocidad. Se requieren equipos con características de alta disponibilidad y redundancia.

- a) Servidor web base para presentación de resultados y tablero de control. Permite al usuario administrador del producto realizar pruebas distribuidas de tráfico, observar las gráficas de tendencias, y las gráficas de desempeño. También se pueden ajustar el intervalo de muestreo (ver Granularidad, escala e intervalo de muestreo. 4.1.1). Es factible generar estímulos para ser enviados a la comunidad de agentes para que realicen una tarea a demanda.

- b) Servidores base para transferencias de archivos *sftp*. Se puede contar con varios servidores ubicados en los centros de alta concentración de servidores o en los nodos más importantes de la red. Se considera un nodo importante en la red, aquellos cuyos el tráfico de red que conecta múltiples edificios, concentra un conjunto importante de las conexiones de red de mayor velocidad o que concentra más de 10% del tráfico de red de una organización.
- c) Servidores para contar con los servicios seguridad, autenticación y distribución del software encargados de alcanzar a cada una de las estaciones, realizar la instalación de los pre-requisitos como el JRE (*Java Runtime Environment*), del software del sistema multi-agente, y mantener la ejecución del agente. Se utilizó el sistema de *Microsoft Windows Active Directory* como controlador general que automatiza estas tareas.
- d) Sistema de registro de agentes Jade mediante el cual se mantiene la lista de cada agente desplegado y en ejecución.

4.3.2 Un servidor de base de datos para guardar los datos de tendencias, muestreos de información, según la ontología de redes (Ver 4.1.5 Captura de flujos)

Cuando se establece una interacción en una red, es importante entender todo el diálogo entre las partes para establecer comportamientos de la capa 6 y 7 del modelo OSI. Se define conversación como el tráfico intercambiado entre dos puntos específicos de forma bi-direccional.

Netflow. Captura de información de trazas de información mediante inspección rápida de cabeceras IP utilizado y patentado por Cisco RFC3954 (Claise, 2004).

Sflow.: Captura de trazas de información con muestreo. No necesariamente analiza todo un flujo de datos. Fue impulsado por los fabricantes de redes como alternativa a *Netflow* de Cisco RFC3176 (Phaal, Panchen, & McKee, 2001).

Ilustración 26. Conversaciones de red.

Conversations: http.pcap

Ethernet: 1 | Fibre Channel | FDDI | IPv4: 15 | IPX | JXTA | NCP | RSVP | SCTP | TCP: 38 | Token Ring | UDP: 1 | USB | WLAN

TCP Conversations - Filter: http or dns

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B
10.211.55.3	60790	193.69.165.21	http	126	72615	63	28028	63
10.211.55.3	60789	193.69.165.21	http	104	64791	52	23330	52
10.211.55.3	41144	128.121.50.122	http	18	9991	9	6794	9
10.211.55.3	41145	128.121.50.122	http	18	8994	9	6885	9
10.211.55.3	45168	193.69.165.57	http	4	3336	2	970	2
10.211.55.3	51236	81.93.163.17	http	2	1876	1	541	1
10.211.55.3	45169	193.69.165.5	http	2	1959	1	1959	1
10.211.55.3	50948	81.7.166.249	http	2	1967	1	1967	1
10.211.55.3	41632	194.237.107.	http	2	1959	1	1959	1
10.211.55.3	51305	193.69.165.2	http	2	1959	1	1959	1
10.211.55.3	58101	80.76.145.17	http	2	1959	1	1959	1
10.211.55.3	41623	209.85.141.99	http	4	1967	1	1967	1
10.211.55.3	43305	62.70.11.43	http	2	1959	1	1959	1
10.211.55.3	51291	193.69.165.29	http	2	1876	1	541	1

Apply as Filter: Selected

Prepare a Filter: Not Selected

Find Packet: ... and Selected

Colorize Conversation: ... or Selected

... and not Selected

... or not Selected

A <-> B

A --> B

A <- B

A <-> ANY

A --> ANY

A <- ANY

ANY <-> B

ANY <- B

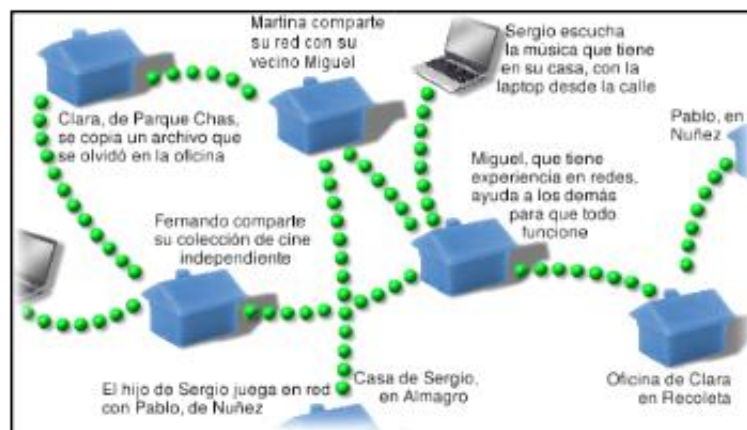
ANY --> B

Name resolution Limit to display filter

Help Copy Close

Mediante un sistema de captura de la información de las conversaciones de red se logran establecer análisis de alto nivel. Si se requiere conocer cual protocolo o servicio consume mayor ancho de banda se requieren sistemas como *netflow* o *sflow*. Los análisis de tráfico de alto nivel nos permiten observar la multiplicidad de intereses o necesidades de los usuarios.

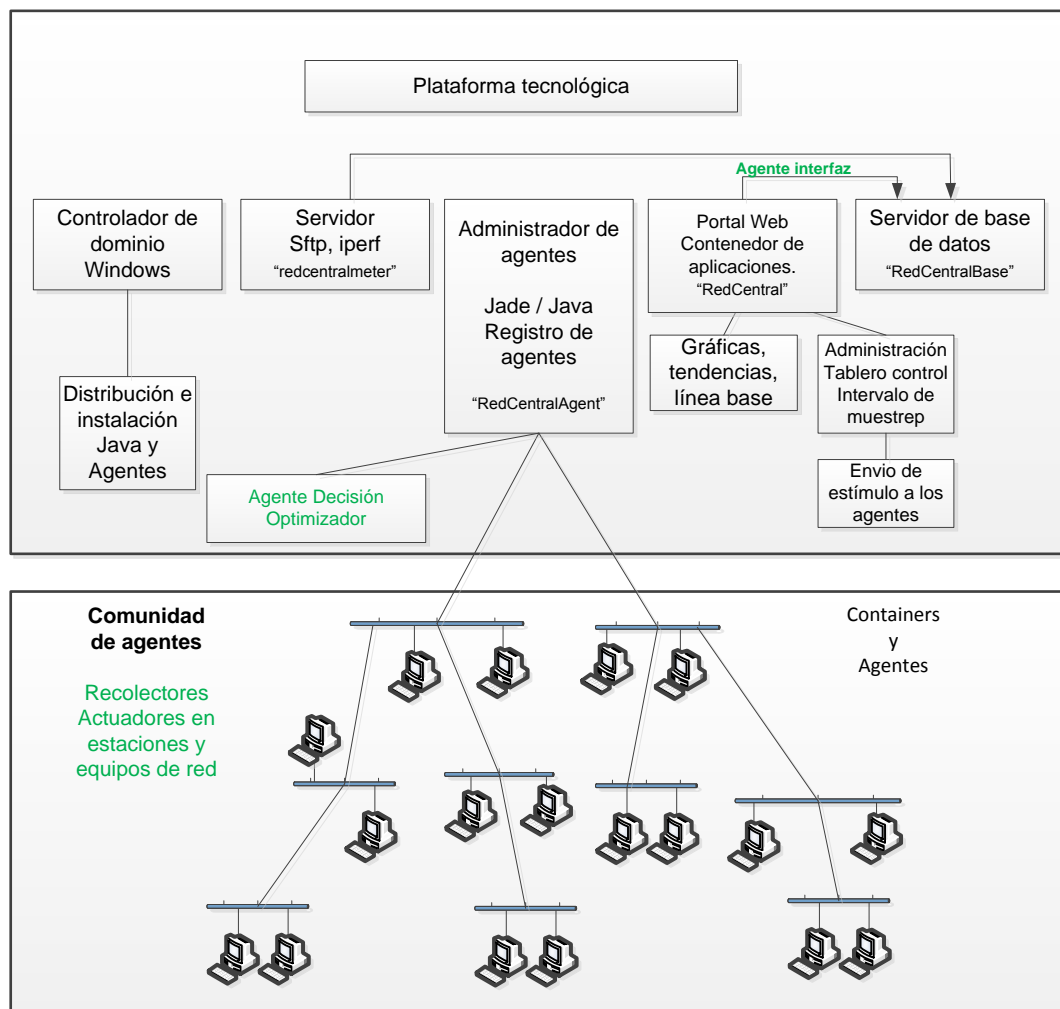
Ilustración 27. Ejemplo de múltiples conversaciones de red.



Es importante anotar, que las capturas de las conversaciones (*flows*) están en la capa 7 del modelo OSI. Una conversación involucra miles de tramas y paquetes IP. Las conversaciones comienzan con el inicio de sesión hasta la respuesta con la información solicitada.

- e) Abstracción de datos y relaciones).
- f) Un servidor para realizar las pruebas de envío y recepción de las ráfagas de tráfico que nos permite medir el ancho de banda libre en todo un camino de red (*path*).

Ilustración 32. Plataforma física para el sistema multi agente.



El desarrollo de software se dividió en dos productos. Uno enfocado a todo el trabajo del sistema multi-agente. El segundo enfocado a la presentación de resultados mediante una interfaz web. Se presenta los nombres de los productos así:

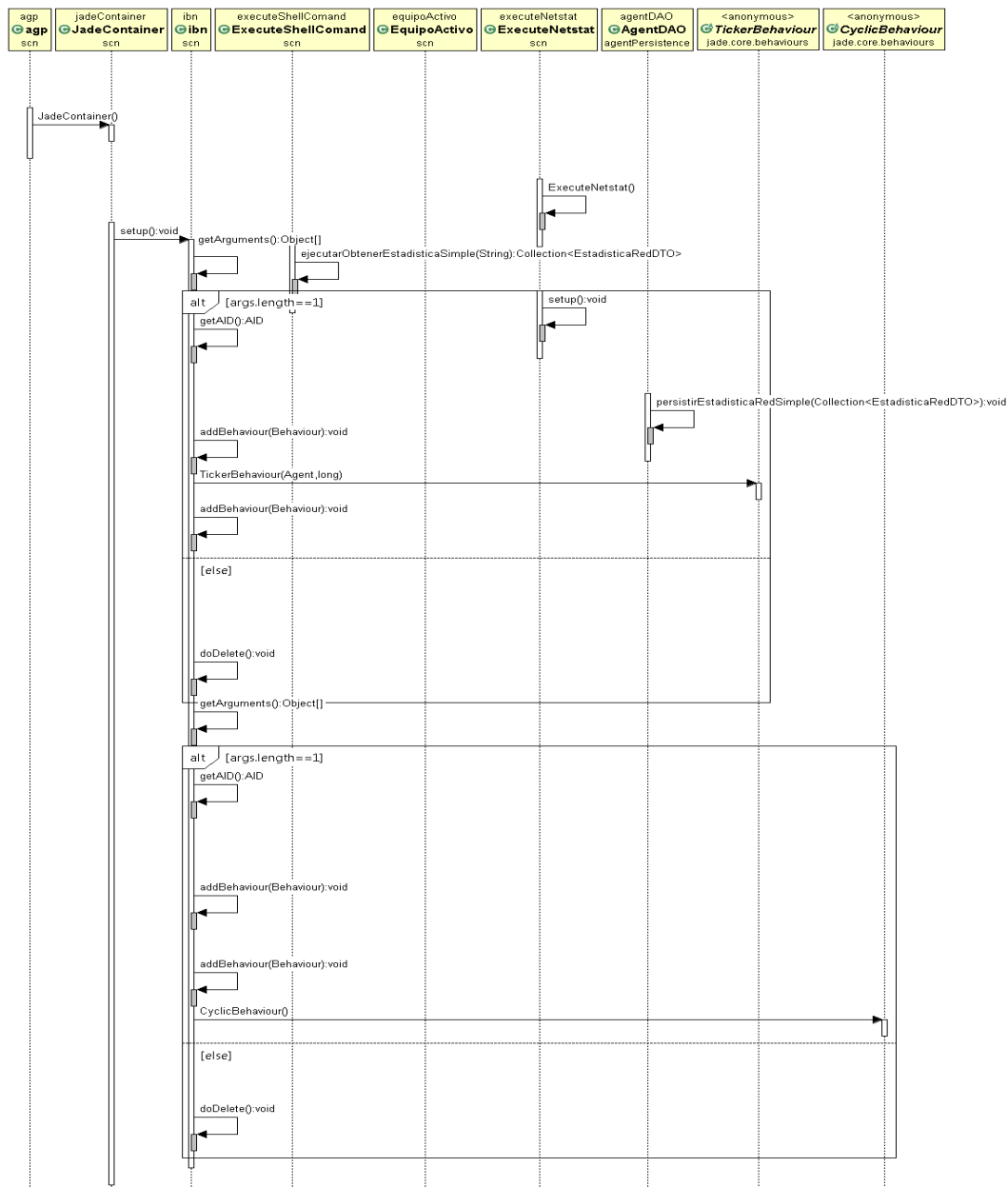
RedCentralModel	
<i>RedCentralAgent</i>	<i>RedCentral</i>
Todos los agentes	Portal de presentación
Base de datos <i>RedCentralBase</i> en mysql	
Trabajo silencioso en muchas estaciones y dispositivos de red	Gráficas, tendencias, altamente visual.

Tabla 4. Productos de software.

4.3.3 Diagrama de secuencia.

El diagrama de secuencia nos presente la interacción, utilización de cada una de las clases en el modelo. De igual forma, los comportamientos de los agentes implementan clases.

Ilustración 33. Diagrama de secuencia.



4.3.4 Agente recolector y de análisis sintáctico.

Agente que permite la recolección de información mediante la ejecución de comandos dentro del sistema operativo de cada máquina en el cual se ejecuta el código.

La recolección funciona en conjunto con el módulo de estandarización de información mediante un análisis sintáctico para escoger la información útil. Se utilizó analizadores

sintácticos para interpretar la información que entregan los comandos del sistema operativo (Alfred V. Aho Monica S. Lam, 2006).

Ejecución de tareas en segundo plano, como ejemplo se muestra la ejecución de un “ping”,

Ilustración 34. Ejecución de comando en el sistema operativo.

```
Haciendo ping a JuanCarlosRiver [127.0.0.1] con 32 bytes de datos:
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128
Respuesta desde 127.0.0.1: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 127.0.0.1:
  Paquetes: enviados = 4, recibidos = 4, perdidos = 0
  (0% perdidos),
  Tiempos aproximados de ida y vuelta en milisegundos:
  Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Las gramáticas de los comandos responden a:

Comando :: <Titulo> <línea de respuesta> > <Resumen final>

Línea :: <Valor1> <separador> <Valor 2>

Los analizadores sintácticos permitieron encontrar los valores valiosos requeridos para este trabajo, separar los separadores de las líneas y los nombres de las variables.

Análisis sintáctico de las respuestas obtenidas por los Agentes.

En los agentes encargados de recolectar la información tales como el *Arp*, *Ip Config*, *NetStat*, *Ping* se realizó un análisis sintáctico lineal de acuerdo a la salida obtenida por la ejecución de los comandos debido a que la ejecución de estos comandos genera tramas planas de resultados.

Por otra parte el agente *TimeMeterSftpAgent* permite realizar el análisis de la información por medio del envío del archivo desde un cliente a un servidor, por lo tanto no fue necesario análisis sintáctico sobre la información.

Mientras que el agente `BandWidthMeterAgent` por medio de `ipertf3` permite la generación de archivos JSON con el resultado de la ejecución de los comandos, por lo tanto sobre estos archivos se realizó el análisis sintáctico `<key, value>` obteniendo la información necesaria para el análisis de gestión de la red.

4.3.5 Agente de Consolidación de datos

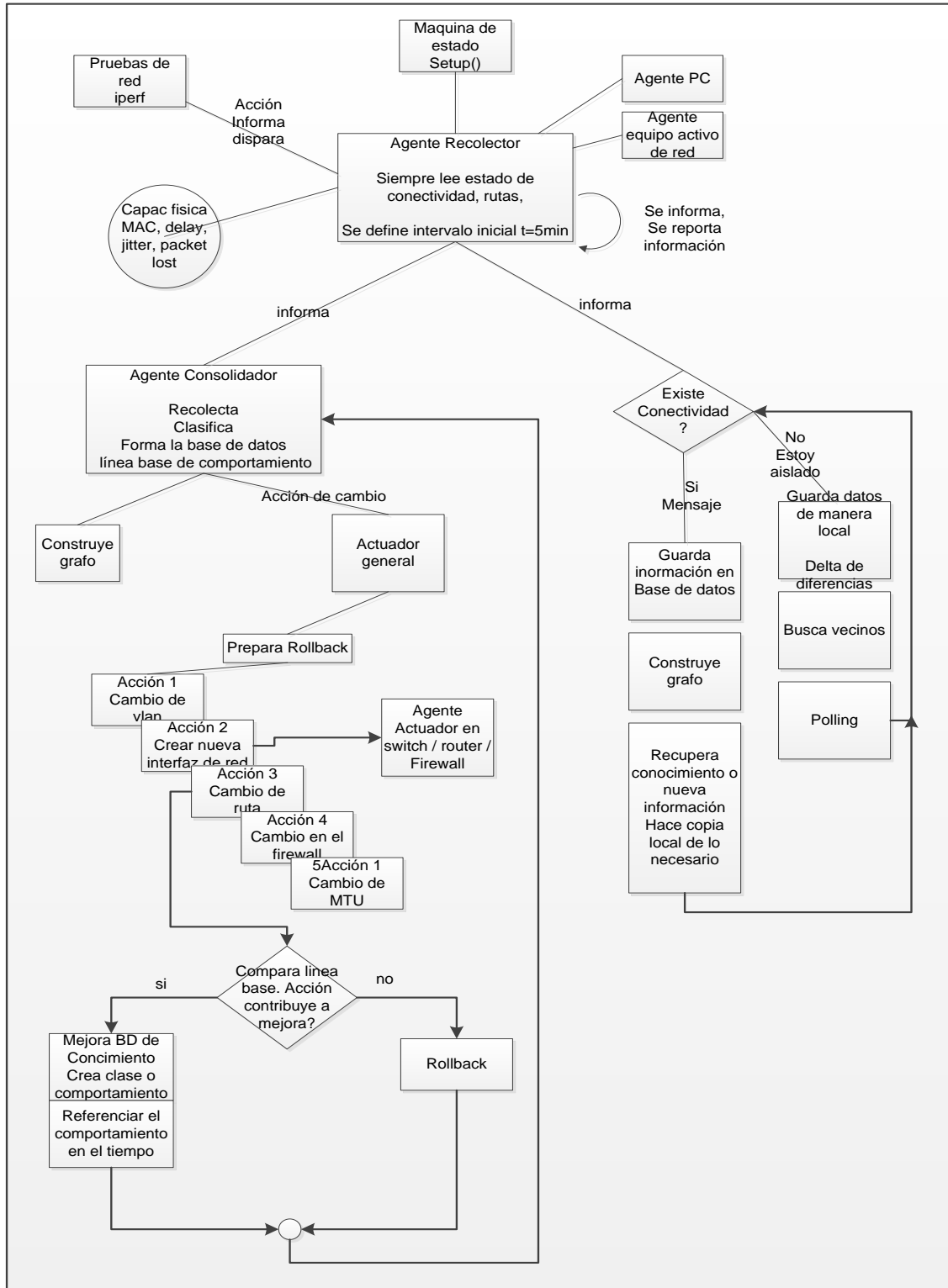
El agente de consolidación permite la recepción de la información desde los clientes y almacenarla en una base de datos relacional. Se escogió `MySql` dentro de un servidor Linux Centos 6.5.

4.3.6 Agente actuador.

Realiza la función principal de Auto-configurador. El agente “actuador” permite modificar los equipos activos de la red mediante comandos propios de la marca de los equipos. En el caso de la UNAL son de marca CISCO. Se puede contar con un manual de sinónimos de comandos para ser ejecutados en equipos de diferentes marcas.

Se emplearon las librerías del lenguaje “`expect`” que permite el acceso a equipos activos de red que tiene una línea de comandos interactiva. “`Expect`” nos permite iniciar un diálogo desde el lenguaje “Java” y capturar la información. Nos permite simular la interacción humana interactuando con una línea interactiva de comandos.

Ilustración 35. Máquina de estado del software.



Existe un agente actuador diferente para una estación y para un equipo de red, dado que el sistema operativo es diferente. Una función que debe implementar el actuador antes de

realizar un cambio, es contar con la información completa para regresar al estado anterior. El actuador debe mantener una copia del histórico de configuraciones de los dispositivos de red y tener un plan denominado “*backtracking*” o “*rollback*”.

Finalmente, el actuador ejecuta el comportamiento de optimizador utilizando una ecuación. Se eligió una ecuación lineal con una sola variable (ver 2.2 La optimización).

4.3.7 Probador de ancho de banda y aplicación

Se utiliza el software “iperf3” desarrollado por *Lawrence Berkeley National Laboratory* para realizar pruebas activas sobre los enlaces de comunicaciones sobre redes activas. Permite realizar las pruebas de ancho de banda en ambos sentidos. Se puede medir ancho de banda, *MSS (Maximum Segment Size)*, *MTU*, tamaño de la ventana TCP. De igual forma se puede realizar pruebas en UDP y TCP. Dado que es multi-plataforma se pueden hacer pruebas cruzadas entre un sistema Unix y un sistema Windows. La forma de utilizar la herramienta es ejecutando en una máquina un ambiente cliente y en otra un servidor (“Iperf,” 2017).

Ilustración 36. Ejemplo de funcionamiento de *iperf*.

```

Simbolo del sistema
iperf Done.
c:\iperf3>iperf3 -c 192.168.189.1 -R
Connecting to host 192.168.189.1, port 5201
Reverse mode, remote host 192.168.189.1 is sending
[ 4] local 192.168.189.1 port 49397 connected to 192.168.189.1 port 5201
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-1.01  sec  768 MBytes  6.36 Gbits/sec
[ 4] 1.01-2.01  sec  758 MBytes  6.37 Gbits/sec
[ 4] 2.01-3.01  sec  750 MBytes  6.30 Gbits/sec
[ 4] 3.01-4.01  sec  766 MBytes  6.43 Gbits/sec
[ 4] 4.01-5.01  sec  772 MBytes  6.49 Gbits/sec
[ 4] 5.01-6.01  sec  764 MBytes  6.42 Gbits/sec
[ 4] 6.01-7.00  sec  769 MBytes  6.45 Gbits/sec
[ 4] 7.00-8.00  sec  772 MBytes  6.49 Gbits/sec
[ 4] 8.00-9.00  sec  776 MBytes  6.52 Gbits/sec
[ 4] 9.00-10.02 sec  786 MBytes  6.50 Gbits/sec
-----
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.00-10.02  sec  7.50 GBytes  6.43 Gbits/sec      sender
[ 4] 0.00-10.02  sec  0.00 Bytes   0.00 bits/sec      receiver
iperf Done.
c:\iperf3>iperf3 -c 192.168.189.1 -R

Simbolo del sistema - iperf3 -s
Server listening on 5201
-----
Accepted connection from 192.168.189.1, port 49396
[ 5] local 192.168.189.1 port 5201 connected to 192.168.189.1 port 49397
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-1.01  sec  759 MBytes  6.28 Gbits/sec
[ 5] 1.01-2.01  sec  758 MBytes  6.37 Gbits/sec
[ 5] 2.01-3.01  sec  749 MBytes  6.29 Gbits/sec
[ 5] 3.01-4.01  sec  766 MBytes  6.44 Gbits/sec
[ 5] 4.01-5.01  sec  771 MBytes  6.48 Gbits/sec
[ 5] 5.01-6.01  sec  764 MBytes  6.42 Gbits/sec
[ 5] 6.01-7.00  sec  769 MBytes  6.46 Gbits/sec
[ 5] 7.00-8.00  sec  771 MBytes  6.48 Gbits/sec
[ 5] 8.00-9.00  sec  776 MBytes  6.52 Gbits/sec
[ 5] 9.00-10.02 sec  786 MBytes  6.51 Gbits/sec
[ 5] 10.02-10.03 sec  10.0 MBytes  5.38 Gbits/sec
-----
[ ID] Interval      Transfer    Bandwidth
[ 5] 0.00-10.03  sec  7.50 GBytes  6.42 Gbits/sec      sender
[ 5] 0.00-10.03  sec  0.00 Bytes   0.00 bits/sec      receiver
-----
Server listening on 5201

```

4.3.8 Integración de productos de software

Para la implementación del software se utilizaron varios grupos de bibliotecas *Open Source* compatibles con Java y por ende con Jade. En los anexos, se puede ver detalles de la instalación y uso de estas herramientas.

Tabla 5. Integración de productos de software.

Producto	Versión	Descripción
Lenguaje programación	Java	Lenguaje de programación
IDE Eclipse	Luna 4.4.2	IDE. Ambiente de desarrollo de software.
Multi-Agent System	Jade 4.1.0	Biblioteca de Sistema Multi Agente.
MySQL	5.1.71	Base de datos relacional sobre Linux.
JFreeChart	1.0.19	Biblioteca para 88realizer gráficas.
Jperf	2.0.0	Biblioteca para realizar pruebas de ancho de banda.
Iperf	3.1	Herramienta para realizar pruebas de ancho de banda.
JgraphT	0.9.1	Biblioteca para 88realizer gráficas.
ObjectAid	1.1.2	UML Diagramas de secuencia http://www.objectaid.com/sequence-diagram
JsCh	0.1.38	Biblioteca de funciones para implementación de transferencias ssh (tcp22) http://www.icraft.com/jsch/ https://sourceforge.net/projects/jsch/
Apache httpd	2.2.15-59.	http://httpd.apache.org/docs/2.4/install.html#download
J2EE	1.4 (Servlet 2.4, Jsp 2.0)	Arquitectura del portal del servidor.
ExpectForJava	1.2.17b	Biblioteca para interfaz interactiva de línea de comandos.

4.4 Validación del modelo de auto-configuración

Escenario 1.

La cantidad de información de tráfico que se queda local en una VLAN.

Escenario 2.

La cantidad de información de tráfico que cruza por los equipos de nivel 3 (enrutadores).

4.5 Estadísticas y análisis

Usuarios por VLAN.

Se realiza la ejecución del software sobre una red con un grupo de usuarios considerable. En este sentido significa que la cantidad de tráfico transportado entre el grupo de usuarios supere el “*backplane*” de un *switch* o de los enlaces de interconexión. La red de la Universidad Nacional de Colombia es un ejemplo útil para probar el modelo.

Se pudo observar que se tienen 600 *switch* de 24 puertos que nos da un máximo teórico de 14.000 equipos en la red fija (“Licitación CON-BOG-007-2016 Universidad Nacional de Colombia,” 2016).

Por parte de la red inalámbrica se cuenta con 500 antenas (APs) cada uno con posibilidad de atender 80 usuarios para comunidad total de 24.000. La capacidad del canal de acceso a INTERNET es de 1.8 Gigabits/segundo. Los enlaces entre los equipos que conforman el corazón de la red son de 10 Gigabits/seg. Los enlaces de segundo nivel entre los equipos de distribución y acceso hacia los diferentes edificios del campus tienen una capacidad de 1 Gigabit/seg.

Este escenario es adecuado para que el software *RedCentral* realice el cálculo de la cantidad de VLANs que deberían configurarse y cuáles son los miembros de cada uno de los segmentos.

Intervalo de muestreo de las iteraciones (1 hora).

Iteración (hora de día)	Cantidad VLAN (antes)	Usuarios x Vlan	Tráfico local (Mbits/s)	Cantidad de VLAN (después)	Cantidad de Vlan	Tráfico local (Mbits/s)
8	60	1024	100	200	200	5000
10	70	1024	100	200	200	5000
12	80	1024	100	200	200	5000
4	60		80			

8	40					
10	20	4000	100			

El tráfico local después de la planeación de cambios propuesta es la suma del tráfico de las estaciones que se encuentran en la misma VLAN.

Resultados del Experimento 1.

Los resultados en el laboratorio de tres *switches*, tres enrutadores y cuatro estaciones. Se comprueba la funcionalidad y el desarrollo del producto. Los tiempos de respuesta de todos los experimentos son perfectos, dado que no existe tráfico, se están en un ambiente controlado. El grado de utilización de procesadores y RAM porque no se logra producir suficiente *stress* en los *switch* de laboratorio.

Resultados del Experimento 2.

Tiempo de recolección de datos accediendo a un equipo de red.

La recolección de datos por parte del agente utiliza recursos de red y es una operación que participa en la congestión de la red; por tanto, también su tiempo de respuesta cambia según el tiempo.

Hora de muestreo 8:00am

Tamaño de consulta de un registro de datos KB	Tiempo (ms)	Número de hilos
1	1	1
10	4	1
100	30	1

Hora de muestreo 8:00am

Tamaño de consulta de un registro de datos KB	Tiempo (ms)	Número de hilos Equipos de red simultáneos
1	1	5
10	4	5
100	30	5

Eficiencia del proceso de recolección y “análizador sintáctico”.

En la captura de datos se utilizan comandos del sistema operativo o sistemas externos. La forma en la cual los datos están son entregados por el sistema operativo están formateados mediante encabezados, separadores, totalizadores. Algunos de los datos de la salida no son relevantes para la gestión de red como caracteres separadores de columnas, líneas en blanco, repeticiones de los nombres de las variables. Este valor mide la eficiencia del proceso dentro de una estación. En la actualidad se usaron máquinas con características de 8Gigas de RAM, 1 procesador de un procesador con 4 núcleos, discos magnéticos de 1 Tera con al menos el 50% del disco disponible. El agente recolector no utiliza espacio en disco pero si una estación no posee espacio en disco tendrá problemas para ejecutar cualquier proceso.

Bytes capturados en bruto (KB)	Bytes útiles Bytes	Análisis sintáctico ms ó ciclos (ms)
10	10	4
20	13	5
30	14	5
100	10k	20

Tiempo de respuesta por cantidad de agentes grabando información en la base de datos.

Se registró el tiempo de respuesta en la grabación de la información en el agente de consolidación. Según el tráfico, el comportamiento generalmente está sometido al tráfico. La hora del día muestra la estacionalidad y afecta el tiempo de respuesta.

Hora de grabación 8:00 am.

Cantidad de agentes	Tiempo ms para Registro (1k)	Tiempo en ms para Registro (10K)	%CPU Consolidador
1	1	7	1
5	1	8	1
10	1	10	2
20	1	11	2

30	1	11	2,5
40	1	12	2,5
100	2	14	4

Tiempo de respuesta por agente sftp.

Se construyó un agente que se despliega desde las estaciones para realizar la transferencia de un archivo. Este agente mide el tamaño del archivo, el tiempo que se demora en ser transferido y la estampilla de tiempo. Se guarda la información mediante el agente de consolidación en la base de datos.

Dirección IP del cliente.	Tamaño archivo (Bytes)	Fecha y hora	Tiempo transferencia (milisegundos)
172.27.35.115	1767249	2017-05-13 22:25:09	1678
172.27.35.115	24326827	2017-05-14 07:14:36	2168
172.27.35.115	25405440	2017-05-14 08:08:07	2227
172.27.35.115	515000007	2017-05-14 16:27:43	47004

Tiempo de respuesta por agente para Jitter.

Se construyó un agente que se despliega desde las estaciones para realizar la ejecución de una trama de tráfico UDP mediante el generador iperf3. Este agente mide el ancho de banda hasta el corazón de la red, el tiempo que se demora en ser transferido y la estampilla de tiempo. Se guarda la información mediante el agente de consolidación en la base de datos.

Dirección IP del cliente.	Jitter (Mili segundos)	Fecha y hora	Ancho de banda en (Mega bytes)
172.27.35.115	0.001	2017-05-13 22:25:09	628
172.27.35.115	0,003	2017-05-14 07:14:36	630
172.27.35.115	0.004	2017-05-14 08:08:07	450
172.27.35.115	0,8	2017-05-14 16:27:43	200

5. Conclusiones

Por medio del desarrollo de este sistema multi-agente se encontró que es viable su implementación utilizando Jade con Java. Se encontró que con los agentes se puede llegar a los bordes de la red y realizar análisis y pruebas. Cuando se conoce la arquitectura y se tiene acceso al código de este sistema de gestión se tiene la certeza de poder fabricar más código o comportamiento a algún agente para entender, recopilar algún otro fenómeno. Como consecuencia de estar desarrollado en Java se tiene acceso a miles de clases, desarrollo adelantados por otros equipos de desarrollo de software, reutilizar código y aumentar la riqueza de las funciones del software. Una vez vencidos los obstáculos de contar con la infraestructura, es mucho fácil modelar otras soluciones de gestión de redes de computación.

En el proceso de decisión que nos conduce a la optimización de una ecuación se encontró que se puede contar con un conjunto de metas disponible y el administrador de red solo tiene que seleccionar que se requiere en un momento determinado. Las ecuaciones multi-objetivo pueden ser calculadas para realizar optimizaciones con más complejas.

Las pruebas del sistema pueden ser realizadas en un ambiente de producción. Dado que pueden existir redes de computación que puede ser muy complejas, este producto de software puede asistir al administrador en diagnósticos avanzados. Algunos administradores de red prefiere que el software sea un asistente que le entrega una plan de trabajo y que lo guie en las mejoras de la red, pero sin utilizar el sistema auto-configuración. Aún existe cierta prevención de entregar el control total a una herramienta de inteligencia artificial.

Respecto a los sistemas se encontró que es necesario contar con una herramienta gráfica que apoye observar el cumplimiento de los objetivos. Además, se encontró con el producto que se pueden realizar análisis estadísticos tanto para el administrador como herramientas de apoyo para que los usuarios finales que tienen un problema realicen pruebas específicas que apoye al administrador de la red. Para el administrador acercarse al borde

de la red es difícil o desplegar una solución es costosa en tiempo y recursos. El sistema multi-agente y esta plataforma de software permiten a realización de pruebas a demanda. En la actualidad se cuenta con acceso a muchas herramientas de virtualización. La herramienta también puede realizar su trabajo en estos ambientes heterogéneos porque los agentes pueden estar omni-presentes. Una vez se cuenta con esta plataforma se pueden realizar ejercicios desafiantes como la utilización de agentes móviles que usen a los agentes base para entrar a cualquier lugar.

Algunas funciones de este sistema requieren contar con elevación de privilegios de administrador. Se recomienda a las organizaciones un buen manejo de estos permisos para poder desarrollar esta implementación.

La ventaja de este producto es brinda una visión de la red en tiempo real con un costo computación razonable. Por las ventajas de los sistemas distribuidos se reduce la complejidad y el punto de coordinación no debe tener recursos de computación ilimitados.

Será necesario trabajar más de cerca con los humanos para que logren encontrar confianza en la optimización o ayuden a construirla. De esta forma permitirán que se le cada vez más control a la inteligencia artificial.

5.1 Recomendaciones

- Para el óptimo funcionamiento e interpretación de la información es necesario contar con un sistema que sincronice los relojes de todos los elementos de red y de las estaciones.
- Se aconseja emplear en el desarrollo de software la metodología de código limpio para que el software pueda ser extensible y reutilizado por otros grupos de desarrollo.
- En todas las partes del código hacer referencias a nombres simbólicos. Nunca dejar fijo en el código direcciones IP.
- Desarrollar y utilizar las bibliotecas de Software libre.

- Contar con un sistema para el laboratorio de voz IP para realizar pruebas y modelar la óptima aplicación de los parámetros de calidad de servicios QoS.
- Manejo de los idiomas en los sistemas operativos por sus modificaciones en los encabezados de las rutinas de captura de datos.
- Diseñar o implementar un lenguaje de red para realizar preguntas al sistema de gestión para generar resultados.

5.2 Alcances y limitaciones

Existen varias limitaciones en esta implementación dado que no se diseñó un sistema de alta disponibilidad para el punto central de registro de los agentes, así como las estaciones centrales para realizar las pruebas de transferencia de archivos o medición de ancho de banda. Jade cuenta con un modelo de alta disponibilidad HA para el contenedor principal.

El tamaño en bytes del software crecerá en cuatro veces si se incluye dentro de todos los agentes el sistema central de Jade. Se puede tener un ligero impacto en los discos duros de los clientes, una vez que se pierda contacto o que la red entre en estado de partición. Será necesario implementar un algoritmo de Quorum para que se pueda elegir una nueva estación de coordinación de manera automática.

5.3 Trabajo futuro

Se sugiere trabajar en el futuro en aumentar la profundidad y detalle de la información recolectada. En el campo de análisis de datos se puede avanzar mucho. Es posible contar con los agentes en un esquema de captura de paquetes distribuido (*sniffer* masivo). De esta forma se puede fabricar un *sniffer* liviano.

Los agentes tienen mucho poder y control sobre las máquinas. La información que transporta puede llegar a ser muy importante. Se hace necesario que los agentes tengan un esquema para ser autenticados y admitidos en el sistema central para evitar alguna filtración de código malicioso.

El transporte entre agentes debe ser cifrado para evitar ataques de hombre en el medio que afecten la seguridad informática.

Se puede construir agentes construidos en otros lenguajes de programación o plataforma como .NET, C# para que interactúe con JADE. En ambientes de Microsoft Windows existen partes del sistema que solo se deben acceder con la herramienta más nativa o próxima al sistema operativo.

Se puede implementar sistemas de optimización más ambiciosos y se propone iniciar pruebas para el modelo "*Nash equilibrium*".

Si se quiere aumentar el control por parte la inteligencia artificial y evolucionar los agentes a un modelo deliberativo, se debe conformar una ontología que defina las redes de datos. Se puede implementar un modelo BDI.

En el cambio de Optimización de la infraestructura se puede avanzar en:

- Manejo de excepciones y reconexión de un agente cuando exista una falla de conectividad en la red.
- Utilización de un modelo de *machine-learning*, *data mining* para refinamiento de solución y encontrar comportamientos emergentes, de auto-configuración de redes, que le permitan a la herramienta construir su propio código.

En el cambio de mejoras en la visualización gráfica, existen los siguientes caminos:

- Portales de lanzamiento de agentes.
- Mostrar las conversaciones entre agentes.
- Modelar un sistema gráfico que muestre los grafos de saltos de una conversación de red (flujo) de una base de datos recolectados.

A. Anexo: Código del producto de software.

El software de este trabajo se puede encontrar en:

1. Ver el repositorio de archivos en Github <https://github.com/jcriveraz/Redcentral>
2. DVD entregado con el libro de tesis.

B. Anexo: Manual de instalación del software.

Instalación de la consola central de Agentes.

Colocar y asignar la dirección IP. Se recomienda que el puerto TCP7778 se encuentre abierto en el servidor y en el firewall para que los agentes se puedan registrar.

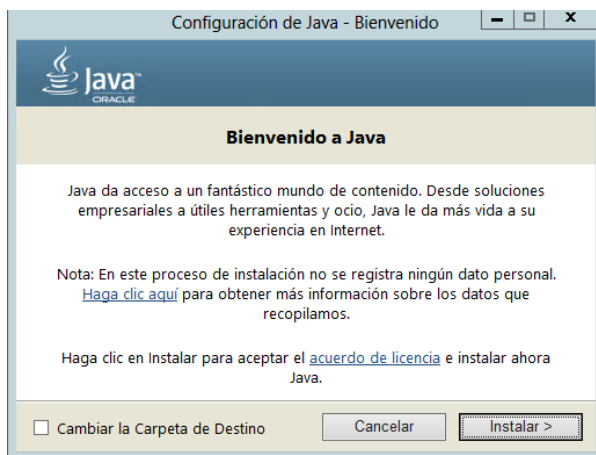
Se recomienda que el portal web, la base de datos y la consola central de agentes se encuentren en la misma subred (VLAN).

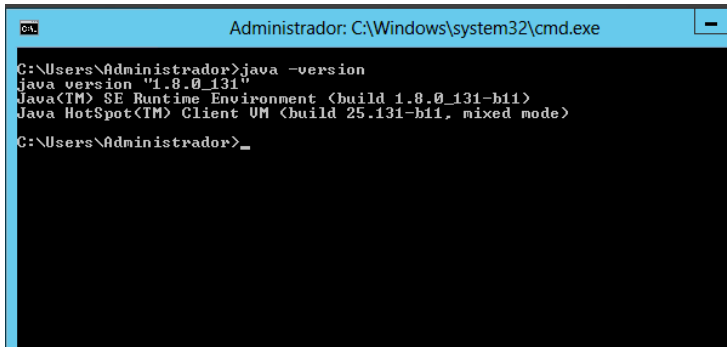
1. Verificar que se encuentre instalado Java. De lo contrario, instalar Java.

Navegar www.java.com

Desactive el Internet explorer security (IE ESC)

Administrador del servidor -> Servidor local -> Propiedades -> Configuración de seguridad mejorada de IE -> on -> Desactivar para administrador y usuarios.





```

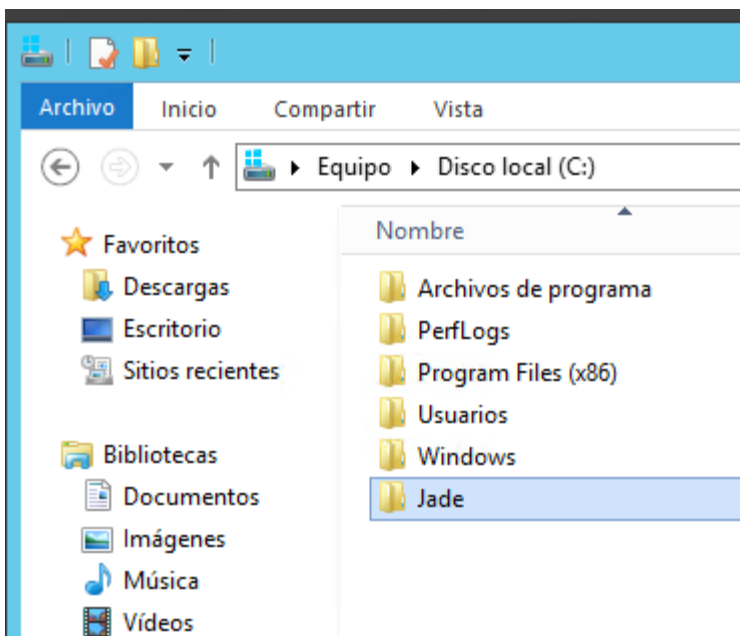
Administrador: C:\Windows\system32\cmd.exe

C:\Users\Administrador>java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) Client VM (build 25.131-b11, mixed mode)

C:\Users\Administrador>_

```

2. Copiar el grupo de librerías en el directorio raíz del sistema Windows.
3. Colocar la carpeta de agentes en la carpeta documentos. De esta forma el usuario con el cual despliegue la aplicación tendrá permisos de ejecución.
4. Verifique la instalación de Java.
5. Copiar Java y colocarlo en a raíz c:\



6. Ejecución del contenedor. Copie las clases compiladas como el `agp.class` que están en su proyecto. En especial toda la carpeta `scn`
 Ventana de comandos "cmd"
 Copy [\\slnxrepo\GrupoSYS\MaestriaJCR\scn](http://slnxrepo\GrupoSYS\MaestriaJCR\scn) c:\document\documents
7. El "starting jade" <http://jade.tilab.com/doc/tutorials/JADEAdmin/startJade.html> debe encontrar la biblioteca *Jade* y la carpeta del Proyecto en este caso "scn"

Coloque estas rutas en la variable “classpath” del sistema operativo ó ejecute así:

- Ventana de comandos “cmd”
cd “Documentos”
java -cp c:\jade\lib\jade.jar;. scn.agp (el “punto” en este path es importante).
- java -cp c:\jade\lib\jade.jar jade.Boot –gui
- Inicio -> Equipo -> Propiedades -> Configuración Avanzada del sistema -> Variables de entorno -> Variables del sistema -> Nueva -> Nombre de variable : classpath Valor de la variable: .;c:\jade\lib\jade.jar;
start java scn.agp

Bibliografía

- 802.1AB-2009 IEEE Standard for Local and Metropolitan Area Networks -- Station and Media Access Control Connectivity Discovery. (n.d.). Retrieved from <https://books.google.com.co/books?id=Zpy5AQAACAAJ>
- 802.1D-2004 IEEE Standard for Local and Metropolitan Area Networks Media Access Control (MAC) Bridges. (n.d.). Retrieved from <https://books.google.com.co/books?id=pKFxAQAACAAJ>
- 802.1Q-2014 IEEE Standard for Local and Metropolitan Area Networks--Bridges and Bridged Networks. (n.d.). Retrieved from <https://books.google.com.co/books?id=j2KPnQAACAAJ>
- Academy, C. N. (2014). *Routing and Switching Essentials Companion Guide* (1st ed.). WebEx Communications.
- Agoulmine, N. (2011). *Introduction to Autonomic Concepts Applied to Future Self-Managed Networks. Autonomic Network Management Principles*.
- Agudelo Rojas, Oscar; Hernández Pérez, G. J. (2006). Caracterización estadística del tráfico para la red de datos de la sede Bogotá en la Universidad Nacional de Colombia.
- Al-fares, M., Loukissas, A., & Vahdat, A. (2008). A scalable, commodity data center network architecture. In *ACM SIGCOMM Conference* (pp. 63–74). <https://doi.org/10.1145/1402958.1402967>
- Alfred V. Aho Monica S. Lam, R. S. J. D. U. (2006). *Compilers - Principles, Techniques, and Tools* (2nd ed.). Pearson/Addison Wesley.
- Ardila Triana, E. (2013). Arquitectura para el manejo de congestión en una red de datos corporativa con participación del usuario, basado en inteligencia computacional.
- Association, S. I. E. E. E. (1998). *ANSI IEEE 802.3 Standard*. Retrieved from <http://standards.ieee.org/getieee802/download/802.3ae-2002.pdf>
- Astuto, B. N., Mendonça, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A Survey

- of Software-Defined Networking: Past, Present, and Future of Programmable Networks (Vol. 16, pp. 1617–1634). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/SURV.2014.012214.00180>
- Berrayana, W., Youssef, H., & Pujolle, G. (2012). A generic cross-layer architecture for autonomic network management with network wide knowledge. *2012 8th International Wireless Communications and Mobile Computing Conference IWCMC*, 82–87. <https://doi.org/10.1109/IWCMC.2012.6314182>
- Boehm, B., & Turner, R. (2015). The Incremental Commitment Spiral Model (ICSM): Principles and Practices for Successful Systems and Software. In *Proceedings of the 2015 International Conference on Software and System Process* (pp. 175–176). New York, NY, USA: ACM. <https://doi.org/10.1145/2785592.2785619>
- Bouabene, G., Jelger, C., Tschudin, C., Schmid, S., Keller, A., & May, M. (2010). The autonomic network architecture (ANA). *Selected Areas in Communications, IEEE Journal on*, 28(1), 4–14. <https://doi.org/10.1109/JSAC.2010.100102>
- Bugenhagen, M. K., Morrill, R. J., & Edwards, S. K. (2008). System and method for displaying a graphical representation of a network to identify nodes and node segments on the network that are not operating normally.
- Case, J. D., Fedor, M., Schoffstall, M. L., & Davin, J. (1990). *Simple Network Management Protocol (SNMP)*. Retrieved from <http://www.rfc-editor.org/rfc/rfc1157.txt>
- Champrasert, P., & Suzuki, J. (2007). Building Self-Configuring Data Centers with Cross Layer Coevolution. *Journal of Software*, 2(5), 29–43. <https://doi.org/10.4304/jsw.2.5.29-43>
- Charles Spurgeon, J. Z. (2014). *Ethernet: The Definitive Guide, 2nd Edition: Designing and Managing Local Area Networks*. O'Reilly Media. Retrieved from <http://gen.lib.rus.ec/book/index.php?md5=936D5800E47B056CCC2810D897A16566>
- Cheng, B. H. C., De Lemos, R., Garlan, D., Giese, H., Litoiu, M., Magee, J., ... Taylor, R. (2010). Fifth workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2010). In *Proceedings - International Conference on Software Engineering* (Vol. 2, pp. 447–448).
- Chiang, M., Low, S. H., Calderbank, A. R., & Doyle, J. C. (2007). Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the*

- IEEE*, 95(1). <https://doi.org/10.1109/JPROC.2006.887322>
- Choi, T., Lee, T.-H., Kodirov, N., Lee, J., Kim, D., Kang, J.-M., ... Hong, J. W.-K. (2011). HiMang: Highly manageable network and service architecture for new generation. *Journal of Communications and Networks*, 13(6), 552–566. <https://doi.org/10.1109/JCN.2011.6157472>
- Chowdhury, N. M. M. K., & Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *Communications Magazine, IEEE*, 47(7), 20–26.
- Claise, B. (2004). *Cisco Systems NetFlow Services Export Version 9*. Retrieved from <http://www.rfc-editor.org/rfc/rfc3954.txt>
- Combs, G. (2017). Wireshark. Retrieved from <https://www.wireshark.org/>
- den Berg, J. L., Litjens, R., Eisenblätter, A., Amirijoo, M., Linnell, O., Blondia, C., ... Schmelz, L. C. (2008). Self-organisation in future mobile communication networks. *Proceedings of ICT Mobile Summit, 2008*.
- Derbel, H., Agoulmine, N., & Salaün, M. (2009). ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks. *Comput. Netw.*, 53(3), 418–430. <https://doi.org/10.1016/j.comnet.2008.10.022>
- Dressler, F. (2008). *Self-Organization in Sensor and Actor Networks (Wiley Series in Communications Networking & Distributed Systems)* (1st ed.).
- Dusia, A., & Sethi, A. S. (2016). Recent Advances in Fault Localization in Computer Networks. *Commun. Surveys Tuts.*, 18(4), 3030–3051. <https://doi.org/10.1109/COMST.2016.2570599>
- Fabio Luigi Bellifemine Giovanni Caire, D. G. (2007). *Developing Multi-Agent Systems with JADE*. Wiley.
- Flach, T., Papageorge, P., Terzis, A., Pedrosa, L., Cheng, Y., Karim, T., ... Govindan, R. (2016). An Internet-Wide Analysis of Traffic Policing. In *Proceedings of the 2016 ACM SIGCOMM Conference* (pp. 468–482). New York, NY, USA: ACM. <https://doi.org/10.1145/2934872.2934873>
- Forde, T. K., Doyle, L. E., & Mahony, D. O. (2005). Self-stabilizing network-layer auto-configuration for mobile ad hoc network nodes. In *IEEE International Conference on Wireless and Mobile Computing, Networking And Communications*. <https://doi.org/10.1109/WIMOB.2005.1512901>
- Gershenson, C. (2010). *Design and Control of Self-organizing Systems: Facing*

Complexity with Adaptation and Self-organization. LAP Lambert Academic Publishing.

- Gilbert, S., & Lynch, N. (2002). Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services. *SIGACT News*, 33(2), 51–59.
<https://doi.org/10.1145/564585.564601>
- Giordano, A., Spezzano, G., & Vinci, A. (2016). *Smart agents and fog computing for smart city applications. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 9704).
https://doi.org/10.1007/978-3-319-39595-1_14
- Glitho, R., & Svensson, B. (2001). Operation and maintenance control point and method of managing a self-engineering telecommunications network. Google Patents.
- GNU. (2017). Retrieved from <https://www.gnu.org/>
- He, J., Zhang-shen, R., Li, Y., Lee, C., Rexford, J., & Chiang, M. (2008). DaVinci: dynamically adaptive virtual networks for a customized internet. In *Conference on Emerging Network Experiment and Technology*.
<https://doi.org/10.1145/1544012.1544027>
- Hu, H., Zhang, J., Zheng, X., Yang, Y., & Wu, P. (2010). Self-configuration and self-optimization for LTE networks. *Communications Magazine, IEEE*, 48(2), 94–100.
- Hurst, H. E., Black, R. P., & Simaika, Y. M. (1965). *Long-term storage : an experimental study / by H.E. Hurst, R.P. Black, Y.M. Simaika*. Constable London.
- IEEE. (2017). Retrieved from <https://www.ieee.org/index.html>
- IEEE Std 802.1ad-2005 (Amendment to IEEE Std 802.1Q-2005): IEEE Standard for Local and Metropolitan Area Networks---Virtual Bridged Local Area Networks---Amendment 4: Provider Bridges*. (2006). IEEE. Retrieved from <https://books.google.com.co/books?id=NJLdnQAACAAJ>
- Iperf. (2017). Retrieved from <https://iperf.fr/>
- ISO/IEC 10039:1991. (1991). Retrieved from <https://www.iso.org/standard/18004.html>
- Jorg P. Muller Wolf Ketter, G. K. G. W. N. B. (eds. . (2015). *Multiagent System Technologies : 13th German Conference, MATES 2015, Cottbus, Germany, September 28 - 30, 2015, Revised Selected Papers* (1st ed.). Springer International Publishing.
- Kim, S. (2011). Forecasting internet traffic by using seasonal GARCH models. *Journal of*

- Communications and Networks*, 13(6), 621–624. Retrieved from <http://dblp.uni-trier.de/db/journals/jcn/jcn13.html#Kim11a>
- Konstantinou, A. V, Florissi, D., & Yemini, Y. (2002). Towards Self-Configuring Networks. In *DARPA Active Networks Conference and Exposition* (pp. 143–156). <https://doi.org/10.1109/DANCE.2002.1003489>
- Kurose James F.; Ross, K. W. (2013). *Computer networking : a top-down approach* (6ed.). Pearson, AW.
- Larus, J., Rajamani, S., & Rehof, J. (2004). Contracts and futures in an asynchronous programming language. Google Patents.
- Leach, P., Mealling, M., & Salz, R. (2005). RFC 4122: A Universally Unique Identifier (UUID) URN Namespace. Retrieved from <http://www.ietf.org/rfc/rfc4122.txt>
- Licitación CON-BOG-007-2016 Universidad Nacional de Colombia. (2016). Retrieved from <http://contratacion.bogota.unal.edu.co/documentos/CON-BOG-007-2016/pdf/CON-BOG-007-2016-PLIEGO DE CONDICIONES CON-BOG-007-2016.pdf>
- Lu, Z., Suizo, N., Nagarajan, R., Villait, A., Lochner, W. M., Ronne, J., ... Desur, G. raj. (2012). Self-configuring network.
- Martin Garcia, L. (2008). Programming with Libpcap --- Sniffing the Network from Our Own Application. *Hackin9 Magazine*, 3(2/2008). Retrieved from <http://www.programming-pcap.albaknocking.com/>
- Mauerer, W. (2008). *Professional Linux kernel architecture*. Wrox/Wiley Pub.
- McCloghrie, K., & Rose, M. T. (1990). *Management Information Base for network management of TCP/IP-based internets*. Retrieved from <http://www.rfc-editor.org/rfc/rfc1156.txt>
- Mearns, H., & Leaney, J. (2013). The use of autonomic management in multi-provider telecommunication services. In *Proceedings of the International Symposium and Workshop on Engineering of Computer Based Systems*. <https://doi.org/10.1109/ECBS.2013.21>
- Mortier, R., & Kiciman, E. (2006). Autonomic network management: Some pragmatic considerations. In *Proceedings of the 2006 SIGCOMM Workshop on Internet Network Management, INM'06* (Vol. 2006, pp. 89–93).
- Nikraz1a, Magid; Caireb, Giovanni ; Bahria, P. A. (2006). A Methodology for the Analysis and Design of Multi-Agent Systems using JADE.

- Nmap. (2017). Retrieved from <https://nmap.org/nping/>
- North, M. J. (2014). A theoretical formalism for analyzing agent-based models. *Complex Adaptive Systems Modeling*, 2(1), 3. <https://doi.org/10.1186/2194-3206-2-3>
- Orozco, J. M. S. (2012). *Applied Ontology Engineering in Cloud Services, Networks and Management Systems*. Springer Publishing Company, Incorporated.
- Park, K., & Willinger, W. (2000). *Self-Similar Network Traffic and Performance Evaluation* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.
- Paxson, V., & Floyd, S. (1995). Wide area traffic: the failure of Poisson modeling. *Networking, IEEE/ACM Transactions on*, 3(3), 226–244.
- Perlman, S. G. (2009). Self-configuring, adaptive, three-dimensional, wireless network.
- Phaal, P., Panchen, S., & McKee, N. (2001). *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*. Retrieved from <http://www.rfc-editor.org/rfc/rfc3176.txt>
- Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. In *IN PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON MULTI-AGENT SYSTEMS (ICMAS-95)* (pp. 312–319).
- Rojas, R. (2015). A Tutorial Introduction to the Lambda Calculus. *CoRR*, *abs/1503.0*. Retrieved from <http://arxiv.org/abs/1503.09060>
- Rose, M. T., & McCloghrie, K. (1990). *Structure and identification of management information for TCP/IP-based internets*. Retrieved from <http://www.rfc-editor.org/rfc/rfc1155.txt>
- Sánchez Cifuentes, J. F. (2012). Modelo de Control de Congestión para Redes IP de area Local.
- Sandvine. (2016). Global Internet Phenomena 2016 Latin America and North America. *Report*, 1, 15.
- Sayama, H. (2015). *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks.
- Shambhu Upadhyaya Abhijit Chaudhury, K. K. M. W. (2002). *Mobile Computing: Implementing Pervasive Information and Communications Technologies (Operations Research Computer Science Interfaces Series)* (1st ed.). Retrieved from <http://gen.lib.rus.ec/book/index.php?md5=F4FF7448352EF33A43173E59A307B551>
- Shehory, O., & Sturm, A. (2014). *Agent-Oriented Software Engineering: Reflections on*

- Architectures, Methodologies, Languages, and Frameworks*. Springer Publishing Company, Incorporated.
- Stallman, R. (2017). No Title. Retrieved from <http://www.fsf.org/es/sabana>
- TCPDump. (2017). Retrieved from <http://www.tcpdump.org>
- Uri Wilensky, W. R. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. The MIT Press.
- Varela, C. A. (2013). *Programming distributed computing systems: a foundational approach*. MIT.
- Viering, I., Dattling, M., & Lobinger, A. (2009). A Mathematical Perspective of Self-Optimizing Wireless Networks. In *IEEE International Conference on Communications* (pp. 1–6). <https://doi.org/10.1109/ICC.2009.5198628>
- Waldbusser, S. (2000). *Remote Network Monitoring Management Information Base*. Retrieved from <http://www.rfc-editor.org/rfc/rfc2819.txt>
- Waldbusser, S. (2006). *Remote Network Monitoring Management Information Base Version 2*. Retrieved from <http://www.rfc-editor.org/rfc/rfc4502.txt>
- Wallace, K. (2015). *CCNP Routing and Switching ROUTE 300-101 Official Cert Guide* (1st ed.). Cisco Press. Retrieved from <http://gen.lib.rus.ec/book/index.php?md5=F0212922D19C1A859BA8519D50E10777>
- Weiss, G. (Ed.). (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA, USA: MIT Press.
- Whittle, J., Sawyer, P., Bencomo, N., Cheng, B. H. C., & Bruel, J.-M. (2009). Relax: Incorporating uncertainty into the specification of self-adaptive systems. In *Requirements Engineering Conference, 2009. RE'09. 17th IEEE International* (pp. 79–88).
- Wooldridge(eds.), M. (2012). *Mobile Agents in Networking and Distributed Computing*.
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems* (2nd ed.).
- Yang, H., Wakamiya, N., Murata, M., Iwai, T., & Yamano, S. (2016). Autonomous and distributed mobility management in mobile core networks. *Wireless Networks*. <https://doi.org/10.1007/s11276-016-1274-3>
- Zhang, H., Wen, X., Wang, B., Zheng, W., & Lu, Z. (2009). A Novel Self-Optimizing Handover Mechanism for Multi-service Provisioning in LTE-Advanced. In *International Conference on Research Challenges in Computer Science*. <https://doi.org/10.1109/ICRCCS.2009.64>

