



Implementación de un prototipo de Red Ad Hoc tipo clúster dentro del campus Universitario de la Universidad Nacional

Jhon Alexander López Velasco

Universidad Nacional de Colombia
Ingeniería Telecomunicaciones
Bogotá D.C, Colombia
2021

Implementación de un prototipo de Red Ad Hoc tipo clúster dentro del campus Universitario de la Universidad Nacional

Jhon Alexander López Velasco

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

Director(a):
Jorge Eduardo Ortiz Triviño, PhD.

Línea de Investigación:
Redes Ad-Hoc
Grupo de Investigación:
Tlön

Universidad Nacional de Colombia
Ingeniería Telecomunicaciones
Bogotá D.C, Colombia
2021

La medida de la inteligencia es la capacidad de cambiar.

Albert Einstein

Agradecimientos

A Joaquín Fernando Sánchez Cifuentes, Profesor de la Universidad Sergio Arboleda y Doctor en Ingeniería de Sistemas y Computación (PhD) de la Universidad Nacional de Colombia, por el apoyo, el seguimiento y el importante enfoque para desarrollar los objetivos trazados en la investigación. Al Grupo de Investigación Tlön por la permanente retroalimentación y los valiosos aportes realizados al trabajo de investigación.

Al profesor Jorge Eduardo Ortiz Triviño PhD. Profesor Asociado de la Universidad Nacional de Colombia y director del Grupo de Investigación Tlön, por la tutoría y orientación académica sobre el problema de investigación.

Resumen

El presente trabajo de investigación propone la implementación de un prototipo de red Ad Hoc tipo Clúster sobre el campus Universitario de la Universidad Nacional de Colombia para zonas con ausencia de redes inalámbricas tradicionales. El prototipo de red ofrece un servicio de mensajería instantánea utilizando el protocolo XMPP, y para su funcionamiento, el servicio opera únicamente a través de un nodo líder o Head, quien hace las veces de Servidor para clientes agrupados sobre un mismo dominio. El prototipo implementado hace parte de la premisa de que cualquier nodo de la red puede asumir el rol de Head sobre un mismo dominio y únicamente a través de este, es posible el intercambio de información por medio de mensajería XMPP con dominios externos. Para dar cumplimiento a lo anteriormente expuesto, es necesaria la construcción de los Clúster Head de la red, y configurar por medio de software las características que permitan establecer los parámetros de operación e interconexión a través de B.A.T.M.A.N del prototipo, garantizando heterogeneidad de Hardware sobre equipos compatibles con kernel Linux. Para el desarrollo de este trabajo de investigación, se utilizaron cinco (5) Raspberry Pi que operaron como los nodos de la red y los cuales en la prueba de campo fueron ubicados geográficamente en dos (2) esquemas diferentes : línea recta y forma circular. El esquema lineal fue el que obtuvo mayor cobertura geográfica y mejores tiempos de respuesta en conectividad.

Palabras clave: Red Ad Hoc, Clúster, Clúster Head, Mensajería XMPP, Cliente, Servidor, Base de Datos distribuida, Capa MAC / OSI.

Abstract

This research project proposes the implementation of an Ad Hoc Cluster network prototype over the National University of Colombia in areas without traditional wireless networks. The network prototype offers an instant messaging service using the XMPP protocol, and for its operation, the service operates only through a head node, which acts as a server for clients grouped in the same domain. The implemented prototype is part of the premise that any node of the network can assume the role of Head on the same domain and only through this, it is possible to exchange information by means of messaging to XMPP with external domains. In order to comply with the above, it is necessary to build the Master Head Clusters of the network, and to configure by means of software the characteristics that allow establishing the operation and interconnection parameters through the B.A.T.M.A.N. of the prototype, guaranteeing hardware heterogeneity on equipment compatible with the Linux kernel. During the development of this work, five (5) Raspberry Pi were used as the nodes of the network, which in the field test were geographically located in two (2) different schemes: straight line and circular shape. The first one scheme was the best in terms of geographic coverage and connectivity.

Keywords: Ad Hoc Network, Cluster, Cluster Head, XMPP Messaging, Client, Server, Distributed Database, MAC / OSI Layer

Esta tesis de maestría se sustentó el 04 de Junio de 2021 a las 2:00 p.m. y fue evaluada por los siguientes jurados:

*Luis Fernando Niño Vásquez, PhD
Afiliación: Profesor Departamento de Ingeniería de Sistemas e Industrial UNAL Bogotá*

*Ingrid Patricia Páez Parra, PhD.
Afiliación: Profesora Departamento de Ingeniería de Sistemas e Industrial UNAL Bogotá*

Contenido

Agradecimientos	7
Resumen	9
Lista de Figuras	1
Lista de Tablas	1
Introducción	2
1. Redes Ad Hoc	5
1.1. Tipos de Redes	6
1.1.1. Red de Petri	6
1.1.2. Red Auto-Configurable	7
1.1.3. Red Adaptativa	7
1.1.4. Red de Auto-Gestión	7
1.1.5. Red de Auto-Análisis	7
1.1.6. Red de Auto-optimización	8
1.1.7. Sistema de auto-estabilización de red	8
1.2. Modelos de Redes	8
1.2.1. Modelo de Gestión Gráfica	8
1.2.2. Modelo ANEMA	8
1.2.3. Modelo de Red DaVinci	9
1.2.4. Modelo ANM	9
1.2.5. Modelo no suponga nada	9
1.3. Enrutamiento de Redes	10
1.4. Necesidad de las redes auto-organizables	11
1.4.1. Computación adaptativa	11
1.4.2. Modelos de Redes auto-organizables	12
1.4.3. Características de las redes autónomas	13
1.5. Mensajería Instantánea-XMPP	15
1.5.1. Protocolo XMPP	15
1.5.2. Características del protocolo XMPP	15
1.5.3. Mensajería instantánea (IM)	17

1.6.	Agrupamiento de Nodos y Clúster	17
1.6.1.	Clasificación de esquemas de agrupamiento	18
1.6.2.	Esquemas de agrupamiento para redes Ad-Hoc móviles	19
1.6.3.	Algoritmos de agrupamiento para redes Ad-Hoc	20
1.7.	Antecedentes e Implementación de redes	22
1.8.	Problema de investigación	25
1.9.	Objetivos	26
1.9.1.	Objetivo General	26
1.9.2.	Objetivos específicos	26
1.10.	Justificación	26
1.11.	Alcances y limitaciones	27
2.	Características de los Nodos, Protocolos y Servicios del Prototipo de Red Ad-Hoc	28
2.1.	Raspberry Pi	28
2.2.	Raspbian	30
2.3.	Pacemaker	30
2.3.1.	Características	31
2.4.	Corosync	32
2.4.1.	Características	32
2.4.2.	Clúster Corosync-Pacemaker	33
2.5.	Ejabberd	33
2.5.1.	Características	34
2.6.	Finch	35
2.7.	B.A.T.M.A.N	36
2.8.	A.L.F.R.E.D	38
2.8.1.	Estructura	39
2.8.2.	Intercambio de datos	41
3.	Metodología y Resultados	42
3.1.	Tlön	42
3.1.1.	Proyecto Tlön	42
3.2.	Rol del prototipo en el modelo Tlön	43
3.3.	Diseño del Prototipo	44
3.4.	Implementación del servicio y definición del Clúster-Head	45
3.4.1.	Funcionamiento de los Head por clúster	45
3.4.2.	Funcionamiento de cada clúster y sus recursos compartidos	45
3.4.3.	Condiciones de operación de mensajería	46
3.5.	Operación del prototipo	46

3.6. Prueba de Campo	47
3.6.1. Ubicación	47
3.6.2. Parámetros Evaluados	48
3.6.3. Pre-requisitos	48
3.7. Métricas de Evaluación	50
4. Conclusiones	54
4.1. Conclusiones y Recomendaciones	54
4.2. Trabajos Futuros	55
4.3. Producción Académica	55
Bibliografía	56
A. Anexo:Instructivos de instalación	63
A.1. Instalar y configurar Raspbian en Raspberry Pi	63
A.2. Instalar y Configurar B.A.T.M.A.N y A.L.F.R.E.D	69
A.3. Instalar y configurar Pacemaker y Corosync (PCS-Pacemaker Cluster Stack)	75
A.4. Instalar y configurar Ejjaberd y Finch	81

Lista de Figuras

1-1. Esquema de Red Ad Hoc	6
1-2. Mensajería instantánea con XMPP [36].	17
1-3. Red de agrupamiento (clustering) [52].	18
1-4. Diferentes maneras de clasificar esquemas de agrupamiento[75].	19
2-1. Algunos componentes centrales en una placa Raspberry Pi[54].	28
2-2. Placa Raspberry Pi modelo A (izquierda) y modelo B (derecha)[54].	29
2-3. Logo de Raspbian [37].	30
2-4. Logo de Pacemaker[21].	31
2-5. Logo de corosync [21].	32
2-6. Logo de Ejjaberd [60].	34
2-7. Logo de Finch	36
2-8. Estructura de una red Alfred [25].	39
2-9. Envío de mensaje del servidor primario a los servidores secundarios para la detección de vecinos [25].	40
2-10. Intercambio de datos en una red Alfred [25].	41
3-1. Proyecto Tlön [73].	43
3-2. Elementos del prototipo asociados al pseudo territorio del modelo Tlön.	43
3-3. Configuración final del prototipo	45
3-4. Operación del clúster head	46
3-5. Configuraciones previas y parámetros obtenidos.	47
3-6. Campo seleccionado para las pruebas de campo, en el Campus de la Universidad Nacional visto desde Google Earth (Círculo Amarillo).	47
3-7. Esquema de la Red con los nodos ubicados en línea recta.	48
3-8. Esquema de la Red con los nodos ubicados de forma circular	49
3-9. Monitoreo del estado de los nodos, servicio XMPP y clústeres a través de Nagios.	49
3-10. Comportamiento de la Red con nodos ubicados en línea Recta en cuanto a Tiempos de Respuesta y Potencia vs Distancia.	50
3-11. Comportamiento de la Red con nodos ubicados de forma circular en cuanto a Tiempos de Respuesta y Potencia vs Distancia.	51
3-12. Comparativos del comportamiento de la Red en cuanto a Potencia versus distancia.	52

3-13. Comparativos del comportamiento de la Red en cuanto a Tiempo de respuesta versus distancia.	52
4-1. Logo Colombian Conference on Communications and Computing (COLCOM-2017).	55

Lista de Tablas

1-1. Algoritmos de agrupamiento	22
2-1. Características de algunos modelos Raspberry comerciales [63].	29

Introducción

Se podría pensar en una red Ad-hoc como el conjunto de dos o más dispositivos equipados con comunicaciones móviles y capacidad de red. Estos dispositivos pueden comunicarse con otros en su área de cobertura directamente y fuera de esa área a través de nodos intermedios. Las redes Ad-Hoc son una serie de host que se comunican entre sí, ya sea directamente o a través de otros equipos que forman parte de la misma red.

Las redes Ad-Hoc son redes inalámbricas soportadas principalmente por señales de radio (muchas sobre en la banda ISM) que permiten que cualquier host de la red sea móvil. La movilidad es prácticamente la principal ventaja para los usuarios de redes Ad-Hoc, así como el principal obstáculo para los investigadores y desarrolladores de protocolos en el enruteamiento.

Este tipo de redes inalámbricas son adaptativas y auto organizativas, se forman y deforman en el aire alterando las rutas que finalmente se han definido. Un ejemplo común son las redes personales entre teléfonos móviles, PDA's e impresoras con transceptores que incluyen tecnología Bluetooth. El término Ad-Hoc como tal se refiere a la movilidad, a la red, la espontaneidad de estos grupos lógicos de computadoras y la falta de requisitos para las estaciones base de control fijas, que son responsables de las tareas de gestión y control del tráfico de la red.

La principal diferencia con las redes inalámbricas basadas en infraestructura consiste generalmente con una disposición de celdas ubicadas en la geografía sobre la cual se pretende brindar el acceso inalámbrico a los usuarios. Estas áreas de cobertura son administradas por un conjunto de dispositivos llamados “puntos de acceso” (como en la figura 1-1), los cuales no poseen ningún tipo de movilidad y que conforman lo que se conoce como “subsistema de radio” cuya función consiste en la gestión de las comunicaciones entre los dispositivos móviles, al igual que las tareas de control que incluyen la gestión de canales, el hand off, el roaming y entre otros. Es por esto que muchos de estos puntos de acceso realizan las tareas de enruteamiento empleando algoritmos similares a los empleados en las redes fijas basadas en medios guiados, mientras que en el ámbito de las redes ad-hoc las tareas de control y gestión de tráfico no son realizadas por un dispositivo especializado para este fin, sino que este rol puede ser asumido por cualquier dispositivo móvil “de usuario” que haga parte de la red y que en ese instante específico, cumple con las condiciones necesarias para realizar dicha función. Dentro

de las condiciones no se requiere hardware especializado sino: ubicación física estratégica del nodo (generalmente asociado a cierto grado de estabilidad sobre un área específica), nivel de potencia/sensibilidad mínimo requerido, entre otros. De manera que, podría afirmarse que en una red Ad-hoc todos los nodos móviles son enrutadores y gestores de tráfico en potencia los cuales cambian de rol dependiendo de las necesidades de comunicaciones de la misma red.

Dentro de las características importantes de las redes Ad-Hoc podría destacarse la heterogeneidad de hardware y software, el bajo consumo de energía de los nodos (debido a que son basados principalmente en baterías o fuentes de energía móviles), transceptores de baja potencia y de corto alcance con niveles bajos de sensibilidad en muchos casos. No obstante, la heterogeneidad de los componentes de la red en algún momento podría llegar a afectar el desempeño de las comunicaciones e incluso la estandarización de los protocolos. La optimización del consumo de energía de los equipos móviles es bastante relevante en el ámbito de las redes de enlace espontáneo, debido a que incluso el reenvío de paquetes por parte de los nodos (principalmente los intermediarios) genera un gran consumo de energía limitando el tiempo de servicio de los nodos en la red. A pesar de la problemática que plantean las redes inalámbricas debido al tipo de comunicación y a la movilidad misma de sus estaciones de trabajo, como la optimización de ancho de banda, el control de potencia, mantenimiento de la topología, el descubrimiento de los dispositivos y los servicios que estos ofrecen al igual que la complejidad del soporte de calidad de Servicio (QoS), hoy en día las redes ad-hoc son consideradas por muchos como la evolución de las redes inalámbricas y traen consigo muchos beneficios y áreas de utilidad. Es posible encontrar en la actualidad redes de PDA (Personal Digital Assistance) armadas remotamente en una conferencia, al igual que redes de enlace espontáneo formadas por equipos de comunicaciones incluidos como parte del apoyo en las comunicaciones de algunas unidades militares e incluso como equipos de apoyo a brigadas de salvamento.

Una utilidad interesante de las Ad-hoc networks son Redes Móviles Ad-Hoc, las redes de sensores llamadas “HAM” o redes Ad hoc híbridas que están enlazadas con centros de monitoreo para recolectar información sobre el estado del clima, detección de propiedades químicas, elementos radiactivos, temperaturas, entre. Los nodos que forman parte de las redes de enlace espontáneo basadas en sensores generalmente realizan cálculos locales basados en los datos recolectados (“Data fusion”) y después , estos realizan estadísticas de los datos recibidos con el objetivo de permitir la toma de decisiones correctas en situaciones críticas. Algunas redes Ad-hoc están enlazadas a redes de infraestructura fija a través de puntos de acceso como las MESH o rooftop Networks [18] con el fin de proveer servicios de acceso a la Internet de forma inalámbrica, incluso con algunos vehículos se crean redes inalámbricas de enlace espontáneo para obtener información sobre el comportamiento del tráfico o para cubrir incluso algún evento deportivo.

Con lo anteriormente mencionado, cabe resaltar que las aglomeraciones de dispositivos en un lugar puede afectar el rendimiento de una red, generando problemas de conectividad, baja tasa de transferencia de datos, tiempos de disminución en la potencia de la señal, respuesta con mayor latencia, entre otros. La Universidad Nacional al ser una institución de educación superior cuenta en su alma máter con una gran cantidad de personas con disposición de dispositivos electrónicos (celulares, portátiles, tablets, etc) que requieren de conectividad a servidores de internet (red o datos) o entre ellos, para la transferencia de archivos, imágenes, mensajería, entre otros. Las redes inalámbricas tradicionales ofrecen un servicio que permiten dicha transferencia de datos, sin embargo, no se puede satisfacer completamente las necesidades de los integrantes de la Universidad Nacional, teniendo en cuenta que este tipo de señales tienen una cobertura que genera que existan zonas con ausencia de estas redes.

Por este razón, en este trabajo de investigación se propuso el diseño de un prototipo de Red Ad Hoc que permitiera la expansión de la cobertura geográfica en el campus de la Universidad Nacional en áreas en ausencia de redes tradicionales, que además ofrece un servicio de mensajería instantánea. Para el diseño del prototipo se usaron cinco (5) dispositivos Raspberry Pi que funcionaron como los nodos de la red y fueron agrupados en dos clústers (cada uno con un clúster head). Estos fueron ubicados en dos esquemas geográficos diferentes: línea recta y forma circular y cuyo desempeño fue evaluado con base en la potencia y tiempos de respuesta en conectividad de la red. Los resultados permitieron concluir que la ubicación lineal demostró un comportamiento superior al del esquema circular, permitiendo una mayor cobertura geográfica con tiempos de respuesta más cortos.

El presente trabajo de investigación está compuesto por cuatro (4) secciones: Redes Ad Hoc (Sección 1), Características de los Nodos, Protocolos y Servicios del Prototipo de Red Ad-Hoc (Sección 2), Metodología y Resultados (Sección 3) y Conclusiones (Sección 4). Finalmente se encuentra la Bibliografía y sus Anexos.

1. Redes Ad Hoc

El conjunto de las redes de comunicaciones inalámbricas, tienen subconjuntos que permiten clasificar este tipo de tecnologías, y los cuales, se encuentran aplicados para problemas particulares. Uno de estos subconjuntos son las redes Ad Hoc, descritas como redes que tienen características especiales, en comparación al concepto de redes de comunicaciones inalámbricas conocido, [58, 65].

Una red Ad Hoc no tiene una infraestructura definida, debido a que los componentes de esta red ingresan e irrumpen de manera aleatoria [2]. Es así, que se genera un comportamiento dinámico y estocástico. Estas propiedades hacen que los diseños de las redes Ad Hoc se tornen complejos para realizar labores de dimensionamiento de recursos, tales como capacidades de los anchos de banda de sus enlaces y capacidades de procesamiento de los dispositivos que involucre la red [65].

Las redes Ad Hoc se formaron como sistemas computacionales auto-organizados, desarrollados por un conjunto de nodos interconectados entre sí por medio de enlaces inalámbricos y los cuales no dependen de una infraestructura pre-existente para funcionar. Este nuevo concepto permitió mitigar la problemática de cobertura, movilidad y dinamismo en las redes de comunicaciones [17].

Entre las características de las redes Ad Hoc antes citadas, se encontró su naturaleza descentralizada, su capacidad de auto-organización, su comportamiento social, su heterogeneidad, su arquitectura plana o jerárquica, sus algoritmos de agrupamiento y su adecuada asignación de tareas [16].

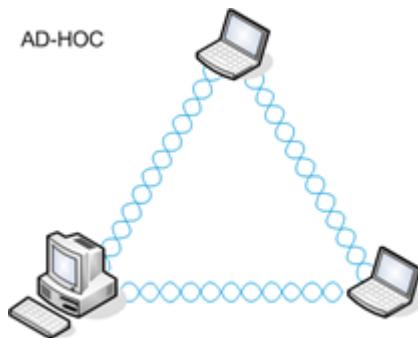


Figura 1-1.: Esquema de Red Ad Hoc

Haciendo un análisis de las características que poseen las redes de comunicaciones inalámbricas tipo Ad Hoc, los diseñadores de redes de comunicaciones en sus estudios encontraron una clasificación taxonómica de las redes autónomas, analizando diferentes modelos y tipos de red, los cuales sirvieron como referenciación para diseñar e implementar un sistema de gestión para estas redes [15].

Actualmente, el reto abierto de mejorar el desempeño puede relacionarse con el acceso al medio, el enrutamiento, los servicios multicast, la eficiencia del consumo de energía, el soporte y el desempeño de los servicios basados en TCP/IP, el soporte de QoS, el descubrimiento de servicios y la seguridad de la información tanto de control como de usuario.

1.1. Tipos de Redes

1.1.1. Red de Petri

Es una generalización de la teoría de autómata que permite expresar un sistema a eventos concurrentes. La red de Petri está formada por lugares, transiciones, arcos dirigidos y marcas o fichas que ocupan posiciones dentro de los lugares [8]. Las reglas son:

- Los arcos conectan un lugar a una transición así como una transición a un lugar.
- No puede haber arcos entre lugares ni entre transiciones.
- Los lugares contienen un número finito o infinito contable de marcas.
- Las transiciones se disparan, es decir, consumen marcas de una posición de inicio y producen marcas en una posición de llegada.
- Una transición está habilitada si tiene marcas en todas sus posiciones de entrada.

En su forma más básica, las marcas que circulan en una red de Petri son todas idénticas. Se puede definir una variante de las redes de Petri en las cuales las marcas pueden tener información que las distingue, un tiempo de activación y una jerarquía en la red [8].

1.1.2. Red Auto-Configurable

Representación de la confiabilidad y desempeño, reduciendo costos con técnicas automatizadas. Una red auto configurable se constituye de elementos que automáticamente son provistos (recursos de red adecuados al servicio ante la ocurrencia de un evento). Para satisfacer un servicio, se modela un perfil con uno o más comandos que configuran una estrategia sobre uno o varios dispositivos de red, los cuales 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 [53].

1.1.3. Red Adaptativa

Representación de un sistema de red que permite la toma de decisiones para adaptar recursos y servicios según los cambios del entorno y las necesidades del usuario. Es necesario entenderlo como un grupo de sensores capaz de adaptarse y entregar soluciones más cercanas al sentir humano [13].

1.1.4. Red de Auto-Gestión

Representación de las redes de comunicación inalámbricas tipo Ad Hoc. Se aplica la adaptabilidad de la red para poder brindar un canal de transporte, pese a la cantidad de cambios en los canales y 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 [62].

1.1.5. Red de Auto-Análisis

Representación del plano de control de mantenimiento y operación de una red como una capa media entre los objetos y la red. Se puede lograr conformar que una capa media (conjunto de equipos, software, entre otros) envíe información de diagnóstico elaborada a los estaciones de gestión (NMS por sus siglas del inglés ,Network Management System), con acciones correctivas y con resultados predictivos. Esto daría mejor efectividad en problemas recurrentes y de esta forma la capa media automáticamente le provee a la red la capacidad de auto-análisis de ingeniería [14, 33].

1.1.6. Red de Auto-optimización

La tendencia mundial de las comunicaciones las está imponiendo 3G-LTE. Para reducir costos operacionales, estas redes están provistas de sistemas avanzados de auto optimización. Poseen un método de traspaso (traspaso de un servicio o usuario a otra antena por efecto de congestión o movimiento handover) basado en la cantidad de celdas sobrepuertas. Se ajustan de manera automática varios parámetros como tiempo de disparo, histéresis e intervalos de medida, mejorando los resultados tradicionales del ping-pong de traspaso. Es interesante observar y comprobar en el laboratorio que existen mejores forma de optimizar una red [49, 79].

1.1.7. Sistema de auto-estabilización de red

Representación de decisiones que generan actividades reactivas de auto-configuración. Por ejemplo, tener la opción de dirigir de forma dinámica el enrutamiento según varios parámetros de decisión. Para interactuar con el protocolo de auto-configuración, es necesario tener un módulo de resolución de conflictos de configuración [29, 45]

1.2. Modelos de Redes

Haciendo un análisis de las características que poseen las redes de comunicaciones inalámbricas tipo Ad Hoc, los diseñadores de redes de comunicaciones en sus estudios encontraron una clasificación taxonómica de las redes autónomas, estudiando diferentes modelos y tipos de red, los cuales sirvieron como referencia para diseñar e implementar un sistema de gestión para estas redes [15], como se muestra a continuación:

1.2.1. Modelo de Gestión 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 para gestión de la red y poder 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, por el contrario, siempre se necesita una tercera estación que capture la traza de la conversación mediante una preselección de las características de comportamiento que se desean observar, enfocadas al comportamiento del usuario [9, 53].

1.2.2. 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, la cual satisface los objetivos de los usuarios. Al 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 [24].

1.2.3. 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) entre todas las redes virtuales, tercerizadas y las reales. Esto mediante la asignación de recursos dinámicos que pueden llevar a riesgos altos de estabilidad y utilización del modelo de asignación de recursos de manera adaptativa, probando que es estable y maximizando su desempeño. La aplicación de la red Davinci prueba la usabilidad y la adaptabilidad de las redes [34].

1.2.4. Modelo ANM

Modelo de red que auto detecta, diagnostica, repara fallas, adapta su configuración, optimiza su desempeño y ofrece calidad de servicio. Desde hace tiempo han existido varios métodos de trabajo autónomos para resolver problemas de red, como el algoritmo de control de enrutamiento de estado de los canales de comunicaciones (link-state) conocido como Open Shortest Path First (OSPF) y Border Gateway Protocol (BGP). En otro escenario, se puede ver el control de congestión de ventana deslizante de las transmisiones de TCP, lo que promueve el uso de las redes autónomas aplicada a la explotación de las redes móviles LTE [11].

1.2.5. Modelo no suponga nada

El modelo plantea que se deben tener acciones y decisiones para cada fenómeno que se pueda presentar. Conociendo el entorno, las variables del medio ambiente y datos de entrada se permite tomar acciones automáticas. Es importante tener presente que la cantidad y calidad de los datos lleva a una buena toma de decisiones. Muchos modelos se construyen sobre supuestos o condiciones ideales pero los cambios en el entorno invalidan todas las suposiciones que se tenían, por esto se debe dar manejo a las fallas comunes y además se debe conocer el ambiente, los escenarios de prueba y comprobar que los comportamientos automatizados controlan de manera adecuada el problema [56].

Teniendo clara la estructura de los modelos aquí citados, los diseñadores de redes de comunicaciones inalámbricas Ad Hoc, debían realizar un análisis de estas redes a partir de sus componentes físicos (hardware) y su lógica (software). Lo anterior responde a que debían estabilizar la topología, el manejo de tráfico y la cobertura de dicha red.

1.3. Enrutamiento de Redes

Después de sintetizar la clasificación taxonómica de las redes (modelos y tipos) y las características de las redes Ad Hoc, se hablará del enrutamiento entre los nodos de las redes Ad Hoc. Estableciendo, definiendo el espacio, y precisando las variables que intervienen en la implementación de una red Ad Hoc se encuentran condiciones de tipo ambiental que pueden generar ruido en la transmisión de datos. Por otra parte, los recursos limitados de cada nodo, la cobertura de estos y finalmente la compatibilidad entre tecnologías deben estar armonizadas con protocolos de enrutamiento [9, 61].

Las redes Ad Hoc se pueden ver como un caso extremo de redes malladas en las que todos los nodos poseen movilidad (sin infraestructura). Los protocolos de enrutamiento de las redes Ad Hoc pueden ser modificados para utilizarlos en redes malladas inalámbricas. Estos cambios se centran en mejorar la escalabilidad, el rendimiento, los requisitos de eficiencia energética y la movilidad, atendiendo a las particularidades de la infraestructura mallada. Posteriormente, se explica específicamente cada una de las características de los protocolos de enrutamiento de las redes Ad Hoc, las cuales son de interés para redes malladas inalámbricas y los aspectos a considerar en su aplicación [43, 65].

- **Número de Saltos:** las redes Ad Hoc suelen utilizar el número de saltos como una medida para el enrutamiento, esta medida es adecuada para las redes Ad Hoc ya que las nuevas rutas deben encontrarse de manera rápida pero no permite distinguir rutas de mejor calidad en un entorno de topologías más estables [65].
- **Tolerancia a fallos:** uno de los objetivos para desarrollar una red mallada inalámbrica ,es garantizar robustez ante fallos en los enlaces. Si un enlace cae, el protocolo de enrutamiento debe ser capaz de seleccionar rápidamente otra ruta para evitar la interrupción del servicio [29].
- **Equilibrado de carga:** uno de los objetivos es compartir los recursos de la red entre muchos usuarios. Cuando una parte de la red experimenta congestión, los nuevos flujos de tráfico no deben ser enviados a través de esas rutas [39].
- **Rendimiento:** el rendimiento es una medida que se usa para saber si la ruta de enrutamiento es óptima. Supongamos que el camino más corto entre dos nodos es el que presenta una baja calidad, si sólo se usa el número de saltos para decidir la ruta, el rendimiento entre estos dos nodos será muy bajo. Para resolver este problema, es importante la relación entre el número de saltos y la calidad, de modo que si se produce congestión, el mínimo número de saltos ya no será una correcta medida de rendimiento [29].
- **Escalabilidad:** medida que indica la destreza de la red para operar ante cambios topológicos, manteniendo la calidad de los servicios. Es una medida importante en to-

pologías dinámicas y en redes que puedan aumentar su tamaño. Para la red mallada es una característica fundamental ya que la topología de la red puede cambiar y debería seguir ofreciendo las mismas calidades ofrecidas antes del cambio, por lo tanto, los protocolos de enrutamiento de las redes Ad Hoc pueden ser utilizados en las redes malladas inalámbricas con algunas consideraciones anteriormente mencionadas. Esto supone una gran ventaja debido a que la infraestructura está desplegada y las operadoras prefieren utilizar un protocolo que ya conocen (aunque modificado) que uno totalmente novedoso. Además si se trabaja sobre un protocolo ya creado se obtiene una base sólida de conocimientos y experiencias a partir de la cual trabajar [58, 2].

1.4. Necesidad de las redes 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 pueden enfrentar cambios en el ambiente y adaptarse a condiciones desconocidas [26]. 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 con el objetivo de mejorar la satisfacción mediante el proceso social [31]. En las redes auto-organizables se estudian los sistemas adaptativos y la capacidad de escalabilidad.

1.4.1. Computación adaptativa

En las redes de datos se han 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 [19]. Se sugiere construir modelos para procesar grandes cantidades de datos de gestión de red y 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.

Por otra parte, existe el reto de propender por la disminución o control de los costos de operación y mantenimiento de una red [4]. Se han enfocado recursos y atención en el diseño de redes con auto-organización, por lo que, se hace necesario estudiar y resolver los mecanismos

para lograr la auto-configuración automatizada [24].

Mark Weiser [76], 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 [68].

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 [5].

Más recientemente, los fabricantes Nokia-Siemens trabajaron en las redes del futuro con su proyecto SON (Self-Organizing Network) y 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 [38]; Los fabricantes y los investigadores asociados a universidades como Harvard, CALTECH y MIT presentaron varios modelos de autonomía en redes como son ANEMA de IBM, DA-VINCI [34], ANM e incluso un modelo gráfico (guiado o asistido por gráficos de color) [9]. 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 [1].

1.4.2. Modelos de Redes auto-organizables

Como se ha mencionado en la sección 1.2, 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.

1.4.2.1. 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 para gestión de la red y poder 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, por el contrario, siempre se necesita una tercera estación

que capture la traza de la conversación mediante una preselección de las características de comportamiento que se desean observar, enfocadas al comportamiento del usuario [9, 53].

1.4.2.2. 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, la cual satisface los objetivos de los usuarios. Al 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 [24].

1.4.2.3. 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) entre todas las redes virtuales, tercerizadas y las reales. Esto mediante la asignación de recursos dinámicos que pueden llevar a riesgos altos de estabilidad y utilización del modelo de asignación de recursos de manera adaptativa, probando que es estable y maximizando su desempeño. La aplicación de la red Davinci prueba la usabilidad y la adaptabilidad de las redes [34].

1.4.2.4. Modelo ANM (Autonomic Network Management)

Modelo de red que auto detecta, diagnostica, repara fallas, adapta su configuración, optimiza su desempeño y ofrece calidad de servicio. Desde hace tiempo han existido varios métodos de trabajo autónomos para resolver problemas de red, como el algoritmo de control de enrutamiento de estado de los canales de comunicaciones (link-state) conocido como Open Shortest Path First (OSPF) y Border Gateway Protocol (BGP). En otro escenario, se puede ver el control de congestión de ventana deslizante de las transmisiones de TCP, lo que promueve el uso de las redes autónomas aplicada a la explotación de las redes móviles LTE [11].

1.4.3. Características de las redes autónomas

Dado que existen muchos problemas y retos en la redes, desde sus componentes físicos (hardware) y su lógica (software y algoritmos) para la estabilidad de su topología y manejo de trabajo; se desarrolló una clasificación de las redes autónomas y las clases de soluciones aplicables al entorno de cada problema, presentadas como las redes autoconfigurables (caso particular de las redes autónomas (Autonomic Networks)).

Adaptabilidad. 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 [13].

Autoconfiguración. Corresponde al aumento de la confiabilidad y desempeño reduciendo costos con técnicas automatizadas [56]. Una red auto configurable se compone de elementos que automáticamente son provistos: los recursos de red están preparados para 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 [53].

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 [62].

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 entregue información de diagnóstico elaborada a las estaciones de gestión con acciones correctivas y con resultados predictivos [33].

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 basado en la cantidad de celdas sobrepuertas. Se ajustan de manera automática varios parámetros como tiempo de disparo, intervalos de medida, histéresis mejorando los resultados tradicionales del traspaso.

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 [29, 45].

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 son claves para la estabilidad, consistencia y desempeño [27].

1.5. Mensajería Instantánea-XMPP

1.5.1. Protocolo XMPP

El protocolo XMPP (eXtensible Messaging and Presence Protocol) es la formalización del protocolo de mensajería instantáneo desarrollado por la comunidad Jabber en 1999 [36]. Este usa tecnología de transmisión de Lenguaje de marcado extensible (XML por sus siglas en inglés) para el intercambio de elementos XML (stanza) entre dos entidades a través de una red. Una stanza XML es una unidad semántica de datos estructurados que existen en el nivel directo del elemento <*stream*>, que representa el inicio de una transmisión XML. El protocolo XMPP define tres núcleos principales [36]:

- **Presence:** Es un mecanismo de notificación de transmisión a través del cual, entidades pueden recibir información de conexión disponible de una identidad a las cuales están suscritos. Es de uso común por aplicaciones de mensajería instantánea para notificar cuando una entidad está en línea, lejana o fuera de línea.
- **message:** Es un mecanismo en el cual una entidad puede enviar información asincrónica de otra entidad. El núcleo mensaje es optimizado para distribución en tiempo real, pero también es compatible almacenamiento y distribución directa asincrónica.
- **iq (info/query):** Es un mecanismo que permite a las entidades realizar preguntas y recibir respuestas otra.

De forma general, el protocolo XMPP se encuentra basado en una arquitectura cliente-servidor de forma descentralizada pero con la extensión comentada XMPP, para establecer una conexión directa de extremo a extremo (P2P por sus siglas en inglés) y entre clientes [59].

En una sesión típica de XMPP, el cliente negocia un broadcast con un servidor, recupera su lista de contactos, notifica y recibe información presente de los contactos online y posteriormente puede iniciar un proceso de mensajería con uno o varios contactos, todo de forma segura y auténtica.

1.5.2. Características del protocolo XMPP

- La principal característica y ventaja con la que cuenta el protocolo XMPP es la disponibilidad pública, abierta y gratuita.
- El protocolo XMPP es estándar como una tecnología de mensajería instantánea e información entre dispositivos informáticos.
- El protocolo XMPP es descentralizado, esto debido a que su arquitectura es similar a la del correo electrónico por lo que su uso queda garantizado en servidores propios.

- El protocolo XMPP es seguro, esto debido a que cualquier servidor XMPP puede ser aislado de una red pública y utilizar protocolos de seguridad adicionales.
- El protocolo XMPP es extensible. XMPP fue creado aproximadamente en el año 1988 y ha sido refinado en el transcurso de la historia a través de nuevas extensiones de uso.
- El protocolo XMPP es flexible, por lo que ha sido utilizado desde aplicaciones de mensajería y presencia, hasta llegar a administración de redes, herramientas de colaboración, juegos, servicios web, computación en la nube, entre otros [36, 59].
- El protocolo XMPP es muy estable para aplicaciones en internet, por lo que ha sido usado en servicios públicos como Google Talk [59].
- La disponibilidad del protocolo XMPP, permite la cooperación entre desarrolladores y empresas, lo que conlleva a que este pueda ofrecer funciones adicionales a mensajería, como lo es la transferencia de listas de contactos, archivos, conversaciones simultáneas, entre otros.

A pesar de que este protocolo cuenta con una variedad de ventajas, también posee algunos defectos que son muy importantes, tales como [59]:

- **Sobrecarga:** Puede existir una sobrecarga de datos de presencia en servidores motivo del tráfico entre ellos (70 %) y alrededor del 60 % en transmisiones redundantes.
- **Ausencia de datos binarios:** El protocolo XMPP está basado en transmisión de datos de 64 bits de forma codificada como único documento XML, lo que genera la imposibilidad de compartir datos de forma binaria.

Clients XMPP

Algunos clientes que implementan el protocolo XMPP son[59]:

- **Windows:** Psi, Gajim, Jabbin, Miranda IM.
- **GNU/Linux:** Ayttm, Jabber.el, Finch [20], Mcabber, GNU Freetalk.
- **Mac OS:** Pidgin, ichat, Center IM, Tkabber, Coccinella.
- **Móviles:** Jabber Mix Client(JME), Talkonaut.

1.5.3. Mensajería instantánea (IM)

La mensajería instantánea es una comunicación en tiempo real, a menudo conocida como chat, entre dos o más participantes usando textos, transferidos vía red. Varias aplicaciones de IM están disponibles como: MSN, Yahoo Messenger, ICQ, AIM y Skype[36]. Sin embargo, estos sistemas son patentados y centralizados.

Por otro lado, XMPP es un protocolo abierto y descentralizado, diseñado para una comunicación eficiente entre pares en tiempo-real, ver figura 1-2. Este hace uso de mecanismos <presence> para estar al tanto de la disponibilidad de correspondientes particulares antes de intercambiar mensajes. Las extensiones básicas de IM proporcionan asistencia para la transferencia de datos, distribución demorada y mecanismos para la captura de presencia extendida de datos como geolocalización, actividad, estado anímico y melodías. Esto ha permitido la negociación de sesiones de multimedia lo que ha generado que XMPP haya sido distribuido en servicios como GoogleTalk y el servicio de chat de Facebook [36].

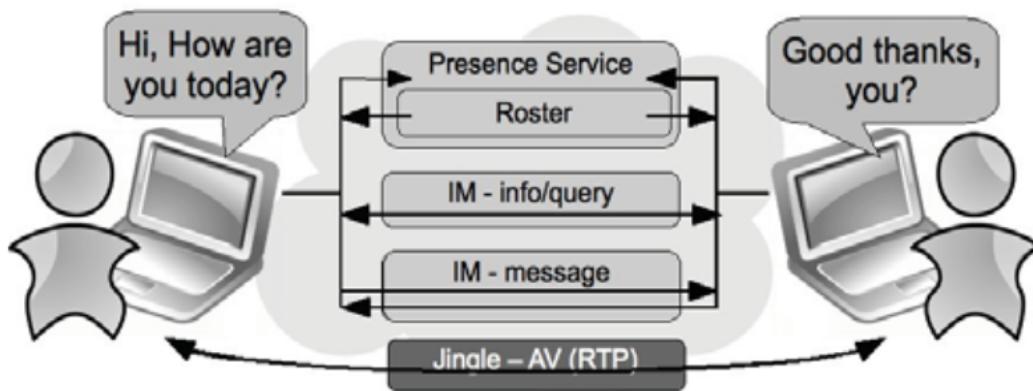


Figura 1-2.: Mensajería instantánea con XMPP [36].

1.6. Agrupamiento de Nodos y Clúster

Una Red Ad-Hoc es un sistema de organización múltiple que pueden comunicarse entre sí sin necesidad de una infraestructura preexistente. En la literatura, se han propuesto dos tipos de estructuras topológicas: topología de estructura plana y estructura de agrupamiento jerárquico [52, 75]. La estructura plana resulta inestable en redes móviles debido a las grandes redes Ad Hoc y a los movimientos de los nodos que pueden cambiar la topología de red. La estructura de agrupamiento jerárquico los nodos están separados en grupos conocidos como clústers, donde generalmente hay tres tipos de nodos en redes de agrupamiento, ver Figura 3-4:

- **Clusterheads (CHs):** En cada clúster, un nodo es elegido como Head (CH) para actuar como controlador local. El tamaño del clúster (número de nodos) depende del rango de transmisión de los nodos en el clúster.
- **The gateway node:** El nodo de puerta de enlace pertenece a más de un clúster y conecta los CH de los grupos. Estos forman una red troncal, aunque la presencia del nodo de puerta de enlace es opcional.
- **The normal node:** El nodo normal transmite datos al CH que transfiere estos paquetes recopilados al siguiente salto.

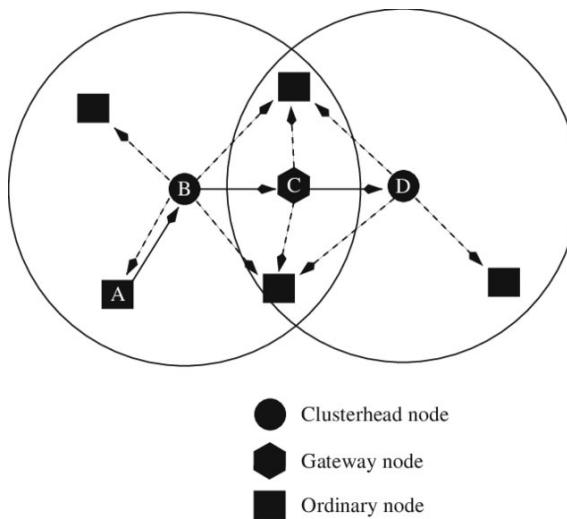


Figura 1-3.: Red de agrupamiento (clustering) [52].

Al agrupar redes Ad Hoc normalmente se usan protocolos de enrutamiento. La conectividad de un clúster solo contiene un pequeño número de nodos y es mantenido por el intercambio de información periódicamente y a través de actualizaciones de información entre nodos vecinos sobre cambios de enlaces. Teniendo en cuenta lo anterior, cuando un nodo envía datos a su CH se utiliza un protocolo de ruta proactivo, basado en una ruta de tablas. No obstante, si el nodo de destino es en un clúster diferente, el CH al cual pertenece el nodo, necesita descubrir la ruta de la red troncal para que el enrutamiento entre clúster sea a demanda, a esto se conoce como enrutamiento reactivo [75].

1.6.1. Clasificación de esquemas de agrupamiento

Hay cuatro formas de clasificar los esquemas de agrupamiento, según el uso de uno o varios saltos, sincronización de la red, nodos móviles o estacionarios o si los esquemas se basan en la información sobre la ubicación de los nodos, ver Figura 1-4.

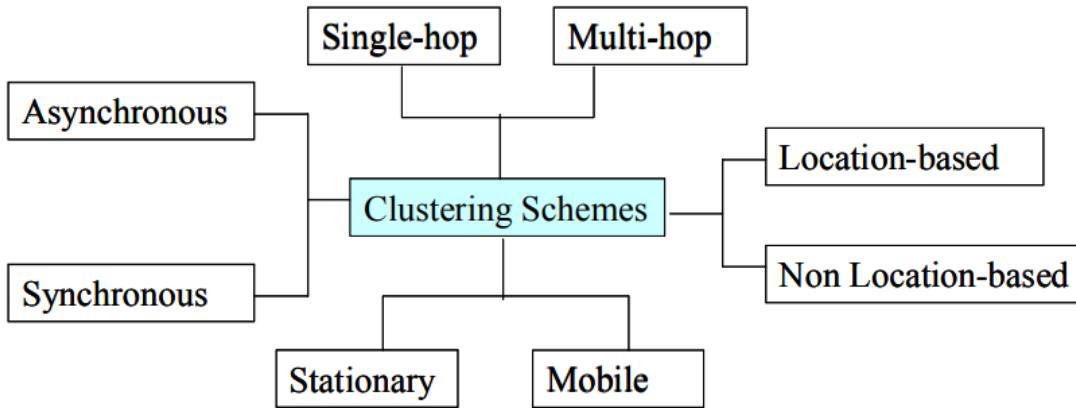


Figura 1-4.: Diferentes maneras de clasificar esquemas de agrupamiento[75].

1.6.2. Esquemas de agrupamiento para redes Ad-Hoc móviles

El movimiento de los nodos en las redes Ad-Hoc puede cambiar rápidamente la topología, lo que puede dar lugar a gastos generales que aumentan dramáticamente. Los esquemas de agrupamiento por lo tanto, tienen como objeto manejar el mantenimiento de la topología, reducir los mensajes de gastos generales o gestionar el movimiento de los nodos [75].

Red troncal de agrupamiento dinámico

Una red troncal de agrupación dinámica puede cambiar de forma adaptable la topología de la red causada por movimiento de nodos. Esta por lo tanto, puede llevar a cabo la formación de clústers y la topología de mantenimiento.

En [46] se establece un esquema de agrupamiento denominado Protocolo de Red troncal Virtual Dinámico (VDBP por sus siglas en inglés). En VDBP, la red troncal dinámica es construida por nodos de un subconjunto de nodos. Aquellos nodos son nodos de la red troncal o vecinos de los nodos de la red troncal, y cualquier par está conectado a través de otra red troncal de nodos.

El VDBP, que tiene tres fases, tiene la intención de proporcionar una red troncal conectada para entornos de red altamente dinámicos. La primera fase, consiste en la selección de un número mínimo de nodos que domina toda la red usando sólo un salto de información del vecindario. Por lo que, los nodos que son relativamente menos en esta fase se seleccionarán un móvil. La segunda fase, construye la red troncal conectando los nodos seleccionados en proceso de selección a la Red troncal y manteniendo la conectividad.

Sin embargo, debido al movimiento de los nodos, los nodos de la red pueden cambiar. La tercera fase, llamada proceso de mantenimiento de la red troncal, reduce los nodos redundantes o los nodos obsoletos de la red troncal para el mantenimiento de esta.

Reducción de gastos

La topología de la red cambia debido al movimiento de los nodos. En las redes de agrupación, se aplica una ruta a demanda en comunicación entre grupos. Los cambios de topología entonces hacen que las inundaciones para el descubrimiento de rutas a pedido sean más serias, lo que aumentará los gastos de mantenimiento de la topología. La agrupación pasiva [48] y el esquema de acceso [72] tratan de reducir los gastos generales. Una agrupación pasiva (PC) es un protocolo de formación de agrupaciones que no utiliza paquetes de control específicos del protocolo o señal [48].

Administración de movimiento

Los nodos móviles abandonarán sus grupos y se unirán a nuevos grupos. El mantenimiento de topología de la red más convenientemente ocurre si es posible predecir la movilidad de los nodos. La predicción de movimiento de los nodos puede ayudar a los diseñadores a manejar el mantenimiento de la topología de manera más eficiente, sin embargo este tipo de tecnología necesita la información de ubicación de los nodos por Tecnología de sistema de posicionamiento global (GPS) o mediante cálculo, lo que aumentará el costo de las redes.

1.6.3. Algoritmos de agrupamiento para redes Ad-Hoc

Los algoritmos de agrupamiento para Redes Ad Hoc, permiten mejorar la calidad en las comunicaciones, el enrutamiento y el control de la topología. Estos métodos permiten reorganizar la red mediante la generación de pequeños grupos de nodos geográficamente adyacentes y conectados entre sí, llamados clústeres. Los algoritmos de agrupamiento tienen como objetivo:

- Proporcionar una utilización más racional de los recursos aumentando la capacidad del sistema. Una asignación adecuada de las tareas permite mejorar el rendimiento de los protocolos de acceso al medio de transmisión, la escalabilidad de la red y una mejor utilización de la energía disponible por los nodos [51].
- La existencia de varios clúster conectados entre sí, mejoran la coordinación de los nodos en la transmisión de paquetes. La generación de rutas más estables reducen el tamaño de las tablas de enrutamiento así como la necesidad de actualizarlas constantemente [40].
- Un clúster genera una visión más simple de la red desde el punto de vista del nodo. El número de miembros de un clúster es menor al total de nodos en la red, como

consecuencia la información que el nodo debe almacenar sobre tablas de enrutamiento y nodos vecinos se ve reducida a una fracción de la red [58].

- Un clúster puede asignar funciones a sus nodos según las necesidades de la red. Los roles más comunes son como clusterhead encargado de administrar las comunicaciones y como gateway ayudando a establecer comunicación con otros clústeres [2].

- La existencia de redes heterogéneas hace necesaria la creación de una estructura jerárquica en la organización de la red. Es necesario asignar tareas a los nodos de manera racional teniendo en cuenta su nivel de recursos disponible. Esta distribución tiene como objetivo encontrar un equilibrio entre el nivel de exigencias del servicio y las restricciones de recursos en la red [2].

Mantener un clúster bajo condiciones dinámicas puede consumir grandes cantidades de recursos. Los algoritmos de agrupamiento se dividen en dos etapas: La primera se encarga de generar los clúster y asignar roles a los nodos; la segunda se encarga del mantenimiento del clúster y tiene que ver con el manejo de eventos especiales como la salida de un nodo o la necesidad de reelegir un nuevo clusterhead. Este tipo de eventos puede afectar el rendimiento de la red debido al gasto adicional de recursos que implica rehacer los clústeres. En la tabla 1-1 se muestran los esquemas de agrupamiento más utilizados con sus principales objetivos. Para los objetivos de este trabajo, se hizo énfasis en los esquemas de agrupamiento para balanceo de carga (Load Balancing Clustering) y combinación de métricas (Combined-Metrics-Based Clustering) [68].

Los Algoritmos de Agrupamiento Bajo Combinación de Métricas, tienen en cuenta factores como la capacidad de almacenamiento, grado de movilidad, número de vecinos y energía disponible por los nodos. Su objetivo principal es generar una combinación de estos parámetros en un solo valor que se convertirá en el criterio para clasificar los nodos según sus capacidades. Una de las ventajas que presentan este tipo de algoritmos es que permiten ajustar las métricas dependiendo del escenario en que se encuentre la red. Entre los algoritmos más representativos encontramos WCA (Weighted Clustering Algorithm) y algunas mejoras propuestas, en donde se mantiene a los nodos con bajos niveles de energía por fuera de la elección del clusterhead minimizando así el número de candidatos [66, 78].

Nombre	Objetivo
DS-Clúster	Busca los nodos dominantes en las rutas de comunicación para ayudar a mantener la tabla de enrutamiento.
Low-Maintenance Clustering	Se enfoca en las capas superiores del modelo OSI buscando el mínimo costo consumo de recursos.
Mobility-Aware Clustering	Utiliza los grados de movilidad en los nodos intentando evitar cambios repentinos en la estructura de la red.
Energy-Efficient Clustering	Evita el consumo innecesario de energía.
Load Balancing Clustering	Realiza un control en la carga de trabajo de la red mediante unos límites en el número de nodos que puede tener un clúster.
Combined-Metrics-Based Clustering	Utiliza diferentes métricas para su configuración consumo de potencia, grados de movilidad, tamaño de los clústeres. Realiza una ponderación de estos valores para agrupar los nodos.

Tabla 1-1.: Algoritmos de agrupamiento

1.7. Antecedentes e Implementación de redes

Realizando una verificación académica en búsqueda de implementaciones reales de redes Ad Hoc en el territorio mundial las cuales incluyeran prototipo, se encontró variedad de proyectos de investigación realizados a nivel internacional, mostrando los avances investigativos sobre redes Ad Hoc [55].

En la Universidad Politécnica de Catalunya en España, se desarrolló una Tesis de Grado en la que se implementó, una red Ad Hoc entre vehículos de una compañía de transporte, cuyo propósito era validar la frecuencia con la que debían salir los vehículos de una ruta de servicio público específica y para la cual los nodos de la red Ad Hoc enviaban información del tráfico del área geográfica cobijada por el sistema [22]. Esto demostró que las Redes Ad Hoc son aplicables a sistemas de redes de sensores, y que a su vez poseen compatibilidad con la tecnología GPS [22]. Evidenciadas estas bondades, esta misma institución educativa decidió profundizar en la investigación inicial sobre las redes Ad Hoc, realizando nuevos trabajos de investigación [57].

Uno de estos proyectos se concentró en simular la asignación de canales en redes Ad Hoc bajo el estándar 802.11 multi-radio, creando enlaces entre puntos. Las redes inalámbricas establecidas, brindaron acceso a internet manejando topologías dinámicas y desligándose de

un conector físico [23].

El siguiente proyecto desarrollado, indagó acerca del estudio y análisis de una red Ad Hoc, en donde los nodos de la red se definieron como sensores inalámbricos y en la cual se buscó que funcionaran punto a punto, sin necesidad de infraestructura pre-existente para operar [74].

Finalmente la Universidad de Catalunya, diseñó e implementó un servidor de video adaptativo en un simulador de redes Ad Hoc, brindando información acerca del manejo del protocolo UDP, como protocolo de transporte para la transmisión de tramas de vídeo sobre redes móviles MANET. Esta simulación ostentó movilidad y flexibilidad mayor que las redes fijas. Además, minimizó la gestión de la propia red al configurarse automáticamente sus nodos, sin necesidad de un elemento centralizador [57].

En el continente Asiático, se ejecutó un estudio en la Jawaharlal Nehru University situada en la India, planteándose una simulación de enrutamiento geográfico para información transmisible por redes Ad Hoc vehiculares (VANETs). En este estudio, se sugirieron diferentes protocolos de enrutamiento a distancia geográfica (GEDIR) en la literatura [47].

Para este mismo país, resultó interesante las aplicaciones de las redes Ad Hoc, y por ello, en la VFSTR University, se simuló una verificación de las ubicaciones de los nodos de una red Ad Hoc Móvil (lo cual es un problema importante en estas redes Ad Hoc) encontrándose la presencia de adversarios destinados a dañar el sistema. Gracias a esto, se verificó el protocolo llamado Posición Vecino (VAN), a fin de comprobar la situación de los vecinos de comunicación y detectando los adversarios de la red Ad Hoc móvil. Este protocolo se utilizó finalmente para el intercambio de mensajes y para validar la posición de la comunicación de los nodos en la red [66].

Otra de las investigaciones del viejo continente, se dieron en la Universidad de Bologna situada en Italia, simulando un estudio sobre las redes inalámbricas móviles Ad Hoc (MANETs), en donde las infraestructuras virtuales dinámicas y adaptativas se obtuvieron por protocolos de agrupación distribuidos, los cuales resultaron ser útiles para soportar comunicaciones más confiables, eficientes en gestión de red y con alta utilización de recursos [6].

Posterior a esto, y citando nuevamente al continente asiático, este decide ahondar en el estudio de las MANETs, en esta ocasión, la Sharif University of Technology situada en Irán, planteó un estudio sobre la Teoría del Proceso de red computarizada bajo protocolos aplicables a redes Ad Hoc móviles (MANETs). Aquí, se exploró la aplicabilidad en dos aspectos: la comprobación del modelo y el razonamiento matemático. La semántica operacional del marco se basó en los sistemas restringidos etiquetados en transición (CLTSs), en el cual,

cada etiqueta de transición estuvo parametrizada con el conjunto de topologías [32].

El continente Americano no se quedó atrás, en Canadá, exactamente en Concordia University, se realizó un estudio sobre la utilización eficiente del ancho de banda y sobre las altas velocidades de datos que impactan el rendimiento de las redes inalámbricas Ad Hoc. Aquí las estaciones vecinas actuaron como nodos de retransmisión, con el fin de transferir los datos de origen, al nodo destino deseado, por medio de un canal de relé independiente. Sin embargo, esto se logró a expensas de perder algunos recursos inalámbricos tales como el ancho de banda. Además, el uso de antenas direccionales demostró ofrecer una manera eficaz para utilizar el ancho de banda eficientemente, y en consecuencia, se realizó un nuevo protocolo de acceso de canal para la transmisión a través de redes Ad Hoc de cooperación. El rendimiento de la red se modeló, utilizando cadenas de Markov continuas. El estado estacionario, la probabilidad de bloqueo de transmisión y el rendimiento de la red promedio, se obtuvieron, mediante el análisis del modelo de Markov derivado [3].

Simultáneamente en New Jersey, el Institute of Technology situado en USA, presentó una métrica para evaluar el desempeño de los sistemas de acceso al medio en redes inalámbricas Ad Hoc, conocidos como la capacidad local. En este artículo, el modelo de red básica y las herramientas analíticas, se aplicaron a una red sencilla, con el fin de derivar la capacidad local de los diversos esquemas de acceso al medio. Además, se analizaron los esquemas de patrón de cuadrícula en donde, transmisores simultáneos se colocaron en un patrón regular de la red. Se analizaron los esquemas de ALOHA, para lo cual transmisores simultáneos, expiraron de acuerdo a una distribución uniforme y exclusión a través de esquemas de Poisson [78].

Basado en los estudios y simulaciones anteriormente relacionados, se concluye que el diseño de la topología de una red Ad Hoc, va de acuerdo a los lineamientos establecidos para cumplir un objetivo de conexión. Finalmente, para realizar la implementación de estas redes se debe garantizar conectividad entre todos los nodos de la red (bajo la posibilidad de utilizar cualquier dispositivo como nodo intermedio de conmutación), y por otra parte, conseguir que el consumo total de las baterías en los dispositivos sea mínimo [50].

Finalmente los investigadores de redes de telecomunicaciones indican que en el presente, las formas de comunicarse han pasado por muchas fases y el gran auge de las comunicaciones ha ayudado al desarrollo de nuevas vías y a la renovación de otras ya existentes. La mensajería instantánea es una prueba de ellos, como una tecnología ya existente en los SMS puede transformarse y pasar a formar parte de una acción más del día a día con Aplicaciones como WhatsApp, Telegram, Line, Hangouts, u otras [67].

Para su puesta en producción el protocolo libre y de mayor inter-operatividad, es el protocolo XMPP (Protocolo extensible de mensajería y comunicación de presencia). Siendo un pro-

tocolo abierto y extensible basado en eXtensible Markup Language (XML) y originalmente ideado para mensajería instantánea [2, 67].

Con el protocolo XMPP queda establecida una plataforma para el intercambio de datos que puede ser usada en aplicaciones de mensajería instantánea. Las características en cuanto a adaptabilidad y sencillez del XML son heredadas de este modo por el protocolo XMPP [67].

A diferencia de los protocolos propietarios de intercambio de mensajes como ICQ, Yahoo y Windows Live Messenger, se encuentra documentado y se insta a utilizarlo en cualquier proyecto. Existen servidores y clientes libres que pueden ser usados sin costo alguno. Obviamente, la limitación del tamaño de la red hace que sea imposible mandar mensajes a gente en otros lugares, pero sería de gran utilidad en caso de grandes eventos, festivales, conciertos o catástrofes, donde no hay señal por colapso o directamente se pierden las instalaciones de comunicaciones [67].

1.8. Problema de investigación

Uno de los principales problemas que enfrentan las redes de comunicaciones inalámbricas actualmente, es el alcance de cobertura geográfica ofrecido; el cual es limitado, debido a las rutas estáticas de sus nodos y la dependencia a un nodo principal, quien adicionalmente enmarca el enrutamiento de la red [24, 28, 33, 35].

Esto afecta la variedad de servicios que viajan por este tipo de redes, como lo son correo electrónico, SMS, Internet, entre otros. Debido a la latencia y desconexión de los nodos al momento de sobrepasar el área de cobertura genera pérdida de información y dependencia para reintentar la conexión [28, 33, 35, 42, 61, 64, 71].

Con los modelos de referencia y la clasificación de los tipos de red autónomas, es posible contrarrestar estas características y optimizar el diseño de las redes de comunicaciones inalámbricas Ad Hoc [12, 30, 44, 70, 75].

Algunas de las causas de la falta de cobertura, es el hecho de que estas redes fueron construidas a partir del modelo de referencia de arquitectura en capas, que no integran distintas tecnologías de acceso al medio inalámbrico y que no se controla la asignación de tareas específicas [7].

Este hecho hace que el usuario califique las redes de comunicaciones inalámbricas como burócratas, de baja señal, lentas, de baja potencia y asocien la baja calidad de servicio a enlaces inalámbricos por hechos que son netamente atribuibles a la configuración dada por

el administrador de la red [42, 44, 45, 61, 70].

Tomando la argumentación anterior se puede plantear la siguiente pregunta problema: ¿La implementación de un prototipo de red Ad Hoc tipo Clúster, mejorará el desempeño de las redes inalámbricas convencionales instaladas en el campus Universitario de la Universidad Nacional, en características de rendimiento como cobertura, movilidad y heterogeneidad?. En el desarrollo de este trabajo de investigación se respondió esta pregunta y se presentan los procesos y técnicas que se utilizaron para la implementación del proyecto, verificando su funcionalidad.

1.9. Objetivos

1.9.1. Objetivo General

Implementar un prototipo de Red Ad Hoc tipo clúster dentro del campus Universitario de la Universidad Nacional.

1.9.2. Objetivos específicos

- Construir los Clústers Head de la Red Ad Hoc.
- Configurar las características de Hardware de los nodos que integran el clúster del prototipo.
- Configurar las características de Software de los nodos que integran el clúster del prototipo, y las cuales permitan establecer los parámetros técnicos de conexión de la red.
- Implementar un servicio de mensajería instantánea usando el protocolo XMPP, dentro del prototipo de Red Ad Hoc.

1.10. Justificación

Finalmente un reto abierto de la comunidad científica y de ingeniería de telecomunicaciones es la tendencia en las redes de comunicaciones inalámbricas Ad Hoc y corresponde a implementar características de inteligencia sobre ellas con el fin de optimizar, regular, descongestionar, auto organizar y dinamizar los nodos de la red para finalmente resolver los problemas de cobertura y movilidad existentes en la actualidad [2, 11, 13, 18, 29, 58, 66]. Una forma de ampliar la cobertura y dar movilidad a las redes de comunicaciones inalámbricas, ha llevado a los diseñadores de redes a formular cambios en la centralización de los

clientes o nodos de una red, volviéndolos auto organizables. Gracias a la asignación de tareas específicas y partiendo de la premisa de que todos los nodos de la red, tienen la misma importancia sobre la misma [2, 16, 17, 24, 45, 75].

1.11. Alcances y limitaciones

- El prototipo a implementar se desarrolló para interconectar las áreas geográficas del campus Universitario, afectadas por falta de cobertura de redes inalámbricas.
- La Heterogeneidad de los nodos que integran los Clúster de la Red Ad Hoc propuesta, se encuentra delimitada para equipos con Sistema Operativo Android y todo aquel que sea compatible con el Kernel de Linux.

2. Características de los Nodos, Protocolos y Servicios del Prototipo de Red Ad-Hoc

2.1. Raspberry Pi

Raspberry Pi es una placa de computación pequeña, potente, económica y orientada a la educación introducida en 2012. Su funcionamiento es igual al de un computador estándar, por lo que requiere de un teclado para ingresar comandos, una unidad de visualización y una fuente de alimentación. Además, posee un tamaño que la hace fácilmente transportable y muy útil en interfaces de varios dispositivos [54]. La mayoría de modelos de placas, poseen características similares, entre las que se pueden destacar: la Unidad Central y Gráfica de procesamiento (CPU y GPU respectivamente), hardware de comunicación y audio, así como otros componentes esenciales (procesador, chip gráfico, memoria, RAM) y otros más opcionales (conectores periféricos), ver figuras 2-1 y 2-2.

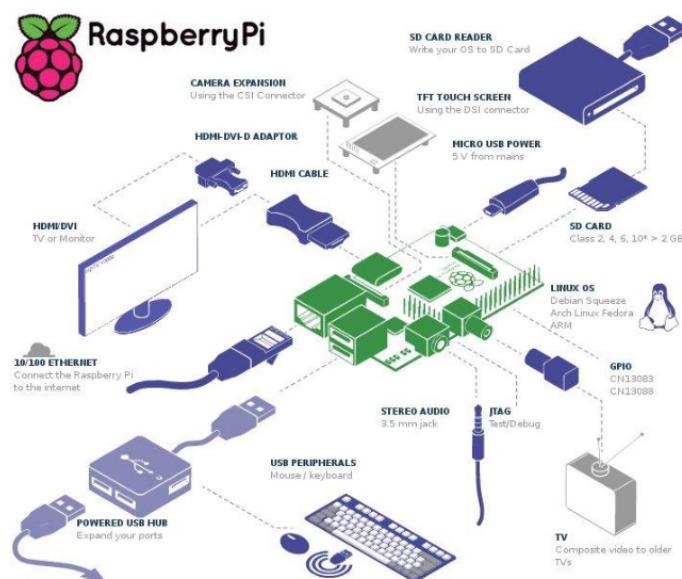


Figura 2-1.: Algunos componentes centrales en una placa Raspberry Pi[54].

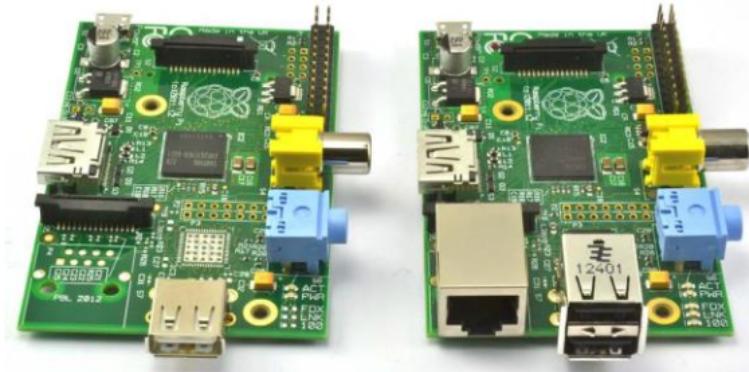


Figura 2-2.: Placa Raspberry Pi modelo A (izquierda) y modelo B (derecha)[54].

Debido a que el comportamiento del Raspberry Pi es igual al de una computadora, requiere de un sistema operativo. Linux (Raspbian) es ampliamente usado en vista de que es un sistema operativo de código abierto y gratuito [54]. Algunos modelos de Raspberry Pi comerciales pueden ser observados en la Tabla 2-1.

Características	Raspberry Pi A+	Raspberry Pi B	Raspberry Pi 2	Raspberry Pi 3
Precio aproximado	35 USD	35 USD	35 USD	35 USD
Tipo de núcleo	ARM1176JZF-S	ARM1176JZF-S	Cortex-A7	Cortex-A53 64-bit
No. de Núcleos	1	1	4	4
RAM	256 MB	512 MB	1 GB	1 GB DDR2
Reloj CPU	700 MHz	700 MHz	900 MHz	1.2 GHz
Rangos de Potencia	200 mA	700 mA @ 5V	800 mA	1.34 A @ 5V

Tabla 2-1.: Características de algunos modelos Raspberry comerciales [63].

Una de las desventajas que posee la placa Raspberry Pi es que no admite la entrada de un dispositivo analógico, esto debido a que no posee convertidor ADC. Para contrarestar esta problemática, Raspberry Pi puede extenderse a través de un conjunto de dispositivos que permitan la entrada de dispositivos o sensores analógicos. Otra desventaja radica en que el dispositivo cuenta con solo un pin que permite hardware PWM, la solución a este inconveniente se puede dar de dos maneras: usar el software PWM o expandir la Raspberry Pi con circuitos PWM que usan tecnología SPI. Sin embargo, estas alternativas pueden ser obsoletas en sistemas de muy alta precisión.

En términos de sistemas operativos, Raspberry Pi puede ejecutar SO basados en Linux, Android, RISC OS y Windows 10. Los lenguajes de programación disponibles son C/C++,

Node.js, Shell-script, Scratch, Java, Python y otros lenguajes presentes en el sistema operativo Linux.

2.2. Raspbian

Un sistema operativo consiste en un conjunto de programas básicos y utilidades que hacen que un dispositivo inteligente funcione. Raspbian es un sistema operativo basado en Debian optimizado para el hardware Raspberry Pi, abierto al público. Además de un sistema operativo, Raspbian proporciona más de 35000 paquetes para un mejor rendimiento del Raspberry Pi, por lo este motivo, Raspberry Pi Foundation lo ha definido oficialmente como el sistema operativo primario para la familia de placas SBC de Raspberry Pi [37].



Figura 2-3.: Logo de Raspbian [37].

En términos técnicos, el sistema operativo es un port no oficial de Debian para el procesador CPU para la placa Raspberry Pi con soporte optimizado para cálculos en coma flotante por hardware, lo que permite mayor rendimiento. Otro factor importante corresponde al menú “raspi-config”, que permite la configuración del SO sin tener que modificar archivos manualmente. Entre sus funciones, puede expandir la partición root (del disco) para que ocupe toda la tarjeta de memoria, puede también aumentar la frecuencia del reloj y la configuración de teclado.

2.3. Pacemaker

Un Pacemaker es un gestor de recursos de clústers de alta disponibilidad. Logra su máxima disponibilidad para sus servicios de clúster (recursos) al detectar y recuperarse de fallas de nivel de nodo y recurso, haciendo uso de las capacidades de mensajería proporcionada por Corosync. Esto es posible con clústers de cualquier tamaño y lleva consigo un poderoso modelo de dependencia que le permite al administrador expresar las relaciones de forma precisa (tanto la ubicación como el orden) entre los recursos del grupo [21]. Algunas configuraciones de nodo comunes que son posibles de configurar con Pacemaker son:

- **Configuración Active/pasivo:** Clúster activo/pasivo de dos nodos con Pacemaker y DRBD

- **Configuración N+1:** Al soportar muchos nodos, el Pacemaker puede reducir los costos del hardware al permitir que varios clústers activos/pasivos se combinen y comparten un nodo de respaldo común.
- **Configuración N to N:** Cuando el almacenamiento compartido es disponible, cada nodo puede usarse para la conmutación por error. Pacemaker puede inclusive distribuir carga de trabajo.

2.3.1. Características

Algunas características de pacemaker son [21]:

- Detección y recuperación de fallas a nivel de máquina y aplicación.
- Permite cualquier configuración de redundancias.
- Soporte de clústeres basados en recursos.
- Admite aplicaciones que deben estar activas en varias máquinas.
- Admite aplicaciones que deben/no deben ejecutarse en la misma máquina.
- Permite aplicaciones con diferentes modos, por ejemplo, aplicaciones con configuraciones maestro-esclavo.
- Admite pedidos de inicio/apagado de aplicaciones, de forma independiente de las máquinas que se encuentran en las aplicaciones.

Pacemaker fue creado en el 2004 a partir de un esfuerzo de colaboración de la comunidad ClusterLabs y desarrolladores de Red Hat y SUSE [21]. Pacemaker se distribuye con la mayoría de distribuciones de Linux y se ha implementado en muchos entornos críticos.



Figura 2-4.: Logo de Pacemaker[21].

2.4. Corosync

Corosync Cluster Engine es un sistema de comunicación grupal con algunas características adicionales para implementar alta disponibilidad en las aplicaciones. Este proporciona cuatro características principales de la interfaz de programación de aplicaciones C/C++ [21]:

- Un modelo de comunicación de grupo de procedimientos cerrados con sincronización virtual que garantiza la coherencia de máquinas de estado replicadas.
- Una base de datos de configuración y estadísticas en memoria que proporciona la capacidad de establecer, recibir y recuperar notificaciones de cambio de información.
- Un administrador de disponibilidad simple que reinicia el proceso de solicitud cuando este falla.
- Un sistema de quorum que notifica a las aplicaciones cuando se alcanza o se pierde el quorum.

Corosync se utiliza como marco de alta disponibilidad en proyectos como Pacemaker y Apache Qpid.

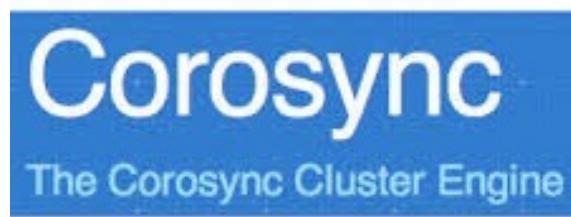


Figura 2-5.: Logo de corosync [21].

2.4.1. Características

- Soporte para redes Ethernet.
- Soporte para IPv4 e IPv6.
- Soporte de autenticación y cifrado.
- Soporte para envío de mensajes.
- Soporte para interfaces redundantes.
- Diagnóstico y análisis de fallas.
- Envío de mensajes multicast y unicast.

2.4.2. Clúster Corosync-Pacemaker

Las características de Corosync, tales como: funcionalidad de estructura de clúster, mensajería, membresía y soporte de la información del quorum, han sido utilizadas por Pacemaker para proporcionar una solución de alta disponibilidad [21]. Los componentes clave de la configuración del clúster corosync-pacemaker son:

- Base de información del clúster (CIB).
- Demonio de gestión de recursos de clúster (CRMd).
- PEngine (PE).
- configurar STONITH (del inglés Shoot The Other Node In The Head).

Los recursos son supervisados según la configuración del CIB. El demonio CRMd en los nodos DC (Designated Coordinator) toma decisiones e inicia comandos. El nodo DC hace referencia al PE y toma la decisión correspondiente. Por ejemplo, un clúster de dos nodos que está configurado para proporcionar el servicio web Apache, realizará las siguientes acciones:

1. Se formará una nueva membresía con nodos disponibles y se elige uno como DC.
2. El PE y el CIB estarán disponibles únicamente en este nodo.
3. El nodo DC verifica el estado del quorum, si se alcanza, procede a continuar el proceso.
4. El CRMd en el nodo DC inicia el comando de inicio de recursos y el administrador de recursos locales en cada nodo recibe la operación, la inicia y reporta el estado al CRMd.
5. El CRMd actualiza CIB y valida el estado, si el resultado no es el esperado, el PE es verificado y se procede con la recuperación o el reinicio.
6. El paso 4 y 5 continúa.
7. La transición de comando y estado utiliza la capa de mensajería de corosync subyacente.
8. Cuando el nodo DC falla, el siguiente nodo disponible es asignado automáticamente.

2.5. Ejabberd

Jabber es un conjunto de protocolos XML de streaming (flujos de descarga) y tecnologías que permiten que dos puntos o entidades en Internet intercambien información de presencia, mensajes y otra información estructurada en tiempo-real o cercanos a este. Jabber tiene como

característica principal, la posibilidad de interpretación por diferentes sistemas operativos y plataformas. Además, brinda soporte SSL (Secure Socket Layer) para comunicaciones cliente/servidor y para algunos clientes soporta la extensión GPG (GNU Privacy Guard) para cifrar comunicaciones punto a punto usando modelo asímetrico [10].

Por otro lado, ejabberd es un servidor Jabber multiplataforma. Este es desarrollado en el lenguaje Erlang, que es un lenguaje funcional utilizado en el desarrollo de aplicaciones distribuidas y que cuenta con soporte para tolerancia a fallos. Ejabberd cuenta con un soporte total de las características del protocolo Jabber [10, 60].



Figura 2-6.: Logo de Ejjaberd [60].

2.5.1. Características

■ Plataforma moderna

- * Soporte multiprotocolo (XMPP, MQTT, SIP).
- * Bibliotecas móviles para iOS, Android.
- * Software de código abierto (Github).

■ Tolerante a Fallas

- * Se ejecuta en un clúster fuera de caja.
- * Todos los nodos están activos.
- * Actualizable mientras se ejecuta.
- * No se pierde ningún mensaje.
- * Desconexiones de red móvil gestionadas a nivel de servidor XMPP.

- **Masivamente Escalable**

- * Simultaniedad masiva (2'000.000 usuarios en un nodo).
- * Clústers con hashing consistente para deshacerse de la replicación de datos.
- * Reducción de la memoria de los datos guardados en la memoria para cada usuario conectado.
- * Optimización de difusión de PubSub con reducción de la transferencia interna de datos en la configuración del clúster.

- **Manejable:** El servidor Ejaber XMPP trae consigo una herramienta de línea de comandos que permite controlar la mayoría de los aspectos del servidor. Las operaciones más comunes se pueden automatizar.
- **Versátil:** El servidor Ejaber XMPP ofrece muchos ganchos de personalización para adaptarlo fácilmente a varios casos de uso. Esto lo convierte en el servidor de elección, sin importar en qué área esté trabajando: mensajería móvil, juegos, Internet de las cosas (IoT), puede hacerlo todo.
- **Modular:** El código del servidor Ejaber XMPP es modular y puede ampliarse a través de una API.

2.6. Finch

Finch es un cliente de mensajería instantánea de código abierto basado en la biblioteca libpurple. Libpurple posee compatibilidad con varios protocolos de mensajería instantánea de uso común, lo que le permite al usuario, el acceso a diferentes servicios desde una sola aplicación [20].

Este programa es de software libre, que puede ser distribuido y modificado bajo los términos de la licencia. Es bastante similar a Pidgin, que consiste en un programa de chat que permite iniciar sesión de forma simultánea y es compatible con Jabber/XMPP, lo que permite la transferencia de archivos, mensajes, notificaciones, entre otros. El uso de finch no es complejo y algunos comandos básicos son:

- Seleccionar la lista de opciones disponible (Alt+A).
- Siguiente ventana (Alt+N).
- Ventana Anterior (Alt+P).
- Lista de ventanas (Alt+W).
- Cerrar ventana (Alt+C).

- Salir (Alt+Q).



Figura 2-7.: Logo de Finch

2.7. B.A.T.M.A.N

Uno de los protocolos dinámicos para redes Ad Hoc, es el protocolo B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking), el cual es un protocolo proactivo que utiliza las tablas de enrutamiento para la toma de decisiones. Este protocolo no calcula rutas completas entre un nodo origen y destino, sino que selecciona un nodo de salto para utilizarlo como Gateway hacia el destino. La función de B.A.T.M.A.N. es encontrar otros nodos B.A.T.M.A.N. y definir el mejor vecino para llegar a ellos. Además, realiza un seguimiento de los nuevos nodos y notifica a sus vecinos de su existencia [69], en otras palabras, cuando un nodo se incorpora a la red, envía un paquete broadcast para avisar de su existencia, este mensaje se va distribuyendo por toda la red. Cuando un nodo recibe este mensaje guarda la dirección por donde le ha llegado la información en menos tiempo, y así si es necesario el intercambio de información utilizará ese nodo (Gateway) para transmitir. El protocolo mantiene la información sobre la existencia de los nodos mientras sean accesibles, por lo tanto, el protocolo B.A.T.M.A.N. busca un nodo vecino hacia el destino para ser utilizado como Gateway para conseguir la comunicación, esto quiere decir que se encarga de elegir el mejor salto para cada destino. Esto provoca que la implementación sea rápida y eficiente, ya que no es necesario calcular la ruta completa [41, 69].

Cuando es necesario decidir la ruta de transmisión se realizan análisis estadísticos de la velocidad de propagación y de la pérdida de paquetes de esa ruta. No se basan en el estado de los nodos ni en la información topológica de otros nodos [41].

Los paquetes de datos del protocolo B.A.T.M.A.N. contienen muy poca información, lo que provoca que sean muy pequeños y que su transmisión sea rápida. Cada nodo percibe y

mantiene la información sobre el mejor salto hacia el resto de nodos, por este motivo, es innecesario el conocimiento global de la red. Estos mensajes son pequeños, el tamaño típico es de cincuenta y dos (52) bytes. Los campos que contienen estos mensajes son la dirección de origen, la dirección del nodo que retransmite, el TTL y el número de secuencia [69].

Es posible que se reciban varias veces el mismo mensaje, ya que hay rutas donde se propagan con retardo o con pérdidas debido a la baja calidad del enlace y habrá otras donde se propagan de manera rápida y fiable. Para solucionar este problema, se etiqueta cada mensaje con un número de secuencia, de este modo se hace un estudio de qué nodo vecino es el que le ha transmitido la información primero y lo selecciona como el vecino de salto, para enviar datos al origen del mensaje y lo configura en su tabla de enrutamiento [69, 77].

Las principales características de este protocolo son:

- **Soporte de interfaces alias:** lo que permite ejecutar otros protocolos en paralelo.
- **Soporte en múltiples interfaces:** se puede utilizar en más de una tarjeta inalámbrica o Ethernet.
- **Interfaz IPC (Inter-Process Communication):** para conectarse al demonio batman. Algunas de las acciones que permite son:
 - Información debugging.
 - Consulta de nodos vecinos.
 - Consulta de nodos Gateway
 - Configuración del Gateway en tiempo de ejecución
 - Mensajes HNA
 - Modificación de interfaces
- **Policy Routing:** A partir de la versión B.A.T.M.A.N. 0.3, se soporta el policy routing y se puede utilizar las funciones especiales de enrutamiento proporcionadas por el kernel de Linux. Con el comando “route” se puede consultar las tablas de enrutamiento. B.A.T.M.A.N. [65, 69]:
 - Tabla “redes”: contiene las entradas de las redes anunciadas (HNA).
 - Tabla “hosts”: contiene todos los nodos alcanzables.
 - Tabla “unreachable”: contiene los nodos que no son alcanzables en la red.
 - Tabla “tunnel”: contiene la ruta por defecto que se utiliza si existe un Gateway disponible.

- **Servidor de visualización:** Ya que el protocolo B.A.T.M.A.N. no contiene la topología de la red, cada nodo envía su vista local de sus vecinos al servidor de visualización y este, con todos los envíos de cada nodo hace una recopilación y los incluye en el formato que utilizan las herramientas de visualización de red [41, 58, 69].
- **Selección de Gateway:** El protocolo B.A.T.M.A.N. permite anunciar una conexión a Internet, e informa del ancho de banda disponible, la velocidad de bajada y de subida. Es posible buscar anuncios de Gateway y decidir alguna conexión teniendo en cuenta los siguientes modos de comportamiento [66, 78]:
 - *Fast Internet Connection:* Considera la calidad del enlace y la clase de Gateway. Mantiene la conexión hasta que ya no se puede.
 - *Stable Internet Connection:* Elige la conexión más estable hacia el Gateway y la mantiene hasta que ya no se puede.
 - *Fast-switching:* Elige la conexión más estable hacia el Gateway pero la cambia cuando encuentra otra mejor.
 - *Late-switching:* Elige la conexión más estable hacia el Gateway pero la cambia cuando encuentra otra mejor.

El cliente de Internet B.A.T.M.A.N. puede detectar y evitar Gateways que posean enlaces con fallas. Además, durante el tiempo de ejecución en Internet se puede desactivar el Gateway, cambiar a otro Gateway o la clase de enrutamiento, sin necesidad de reiniciar el demonio [6].

2.8. A.L.F.R.E.D

Almighty Lightweight Fact Remote Exchange Daemon o A.L.F.R.E.D, es un demonio de espacio de usuario utilizado para distribuir información local arbitraria a través de la red de manera descentralizada. Estos datos pueden distribuir información diversa como nombres de host, guías telefónicas, información de administración, información de DNS, entre otros [25].

Por lo general, A.L.F.R.E.D se ejecuta como demonio unix en el fondo del sistema. Un usuario puede agregar información usando el binario alfred en la línea de comando o con programas escritos personalizados. Una vez los datos locales son recibidos, Alfred se encarga de distribuir esta información a otros servidores Alfred en otros nodos en alguna ubicación de la red, ver Figura 2-8. Normalmente, direcciones de multidifusión local de enlace IPv6 son usadas como esquema de direccionamiento debido a que no requieren ninguna configuración manual. El usuario puede solicitar datos de alfred y recibirá información disponible de todos los servidores alfred de la red [25].

2.8.1. Estructura

Una red Alfred es utilizada para intercambiar datos y se construye a través de diferentes nodos que ejecutan un servidor Alfred. Los servidores se detectan entre sí y se conectan automáticamente.

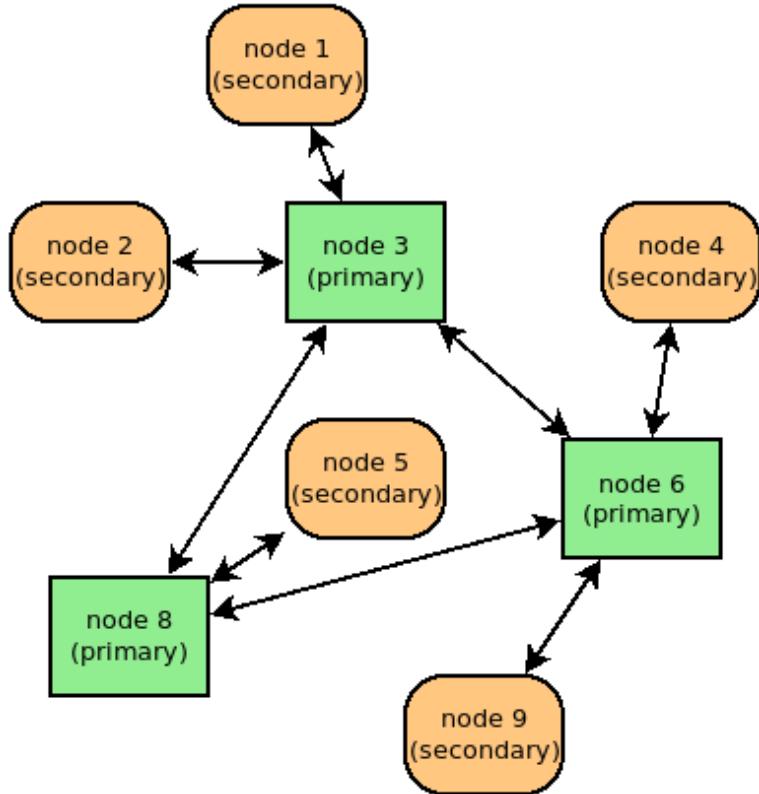


Figura 2-8.: Estructura de una red Alfred [25].

Esta red posee un servidor primario, que son utilizados como almacenamiento de datos globales y ayudan en la sincronización con otros nodos. Este papel es tan importante que anuncian activamente su presencia en la totalidad de nodos. Otros servidores primarios sincronizan automáticamente sus datos almacenados para distribuirlos a todos los nodos primarios accesibles, ver Figura 2-8.

La red posee otros servidores que son conocidos como servidores secundarios, estos son aquellos que no intentan recopilar todos los datos almacenados en la red Alfred. Estos al contrario, se conectan a su mejor servidor primario para solicitar datos cuando es necesario y para enviar datos de los clientes locales a la red Alfred a través de los servidores primarios, ver Figura 2-8.

Toda comunicación entre servidores en una red Alfred se realiza mediante mensajes UDP de

enlace local IPv6 (Puerto 0x4242). Esto limita la comunicación entre la subred actual, pero los mensajes no se enrutarán entre diferentes subredes.

La capa subyacente puede ser Batman-adv, lo que genera que todos los nodos en una malla parezcan estar conectados a un gran interruptor. Esto permite llegar a todos los nodos, por lo que los mensajes de enlace local son suficientes para este escenario.

Detección de Vecinos

El servidor primario se anunciará usando mensajes de multidifusión local de enlace IPv6. Esto permite llegar a todos los nodos con un mensaje, mientras la conexión no se caiga, ver Figura 2-9.

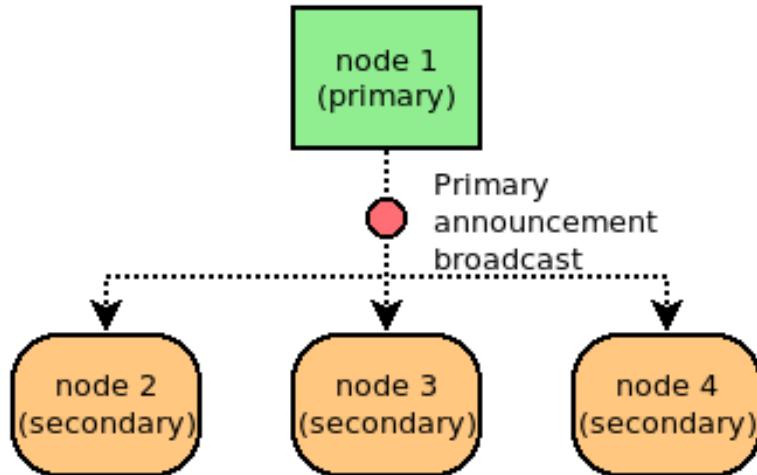


Figura 2-9.: Envío de mensaje del servidor primario a los servidores secundarios para la detección de vecinos [25].

Los mensajes de multidifusión son transmitidos de forma periódica cada 10 segundos y son recibidos por los otros servidores Alfred y se almacenarán para más adelante. Los servidores primarios usan esta información para la sincronización de datos y los servidores secundarios la usan para tener acceso a los datos globales.

Por lo tanto, los servidores secundarios tienen que elegir su mejor servidor primario. Las redes batman-adv proporcionan una métrica TQ para cuantificar la calidad de un servidor primario. El mejor servidor primario es el servidor con la mejor TQ para un nodo secundario.

Los servidores vecinos detectados se eliminan automáticamente después de un tiempo de espera de 60 segundos desde que se recibió el último mensaje.

2.8.2. Intercambio de datos

La comunicación entre el cliente y el servidor Alfred se realiza a través de sockets unix en el mismo nodo. El formato paquete utilizado se comparte entre el servidor *Server to Server* y la comunicación *Client to Server*. La comunicación más fácil es el envío de datos a un servidor. El cliente crea un paquete *alfred push data* que el servidor recibe y almacena, cierra la conexión del cliente y controla el reenvío de datos a los otros servidores Alfred.

Por otro lado, la solicitud de datos de un servidor primario se da de la siguiente manera: El cliente tiene que enviar un paquete *alfred request* al servidor primario. La respuesta será un número arbitrario de paquetes *alfred push data* del tipo solicitado con los datos establecidos por los clientes de este nodo y de los otros nodos.

La solicitud de Alfred a un servidor secundario es similar a la de un servidor primario. Pero el servidor secundario no almacena todos los datos actuales de la red Alfred. El servidor secundario tiene que pedirle dicha información a su servidor primario. Esto se hace de la misma manera que un cliente pero a través de un mensaje UDP/IPv6 de enlace local. El servidor responde a la solicitud con los paquetes *alfred push data* y un único paquete final *alfred status* que contiene el número de paquetes *alfred push data* enviados, ver Figura 2-10.

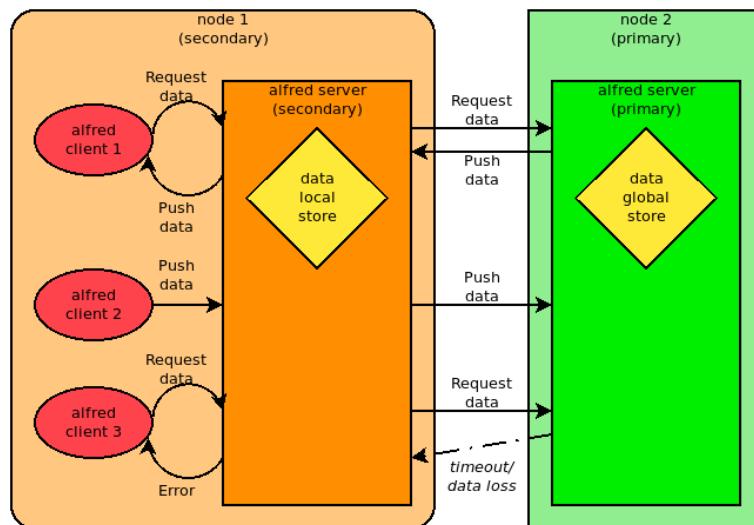


Figura 2-10.: Intercambio de datos en una red Alfred [25].

3. Metodología y Resultados

3.1. Tlön

Este proyecto, hizo parte de las ramas de estudio del grupo de investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos: **Tlön**. El objetivo de este grupo de investigación es buscar soluciones de integración e intercambio de información en ambientes heterogéneos y distribuidos con el fin de resolver las problemáticas planteadas por las nuevas tecnologías, generar conocimiento y aprovechar oportunidades brindadas por tecnología informática y así aumentar la competitividad de la academia y las empresas colombianas. Esto se logra por medio de la adquisición, generación, y transferencia de conocimiento en las líneas de investigación [73].

3.1.1. Proyecto Tlön

El proyecto del Grupo de investigación en redes de Telecomunicaciones Dinámicas Lenguajes de Programación Distribuidos-Tlön, propone un esquema de computación inspirado en modelos Sociales, inviables en la práctica pero muy posibles en entornos artificiales controlados, este sistema basado en los conceptos de Justicia de Jhon Rawls, Inmanencia de Baruch de Spinoza, Paradigma Thomas Kuhn, Estado Thomas Hobbes y las concepciones de existencia y esencia de Jean Paul Sartre, generan una analogía completa de un esquema de virtualización inalámbrica, necesaria para implementar estos modelos sociales en sistemas computacionales, este modelo social inspirado, es una abstracción superior a los modelo bio-inspirados, en la Figura 3-1 se pueden observar los componentes del modelo propuesto el cual además funcionará sobre una red Ad Hoc con todas sus condiciones dinámicas, estocásticas e inalámbricas [73].

La mejor forma de mostrar las dimensiones del proyecto Tlön es el modelo de capas, en la primera está la infraestructura, donde se encuentra la Red Ad Hoc, la segunda capa contiene la virtualización inalámbrica y el Sistema Operativo, en tercer lugar se cuenta con el sistema Multiagente, donde operaran comunidades de agentes que proveerán servicios a lo largo de la red Ad Hoc en el esquema de virtualización y finalmente una capa de aplicaciones específicas del sistema de cómputo. De forma transversal se tiene el lenguaje del sistema y la ontología propia del mismo, generando las interacciones en todas las capas.

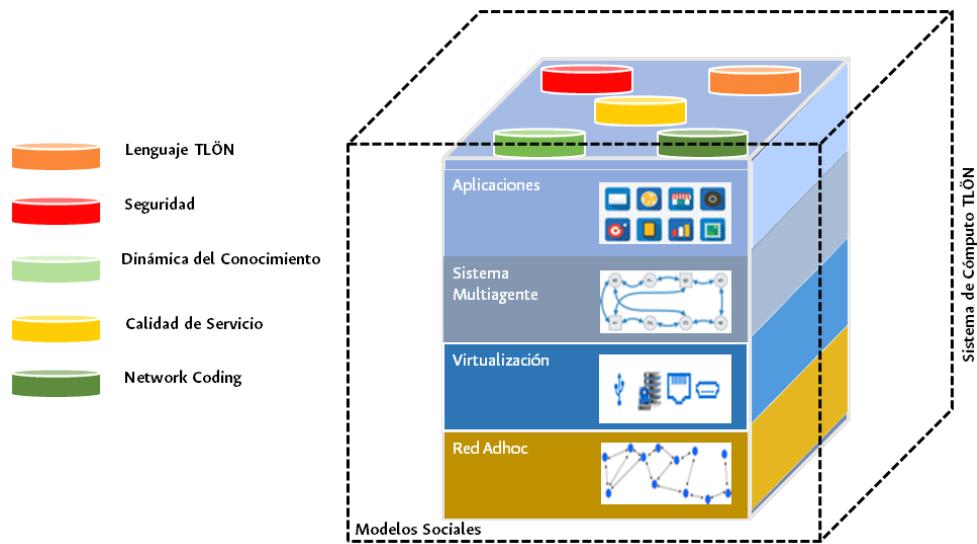


Figura 3-1.: Proyecto Tlön [73].

3.2. Rol del prototipo en el modelo Tlön

En la búsqueda de Tlön por incorporar de forma artificial los modelos sociales en un sistema de cómputo distribuido, en el desarrollo de este proyecto se diseñó un prototipo que se define como un “pseudoterritorio” dentro del pseudo-estado que propone el grupo Tlön. Este pseudoterritorio consistió en un conjunto de elementos hardware y software que forman el prototipo de una Red Ad Hoc, cuyo funcionamiento fue basado en clústers y mensajería instantánea, ver Figura 3-2.

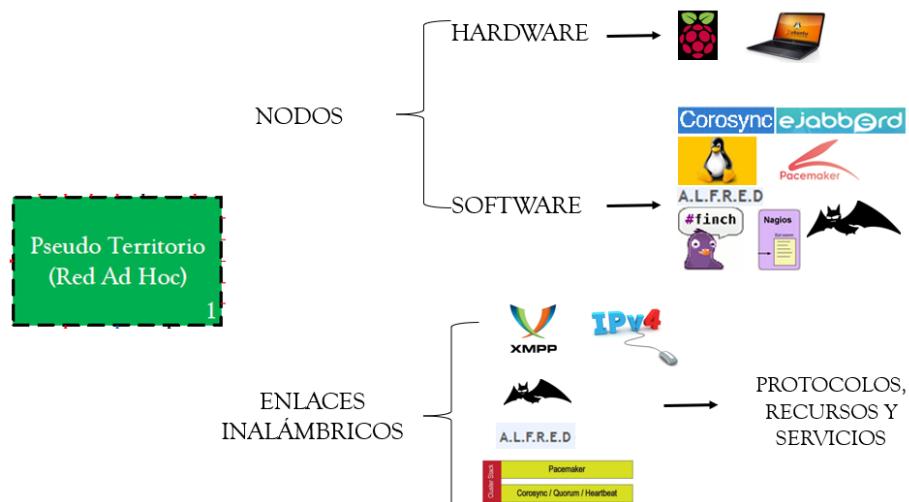


Figura 3-2.: Elementos del prototipo asociados al pseudo territorio del modelo Tlön.

3.3. Diseño del Prototipo

Para los elementos hardware, se usaron cinco (5) Raspberry Pi versiones 1,2 y 3 que corresponden a los nodos físicos junto con un computador portátil para monitorizar el funcionamiento de la Red. También se usaron 5 bancos de energía para proveer alimentación a los Raspberry Pi.

Por otro lado para el software, se usó el sistema operativo Linux para el computador y Raspbian para los Raspberry Pi. Los protocolos de enrutamiento se basaron en el protocolo Batman-Adv e IPv4, además del demonio Alfred para la aplicación de distribución de información aleatoria basada en clúster lógicos.

A través de Pacemaker se realizó la configuración de la agrupación de los nodos físicos y con Corosync se realizó la sincronización de los nodos que integran un clúster y la comunicación entre ellos.

Finalmente para la incorporación del sistema de mensajería instantánea, se usó Jabberd como servidor y la consola Finch como cliente, permitiendo que el funcionamiento de la Red sea a través de la estructura Cliente-Servidor mediante el protocolo de mensajería XMPP.

Alfred fue implementado como una aplicación en segundo plano, la cual corre sobre todos los nodos y en la Red propuesta fue utilizado para la distribución de información de forma descentralizada. Como resultado de esto, cualquier nodo puede insertar información mediante el uso de binarios a través de la línea de comandos. Para este caso la red de direccionamiento consistió en IPv6 (envío simultáneo de información de uno a varios). Dentro del prototipo cumple la función de contingencia en caso de falla de la mensajería instantánea y permite que un nodo pertenezca a más de un clúster simultáneamente.

Finalmente, la configuración final del prototipo se puede observar en la Figura 3-3.

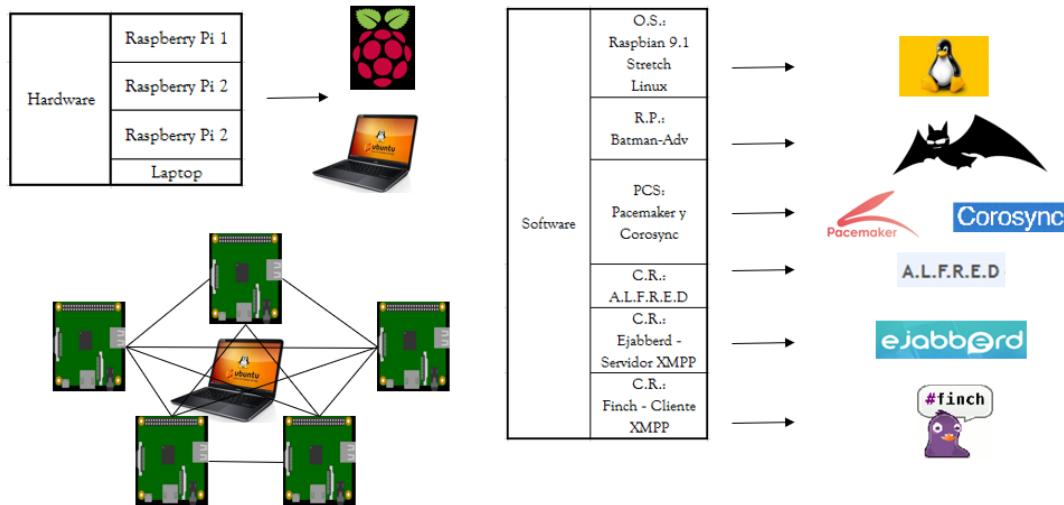


Figura 3-3.: Configuración final del prototipo

3.4. Implementación del servicio y definición del Clúster-Head

3.4.1. Funcionamiento de los Head por clúster

- Enrutamiento XMPP, donde se usan dos nodos como head: Tlon1.net y Tlon2.net. En otras palabras estos dos son los servidores XMPP.
- Distribución A.L.F.R.E.D, donde dos nodos son maestros Alfred y los otros son esclavos Alfred.

3.4.2. Funcionamiento de cada clúster y sus recursos compartidos

- El nodo Head activo se le asigna la dirección IP flotante del clúster al cual pertenece.
- El nodo Head activo conmuta a través del comando *Mode Switch* de esclavo a maestro sobre A.L.F.R.E.D.
- El nodo Head activo asume el Rol de servidor XMPP de su respectivo clúster.

Finalmente, la operación del clúster head se puede observar en la Figura 3-4.

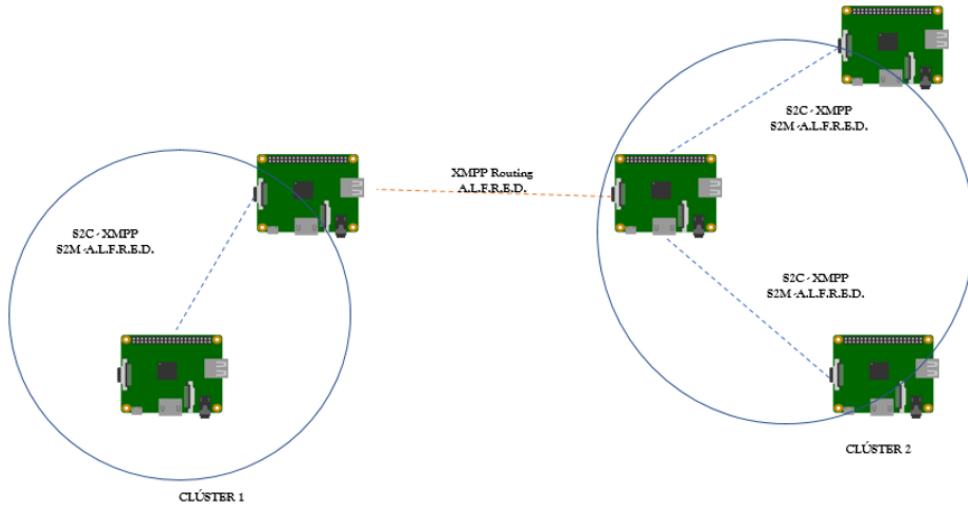


Figura 3-4.: Operación del clúster head

3.4.3. Condiciones de operación de mensajería

- Creación de usuarios por clúster de acuerdo a su dominio XMPP. Por ejemplo Laura@Tlon1.net y Juan@Tlon2.net.
- Configuración de cada nodo a un clúster específico.
- Cada dominio debe tener una dirección IP de servicio que asume el Head o nodo activo del clúster, que debido a pacemaker y corosync, comparte recursos.
- El servidor XMPP (ejabberd) funciona como una base de datos replicable entre nodos, la cual contiene información asociada a contraseñas de usuario, usuarios y datos de mensajería (chat).
- Se realiza un enrutamiento entre dominios XMPP, por ejemplo: entre Laura@Tlon1net y Juan@Tlon2.net.

3.5. Operación del prototipo

Inicialmente se realizan pruebas y configuraciones previas a la prueba final de campo y se obtienen los parámetros técnicos expuestos en la Figura 3-5.



Hostname	Hardware	Software	IP Wlan0 (Enrutamiento XMPP)	IP bat0 (Basada MAC) SSID = TLON	Ip Gestión (Eth0) [Chequeo Fallas y Configuración de cada Nodo]	IP Servicio XMPP	Dominio XMPP
RP1-1	Raspberry Pi 1 Modelo B+	B.A.T.M.A.N	192.168.20.1	fe80::7089:4aff:fea7:3b6d /64	192.168.13.31	192.168.20.101 192.168.20.102	Tlon1.net Tlon2.net
RP3-1	Raspberry Pi 3 Modelo B+	A.L.F.R.E.D RASPBIAN	192.168.20.4	fe80::64f1:baff:fe54:216a /64	192.168.13.32		
RP2	Raspberry Pi 2 Modelo B+	EJABBERD FINCH	192.168.20.3	fe80::68e2:c9ff:fe7b:10da /64	192.168.13.33		
RP3-2	Raspberry Pi 3 Modelo B+	PACEMAKER COROSYNC	192.168.20.5	fe80::e412:64ff:fe59:d83e /64	192.168.13.34		
RP3-3	Raspberry Pi 3 Modelo B+	PCS	192.168.20.6	fe80::c80d:ffff:fe6e:5b8e/64	192.168.13.35		

Figura 3-5.: Configuraciones previas y parámetros obtenidos.

3.6. Prueba de Campo

3.6.1. Ubicación

La evaluación del prototipo en campo abierto, se realizó en un lugar específico del Campus de la Universidad Nacional, señalado en la Figura 3-6.



Figura 3-6.: Campo seleccionado para las pruebas de campo, en el Campus de la Universidad Nacional visto desde Google Earth (Círculo Amarillo).

3.6.2. Parámetros Evaluados

- Cambio del Rol en cada Clúster.
- Intercambio de Mensajería XMPP Intra Clúster e Inter clúster.
- Disponibilidad de cada nodo del prototipo simulando movilidad
- Alcance de cobertura de la Red en el Campus.
- Tiempos de respuesta de la conectividad entre nodos.
- Comportamiento de los enlaces inalámbricos del prototipo al utilizar un Canal Saturado de señales.

3.6.3. Pre-requisitos

- Conectividad de Enlace y de Red, entre los nodos.
- Servicio XMPP y software operativo en cada uno de los nodos.
- Clústers operativos y base de datos distribuida en cada clúster.
- Bancos de energía totalmente cargados.
- Equipo de monitoreo totalmente cargado

Escenarios

Los nodos se ubicaron en dos escenarios diferentes.

- Nodos de la Red ubicados en línea recta sobre el campus, ver Figura 3-7.

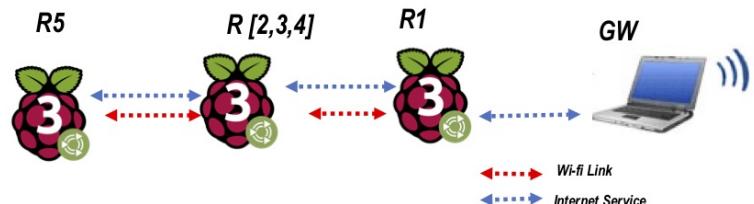


Figura 3-7.: Esquema de la Red con los nodos ubicados en línea recta.

- Nodos de la Red ubicados de forma circular sobre el campus, ver Figura 3-8.

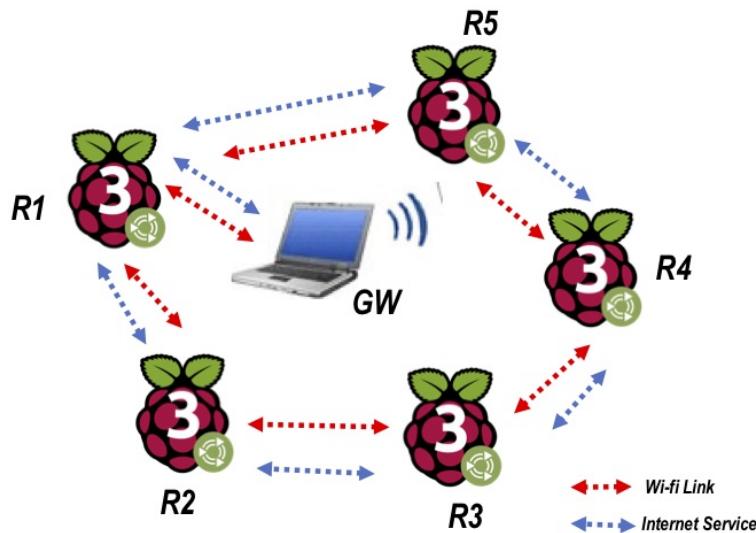


Figura 3-8.: Esquema de la Red con los nodos ubicados de forma circular

Para el equipo de monitoreo adicionalmente, fue necesario instalar en el computador portátil la aplicación **Nagios** que permite la monitorización del estado de salud de los nodos, del servicio XMPP y de los clústeres en tiempo real, ver Figura 3-9.

Las capturas de pantalla muestran la interfaz web de Nagios, que incluye:

- Host Status Details For All Host Groups:** Muestra la lista de hosts (RPI1-RPI5) con su último check, duración y status.
- Service Status Details For All Hosts:** Muestra la lista de servicios para cada host, incluyendo XMPP Service y MAC Address.
- Current Network Status:** Muestra el estado actual de los clústeres (Cluster RPI1.net, Cluster RPI2.net, Cluster RPI3.net, Cluster RPI4.net, Cluster RPI5.net) y sus servicios.
- Service Overview For All Host Groups:** Muestra un resumen detallado de los servicios y su estado.

Figura 3-9.: Monitoreo del estado de los nodos, servicio XMPP y clústeres a través de Nagios.

3.7. Métricas de Evaluación

Por último se evaluó el comportamiento de la potencia y de los tiempos de respuesta de la Red, respecto a la variación de la distancia de ubicación de los nodos.

Para los nodos ubicados en línea recta el comportamiento se puede observar en la Figura 3-10.

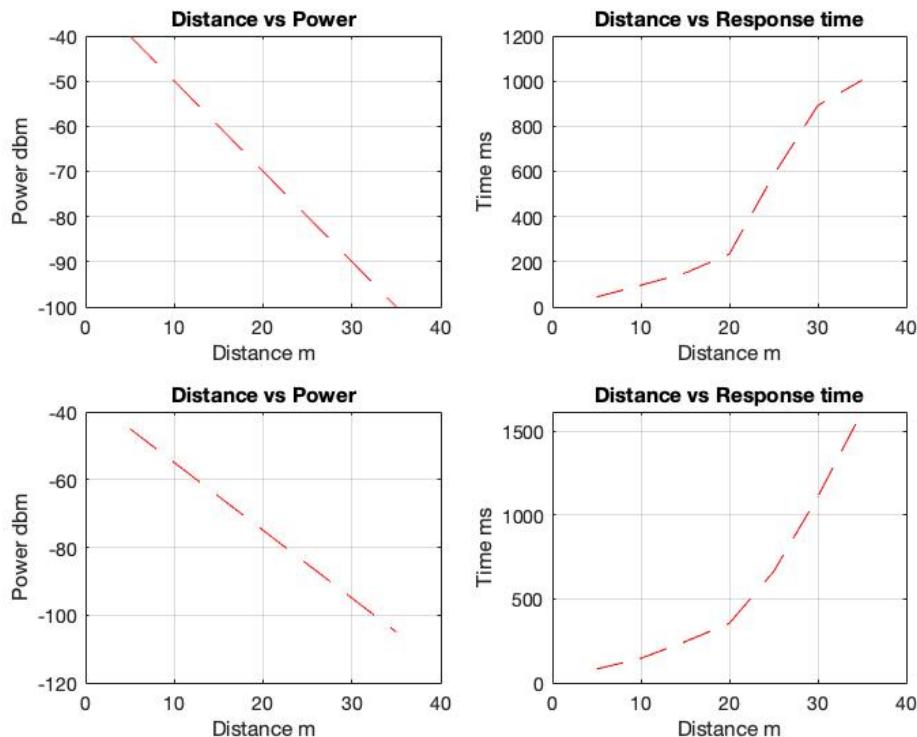


Figura 3-10.: Comportamiento de la Red con nodos ubicados en línea Recta en cuanto a Tiempos de Respuesta y Potencia vs Distancia.

Para los nodos ubicados en forma circular el comportamiento se puede observar en la Figura 3-11.

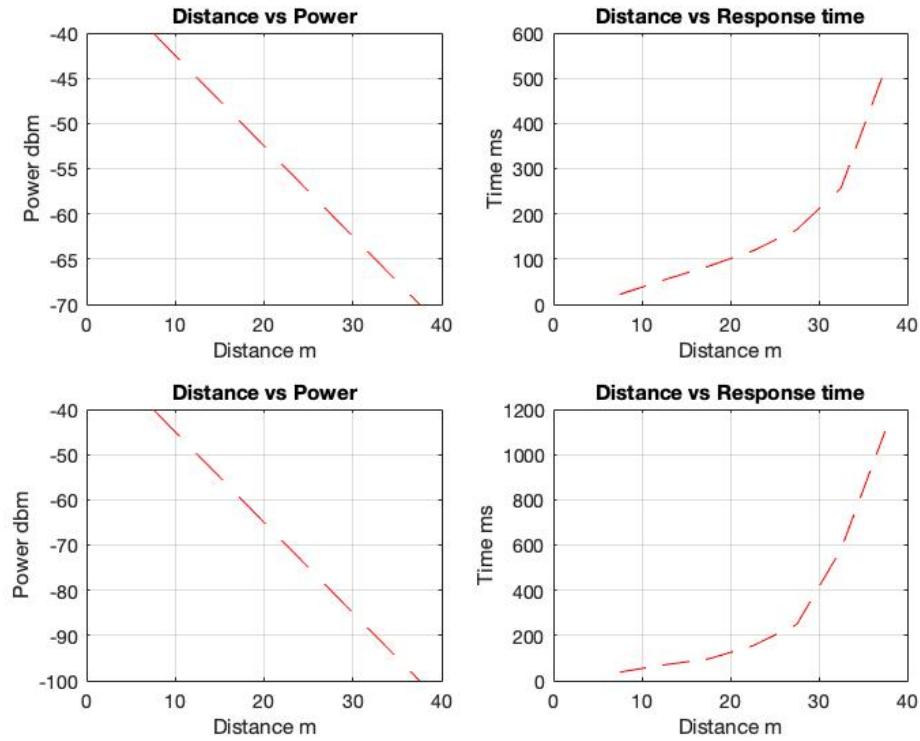


Figura 3-11.: Comportamiento de la Red con nodos ubicados de forma circular en cuanto a Tiempos de Respuesta y Potencia vs Distancia.

Estas métricas de evaluación se utilizaron para comparar el comportamiento del prototipo de la Red propuesta en los dos esquemas evaluados: Línea recta y forma circular. La comparación de la respuesta en potencia versus la distancia para los dos esquemas se puede observar en la Figura 3-12, mientras que el tiempo de respuesta versus la distancia para dichos esquemas se puede observar en la Figura 3-13.

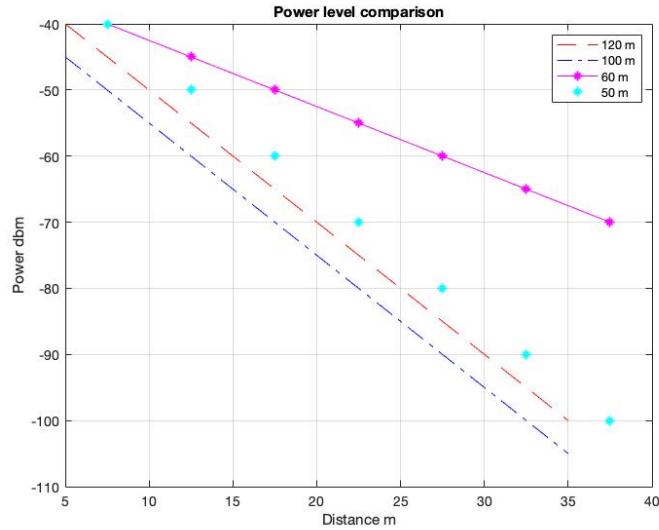


Figura 3-12.: Comparativos del comportamiento de la Red en cuanto a Potencia versus distancia.

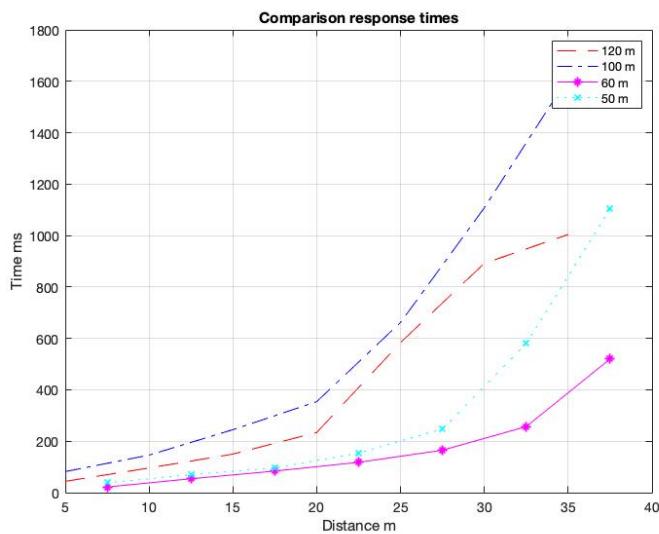


Figura 3-13.: Comparativos del comportamiento de la Red en cuanto a Tiempo de respuesta versus distancia.

Al analizar estas métricas se pudo determinar que cuando los nodos son ubicados de forma lineal la Red tiene mayor cobertura a comparación de cuando los nodos de forma circular fueron ubicados. En la Red con nodos ubicados linealmente permitió un alcance de hasta 100 metros mientras que los que fueron ubicados de forma circular, el alcance se redujo en un factor de 2 (50 metros). Respecto al los tiempos de respuesta, la conectividad lineal

obtuvo tiempo de respuesta de conectividad más altos a comparación de la ubicación circular. Con base en las Figuras 3-12 y 3-13 es posible establecer la relación matemática entre las variables distancia, potencia y latencia para los nodos ubicados de forma circular y lineal:

- **Distancia vs Potencia:** Ecuación Lineal, a mayor distancia menor potencia de radiación de señal.
- **Distancia vs Latencia:** Ecuación cuadrática (segundo grado), a mayor distancia se da un incremento parabólico (vertical hacia arriba), de latencia.

4. Conclusiones

4.1. Conclusiones y Recomendaciones

Es necesario en el momento de ejecutar la implementación de un tangible, contar con un ambiente de pruebas virtual con el objetivo de descartar puntos de falla generados por el desempeño del Hardware. Lo anterior teniendo en cuenta que en la práctica se hacen evidentes fallas ocasionadas por el rendimiento de este.

Adicionalmente y apuntando hacia el objetivo antes descrito, es útil el uso de software libre, bajo en almacenamiento y con un rendimiento adecuado a la capacidad de Hardware a utilizar.

El enfoque desarrollado en el presente trabajo de investigación definió dentro de su alcance, que la operación, la arquitectura y el desempeño de las redes inalámbricas, se enfocara en cubrir necesidades especiales de cobertura geográfica.

Las redes tradicionales WiFi normalizadas por el estándar 802.11 entre todos sus tipos (802.11a, 802.11b, 802.11g, 802.11n y 802.11ac), generan una cobertura geográfica de hasta cien (100) metros. Sin embargo, el prototipo de red implementado, que entre otras cosas se concibe como una red totalmente descentralizada, heterogénea y la cual opera en modo Ad-Hoc con cinco (5) nodos, cubrió la misma distancia geográfica que todos los tipos del estándar que cubren las redes WiFi. Podemos concluir que al ampliar la cantidad de nodos del prototipo, la cobertura geográfica será superior y en consecuencia también es una red escalable.

Adicionalmente y como pilar del trabajo de investigación, el prototipo de red implementado debido a su naturaleza Ad-Hoc, opera sin Infraestructura preexistente, es descentralizado, contiene información distribuida para el servicio de mensajería que presta y como ya se mencionó, puede cubrir un área geográfica de hasta mil (1000) metros con cincuenta (50) nodos. Con lo anterior, se infiere que el prototipo tiene un rendimiento cuantitativo equivalente a una relación de veinte (20) metros por nodo (20 m/nodo).

4.2. Trabajos Futuros

A pesar de que el O.S. Android es compatible, la cantidad de versiones que posee, los limitantes de configuración sobre este y la imposibilidad de manipular el Hardware debido a la reserva y patentes de cada uno de los Fabricantes de estos dispositivos (Samsung, Motorola, entre otros); imposibilita incluirlos en el prototipo debido a su nivel de complejidad.

Por lo anterior, como trabajo futuro es necesario implementar una versión de prueba de O.S. Android sobre Raspberry Pi con el objetivo de garantizar mayor heterogeneidad de dispositivos.

Finalmente y para observar el comportamiento del prototipo en un ambiente saturado de señales, vale la pena operar la red en un escenario con saturación y otro libre de ella; apoyándonos con un analizador de espectros.

4.3. Producción Académica

Con base en los resultados obtenidos de esta Tesis, la producción académica titulada “**Proposed ad hoc network to expand coverage of IEEE 802.11n network**” fue presentando en “**Colombian Conference on Communications and Computing (COLCOM)-2017**” disponible en la base de datos del IEEE: <http://ieeexplore.ieee.org/document/8088196/>.



Figura 4-1.: Logo Colombian Conference on Communications and Computing (COLCOM-2017).

Bibliografía

- [1] ASTUTO, B. N. ; MENDONZA, M. ; NGUYEN, X. N. ; OBRACZKA, K. ; TURLETTI, T.: A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. En: *IEEE* 16 (2014), p. 1617–1634
- [2] BASAGNI, S. ; CONTI, M. ; GIORDANO, S. ; STOJMENOVIC, I.: *Mobile Ad Hoc Networking*. 1. John Wiley, 2004. – ISBN 0471373133,9780471373131
- [3] BASYOUNI, A. ; HAMOUDA, Walaa: Improved channel access protocol for cooperative ad hoc networks. En: *Communications, IET* 4 (2010), 06, p. 915 – 923
- [4] DEN BERG, J. L. ; LITJENS, R. ; EISENBLATTER, A. ; AMIRIJOO, M. ; LINNELL, O. ; BLONDIA, C. ; SCHMELZ, L. C.: Self-organisation in future mobile communication networks. En: *Proceedings of ICT Mobile Summit*, 2008
- [5] BERRAYANA, W. ; YOUSSEF, H. ; PUJOLLE, G.: A generic cross-layer architecture for autonomic network management with network wide knowledge. En: *8th International Wireless Communications and Mobile Computing Conference IWCMC*, 2012, p. 82–87
- [6] BONONI, Luciano ; FELICE, Marco ; DONATIELLO, Lorenzo ; BLASI, Danilo ; CACACE, Vincenzo ; CASONE, Luca ; ROTOLI, Salvatore: Design and performance evaluation of cross layered MAC and clustering solutions for wireless ad hoc networks. En: *Performance Evaluation* 63 (2006), 11, p. 1051–1073
- [7] BRAKMO, Lawrence S. ; O'MALLEY, Sean W. ; PETERSON, Larry Lee: TCP vegas: New techniques for congestion detection and avoidance. En: *Proceedings of the Conference on Communications Architectures, Protocols and Applications, SIGCOMM 1994*, Association for Computing Machinery, Inc, Oktober 1994. – 1994 Conference on Communications Architectures, Protocols and Applications, SIGCOMM 1994 ; Conference date: 31-08-1994 Through 02-09-1994, p. 24–35
- [8] BRAMS, G. W.: *REDES DE PETRI MODELIZACION Y APLICACIONES*. MASSON SA, 1996
- [9] BUGENHAGEN, M. K. ; MORRILL, R. J. ; EDWARDS, S. K.: *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*. 2008

- [10] CAICEDO RENDON, Oscar ; CAICEDO, Dario ; FIGUEROA, Edwin ; MARTINEZ, Orlando ; HURTADO, Javier: Universal Access Platform to Mobile Instant Messaging. En: *Latin America Transactions, IEEE (Revista IEEE America Latina)* 5 (2007), 07, p. 192 – 201
- [11] CHAMPRASERT, P. ; SUZUKI, J.: Building Self-Configuring Data Centers with Cross Layer Coevolution. En: *Journal of Software* 2 (2007), Nr. 5, p. 29–43
- [12] CHANG, Ray-I ; CHUANG, C.-C: A new spatial IP assignment method for IP-based wireless sensor networks. En: *Personal and Ubiquitous Computing* 16 (2012), 10
- [13] CHENG, B. H. C. ; DE LEMOS, R. ; GARLAN, D. ; GIESE, H. ; LITOIU, M. ; MAGEE, J. ; TAYLOR, R.: Fifth workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2010). En: *International Conference on Software Engineering* Vol. 2, 2010, p. 447–448
- [14] CHENG JIN ; WEI, D. ; LOW, S. H. ; BUNN, J. ; CHOE, H. D. ; DOYLL, J. C. ; NEWMAN, H. ; RAVOT, S. ; SINGH, S. ; PAGANINI, F. ; BUHRMASTER, G. ; COTTRELL, L. ; MARTIN, O. ; WU-CHUN FENG: FAST TCP: from theory to experiments. En: *IEEE Network* 19 (2005), Nr. 1, p. 4–11
- [15] CHIANG, M. ; LOW, S. H. ; CALDERBANK, A. R. ; DOYLE, J. C.: Layering As Optimization Decomposition: Current Status and Open Issues. En: *2006 40th Annual Conference on Information Sciences and Systems*, 2006, p. 355–362
- [16] CHIANG, M. ; LOW, S. H. ; CALDERBANK, A. R. ; DOYLE, J. C.: Layering As Optimization Decomposition: Framework and Examples. En: *2006 IEEE Information Theory Workshop - ITW '06 Punta del Este*, 2006, p. 52–56
- [17] CHIANG, M. ; LOW, S. H. ; CALDERBANK, A. R. ; DOYLE, J. C.: Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. En: *Proceedings of the IEEE* 95 (2007), Nr. 1, p. 255–312
- [18] CHIANG, MUNG: Balancing transport and physical Layers in wireless multihop networks: jointly optimal congestion control and power control. En: *IEEE Journal on Selected Areas in Communications* 23 (2005), Nr. 1, p. 104–116
- [19] CHOWDHURY, N. ; BOUTABA, R.: Network Virtualization: State of the Art and Research Challenges. En: *Communications Magazine, IEEE* 47 (2009), 08, p. 20 – 26
- [20] CLOWDHURY, Sadrul. *finch*. url <https://linux.die.net/man/1/finch>
- [21] CLUSTERLABS. *PaceMaker*. url <https://wiki.clusterlabs.org/wiki/Pacemaker>. 2018
- [22] CORONADO, M. J. *Asignacion de canales en redes ad-hoc 802.11 multi-radio*. 2006

- [23] CORONADO, M. J. *Redes Ad Hoc entre vehiculos.* 2008
- [24] DERBEL, H. ; AGOULMINE, N. ; SALAÁ¹N, M.: ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks. En: *Comput. Netw* 53 (2009), Nr. 3, p. 418–430
- [25] DEVELOPERS, B.A.T.M.A.N. *A.L.F.R.E.D - Almighty Lightweight Fact Remote Exchange Daemon.* url <https://www.open-mesh.org/doc/alfred/index.html>. 2018
- [26] DRESSLER, F.: *Self-Organization in Sensor and Actor Networks.* Wiley Series in Communications Networking Distributed Systems, 2008
- [27] DUSIA, A. ; SETHI, A. S.: Recent Advances in Fault Localization in Computer Networks. En: *Commun. Survey Tuts* 18 (2016), Nr. 4, p. 3030–3051
- [28] FELDMANN, Anja ; GILBERT, Anna ; HUANG, Polly ; WILLINGER, Walter: Dynamics of IP traffic: A study of the role of variability and the impact of control. En: *Computer Communication Review* 29 (1999), 06
- [29] FORDE, T. K. ; DOYLE, L. E. ; MAHONY, D. O.: Self-stabilizing network-layer auto-configuration for mobile ad hoc network nodes. En: *IEEE International Conference on Wireless and Mobile Computing, Networking And Communications.,* 2005
- [30] FUJISHIRO, T. ; NAKANO, H. ; MIYAUCHI, A.: A routing algorithm for mobile ad Hoc networks based on multi-agent cost learning. En: *ICIC Express Letters* 7 (2013), 01, p. 67–72
- [31] GERSHENSON, C.: *Design and Control of Self-organizing Systems: Facing Complexity with Adaptation and Self-Organization.* LAP Lambert Academic Publishing, 2010
- [32] GHASSEMI, F. ; FOKKINK, W. *Sinab*
- [33] GLITHO, R. ; SVENSSON, B.: *Operation and maintenance control point and method of managing a self-engineering telecommunications network.* Google Patents, 2001
- [34] HE, J. ; ZHANG-SHEN, R. ; LI, Y. ; LEE, C. ; REXFORD, J. ; CHIANG, M.: DaVinci:dynamically adaptive virtual networks for a customized internet. En: *Conference on Emerging Network Experiment and Technology,* 2008
- [35] HEGDE, Sanjay ; LAPSLEY, David ; WYDROWSKI, Bartek ; LINDHEIM, Jan ; WEI, David ; JIN, Cheng ; LOW, Steven ; NEWMAN, Harvey: FAST TCP in high-speed networks: An experimental study. (2008), 08
- [36] HORNSBY, A. ; WALSH, R.: From instant messaging to cloud computing, an XMPP review. En: *IEEE International Symposium on Consumer Electronics (ISCE 2010),* 2010, p. 1–6

- [37] HOSTING, Bytemark. *Welcome to Raspbian*. url <https://www.raspbian.org/>. 2012
- [38] HU, H. ; ZHANG, J. ; ZHENG, X. ; YANG, Y. ; Wu, P.: Self-configuration and self-optimization for LTE networks. En: *IEEE Communications Magazine* 48 (2010), Nr. 2, p. 94–100
- [39] JIANTAO WANG ; LUN LI ; LOW, S. H. ; DOYLE, J. C.: Cross-layer optimization in TCP/IP networks. En: *IEEE/ACM Transactions on Networking* 13 (2005), Nr. 3, p. 582–595
- [40] JOHANSSON, Bjorn ; JOHANSSON, Mikael: Primal and dual approaches to distributed cross-layer optimization. (2005), 07
- [41] JOHNSON, D ; NTLATLAPA, N ; AICHELE, C: Simple pragmatic approach to mesh routing using BATMAN. En: *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, 2008
- [42] KATABI, Dina ; HANDLEY, Mark ; ROHRS, Charles: Congestion control for high bandwidth-delay product networks, 2002, p. 89–102
- [43] KELLY, F.P.: Fairness and Stability of End-to-End Congestion Control*. En: *European Journal of Control* 9 (2003), 12
- [44] KELLY, F.P. ; VOICE, Thomas: Stability of end-to-end algorithms for joint routing and rate control. En: *Computer Communication Review* 35 (2005), 04, p. 5–12
- [45] KONSTANTINOU, A. V. ; FLORISSI, D. ; YEMINI, Y.: Towards Self-Configuring Networks. En: *DARPA Active Networks Conference and Exposition*, 2002, p. 143–156
- [46] KOZAT, U. C. ; KONDYLIS, G. ; RYU, B. ; MARINA, M. K.: Virtual dynamic backbone for mobile ad hoc networks. En: *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)* Vol. 1, 2001, p. 250–255
- [47] KUMAR, S. ; LOBIYA, D. K. *Performance Improvement in Geographic Routing for Vehicular Ad Hoc Networks*. 2014
- [48] KWON, Taek ; GERLA, Mario: Efficient flooding with Passive Clustering (PC) in ad hoc networks. En: *Computer Communication Review* 32 (2002), 01, p. 44–56
- [49] LAZAR, A. A. ; PACIFICI, G.: Control of resources in broadband networks with quality of service guarantees. En: *IEEE Communications Magazine* 29 (1991), Nr. 10, p. 66–73
- [50] LEMA, Maria: Diseño de la topología de un área de red – *hoc*. (2008), 07

- [51] LIFANG LI ; GOLDSMITH, A. J.: Capacity and optimal resource allocation for fading broadcast channels .I. Ergodic capacity. En: *IEEE Transactions on Information Theory* 47 (2001), Nr. 3, p. 1083–1102
- [52] In: LIPMAN, Justin ; LIU, Hai ; STOJMENOVIC, Ivan: *Broadcast in Ad Hoc Networks*. 2009, p. 121–150
- [53] LU, Z. ; SUIZO, N. ; NAGARAJAN, R. ; VILLAIT, A. ; LOCHNER, W. M. ; RONNE, J. ; DESUR, G.: *Self-configuring network*. 2012
- [54] MAKSIMOVIC, Mirjana ; VUJOVIC, Vladimir ; DAVIDOVIC, Nikola ; MILOSEVIC, Vladimir ; PERISIC, Branko: Raspberry Pi as Internet of Things hardware: Performances and Constraints. En: *IcETRAN 2014*, 2014
- [55] MASSOULIÉ, Laurent ; ROBERTS, James: Bandwidth sharing and admission control for elastic traffic. En: *Telecommunication Systems* 15 (2000), p. 185–201
- [56] MORTIER, R. ; KICIMAN, E.: Autonomic network management: Some pragmatic considerations. En: *the 2006 SIGCOMM Workshop on Internet Network Management, INM06*, 2006, p. 89–93
- [57] MUNOS, J. *Diseno e implementacion de un servidor de video adaptativo en simulador de redes ad hoc*. 2009
- [58] NI, Yang ; KREMER, Ulrich ; STERE, Adrian ; IFTODE, Liviu: Programming Ad-hoc Networks of mobile and resource-constrained devices, 2005, p. 249–260
- [59] NOGUERA, José A.: *Sistema de diálogo basado en mensajería instantánea para el control de dispositivos en el internet de las cosas*, Tesis de Grado, 01 2016
- [60] ONE, Process. *Ejabberd*. url <https://www.ejabberd.im/>
- [61] PAGANINI, F. ; ZHIKUI WANG ; DOYLE, J. C. ; LOW, S. H.: Congestion control for high performance, stability, and fairness in general networks. En: *IEEE/ACM Transactions on Networking* 13 (2005), Nr. 1, p. 43–56
- [62] PERLMAN, S.G.: *Self-configuring, adaptive, three-dimensional, wireless network*. 2009
- [63] PI, Raspberry. *Raspberry Pi*. url [ps://www.raspberrypi.org](http://www.raspberrypi.org). 2012
- [64] PRYTZ, Mikael: On Optimization in Design of Telecommunications Networks with Multicast and Unicast Trafic. (2002), 01
- [65] QU, Da-peng ; WANG, Xing-wei ; HUANG, Min: Component based ant routing protocols analysis over mobile ad hoc networks. En: *Journal of Central South University* 20 (2013), 09, p. 2378–2387

- [66] RAO, R. ; VEERAIAH, Darnasi ; MANDHALA, Venkata ; KIM, Tai-hoon: Neighbor Position Verification with Improved Quality of Service in Mobile Ad-hoc Networks. En: *International Journal of Control and Automation* 8 (2015), 01, p. 83–92
- [67] SAINT-ANDRE, Peter ; SMITH, Kevin ; TRONÑON, Remko: *XMPP - The Definitive Guide: Building Real-Time Applications with Jabber Technologies*. 2009. – ISBN 978-0-596-52126-4
- [68] SHAMBHU UPADHYAYA, K. ; CHAUDHURY, M. W.: *Mobile Computing: Implementing Pervasive Information and Communications Technologies (Operations Research Computer Science Interfaces Series)*. 2002
- [69] SOCIETY, IEEE C.: *IEEE Transactions on Wireless Communications*. 2014
- [70] TANESBAUM, A.S ; WETHERALL, D.J.: *Redes de computadoras*. Pearson, 2014
- [71] TANG, Ao ; WANG, Jiantao ; HEGDE, Sanjay ; LOW, Steven: Equilibrium and fairness of networks shared by TCP Reno and Vegas/FAST. En: *Telecommunication Systems* 30 (2005), 12, p. 417–439
- [72] TING-CHAO HOU ; TZU-JANE TSAI: A access-based clustering protocol for multihop wireless ad hoc networks. En: *IEEE Journal on Selected Areas in Communications* 19 (2001), Nr. 7, p. 1201–1210
- [73] TLON. *TLON - Grupo de Investigación en Redes de Telecomunicaciones Dinámicas y Lenguajes de Programación Distribuidos*. url <http://www.tlon.unal.edu.co/>
- [74] VILARNAU, A. ; AMELL, E. *Desarrollo y analisis de una red ad-hoc con sensores inalámbricos*. 2007
- [75] WEI, D. ; CHAN, H. A.: Clustering Ad Hoc Networks: Schemes and Classifications. En: *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks* Vol. 3, 2006, p. 920–926
- [76] WEISER, Mark: Source code. En: *IEEE Computer* 20 (1987), Nr. 11, p. 66–73
- [77] XIAOJUN LIN ; SHROFF, N. B. ; SRIKANT, R.: A tutorial on cross-layer optimization in wireless networks. En: *IEEE Journal on Selected Areas in Communications* 24 (2006), Nr. 8, p. 1452–1463
- [78] ZHANG, Congzhe ; ZHOU, Mengchu ; YU, Ming: Ad hoc network routing and security: A review. En: *International Journal of Communication Systems* 20 (2007), 08, p. 909 – 925
- [79] ZHANG, H. ; WEN, X. ; WANG, B. ; ZHENG, W. ; LU, Z.: A Novel Self-Optimizing Handover Mechanism for Multi-service Provisioning in LTE-Advanced. En: *International Conference on Research Challenges in Computer Science*, 2009

A. Anexo:Instructivos de instalación

A.1. Instalar y configurar Raspbian en Raspberry Pi

Rasbian es un sistema operativo adaptado para los dispositivos Raspberry PI que se a convertido en la distribución preferida por los usuarios para estos dispositivos. Raspbian es una adaptación de Debian/Linux para arquitectura ARM, con los controladores para las Raspberry PI, además de otras adaptaciones como las herramientas que vienen pre-instaladas en la distribución.

Para entender mejor el contenido de este apartado, se propone una explicación detallada de los términos relacionados anteriormente, explicando qué es RaspberryPI, tecnología ARM, GNU/Linux y finalmente Raspbian.

Raspberry Pi

La fundación Raspberry PI es una organización que crea computadoras de bajo costo y alto rendimiento con el objetivo de aprender, resolver problemas y divertirse ¹. Hasta el momento se han publicado tres (3) versiones estándar, algunas versiones de computo de integración de varias tarjetas y diferentes accesorios para estos equipos. Los tres (3) modelos mas conocidos son:

- Raspberry Pi 1 modelo B+.
- Raspberry Pi 2 modelo B.
- Raspberry Pi 3 modelo B.

La gran diferencia entre estos tres (3) modelos está en capacidad de procesamiento, capacidad de memoria RAM y conectividad inalámbrica (Bluetooth y WiFi).

Estos dispositivos tienen las características de un computador de bajo costo, en un tamaño muy pequeño (similar a una tarjeta de crédito), con un procesador ARM (broadcom), almacenamiento en disco en una MicroSD, puertos USB, puerto Ethernet, conector para cámara (PI Camera), puerto de audio (3.5mm), puerto de video HDMI, alimentación de energía

¹<https://www.raspberrypi.org/about/>

MicroUSB (como el conector de energía de smartphones) y una característica que le aporta gran flexibilidad, el puerto GPIO (Pines de propósito general, digitales y análogos) para conexiones con circuitos eléctricos.

Tecnología ARM

La tecnología ARM es una arquitectura de procesadores RISC (Reduced Instruction Set Computer / Ordenador con Conjunto Reducido de Instrucciones) desarrollada por ARM Holding. Los procesadores ARM RISC requieren una cantidad de transistores mucho menor que los procesadores x86 CISC, lo que conlleva a una reducción en los costos, generación de calor y consumo energético. Así mismo, la simplicidad que ofrecen los hace ideales para aplicaciones de bajo consumo de cómputo.

Las características y ventajas de los procesadores ARM han hecho que se conviertan rápidamente en los procesadores por defecto de smartphones, tablets, smartwatches, enrutadores, computadores de bajo costo, entre otros.

ARM Holding genera licencias para que compañías fabricantes de chips puedan crear sus propios procesadores y microcontroladores basados en esta tecnología, por lo que encontramos en el mercado procesadores licenciados como Broadcom, Qualcom Snapdragon, Samsung Exynos, Nvidia Tegra, entre otros.

GNU/Linux

GNU/Linux es un sistema operativo de libre distribución y código abierto patrocinado por la Free Software Fundation ². El sistema operativo es el resultado del proyecto GNU ³ que utiliza un kernel llamado Linux ⁴.

Existen varias distribuciones de GNU, entre las mas conocidas tenemos a Debian ⁵, Redhat ⁶ y Suse ⁷. Estas distribuciones de GNU pueden utilizar el kernel de Linux (es el caso mas común) o pueden ejecutar kernel de otras arquitecturas como Hurd o kFreeBSD ⁸.

Teniendo en cuenta lo anterior, la correcta referenciación del sistema operativo sería Debian/Linux, Debian/kFreeBSD, Redhat/Linux, etc.

²<https://www.fsf.org/es>

³<https://www.gnu.org/gnu/gnu.html>

⁴<https://www.kernel.org/category/about.html>

⁵<https://www.debian.org/index.es.html>

⁶<https://www.redhat.com/es/technologies/linux-platforms/enterprise-linux>

⁷<https://www.suse.com/es-es/solutions/enterprise-linux/>

⁸<https://wiki.debian.org/es/Kernel>

Los sistemas operativos GNU/Linux son los más usados en ambientes de investigación y desarrollo, laboratorios, entidades educativas y ambientes donde se necesite tener una flexibilidad y control total sobre el sistema operativo, por ser de libre distribución y código abierto.

Raspbian

Raspbian es un sistema operativo derivado de Debian/Linux y optimizado para el hardware de la Raspberry PI con mas de 35.000 paquetes (aplicaciones) disponibles. Raspbian está basado específicamente en “Debian Wheezy armhf” con ajustes de compilación para los dispositivos Raspberry PI ⁹.

Raspbian es el sistema operativo (distribución) recomendado por la Fundación Raspberry PI para sus dispositivos ¹⁰.

También existe una compilación llamada NOOBS 11 (New Out Of the Box Software/Software Nuevo Fuera De La Caja) que viene en una imagen para la SD en la cual vienen precargados varios sistemas operativos, entre esos, Raspbian.

Instalación de Raspbian

La instalación del sistema operativo Raspbian en los dispositivos Raspberry PI se realiza de diferentes formas, dependiendo del sistema operativo que se use en el Computado Personal (PC). Existe un paso para instalar Raspbian en una Raspberry PI desde un PC Windows, otro proceso para instalar Raspbian en una Raspberry PI desde un MAC y otro proceso para realizar la instalación desde un PC GNU/Linux.

La imagen de Raspbian se puede descargar desde el siguiente link: <https://www.raspberrypi.org/downloads/raspbian/>.

La guía de instalación, con los pasos para cada sistema operativo de PC, se encuentra en el siguiente link: <https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

En este caso, la instalación de Raspbian se realizará desde un PC con GNU/Linux. Los requerimientos necesarios son:

- Un dispositivo Raspberry Pi.
- Una tarjeta MicroSD con una capacidad mínima recomendada de 4GB.

⁹<https://www.raspbian.org/RaspbianAbout>

¹⁰<https://www.raspberrypi.org/help/faqs/softwareOS>

- Imagen de Raspbian (https://downloads.raspberrypi.org/raspbian_latest)

Lo primero que se debe realizar es verificar que la imagen de Raspbian descargada esté bien y no haya quedado corrupta, para lo cual compararemos el hash SHA-1 publicado en la página, con hash calculado:

```

1 $ echo b4c0dedd56c2cfb7913322cadb9ba81e19b6daee
2 b4c0dedd56c2cfb7913322cadb9ba81e19b6daee
3 $ shasum 2017-03-02-raspbian-jessie.zip
4 b4c0dedd56c2cfb7913322cadb9ba81e19b6daee 2017-03-02-raspbian-jessie.zip
5 $
```

Como se puede ver en los comandos anteriores, el hash SHA-1 es el mismo, tanto el publicado como el calculado, lo cual indica que la imagen no está corrupta y podemos proceder con la instalación.

El siguiente paso es conectar la tarjeta MicroSD a nuestro PC (con la ayuda de algun adaptador MicroSD/SD) y verificar el nombre con el que la reconoce el sistema operativo. Generalmente el nombre asignado a la tarjeta SD en GNU/Linux es /dev/mmcblk0p1. Teniendo identificado el dispositivo, se procede a desmontar la unidad. En el siguiente recuadro podremos ver la información relativa al dispositivo como desmontarlo.

```

1 # df -h
Filesystem      Size  Used Avail Use% Mounted on
...
/dev/mmcblk0p1  3.8G  0   3.8G  0% /media/user/D1356-D714
2 # umount /media/user/D1356-D714
3 #
```

Este procedimiento es necesario realizar por dos razones, la primera es porque el sistema operativo auto-monta la tarjeta SD automáticamente y lo segundo y más importante, es que la herramienta que se utiliza para escribir la imagen en la SD realiza la copia “en crudo” (raw), o sea bit a bit, y este proceso necesita que el dispositivo esté libre (no esté montado o en uso).

La herramienta que realiza la escritura de la imagen es dd. DD es una herramienta que puede leer y escribir entre dos archivos (archivos de imagen, discos duros, tarjetas SD, etc). La sintaxis es la siguiente:

```
dd if=archivo_para_leer of=archivo_para_escribir bs=cantidad_bytes_escritos_por_paso
```

Para mas información del comando dd, consultar el manual (man dd). Los pasos para descomprimir la imagen descargada y escribirla en la tarjeta SD, se describen a continuación:

```
1 # unzip 2017-03-02-raspbian-jessie.zip
Archive: 2017-03-02-raspbian-jessie.zip
      inflating: 2017-03-02-raspbian-jessie.img

2 # dd bs=4M if=2017-03-02-raspbian-jessie.img of=/dev/mmcblk0p1
1098+1 records in
1098+1 records out
4393533440 bytes (4,4 GB, 4,1 GiB) copied, 340,675 s, 12,5 MB/s

3 # sync
```

Después de estos pasos la tarjeta SD tiene instalada la imagen de Raspbian. El paso a seguir es introducir la MicroSD en la Raspberry PI, encenderla y esperar a que el sistema operativo inicie por primera vez y ajuste automáticamente algunos parámetros dependiendo del dispositivo RaspberryPI. En vez de utilizar la herramienta dd también es posible utilizar una herramienta gráfica llamada etcher, que esta disponible para varios sistemas operativos.

Cuando termine de iniciar el sistema operativo, podremos iniciar sesión y comenzar a utilizarlo. Las credenciales por defecto son:

- **Usuario:** Pi.
- **Contraseña:** raspberry.

A.2. Instalar y Configurar B.A.T.M.A.N y A.L.F.R.E.D

Para instalar y configurar un cluster mesh se requiere configurar una red que comunique a varios nodos de una red. Primero se explicarán los términos para entender exactamente cuales son los procedimientos que se realizarán.

Los clúster o las agrupaciones de equipos es una tecnología cada vez mas usada por sus beneficios. Los clúster ofrecen una mayor capacidad de procesamiento, descentralización de datos críticos para ofrecer redundancia, capacidades de escalamiento horizontal dinámico, balanceo de carga, altos niveles de procesamiento, alta disponibilidad y procesamiento distribuido.

Las redes inalámbricas o wireless permiten conectar equipos en una red sin cables. En ambientes de área local son conocidas como redes Wi-Fi.

Las redes tipo mesh o malla son redes donde todos los nodos que pertenecen a la red están interconectados mutuamente. Esto asegura un algo grado de disponibilidad de conexiones. Hay una variación de las redes mesh, las redes mesh móviles, estas redes son una combinación entre las redes Wi-Fi y las redes mesh, permitiendo interconectar varios nodos cercanos de forma inalámbrica entre ellos. La característica de móvil hace referencia a que los nodos de la red pueden salir y entrar de forma dinámica de la red, sin afectar el funcionamiento de la red.

Las redes mesh móviles requieren un protocolo de enrutamiento dinámico que permita la comunicación entre todos los nodos de la red, teniendo en cuenta que la red no siempre tendrá ni los mismos nodos ni la misma cantidad de nodos. Para enrutar los datos de los nodos de este tipo de redes existe BATMAN. BATMAN es un protocolo de enrutamiento para redes mesh ad-hoc, multi-hop, desarrollado bajo el proyecto Open Mesh. Open Mesh es un proyecto que ofrece herramientas para construir redes mesh abiertas y libres.

BATMAN es un acrónimo de “Better Approach To Mobile Ad-hoc Networking” (Mejor enfoque para redes ad-hoc móviles) y está destinado a reemplazar el protocolo de enrutamiento más usado en redes mesh, OLSR (Optimized Link State Protocol), mejorando la descentralización de la mejor ruta a través de la red, no permitiendo que un solo nodo tenga todos los datos ¹¹.

El enfoque del algoritmo BATMAN es dividir el conocimiento sobre las mejores rutas de extremo a extremo entre nodos de la mesh a todos los nodos participantes. Cada nodo percibe y mantiene solamente la información sobre el mejor salto siguiente hacia todos los otros nodos. Por lo tanto, la necesidad de un conocimiento global sobre los cambios de topología local resulta innecesaria. Además, un mecanismo de inundación basado en eventos perma-

¹¹<https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept>

nente, evita la acumulación de información de topología contradictoria (la razón habitual de la existencia de bucles de enrutamiento) y limita la cantidad de mensajes de inundación de topología en la mesh (evitando así sobrecargas de tráfico de control). El algoritmo está diseñado para tratar con redes basadas en enlaces no confiables.

BATMAN ó B.A.T.M.A.N. es un protocolo que funciona en la capa 3 del modelo OSI. Open Mesh desarrolló una versión avanzada de BATMAN llamada BATMAN-ADV, la cual pasa a funcionar en Capa 2, en forma de módulo del kernel de linux.

La mayoría de otras implementaciones de protocolos de enrutamiento inalámbrico (por ejemplo, el demonio de batman) operan en la capa 3, lo que significa que intercambian información de enrutamiento enviando paquetes UDP y poniendo en práctica su decisión de enrutamiento manipulando la tabla de enrutamiento del kernel. Batman-adv opera completamente en OSI Layer 2, no sólo la información de enrutamiento se transporta utilizando tramas de ethernet sin procesar, sino también el tráfico de datos es manejado por batman-adv.

Encapsula y reenvía todo el tráfico hasta que llega al destino, emulando así un switch virtual para todos los nodos participantes. Por lo tanto, todos los nodos se ven como enlaces locales y no son conscientes de la topología de red, así como no son afectados por los cambios de red.

Este diseño permite tener algunas características interesantes:

- Es agnóstico de la capa de la red-puede usar el protocolo de red que se desee encima de batman-adv: IPv4, IPv6, DHCP, IPX.
- Los nodos pueden participar en una red mesh sin tener una dirección IP.
- Roaming de clientes que no son de la red mesh
- Optimización del flujo de datos a través de la mesh (por ejemplo, alternación de interfaces, multidifusión, corrección de errores hacia adelante, etc.)
- Ejecutar protocolos basados en difusión/multidifusión sobre los clientes de mesh y clientes que no pertenezcan a la mesh (vecindario de Windows, mDNS, streaming, etc.)

BATCTL

batctl o “batman control” es una herramienta útil para configurar y depurar el módulo del kernel batman-adv. Ofrece una interfaz conveniente para todos los ajustes del módulo, así como información de estado. También contiene una versión de la capa 2 de ping, traceroute y tcpdump, ya que el switch virtual es completamente transparente para todos los protocolos

anteriores a la capa 2.

Configurar Batman-ADV

Batman-adb es un módulo del kernel linux, por lo que no es necesario instalarlo, únicamente activarlo para que el sistema operativo lo cargue en “tiempo de inicio” (boot time) del kernel. Para esto, agregamos “batman-adv” al archivo “/etc/modules” y reiniciamos el sistema operativo. Después, podemos validar que el modulo quedó montado con el comando “lsmod” el cual lista todos los módulos cargados en el kernel.

```
1 # echo batman-adv >> /etc/modules
2 # reboot
3 # lsmod | grep batman
batman_adv      184048  0
4 #
```

Ahora, el paso a seguir es configurar la interfaz inalámbrica, configurar la interfaz virtual “bat0”, y asociar estas dos interfaces de red. Esta tarea se realiza sobre el archivo de configuración de interfaces “/etc/network/interfaces”.

Configuración de la Interfaz inalámbrica

Para saber que interfaz WiFi tiene el dispositivo utilizaremos el comando “iwconfig” que nos indica cual interfaz de red tiene IEEE 802.11 (WiFi). Teniendo identificada la interfaz WiFi, se procede a configurar los parámetros de la red mesh. A continuación se muestra el proceso:

```
1 # iwconfig

lo      no wireless extensions.

wlan0   IEEE 802.11 ESSID:"Mi red Wi-Fi"
        Mode:Managed Frequency:2.462 GHz Access Point: 54:A6:19:4B:BF:94
        Bit Rate=6.5 Mb/s Tx-Power=31 dBm
        Retry short limit:7 RTS thr:off Fragment thr:off
        Encryption key:off
        Power Management:on
        Link Quality=37/70 Signal level=-73 dBm
        Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
        Tx excessive retries:615 Invalid misc:0 Missed beacon:0

eth0    no wireless extensions.

2 # vim /etc/network/interfaces

auto wlan0

iface wlan0 inet6 manual
        mtu 1500
        wireless-channel 1
        wireless-essid my-mesh-network
        wireless-mode ad-hoc

3 #
```

Finalmente, se asocia la interfaz bat0 con la interfaz wlan0 y se asigna una dirección IP a bat0 (esto es opcional).

```

1 # vim /etc/network/interfaces

...
auto bat0

iface bat0 inet static
    pre-up /usr/sbin/batctl if add wlan0
    address 192.168.20.4/24
...
3 #

```

Con estos pasos queda “instalado” y configurado batman-adv, con lo que permite la conectividad en red mesh de todos sus nodos.

Para realizar pruebas de conectividad entre todos los nodos de la red mesh se puede inicialmente utilizar el comando batctl para revisar la visibilidad entre los nodos de la mesh y revisar la conectividad a nivel de capa 2. Posteriormente, se pueden hacer pruebas de conectividad a nivel de capa 3 con el comando ping.

Con el siguiente comando se muestran los vecinos con los cuales se tiene visibilidad a nivel de capa 2, para esto se utiliza el comando batcl (batman control) con el parámetro *n* de neighbors:

```

1 # batctl n
[B.A.T.M.A.N. adv 2016.4, MainIF/MAC: wlan0/06:00:27:37:17:3c (bat0/e6:b6:a8:e9:1a:ec BATMAN_IV]
IF   Neighbor      last-seen
wlan0 06:00:27:48:16:7e  0.612s
wlan0 06:00:27:70:b4:8d  0.612s
wlan0 06:00:27:5d:e0:6f  0.612s
wlan0 06:00:27:b0:b6:df  0.612s
wlan0 06:00:27:40:98:fe  0.612s
wlan0 06:00:27:46:16:7d  0.612s
2 #

```

Si se desea hacer una prueba de conectividad entre nodos, a nivel de capa 2, el comando batctl tiene la funcionalidad de un ping capa 2. A continuación se muestra el resultado del comando:

```
1 # batctl p 06:00:27:48:16:7e
PING 06:00:27:48:16:7e (06:00:27:48:16:7e) 20(48) bytes of data
20 bytes from 06:00:27:48:16:7e icmp_seq=1 ttl=50 time=0.28 ms
20 bytes from 06:00:27:48:16:7e icmp_seq=2 ttl=50 time=0.40 ms
20 bytes from 06:00:27:48:16:7e icmp_seq=3 ttl=50 time=0.46 ms
20 bytes from 06:00:27:48:16:7e icmp_seq=4 ttl=50 time=0.30 ms
20 bytes from 06:00:27:48:16:7e icmp_seq=5 ttl=50 time=0.43 ms
--- 06:00:27:48:16:7e ping statistics ---
5 packets transmitted, 5 received, 0% packet loss
rtt min/avg/max/mdev = 0.281/0.375/0.461/0.070 ms
2 #
```

Para más información sobre configuración de batman-adv, puede consultar la guía rápida:
<https://www.open-mesh.org/projects/batman-adv/wiki/Quick-start-guide>

A.3. Instalar y configurar Pacemaker y Corosync (PCS-Pacemaker Cluster Stack)

La palabra clúster viene del inglés “cluster”, que significa “racimo” o “grupo” que hace referencia a una agrupación de equipos que se comportan como un solo equipo. Estos clusters son creados para distribuir información entre los equipos o nodos que conformen el clúster con el fin de ofrecer alto rendimiento, alta disponibilidad, balanceo de carga o escalabilidad.

Existen diferentes soluciones en el mercado para crear clústers de alta disponibilidad, algunas privativas y otras software libre; para el caso de este proyecto se utilizará una solución de software libre. Las soluciones de software libre más conocidas para clusters de alta disponibilidad son pacemaker y heartbeat. Cuando se consulta la documentación de pacemaker aparece dentro del proyecto ClusterLabs. ClusterLabs es un stack para clusters de alta disponibilidad que está compuesto principalmente por pacemaker y corosync. Heartbeat es soportado hasta la versión 2.0 de pacemaker, en las versiones posteriores solo está soportado corosync.

Pacemaker es un administrador de recursos de cluster, es decir, es la lógica responsable del ciclo de vida del software implementado. Controla un conjunto de computadoras (también llamados Nodos) a partir de reglas prescritas. Con Pacemaker se busca obtener la máxima disponibilidad para sus servicios de cluster (también llamados recursos) mediante la detección y recuperación de fallas a nivel de nodos y recursos mediante el uso de las capacidades de mensajería y membresía proporcionadas por su infraestructura de cluster preferida (Corosync o Heartbeat), y posiblemente mediante la utilización otras partes del stack global de cluster.

Las principales características de Pacemaker incluyen:

- Detección y recuperación de fallas de nodo y nivel de servicio.
- Almacenamiento independiente, sin requisitos de almacenamiento compartido.
- Es agnóstico de los recursos, todo lo que se puede secuenciar puede agruparse.
- Admite STONITH (“Shoot The Other Node In The Head” “Disparar al otro nodo en la cabeza” o “Dispara al nodo ofensivo en la cabeza”), a veces llamado STOMITH (“Dispara al otro miembro/máquina en la cabeza”), es una técnica cercamiento (fencing) en clusters de computadoras. El cercamiento (fencing) es el aislamiento de un nodo fallido para que no cause interrupciones en un cluster de computadoras. Como su nombre lo sugiere, STONITH mata los nodos fallidos reiniciando o apagando el nodo fallido.
- Admite clusters grandes y pequeños.

- Compatible con clusters basados en quórum y basados en recursos
- Admite prácticamente cualquier configuración de redundancia.
- La configuración se replica automáticamente y se puede actualizar desde cualquier nodo.
- Posibilidad de especificar prioridades en los servicios de todo el clúster.
- Herramientas de administración de clusters unificables y programables.
- Soporte para tipos de servicio avanzado, clones y multi-estado (maestro/esclavo)

Corosync¹² es un motor de clúster, en otras palabras, es un sistema de comunicación grupal con funciones adicionales para implementar alta disponibilidad dentro de las aplicaciones. Corosync ofrece, entre otras, cuatro características principales:

- Un modelo de comunicación grupal de procesos cerrado con garantías de sincronía virtual para crear máquinas de estado replicadas.
- Un administrador de disponibilidad simple que reinicia el proceso de solicitud cuando falla.
- Una base de datos en memoria de la configuración y estadísticas, que proporciona la capacidad de establecer, recuperar y recibir notificaciones de cambio de información.
- Un sistema de quórum que notifica las aplicaciones cuando se logra o se pierde el quórum.

Instalación de pacemaker y corosync

Pacemaker y Corosync se encuentran disponibles en los repositorios oficiales de la mayoría de distribuciones de GNU/Linux, incluidos Debian, Fedora, Red Hat Enterprise Linux, Suse Linux Enterprise Server y Ubuntu LTS. También es posible descargar el código fuente, compilarlo e instalarlo.

Para nuestro caso, estamos trabajando con Raspbian en el cual tiene los paquetes en sus repositorios oficiales, por lo tanto bastará con ejecutar el siguiente comando:

```
1 # apt install pacemaker corosync crm pcs
```

Cuando la instalación termine se abrán creado los archivos de configuración en /etc.

¹²<http://corosync.github.io/corosync/>

Configuración de pacemaker y corosync

Para que corosync pueda comunicarse con todos los miembros de un cluster deben compartir la misma configuración de cluster y una llave para poderse autenticar entre ellos. A continuación se muestra la ruta del archivo de configuración y la llave:

- /etc/corosync/corosync.conf
- /etc/corosync/authkey

La llave de autenticación puede generarse con el comando corosync-keygen.

Para este caso, vamos a usar la siguiente configuración de corosync (/etc/corosync/corosync.conf):

```
1  totam {  
2      version: 2  
3      secauth: off  
4      cluster_name: cluster_dlon1  
5      transport: udpu  
6      interface {  
7          ringnumber: 0  
8          bindnetaddr: 192.168.20.0  
9          broadcast: yes  
10         mcastaddr: 239.255.1.1  
11         mcastport: 5405  
12         ttl: 1  
13     }  
14 }  
15 nodalist {  
16     node {  
17         name: RP12  
18         ring0_addr: 192.168.20.3  
19     }  
20     node {  
21         name: RP13-3  
22         ring0_addr: 192.168.20.6  
23     }  
24     node {  
25         name: RP13-4  
26         ring0_addr: 192.168.20.7  
27     }  
28 }  
29     quorum (real_ip ocf:heartbeat:IPaddr2 par  
30         provider: corosync_votequorum|
```

```

31    expected_votes: 2
32    two_node: 1
33  ]
34  logging [
35    to_logfile: yes
36    logfile: /var/log/corosync/corosync.log
37    to_syslog: yes
38    fileline: off
39    timestamp: on
40    logger_subsys {
41      subsys: QUORUM
42      debug: on
43    }
44  ]
45  service {
46    name: pacemaker
47    ver: 0
48  }

```

Ahora se debe unificar en todos los nodos del mismo cluster el archivo de configuración y la llave de autenticación de corosync. Con este procedimiento ya debería crearse el cluster y podemos ver el estado con el siguiente comando:

```

1 # crm status
Stack: corosync
Current DC: RPI2 (version 1.1.16-94ff4df) - partition with quorum
Last updated: Thu Mar 22 22:24:51 2018
Last change: Thu Mar 22 22:19:02 2018 by hacluster via crmd on RPI2

3 nodes configured
0 resources configured

Online: [ RPI2 RPI3-3 RPI3-4 ]

No resources
2 #

```

Ya se tiene el cluster creado y configurado. Como tenemos un cluster de tres (3) nodos es necesario que deshabilitemos el quorum y como es sobre una red ad'hoc que tiene alta

latencia también se deshabilitará el STONITH. A continuación se muestran los comando para deshabilitar el stonith, el quorum y agregar el único registro que es una dirección IP flotante.

```
1 #crm configure property stonith-enabled=false
2 #crm configure property no-quorum-policy=ignore
3 #crm configure primitive virtual_ip ocf:heartbeat:IPaddr2 params ip="192.168.20.101" cidr_netmask="32" op monitor interval="10s" meta migration-threshold="10"crm status
```

Con estos pasos ejecutados satisfactoriamente se tiene un cluster instalado, configurado y con un recurso de dirección IP flotante. A continuación se muestra el resultado del comando `crm status`.

```
1 #crm status
Stack: corosync
Current DC: RPI2 (version 1.1.16-94ff4df) - partition with quorum
Last updated: Thu Mar 22 22:34:51 2018
Last change: Thu Mar 22 22:33:52 2018 by hacluster via crmd on RPI2
```

```
1 resources configured

Online: [ RPI2 RPI3-3 RPI3-4 ]

Full list of resources:
virtual_ip      (ocf:heartbeat:IPaddr2):    Started RPI2
```

Para mas información sobre configuración de pacemaker y corosync, puede la documentación en: https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/2.0/html/Clusters_from_Scratch/index.html

A.4. Instalar y configurar Ejjaberd y Finch

En este proyecto se utiliza mensajería instantánea como el servicio que usa los clúster de alta disponibilidad, las redes mesh y las redes ad-hoc. La mensajería instantánea es una solución de software que se puede usar desde múltiples dispositivos y sistemas operativos, y existen muchos productos y protocolos para este servicio, uno de ellos es XMPP. XMPP es un protocolo de mensajería instantánea que permite el envío de mensajes de texto entre los contactos de una organización y adicional permite enviar contenido multimedia como imágenes, audio, video y otros archivos adjuntos.

Para la implementación de XMMP se utilizará ejabberd y finch. Ejabberd y Finch son dos (2) soluciones de software libre para mensajería instantánea con XMPP. Ejabberd es un servidor XMPP muy usado actualmente, incluso WhatsApp en sus primeras versiones estuvo basado en ejabberd. Finch es un cliente de línea de comandos multiprotocolo para mensajería instantánea que comparte las funcionalidades de otro cliente multiprotocolo gráfico llamado pidgin.

Ejabberd jabberd es el servidor XMPP de facto en el mundo. El hecho de que se utilice para alimentar las implementaciones más grandes del mundo (como WhatsApp) no debería intimidarle. ejabberd es igualmente adecuado para pequeños casos. Ejabberd se ha diseñado desde cero, desde 2002, para una implementación empresarial robusta. El objetivo siempre ha sido disparar para la luna y eso lo convirtió en un éxito duradero. Ejabberd está diseñado específicamente para fines empresariales: es tolerante a fallos y puede utilizar los recursos de múltiples máquinas agrupadas en clústeres, y se puede escalar fácilmente cuando se requiere más capacidad (solo agregando un equipo/VM). Se desarrolló en un momento en el que el protocolo XMPP todavía se conocía como "Jabber", pero adoptó rápidamente la evolución del protocolo para admitir las diversas versiones de las RFC de XMPP. También fomenta la innovación y la experimentación al admitir la mayoría, si no todas, las extensiones producidas por el XSF.

Finch es un programa de mensajería instantánea basado en la consola que le permite iniciar sesión en Jabber, GoogleTalk, IRC y otras redes de mensajería instantánea. Se ejecuta en sistemas Unix's. Utiliza GLib y ncurses. Ejabberd está desarrollado en erlang, por lo tanto usa Erlang Run-Time System Application (erts) y Erlang Virtual Machine (beam) para ejecutarse. También usa una base de datos distribuida llama Mnesia, que es utilizada para almacenar los usuarios, grupos, información de chats, información del cluster, entre otros.

Instalación Ejjaberd y Finch

Ejabberd y Finch se encuentran disponibles en los repositorios oficiales de la mayoría de distribuciones de GNU/Linux, incluidos Debian, Fedora, Red Hat Enterprise Linux, Suse Linux

Enterprise Server y Ubuntu LTS. También es posible descargar el código fuente, compilarlo e instalarlo.

Para nuestro caso, estamos trabajando con Raspbian en el cual tiene los paquetes en sus repositorios oficiales, por lo tanto bastará con ejecutar el siguiente comando:

```
1 # apt install ejabberd finch
```

Configuración de Ejabberd y Finch

La configuración de Ejabberd se hace igual para todos los nodos del mismo cluster en el cual se especifica que el servicio escuche peticiones por todas las interfaces (0.0.0.0) para que sea transparente para todos los nodos. Adicional, es necesario que todos los nodos del mismo cluster compartan (debe copiarse el mismo archivo en cada nodo del mismo cluster) la misma cookie de erlang, la cual se puede encontrar en la siguiente ruta “/var/lib/ejabberd/.erlang.cookie”. El archivo de configuración ejabberd se encuentra en la ruta “/etc/ejabberd/ejabberd.yml”. Para simplificar la configuración se provee una copia del archivo de configuración y se indican las líneas que deben modificarse.

Para el siguiente archivo, únicamente es necesario modificar las líneas 7, 69 y 70. La modificación consiste en cambiar el dominio del cluster al que pertenece.

```
1 loglevel: 4
2 log_rotate_size: 0
3 log_rotate_date: ""
4 log_rate_limit: 100
5
6 hosts:
7 - "tlon1.net"
8
9 listen:
10 -
11 port: 5222
12 ip: "0.0.0.0"
13 module: ejabberd_c2s
14 certfile: "/etc/ejabberd/ejabberd.pem"
15 starttls: true
16 protocol_options:
17 - "no_sslv3"
18 max_stanza_size: 65536
19 shaper: c2s_shaper
```

```
20 access: c2s
21 zlib: true
22 resend_on_timeout: if_offline
23 -
24 port: 5269
25 ip: "0.0.0.0"
26 module: ejabberd_s2s_in
27 shaper: s2s_shaper
28 max_stanza_size: 131072
29 -
30 port: 5280
31 ip: "0.0.0.0"
32 module: ejabberd_http
33 http_bind: true
34 web_admin: true
35 -
36 -
37 port: 5281
38 ip: "0.0.0.0"
39 module: ejabberd_http
40 request_handlers:
```

```
41     "/websocket": ejabberd_http ws
42     http_bind: true
43     tls: true
44     certfile: "/etc/ejabberd/ejabberd.pem"
45
46     disable_sasl_mechanisms: "digest-md5"
47     s2s_use_starttls: optional
48     s2s_certfile: "/etc/ejabberd/ejabberd.pem"
49
50     s2s_protocol_options:
51       - "no_sslv3"
52
53     outgoing_s2s_families:
```

```
54   - ipv4
55   outgoing_s2s_timeout: 10000
56   auth_method: internal
57   auth_password_format: scram
58   shaper:
59     normal: 1000
60     fast: 50000
61   max_fsm_queue: 1000
62   acl:
63     admin:
64       user:
65         - "admin@tlon1.net"
66         - "jhon@tlon1.net"
67     local:
68       user_regex: ""
69   loopback:
70     ip:
71       - "127.0.0.0/8"
72   access:
73   configure:
74     admin: allow
```

```
75 shaper_rules:
76   max_user_sessions: 10
77   max_user_offline_messages:
78     - 5000: admin
79     - 100
80   c2s_shaper:
81     - none: admin
82     - normal
83   s2s_shaper: fast
84
85 access_rules:
86   local:
87     - allow: local
```

```
88   c2s:
89     - deny: blocked
90     - allow
91   announce:
92     - allow: admin
93   configure:
94     - allow: admin
95   muc_create:
96     - allow: local
97   pubsub_createnode:
98     - allow: local
99   register:
100    - allow
101   trusted_network:
102    - allow: loopback
103
104   language: "en"
105
106   modules:
107     mod_adhoc: []
108     mod_admin_extra: []
```

```
109 mod_announce: # recommends mod_adhoc
110 access: announce
111 mod_blocking: {} # requires mod_privacy
112 mod_caps: {}
113 mod_carboncopy: {}
114 mod_client_state: {}
115 mod_configure: {} # requires mod_adhoc
116 mod_disco: {}
117 mod_echo: {}
118 mod_irc: {}
119 mod_http_bind: {}
120 mod_last: {}
121 mod_muc:
```

```
122 access:
123   - allow
124 access_admin:
125   - allow: admin
126 access_create: muc_create
127 access_persistent: muc_create
128 mod_muc_admin: {}
129 mod_offline:
130   access_max_user_messages: max_user_offline_messages
131 mod_ping: {}
132 | mod_privacy: {}
133 mod_private: {}
134 mod_pubsub:
135   access_createnode: pubsub_createnode
136   ignore_pep_from_offline: true
137   last_item_cache: false
138   plugins:
139     - "flat"
140     - "hometree"
```

```
137 last_item_cache: false
138 plugins:
139 - "flat"
140 - "hometree"
141 - "pep" # pep requires mod_caps
142 mod_roster:
143 versioning: true
144 mod_shared_roster: []
145 mod_stats: []
146 mod_time: []
147 mod_vcard:
148 search: false
149 mod_version: []
150
151
152 allow_contrib_modules: true
```

Posterior a tener el archivo de configuración y la cookie unificada en cada equipo del mismo cluster, debe configurarse el grupo “everybody” y los miembros son “@all” para que todo los usuarios del mismo cluster puedan verse. Esta configuración se realiza desde la interfaz web que es accesible desde la siguiente URL “<http://IP:5280/admin>”. En el formulario de grupos en la interfaz web de administración, poner los siguientes datos:

- Nombre: everybody
- Descripción: This group contains everybody
- Miembros: @all@
- Mostrar grupos: everybody

A continuación se muestra una imagen de la interfaz web con estos datos:

Iniciar y administrar el servicio ejabberd

El servicio o demonio de ejabberd se puede gestionar (iniciar, detener, reiniciar) con el comando del sistema “systemctl”, sin embargo el comando “ejabberdctl” tiene muchas mas opciones para gestionar el servicio. A continuación se muestran las sentencias de ejabberdctl para gestionar el servicio.

- Mostrar el estado del servicio

```
# ejabberdctl status
The node 'ejabberd@RPI1-1' is started with status: started
ejabberd 16.09 is running in that node
#
```

- Iniciar el servicio:

```
# ejabberdctl stop
#
```

- Detener el Servicio

```
# ejabberdctl stop
#
```

- Crear un usuario de chat

```
# ejabberdctl register admin tlon1.net jh0n.  
User admin@tlon1.net successfully registered  
#
```

- Mostrar usuarios de chats existentes:

```
# ejabberdctl unregister admin tlon1.net  
#
```

- Eliminar usuarios de chats existentes

```
# ejabberdctl --no-time join_cluster ejabberd@RPI3-1  
#
```

- Crear cluster de ejabberd

```
# ejabberdctl --no-time join_cluster ejabberd@RPI3-1  
#
```

- Listar los nodos del cluster ejabberd

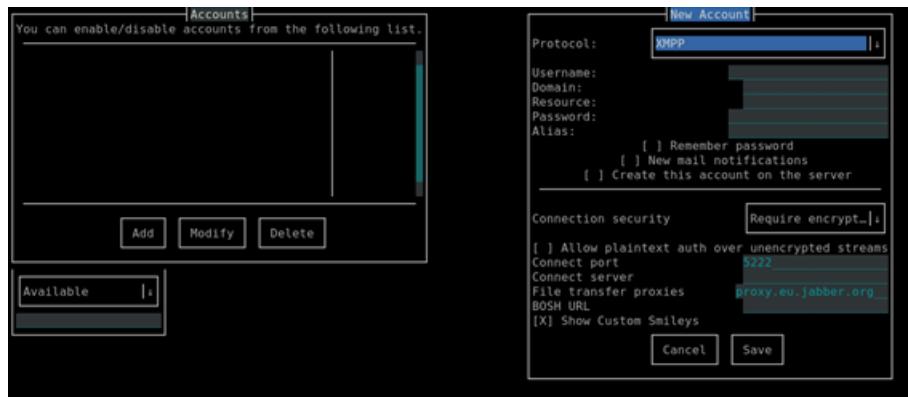
```
# ejabberdctl list_cluster  
'ejabberd@RPI1-2'  
'ejabberd@RPI3-1'  
ejabberd@RPI1-1  
#
```

Uso de cliente de mensajería Finch

Finch es un cliente de mensajería instantánea derivado del famoso cliente de mensajería llamado Pidgin¹³. Finch funciona en la consola de comandos (no requiere X) y puede usarse, entre otros protocolos, para XMPP.

Para ejecutar finch, se llama únicamente el binario “finch” y se mostrará una interfaz de usuario basada en la consola, como se muestra en la siguiente imagen:

¹³<https://www.pidgin.im/>



Para navegar entre las opciones de las ventanas de finch usar TAB. Para expandir los menús contextuales o seleccionar opciones usar BARRA ESPACIADORA.

Cómo es una interfaz basada en la consola, es necesario utilizar combinación de teclas o atajos de teclado. En esta combinación de teclas es necesario identificar cuál es la tecla que corresponde a META que generalmente es ALT. Esta tecla META será representada con la letra M en mayúscula en las siguientes combinaciones de teclas, seguida de la tecla que debe combinarse para ejecutar una tarea.

Combinación de teclas:

- Cambiar entre ventanas: M+n / M+p.
- Listar ventanas: M+w.
- Cerrar ventana: M+c.
- Mostrar ventana del menú: F10.
- Abrir el menú de una ventana: F11

Teniendo en cuenta lo anterior, solo queda utilizar la interfaz de usuario basada en la consola de comandos de finch para configurar los datos del usuario con los siguientes datos:

- Usuario.
- Dominio.
- Contraseña.
- Servidor para conectarse.

A continuación se muestra una imagen de como se ve la ventana para configurar una nueva cuenta:



Para mas información sobre configuración de ejabberd, puede la documentación en: <https://docs.ejabberd.im/>.

Para mas información sobre configuración de finch, puede la documentación en: <https://developer.pidgin.im/wiki/Using%20Finch>.