ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

Diseño e implementación de una plataforma de gestión mediante LibreNMS: Monitorización y control de la red privada del laboratorio de Telemática (GIT – UNICAN)

(Design and implementation of a management platform using LibreNMS: Monitoring and control of the Telematics laboratory network (GIT – UNICAN))

Para acceder al Título de

Graduado en Ingeniería de Tecnologías de Telecomunicación

Autor: Juncal Cantos San Emeterio

Septiembre - 2021

E.T.S. DE INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Juncal Cantos San Emeterio
Director del TFG: José Ángel Irastorza Teja
Título: "Diseño e implementación de una plataforma de gestión mediante LibreNMS: Monitorización y control de la red privada del Laboratorio de Telemática (GIT – UNICAN)"
Title: "Design and implementation of a management platform using LibreNMS: Monitoring and control of the Telematics laboratory network (GIT – UNICAN)"

Presentado a examen el día: 29 de septiembre de 2021

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:
Presidente (Apellidos, Nombre): García Arranz, Marta
Secretario (Apellidos, Nombre): Irastorza Teja, José Ángel
Vocal (Apellidos, Nombre): Sánchez González, Luis

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

Agradecimientos

Antes de nada, quisiera dedicar unas palabras para dar las gracias a aquellas personas que me han tendido una mano para ayudarme a lograr uno de mis objetivos más importantes hasta ahora.

En primer lugar a mi familia por todos sus esfuerzos y facilitarme todo cuanto era posible para que estudiase lo que me hiciese sentir más realizada. A mi Ama por preocuparse continuamente por mí. Y a mi Aita quien me enseñó el significado de lo que conlleva el sacrificio, la perseverancia y la disciplina. Siempre apostó por mi capacidad de superación y supo cómo apoyarme en mis momentos más débiles durante la carrera. A mi hermana Ainhoa, por preocuparse con sus continuas llamadas preguntándome por los resultados de los exámenes. A Juan, por hacerme de guía en la organización de mis estudios. A mis Aitites, Daniel y Lali, por darme tanto cariño y presumir de nieta. Y, por supuesto, a Rigo, mi pareja, quien me ha enseñado a no tirar la toalla y tener ambiciones en esta vida.

También quisiera tener en cuenta a compañeros, algunos de ellos ya amigos, por compartir tantos momentos de estudio y distracción durante estos años. Así como a los profesores que me han aportado el conocimiento necesario para poder avanzar en mi camino a la meta. Y como no, a mí tutor José Ángel, por su apoyo, paciencia y darme ánimos para terminar esta etapa tan especial.

Os estaré siempre muy agradecida.

Resumen

La utilización de protocolos y estándares han facilitado el control de las redes informáticas desde sus inicios. Siendo hoy éstas cada vez más heterogéneas, proporcionando mayor número de servicios, su gestión ha sido una ardua tarea por implementar para poder adaptarse a su evolución continua.

Los técnicos o administradores de las redes han tenido que buscar métodos alternativos para cubrir las necesidades de gestión de las organizaciones, ya que se espera que todas y cada una de sus áreas principales funcionen de manera óptima y garanticen las operaciones de los usuarios con la red de datos. A través de estos, debe ser posible medir, mantener y mejorar los procesos informáticos.

Gracias a la utilización de la plataforma de gestión LibreNMS, será posible lograr una mejor calidad de servicio, dado que su sistema permite administrar una base de datos con todas las configuraciones de los equipos, pudiendo monitorizarlos gráficamente, detectar incidencias mediante el aviso de alertas que se obtiene de los mensajes SNMP, como prevenir futuros fallos de red y actuar con la mayor brevedad posible.

Palabras clave: Monitorización, LibreNMS, sistema, red, agente, gestión, servicio, SNMP, router, switch, servidor, notificación, registro de eventos, gráficos, informe, widget

Abstract

The use of protocols and standards have provided the control of computer networks since its their very beginning. These days, the networks are increasingly heterogeneous, providing a greater number of services, their management has been an arduous task to implement in order to adapt to their continuous evolution.

Technicians or network administrators have had to look for alternative methods to meet the management requirements of organizations, since each and every one of their main areas is expected to work optimally and guarantee the operations of users with the data network. Through these, it must be possible to measure, maintain and improve IT processes.

Thanks to the use of the LibreNMS management platform, it will be possible to achieve a better quality of service, since its system allows to manage a database with all the configurations of the equipment, being able to monitor them graphically, detect problems through the warning of alerts that is obtained from the SNMP messages, as well as preventing future network failures and act as soon as possible.

Keywords: Monitoring, LibreNMS, network, system, network, agent, management, service, SNMP, router, switch, server, notification, eventlog, graphics, report, widget

Índice

Índice de figuras

Siglas

SNMP	Simple Network Management Protocol
PC	Personal Computer
GNU	GNU's Not Unix
CMIP	Common Management Information Protocol
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
OSI	Model Open Systems Interconnection
ISO	Modelo de Interconexión de Sistemas Abiertos
TMN	Telecommunication Management Network
RDSI	Red Digital de Servicios Integrados
GSM	Global System for Mobile Communications
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
BGP	Border Gateway Protocol
OSPF	Open Shortest Path First
FDP	Foundry Discovery Protocol
LLDP	Link Layer Discovery Protocol
ARP	Address Resolution Protocol
SMI	Structure of Management Information
MIB	Management Information Base
IAB	Internet Architecture Board
SGMP	Simple Gateway Monitoring Protocol
RFC	Request For Comments
IETF	Internet Engineering Task Force
RMON	Remote MONitor
NMS	Network Management Station
OID	Object Identifier
ASN	Abstract Syntax Notation One
GUI	Graphical User Interface
DBMS	DataBase Management System
API	Application Programming Interfaces
REST	Representational State Transfer
CPD	Data Center
TIC	Tecnologías de la Información y la Comunicación
SMS	Short Message Service
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
MAC	Media Access Control
LDAP	Lightweight Directory Access Protocol
RADIUS	Remote Authentication Dial In User Service
RANCID	Really Awesome New Cisco confIg Differ
CLI	Common Line Interface
ICMP	Internet Control Message Protocol
PING	Packet Internet Groper
CPU	Central Processing Unit

1. Introducción

El objetivo de este capítulo es servir de introducción a este documento. Durante su desarrollo se expondrá el contexto, que servirá para enmarcar el escenario en el que se implementará la plataforma de gestión, la motivación de la que surge su creación, y los objetivos, aquellos de carácter general que se pretenden cumplir, y un resumen de la estructura de la memoria, que ayudará a situar al lector a lo largo del trabajo fin de grado.

1.1. Contexto

La gestión de redes abarca hoy en día muchos aspectos, que pueden resumirse o sintetizarse en tareas de despliegue, integración y coordinación del hardware, software y los elementos humanos para monitorizar, probar, sondear, configurar, analizar, evaluar y controlar los recursos de una red para conseguir niveles de trabajo y de servicio adecuados a los objetivos de una instalación y de una organización.

La necesidad de gestionar redes desde el punto de vista de la gestión y de los servicios es, hoy por hoy, una ardua tarea debido a que se necesitan profesionales capaces de conocer tecnologías asociadas a redes y servicios, y posean una fuerte capacidad sistémica de integrar conocimiento de telecomunicaciones con conocimiento de hardware y software informático de redes y de datos.

A su vez, esta gestión ha ido adquiriendo cada vez mayor complejidad dado que lo que en un principio sólo estaba constituida por una red LAN y los enlaces WAN, en la actualidad las redes son más heterogéneas, enlazando sistemas de telecomunicaciones, equipos informáticos, Internet, servicios multimedia como videoconferencia, las aplicaciones remotas y hasta las compras a través de medios electrónicos.

Por consiguiente, la constante evolución de todos y cada uno de estos servicios ha involucrado de forma paralela la ejecución de estrategias y métodos para el control de esta información al tiempo de proveer de mecanismos de seguridad para proteger la integridad de los datos en la red. Así es como han ido surgiendo múltiples aplicaciones para dar respuesta a este tipo de gestión, y facilitar la labor a quien se encarga de administrar y controlar la red. Estas herramientas software hacen posible prevenir y detectar problemas en la red con la mayor anticipación posible, buscando minimizar errores, y obtener un funcionamiento adecuado.

1.2. Motivación y objetivos

Se parte del escenario de la red del laboratorio del Grupo de Telemática de la Universidad de Cantabria, en el que se lleva a cabo la realización del contenido práctico de diversas asignaturas, y que se emplea por un considerable número de alumnos y docentes, los cuales pueden cambiar algún parámetro de la configuración o hurgar en el cableado provocando incidencias o resultando en un completo colapso, dejando la red inactiva.

Este Trabajo Fin de Grado nace de la necesidad de disponer de una plataforma software para la gestión de los dispositivos (routers, switches, servidores, PCs, impresoras, inalámbricos, hubs y demás dispositivos gestionables) siendo capaz de monitorizarlos, de forma que garantice su correcto funcionamiento.

Lo primero que se debe plantear es el estudio del tipo de información que será necesaria gestionar, poniendo hincapié en la gestión de los fallos, la gestión de configuración y la gestión de contabilidad. Para ello, se describirá en detalle cómo funciona SNMP, el protocolo de gestión estándar para la administración de dispositivos de redes.

A continuación, se hará una comparativa de con varias plataformas de gestión existentes en el mercado para escoger la que mejor se adapta a los requisitos por cubrir. También se describirá detalladamente el entorno de red del laboratorio de telemática, objeto de la gestión a realizar.

Igualmente, el trabajo defenderá de modo bien fundamentado la situación del gestor y sus posibles implementaciones: sistema operativo base o virtualizado, implementación sobre nube, utilización de contenedores, etc., así como los requerimientos hardware del equipo gestor.

Por último, una vez instalado el software, se diseñará un panel de control o dashboard sobre el cual quedarán reflejados los estados de los distintos dispositivos de red y los principales parámetros de gestión elegidos en la fase de diseño. Y se mostrarán algunas capturas con los datos que la plataforma es capaz de obtener y procesar, por ejemplo, en forma gráfica. Además, se la dotará de nuevas funcionalidades con la incorporación de plugins.

1.3. Estructura de la memoria

A continuación, se expone un resumen de los diferentes capítulos que contiene la memoria de este proyecto, siendo brevemente explicados:

- Capítulo 1: Introducción. En este capítulo se hace una breve introducción y se describe el contexto, la motivación y objetivos que han hecho favorable la elaboración de este trabajo.

- Capítulo 2: La gestión de las redes. Sirve como referencia para el marco teórico en el que se engloba el software de gestión. Por un lado, se hace una introducción histórica del protocolo SNMP, base del trabajo, definiendo para qué se emplea y analizando sus características. Por otro, se pone en valor la utilización de las plataformas de gestión, relatando sus funcionalidades y cómo llevan a cabo la monitorización de equipos, con una posterior presentación de varias de las más destacables en el mercado.

- Capítulo 3: Implementación en el laboratorio. Este capítulo contextualiza las bases del trabajo en el laboratorio. Contendrá la descripción del entorno de trabajo y una argumentación de cómo se va a desplegar la topología de gestión sobre la red. Se definirán los principales agentes implicados, tanto gestor como nodos gestionados, objeto de la monitorización de LibreNMS.

- Capítulo 4: Resultados de la monitorización. Partiendo del capítulo anterior, se mostrarán los resultados y pruebas más relevantes a los que ha sido sometida la herramienta. Además, se incluirá la descripción de dos plugins que acompañan a la plataforma, dotándola de nuevas funcionalidades.

- Capítulo 5: Conclusiones y líneas futuras. Finalmente se muestran las conclusiones obtenidas tras la realización del trabajo fin de grado. Asimismo, se incluirán una serie de mejoras de la parte implementada por las que el proyecto podría avanzar.

En las últimas páginas, se incluirán una serie de anexos con objeto de completar ciertas partes del texto con el código de instalación y configuración del software.

2. La gestión de las redes

2.1. Introducción

"***Gestión*** *es la tarea que cubre todas las precauciones y actividades que aseguren el uso eficiente y efectivo de procesos y recursos distribuidos, los cuales pueden constituir una red de comunicaciones o un sistema distribuido*". [1]

La gestión de red trata sobre la planificación, la organización, la supervisión y el control de elementos de comunicaciones para garantizar un adecuado nivel de servicio, y de acuerdo con un determinado coste. Los objetivos principales de la gestión de red consisten en mejorar la disponibilidad y el rendimiento de los elementos del sistema, así como incrementar su efectividad [5].

Desde el momento en que las redes se consideran cada vez más una parte importante y estratégica de las empresas, industrias u otros tipos de instituciones y como resultado de las cada vez mayores dimensiones que están adoptando, resulta pues más importante su control y gestión con el fin de obtener la mejor calidad de servicio posible.

Tradicionalmente, en la gestión de las redes se ha partido de soluciones propietarias y cerradas con un ámbito de actuación limitado a la propia empresa o dominio de la organización. Con el tiempo, la evolución tecnológica ha permitido la entrada de múltiples fabricantes de equipos, de la misma forma que otros fabricantes de reputado nombre han desaparecido y, en consecuencia, también el apoyo que prestaban a sus soluciones de red. Por tanto, bien sea porque ha ocurrido la absorción de empresas o bien por diversificación de las fuentes de los equipos, las redes actuales son cada vez más heterogéneas en equipos.

Uno de los problemas más graves que tienen estas redes es que los equipos que las constituyen son de fabricantes distintos, con lo cual la única forma de gestionarlas es a partir de sistemas de gestión que utilicen estándares abiertos con el fin de compatibilizar protocolos e información. De esta forma, durante la década de los noventa, se fueron desarrollando diversas iniciativas con el objetivo de ofrecer recomendaciones y estándares abiertos para tratar de dar solución a estas nuevas problemáticas, como, por ejemplo, mediante el protocolo de gestión SNMP o el CMIP (*Protocolo de administración de información común*).

Para proporcionar una calidad de servicio adecuada mediante la gestión de redes, se parte de unos recursos humanos que mediante una serie de herramientas aplican unas determinadas metodologías a la red. Este texto versa sobre una de las herramientas disponible actualmente y los métodos que se pueden emplear en la gestión de red. Las recomendaciones sobre esta temática provienen de diversos grupos de estandarización. La más importante, la ITU-T, ha definido la *red de gestión de las telecomunicaciones*

(TMN). Estas recomendaciones definen cinco áreas funcionales para la gestión de red; las de fallos, configuración, tarificación, prestaciones y seguridad.

La organización de la gestión puede estructurarse también según un criterio temporal. De esta forma, se puede hablar de un control operacional que opera a muy corto plazo y a bajo nivel, una administración que opera a corto plazo y a bajo-medio nivel, un análisis de la gestión que opera a medio plazo y a medio-alto nivel y, finalmente, una planificación a largo plazo y a más alto nivel.

En el *control operacional*, las operaciones realizadas a este nivel deben quedar registradas, para su posterior análisis por el administrador de red. Es el caso de operaciones tales como la recogida de datos sobre prestaciones y utilización de la red, la evaluación de alarmas, la diagnosis de problemas, el arranque y la parada de los componentes de la red, la ejecución programada de pruebas preventivas, la modificación de configuraciones o la carga de nuevas versiones de software.

Las funciones principales de la *administración* consisten en seguir las tareas de control operacional y en elaborar informes periódicos para su posterior análisis. Por ello se ocupa de tareas como la evaluación de la calidad de servicio, la evaluación de tráfico, el mantenimiento de registro histórico de problemas, el mantenimiento de inventario, el mantenimiento de configuraciones, la contabilidad de red y de control de acceso. El objetivo del análisis es garantizar la calidad de servicio y, finalmente, la *planificación* se encarga de las decisiones dependientes del negocio al que se dedica la empresa u organización.

Llegados a este punto, sería conveniente distinguir entre monitorización, control y gestión de una red. Se utiliza el término *monitorización* para designar el tipo de acciones consistentes en obtener información de la red con el fin de detectar anomalías. Estas acciones son pasivas y su único objetivo es conocer el comportamiento respecto al tráfico del sistema. Una vez se conoce el sistema se puede proceder al *control*. Para ello, se establece una señalización o plano de control. Un ejemplo de red de control es el sistema de señalización nº 7. Finalmente, la *gestión* se define a partir del plano de gestión que integran las redes más avanzadas como RDSI, GSM, etc. Como ejemplo de red de gestión se puede citar la TMN definida por el ITU-T.

Según las áreas funcionales de gestión definidas por la ITU-T, la monitorización de red se utiliza para proporcionar información en la gestión de las funciones de prestaciones, fallos, contabilidad y en determinados aspectos de configuración, mientras que el control de red se aplica a las funciones de configuración y seguridad.

En el proceso de monitorización de la red se consideran una serie de aspectos como son: en primer lugar, una definición de la información de gestión que se monitoriza, una forma de acceso a la información de monitorización, un diseño de los mecanismos de monitorización y, finalmente, un procesado de la información de monitorización obtenida. Por otra parte, la información de monitorización puede

clasificarse según su naturaleza temporal en: información estática que se almacena en los elementos monitorizados (por ejemplo, inventario); información dinámica que se almacena en los propios elementos o en equipos especializados (por ejemplo, cambios de estado o fallos) e información estadística que se genera a partir de la información dinámica y que puede residir en cualquier lugar que tenga acceso a la información dinámica (por ejemplo, rendimientos). Los mecanismos de monitorización se basan fundamentalmente en un sondeo o *polling* por parte de la estación gestora, esto es, en un acceso periódico a la información de gestión almacenada en los nodos gestionados. Este método tiene la ventaja de que los objetos que se gestionan únicamente deben estar preparados para responder, con lo que es más simple. Otros mecanismos que se emplean, desde el punto de vista del agente gestionado, se denominan *event reporting* o notificaciones, donde son los propios recursos quienes envían mensajes bajo ciertas condiciones. De esta forma, tienen como ventaja el hecho que se minimiza el tráfico de gestión por la red. Otros métodos son mixtos, se basan en proxies, sondas, etc., y combinan los dos mecanismos anteriores.

Una arquitectura de gestión describe un marco general de un modelo de gestión integrado en un entorno heterogéneo. Así mismo, se pueden clasificar cuatro submodelos dentro de la arquitectura [3]:

- Modelo de información: En él se describe la estructuración y definición de la MIB.
- Modelo organizativo: se basa en el manejo y soporte de los aspectos organizativos (roles y responsabilidades).
- Modelo de comunicación: describe los protocolos de comunicación a emplear y el acceso a la MIB.
- Modelo funcional: se centra en la estructuración de las tareas de gestión.

A continuación, dada la importancia de las características que engloba este último submodelo mencionado, se procede a una descripción de este de manera más detallada.

Modelo funcional

Este modelo divide las tareas de gestión en un conjunto de áreas funcionales, describiendo para cada área su tarea principal, los servicios necesarios para proporcionar dicha funcionalidad, los objetos o recursos de interés gestionados en el área y las posibles subfunciones. El marco de gestión OSI ISO 7498-4, ITU-T X.701 [8] define cinco áreas funcionales, en las que se divide la gestión de red como se aprecia en la *Figura 1*:

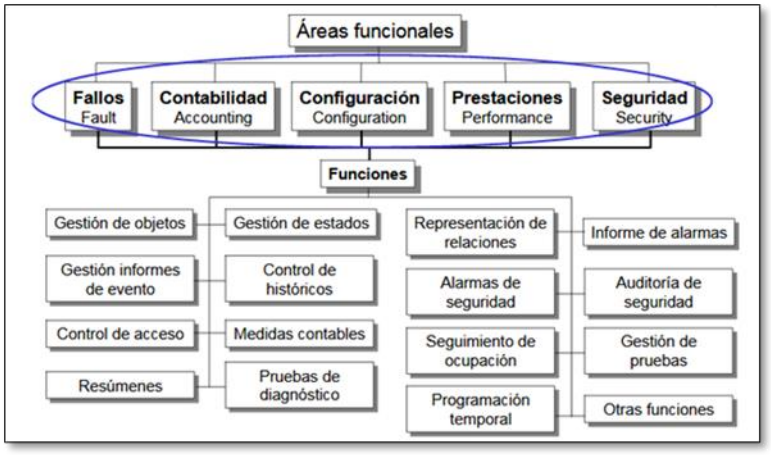


Figura 1. Jerarquía de las áreas funcionales de la arquitectura de gestión.

- **Fallos**: Mantiene y examina los registros de errores. Acepta y actúa frente a las notificaciones de errores. Localiza e identifica las averías, además de ser capaz de corregirlas. Lleva a cabo secuencias de prueba de diagnóstico.
- **Contabilidad**: Informa a los usuarios de los costes en los que han incurrido y de los recursos consumidos. Establece los límites de costes y programas de tarifas asociados al uso de los recursos. Combina costos de varios recursos cuando un servicio se presta en base a múltiples elementos.
- **Configuración**: Establece parámetros de operación. Asocia nombres a objetos. Activa y desactiva objetos. Recoge información sobre el estado actual. Recoge avisos de cambios significativos. Cambia la configuración del sistema.
- **Prestaciones**: Reúne información estadística. Mantiene y explota registros históricos del estado del sistema. Determina las prestaciones del sistema en condiciones naturales y artificiales. Modifica la operación del sistema para una correcta gestión de prestaciones.
- **Seguridad**: permite crear, borrar y controlar los servicios y mecanismos de seguridad. Distribuye la información de seguridad. Informa de los sucesos relativos a la seguridad del sistema. Protege la red y la información que transporta, ante accesos y usos no autorizados.

2.2. Protocolo SNMP

La gestión de redes en Internet inicialmente no se encuentra normalizada, sino que es una solución de facto. Se basa en un modelo estructurado cuyos niveles guardan cierta correspondencia con el modelo OSI de ISO [12].

Uno de los objetivos importantes de los estándares de gestión de red es desarrollar e integrar un conjunto de procedimientos y estándares que apliquen igualmente en diferentes dispositivos de red. La elaboración de estándares en ITU-T y la ISO se ha caracterizado por una gran lentitud, debido a la necesidad de llegar a un consenso entre muchos participantes y a procedimientos excesivamente complejos y

burocratizados. Debido a estos inconvenientes y gracias a la simplicidad y a una serie de aspectos importantes del protocolo SNMP, han hecho que se convierta en el estándar para la administración de dispositivos de redes.

El marco de gestión de red estándar de Internet se definió originalmente por tres documentos, tal y como se puede ver en detalle de la *Figura 2* que ilustra la estructura el modelo de gestión de redes utilizado en Internet:

- Structure of Management Information (SMI). RFC 1155.
- Management Information Base (MIB). RFC 1213
- Simple Network Management Protocol (SNMP). RFC 1157.

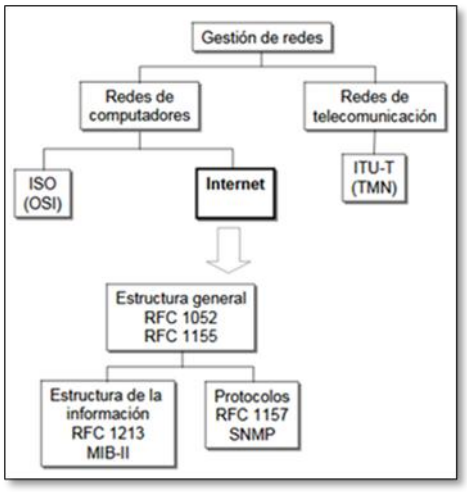


Figura 2. Modelo estructurado de gestión de redes.

El Protocolo de gestión de red simple o SNMP, fue desarrollado en 1988 aprobado por el IAB (*Internet Architecture Board)* y está basado en SGMP (*Simple Gateway Monitoring Protocol*, RFC 1028). Es un protocolo de la familia TCP/IP, situado en la capa de aplicación, que se emplea para acceder a la información de gestión de los equipos que forman la red. Permite a los administradores de red poder configurar los equipos, supervisar la operación de red, analizar las prestaciones de los equipos y, además, encontrar y resolver fallos.

Existen tres versiones del protocolo SNMP y cada una recopila los estándares del IETF (*Internet Engineering Task Force)* publicados en los documentos RFC (*Request For Comment*):

- **SNMPv1**: Año 1988. RFC 1065, 1066, 1067. Define la arquitectura física (gestor-agente), el modelo de información de gestión (SMIv1) y las operaciones básicas del protocolo. Es muy inseguro. Se transmiten las contraseñas en texto claro.
- **SNMPv2**: Año 1993. RFC 1441, 1452. Amplía el modelo de información (SMIv2), añade algunas operaciones (Trap), reduce la carga de tráfico adicional

para la monitorización (con los GetBulk e Informs) y soluciona los problemas de monitorización remota o distribuida (con las RMON, Remote MONitor). Mejora la seguridad, pero resulta muy complicado y apenas se utiliza. En la práctica, se sigue empleando una versión simplificada (SNMP v2c, RFC 1901, 1908), que mantiene el envío de contraseñas en abierto. Las versiones de SNMP son compatibles, en el sentido que SNMPv2 puede leer SNMPv1.

- **SNMPv3**: Año 2002. Mejora los aspectos de seguridad. Incluye autenticación previa a efectuar operaciones de lectura o escritura, confidencialidad e integridad. Rehace toda la notación, aunque mantiene funcionalidad. Muchas cuestiones aun por desarrollar. La descripción más actualizada de SNMPv3 se encuentra en las RFC 3410 a 3415.

La arquitectura SNMP consta de los siguientes componentes (véase *Figura 3*):

- Estación de gestión o gestor (NMS, Network Management Station).
- Agente.
- Base de información de gestión (MIB).
- Protocolo de gestión de red.

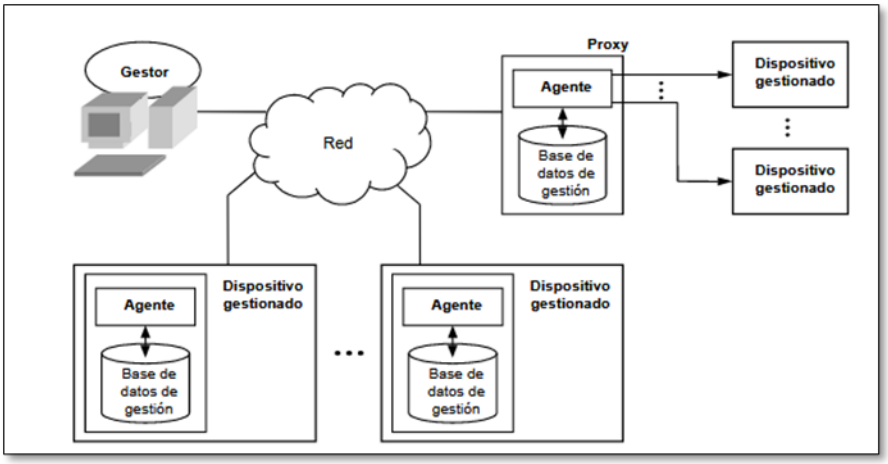


Figura 3. Esquema de una red gestionada con SNMP.

En la *Figura 4* se encuentran representados todos los elementos que intervienen en la arquitectura de SNMP, en ella se menciona al administrador de red que representa a la estación de gestión, un router y dos host, que representan los procesos de administración y de agentes sobre estos dispositivos administrados, y finalmente las líneas, que representan los protocolos de comunicación a través de los cuales se hace efectivo el llenado de la MIB central que es la base de datos que recopila toda la información de administración.

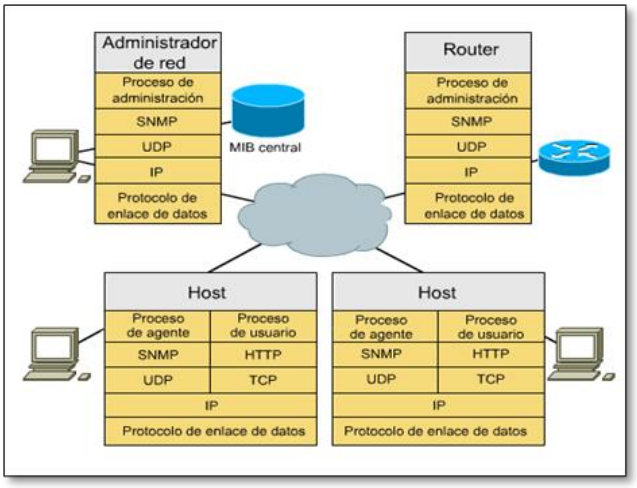


Figura 4. Componentes de la arquitectura SNMP.

Cabe destacar que esta estructura de componentes es común para todas las versiones del protocolo SNMP, tal como SNMPv1, SNMPv2 y SNMPv3.

El gestor o estación de gestión es normalmente un dispositivo autónomo, pero puede ser implementado en un sistema compartido. En cualquier caso, la estación de gestión sirve de interfaz entre el gestor de red humano y el sistema de gestión de red. La estación de gestión tendrá, como mínimo:

- Un conjunto de aplicaciones de gestión para el análisis de los datos, recuperación de fallos, etc.
- Una interfaz a través de la cual el gestor de red puede monitorizar y controlar la red.
- La capacidad de trasladar los requisitos del gestor de red a la monitorización y control real de los elementos en la red.
- Una base de datos de información de gestión de red extraída de las bases de datos de todas las entidades gestionadas en la red.

Un dispositivo gestionado es cualquier nodo de la red que resulte de interés desde el punto de vista de gestión. Para que forme parte del entorno de gestión, el dispositivo debe incorporar un agente SNMP.

Por otro lado, un agente es un módulo software que se encarga de mantener la información de gestión y de interaccionar con los gestores. Se ubica en el mismo dispositivo a gestionar y tiene acceso a sus parámetros internos (la información a gestionar).

Con frecuencia el agente no está ligado físicamente al dispositivo gestionado (por ejemplo: dispositivos no SNMP). Un *proxy* es un agente que reside en un equipo externo al dispositivo y que se encarga de traducir entre SNMP y el mecanismo de gestión del dispositivo (por ejemplo: emulación de terminal). (Véase *Figura 5)*.

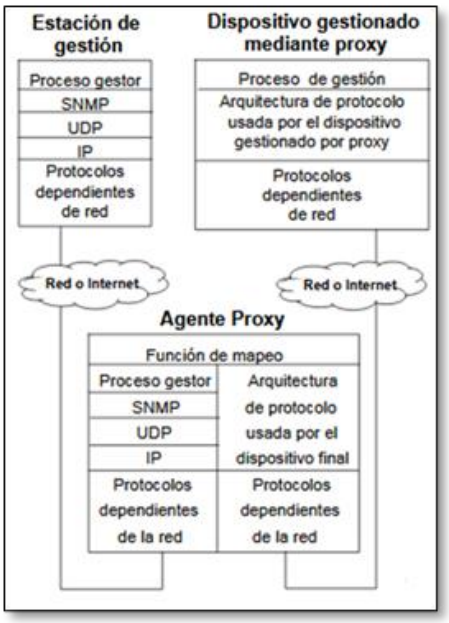


Figura 5. Arquitectura de gestión de red a través de un agente proxy.

Modelo de Información SNMP

El medio por el cual se pueden gestionar (monitorizar y controlar) los recursos de una red son representando estos recursos como objetos. Cada objeto es, esencialmente, una variable de datos que representa un aspecto del agente de gestión. La colección de objetos se conoce como base de información de gestión (MIB). La MIB funciona como una colección de puntos de acceso al agente por parte de la estación de gestión. Estos objetos están normalizados a través de los sistemas de una clase particular (por ejemplo, todos los bridges contienen los mismos objetos de gestión). La estación de gestión lleva a cabo la función de monitorización mediante el acceso a los valores de los objetos MIB. Una estación de gestión puede causar que una acción tenga efecto en un agente o puede cambiar la configuración de un agente mediante la modificación de los valores de variables específicas.

Las MIBs se dividen en tres tipos; públicas, experimentales y privadas. Las públicas están definidas mediante estándares y proporcionan información general del sistema. Por otro lado, las experimentales son las MIB consideradas en fase de desarrollo por los grupos de trabajo de Internet. Por último, las privadas corresponden a las MIBs de productos específicos, generadas por los distintos fabricantes, y ofrecen información más detallada y concreta, añadiendo de este modo funcionalidad a las MIB estándar. Generalmente, los fabricantes las hacen públicas, poniéndolas accesibles por Internet.

Una de las características más importantes de SNMP es el hecho de que la MIB ha sido diseñada de manera que puede ir creciendo y así proporciona flexibilidad para incorporar nuevos objetos. En la *Figura 6*, donde se muestra la estructura de la MIB,

hay una rama *private* a la que se pueden añadir extensiones. Esto permite a los fabricantes incorporar objetos asociados en la gestión específica de sus productos. Gracias a la estandarización del SMI, aunque se creen objetos privados éstos se pueden gestionar desde cualquier plataforma de gestión, es decir, debe existir una interoperabilidad que llegue hasta las extensiones privadas de la MIB.

Por defecto, las estaciones de gestión sólo conocen la MIB estándar. Por lo tanto, para poder gestionar objetos MIB privados es necesario previamente se cargue la estructura de la MIB privada en la plataforma de gestión.

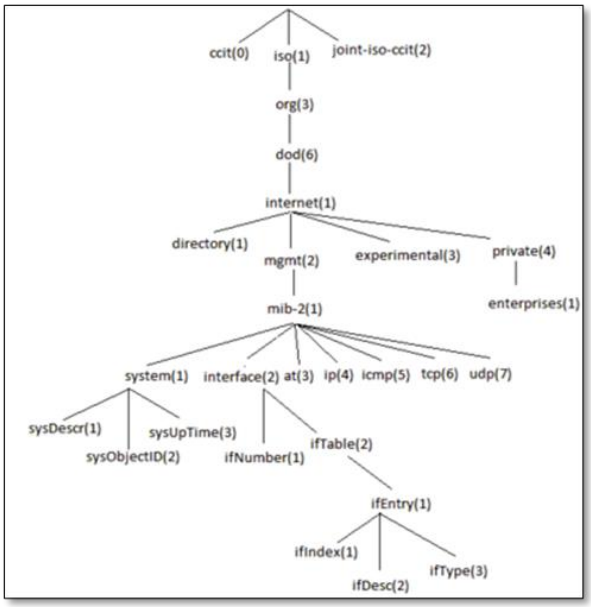


Figura 6. Árbol de registro de la MIB.

Cada objeto manejado en un MIB tiene un identificador de objeto único (OID), formado por una secuencia de números y puntos, representando cada uno un salto de nivel dentro del árbol que compone la MIB, como se muestra en la Figura 6. El OID incluye el tipo de objeto, el nivel de acceso, restricciones de tamaño, y la información del rango del objeto. Los objetos de una MIB se definen usando un subconjunto del ASN.1 (Abstract Syntax Notation One), la versión 2 de la estructura de la información de gestión (SMIv2) definido en el RFC 2578.

El nombre de los objetos se define como un OBJECT IDENTIFIER que se usa para nombrar a los objetos gestionados. Este identificador puede estar en los tres tipos de MIBs definidas anteriormente:

```
MIB estándar de Internet
        mib OBJECT IDENTIFIER
                ::= { internet mgmt(2) 1 }
MIB experimental
        experimental OBJECT IDENTIFIER
                ::= { internet 3 }
```

MIB privadas
entrerprises OBJECT IDENTIFIER
::= { internet private(4) 1 }

Cada objeto se describe utilizando la macro OBJECT-TYPE. Cada objeto se descompone en una serie de campos. A continuación, se describen brevemente los campos obligatorios:

- Descriptor: nombre del objeto.
- Value: nombre del objeto en forma de OBJECT IDENTIFIER.
- Syntax: especifica el tipo de dato del que se trata.
- Access: indica el nivel máximo de acceso (lectura, escritura, ...).
- Status: muestra la vigencia de la definición del objeto.
- Description: se trata de un texto que describe el significado del objeto.

```
OBJECT-TYPE MACRO ::=
BEGIN
TYPE NOTATION ::= "SYNTAX" type (TYPE ObjectSyntax )
                  "ACCESS" Access
                   "STATUS" Status
VALUE NOTATION ::= value (VALUE ObjectName)
Access ::= "read-only"
         | "read-write"
         | "write-only"
         | "non-accessible"
Status ::= "mandatory"
         | "optional"
         | "obsolete"
END
```

A partir de ésta se crean los tipos específicos, lo que denominamos "instancias macro":

```
OBJECT:
       object descriptor object identifier
SYNTAX:
       sintaxis ASN.1 de los objetos
DEFINITION:
       descripción del objeto
ACCESS:
       sólo lectura, lectura-escritura o no accesible
STATUS:
       obligatorio, opcional u obsoleto
```

La MIB-II es la base de datos común para la gestión de equipos en Internet. Se apoya en el modelo de información estructurada definido en el RFC 1155, que establece las bases para definir la MIB, indica los tipos de objetos que se pueden usar y define el uso de ASN.1. Esta cuelga del nodo 1.3.6.1.2.1 del árbol de registro.

Dentro de la MIB II existen dos tipos de nodos: estructurales y de información.

- **Nodos estructurales**: Son aquellos que sólo tienen descrita su posición en el árbol. Son "ramas". Por ejemplo:

```
ip OBJECT IDENTIFIER : := { 1 3 6 1 2 1 4 }
```

Figura 7. Ejemplo de nodo estructural.

- **Nodos con información**: Son nodos "hoja". De ellos no cuelga ningún otro nodo. Estos nodos están basados en la macro OBJECT TYPE, por ejemplo:

```
ipInReceives OBJECT TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION " Texto indicando para que vale "
: :={ ip 3 }
```

Figura 8. Ejemplo de nodo de información.

Este fragmento ASN.1 (*Figura* 8) nos indica que el objeto "ipInReceives" es un contador de sólo lectura que es obligatorio incorporar si se quiere ser compatible con la MIB-II (aunque luego no se utilice) y que cuelga del nodo ip con valor tres. Previamente se ha mostrado el nodo estructural "ip" con su valor absoluto. En este se puede ver que el identificador del objeto "ipInReceives" es "1.3.6.1.2.1.4.3".

Tipos de operaciones/mensajes

La comunicación gestor-agente se realiza mediante un protocolo de gestión de red al nivel de aplicación. En las redes TCP/IP, como ya se ha mencionado anteriormente, este protocolo es SNMP, y permite el intercambio de información entre gestores y agentes mediante un conjunto de operaciones sencillas de gestión. Estas operaciones se llevan a cabo gracias al intercambio de los correspondientes mensajes SNMP. Una entidad SNMP admite que se realicen en ella tres tipos de operaciones:

- El primero es el de *lectura*, en el cual un gestor recupera instancias de objetos gestionados de un agente por medio de los comandos: Get, GetNext, GetBulk.
- El siguiente tipo es el de *escritura*. A través del comando Set un gestor puede modificar o crear nuevas instancias de objetos de un agente. Además, permite al gestor actuar sobre el dispositivo gestionado.
- El último tipo de mensaje es el de *notificación*, mediante el cual un agente notifica a un gestor de la ocurrencia de una situación anómala por medio de los comandos: Trap, SNMPv2-Trap, Inform. El gestor es informado inmediatamente, evitando los retardos debidos al sondeo.

En la siguiente figura (*Figura 9*) se puede observar el esquema de comunicación entre un sistema agente y un sistema gestor.

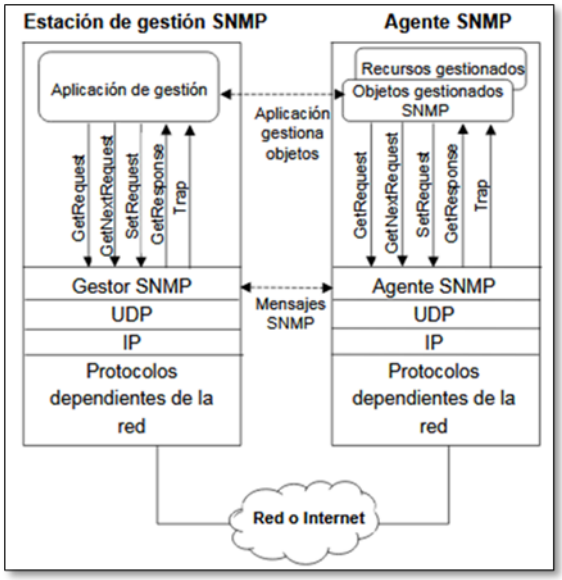


Figura 9. Esquema de mensajes SNMP intercambiados entre el gestor y el agente.

SNMP define ocho mensajes que pueden enviarse:

- Get Request: Solicita uno o más (lista) atributos (valores) de un objeto (o variable). Transmitida por el nodo administrador y recibida por el agente que contesta.
- Get Next Request: Solicita el siguiente atributo de un objeto. Transmitida por el nodo administrador y recibida por el agente que contesta.
- Get Bulk Request (en SNMP v2): Solicita un conjunto amplio de atributos en vez de solicitar uno a uno. Transmitida por el nodo administrador y recibida por el agente que contesta.
- Set Request: Actualiza uno o varios atributos de un objeto. Transmitida por el nodo administrador y recibida por el agente.
- Set Next Request: Actualiza el siguiente atributo de un objeto. Transmitida por el nodo administrador y recibida por el agente.
- Get Response: Devuelve los atributos solicitados. Transmitida por el agente y recibida por el nodo administrador.
- Trap: Informa fallos como la perdida de comunicación con un vecino. Transmitida por el agente y recibida por el nodo administrador.
- Inform Request (en SNMP v2): Describe la base local de información de gestión MIB para intercambiar información de nodos de administración entre sí. Transmitida por el nodo administrador y recibida por otro nodo administrador.

2.3. Plataformas de gestión

Antes, la gestión de red se realizaba mediante un conjunto de programas aislados, cada uno encargado de gestionar un conjunto específico de componentes (dispositivos o datos de gestión) de la red. Restricciones de coste, espacio físico y disponibilidad de técnicos plantean la necesidad de una gestión integrada desde un solo sistema, para poder adaptarse al entorno de los elementos de red que se quieran gestionar, que debería presentar sus interconexiones en un mapa de la red.

Un sistema de gestión de red está formado por dos componentes principales: la plataforma y las aplicaciones. Para la elección de este se debería realizar un inventario de dispositivos, de este modo conocer de un vistazo la foto global de la instalación y saber cuándo se necesita más hardware y cuándo está sobredimensionada. Además, se debería tener en cuenta la priorización de las áreas funcionales y analizar las aplicaciones de gestión de red necesarias. Por último, se debe escoger una plataforma de gestión de red.

Una plataforma de gestión de red es una aplicación software que proporciona la funcionalidad básica de gestión de red para los diferentes componentes de una red. El objetivo de la plataforma es proporcionar una funcionalidad genérica para gestionar dispositivos de red diversos. Las funcionalidades básicas que debe incluir son la interfaz gráfica de usuario (GUI) con un menú del sistema configurable. Un mapa de la red y un sistema gestor de bases de datos (DBMS). Además de un método estándar de consulta de dispositivos (protocolo) y, es imprescindible disponer de un registro de eventos (eventlog). Otras características adicionales que puede tener son la herramienta de gráficos, una interfaz de programación de aplicaciones (API) y un control de seguridad del sistema.

Las plataformas de gestión utilizan una integración de aplicaciones para poder adaptarse al entorno cambiante y complejo de los elementos de red que se quieran gestionar. Entre las aplicaciones más usuales que se incorporan, destacan los *MIB browser* (navegadores u ojeadores de MIB) como interfaces de usuario del protocolo SNMP; el *discover*, que permite autodescubrir equipos y topologías de la red; la programación de sondeos de variables de la MIB; la programación de acciones ante alarmas; y, finalmente, los visualizadores gráficos de valores de variables de la MIB.

Por otro lado, posibilitan mayor grado de integración multifabricante que el esquema gestor de gestores. Las interacciones con otros sistemas de gestión de diferentes fabricantes se realizan a través de un interfaz de programación de aplicaciones estándares y un conjunto estándar de definiciones de datos de gestión.

Una visión genérica de cómo se estructura una plataforma de gestión se puede observar en la siguiente ilustración (*Figura 10)*:

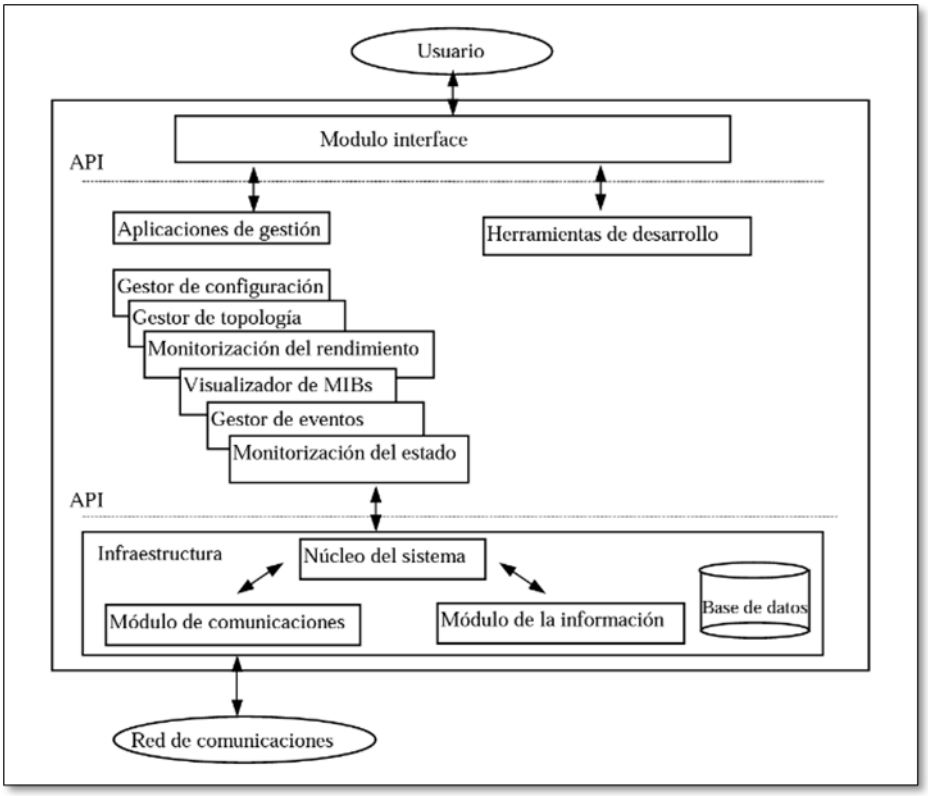


Figura 10. Estructura de las plataformas de gestión.

El proceso de evaluación de una herramienta de gestión de red es complejo y equivocarse puede llevar a perder tiempo, dinero y complicar la infraestructura. Mientras que, si se elige la plataforma adecuada, teniendo en cuenta una serie de factores dependiendo de las necesidades de cada negocio, la reducción de costes será notable. Es importante valorar que las herramientas que se están eligiendo sean versátiles, a corto y largo plazo, para que su único objetivo no sea únicamente la monitorización, sino que esto se trata de un punto de apoyo en un tipo de rendimiento superior. Existen herramientas de monitorización que incluyen las redes como uno de los elementos que se gestionan, y además realizan el proceso de administración de las distintas aplicaciones que se usen dentro del entorno en que se utilizan. Esto no hará más que potenciar la monitorización añadiendo aplicaciones y servidores que ya están instalados en la red.

Previo a la elección de la herramienta de gestión de red específica, es conveniente comprobar cuál es su método de instalación y configuración. Hay herramientas que son demasiado planas y básicas en las que simplemente se roza el rendimiento avanzado y se conforman con métodos mucho más simples en cuanto al uso. Mientras que otras ofrecen complejas configuraciones y muy poco estándar, que provocan que los administradores pasen a ser indispensables, teniendo prácticamente que tirar el sistema de monitorización de redes en el caso que estos abandonen.

Lo que sí se debe tener en cuenta es si la herramienta dispone de un sólido sistema con el cual se proporcionen alertas, dado que esto es indispensable para el

administrador de la red. Es muy importante que la herramienta permita utilizar múltiples canales de comunicación y que éstos puedan ser flexibles, no solo en cuanto a contenido sino también en cuanto a políticas de envío.

En este mismo aspecto el sistema de monitorización a través de control remoto también es fundamental, dado que la red o redes que se van a monitorizar puede estar ubicadas en diferentes localizaciones o con diferentes dueños, incluso que los equipos que la conforman pueden ser inaccesibles.

Asimismo, la monitorización de redes puede llegar a abarcar miles de nodos o elementos a monitorizar. Vivimos en un mundo interconectado y muchas organizaciones necesitan monitorizar elementos conectados a sus redes internas como dispositivos móviles, vehículos, cajeros, etc. Es en estos casos cuando la red a monitorizar crece y es vital que la herramienta monitorice miles y miles de dispositivos con la misma eficiencia y rendimiento. Hay que tener en cuenta la posibilidad de escalado y rendimiento que ofrece la herramienta y sobre todo qué hardware necesita, para poder evaluar los costes.

Otra opción considerable para tener en cuenta es la posibilidad de ofrecer la monitorización con o sin agentes. Debido a la naturaleza de diferentes redes o elementos de las redes, en ocasiones no se podrá instalar un agente y se requerirá de una monitorización sin instalación de agentes en el destino a monitorizar.

Una característica crucial es la creación de informes y envíos de estos. Muy probablemente se pedirán informes del estado de la monitorización de la red y estos deben ser claros, exportables y aptos para todo tipo de perfil.

Adicionalmente, otro punto en el que incidir es la posibilidad de ofrecer la monitorización con o sin agentes. Debido a la naturaleza de diferentes redes o elementos de las redes, en ocasiones no se podrá instalar un agente y se requerirá de una monitorización sin instalación de agentes en el destino a monitorizar.

Como se ha mencionado anteriormente, cada vez es mayor la necesidad de integrar elementos externos en el sistema de monitorización. Para ello, es destacable que la herramienta elegida disponga de una API que permita integrar otras aplicaciones.

Si la infraestructura de red dispone de sistemas virtualizados, es muy importante que la monitorización de redes abarque también la monitorización de las instalaciones virtualizadas. Así mismo, y con la idea de poder aumentar en un futuro la monitorización, es conveniente que se pueda monitorizar no solo las máquinas que residen en el sistema virtualizado, sino que también permita monitorizar la propia infraestructura de virtualización. Para los contenedores se deberá realizar el mismo ejercicio.

Cada vez son más las organizaciones que deciden mover parte o toda su infraestructura a la nube. Por tanto, es necesario verificar que la herramienta de

monitorización de redes elegida permitirá hacer una monitorización híbrida (la del propio CPD y la instalación de la nube) y agrupar todo en el mismo panel.

La capacidad de guardar y analizar históricos es una de las características imprescindibles en la herramienta de monitorización de red. Es necesario saber qué está pasando en tiempo real, pero también poder analizar lo que pasó en el pasado y así poder aprender y modificar la herramienta acorde a lo que ha pasado a lo largo del tiempo.

Algunas herramientas ofrecen múltiples tipos de licencias y cuando se requiere aumentar el número de elementos monitorizados o el tipo de elementos a monitorizar (aplicaciones, procesos de negocio, determinados servidores, etc.) algunas empresas aprovechan para aumentar los costes de forma desproporcionada.

Finalmente, se debe comprobar que el panel donde se presenta la monitorización de redes y sus resultados se adapta a las necesidades y a las posibles necesidades futuras a las que haya que enfrentarse cuando se escale el sistema. Y, por otro lado, que las personas que se ocupan de gestionar y administrar la información de la monitorización puedan acceder a estos datos en cualquier sitio y que además dispongan de un sólido sistema de historial y archivo se convierte en un aspecto muy positivo a la hora de aumentar el rendimiento y los buenos resultados.

En definitiva, se puede ver que la elección de una herramienta de monitorización de redes no es tan simple como se pueda imaginar, pero que una vez realizadas las comprobaciones necesarias tampoco es tan difícil llegar a aquello que se puede requerir en cada tipo de situación. Con todo lo anterior tenido en cuenta, se podrán detectar cuellos de botellas en la red y averiguar cuál de los equipos es el causante para solucionarlo. Igualmente, una buena herramienta de gestión y una correcta implementación podrá detectar tráfico intruso o mal intencionado. Será capaz de anticiparse a problemas y evitar que lleguen a más, y generar logs y analizar el rendimiento de la instalación a lo largo del tiempo, pudiendo detectar problemas y asociarlos a las modificaciones hechas en la red.

En las líneas siguientes se analizan algunas de las herramientas de gestión con más prevalencia en el mercado.

2.3.1. Nagios

Nagios es una herramienta de monitorización de redes, de código abierto, potente, flexible y escalable que permite que los equipos de TIC detecten problemas en las infraestructuras con bastante antelación, evitando así que afecten los procesos de la infraestructura de red. Las opciones, que son personalizables, permiten monitorizar una gran variedad de parámetros y enviar alertas mediante informes detallados, secuencias de comandos personalizados, correos electrónicos o SMS.

Es una de las herramientas que goza de mayor popularidad y su uso es muy común en las áreas de finanzas, sanidad y educación. Al ser una de las herramientas pioneras en la monitorización de redes, Nagios es una de las herramientas de mayor uso. Aun así, es interesante ver como la tendencia de su demanda en Internet ha ido disminuyendo con el paso del tiempo. Lo que antes fue una de las más potentes y conocidas herramientas de red está perdiendo terreno.

Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix.

Ventajas

- Se encuentran muchos perfiles con experiencia Nagios.
- Si se tiene gran conocimiento de la herramienta, la configuración manual puede darle mucha potencia a la hora de monitorizar casos aislados y particulares.
- Ofrece muchos plugins para adaptar Nagios a las necesidades del usuario.
- Para la configuración básica es muy fácil.

Desventajas

- Configuración y edición compleja, debido a la necesidad de hacer modificaciones de forma manual para dejar lista la herramienta.
- El interfaz gráfico carece de una buena usabilidad.
- Coste de aprendizaje elevado.
- Cada instalación al final resulta un "puzle" en el que más que un producto estándar tenemos una implementación propia, con cientos de parches, código propio o de terceros y complicada de evolucionar o de mantener por terceros.
- Informes sencillos.
- Muy pobre en su tratamiento de SNMP, tanto de polling como de gestión de traps.

En resumen, Nagios fue el origen de la monitorización y, de hecho, muchas nuevas herramientas de monitorización de redes han heredado el código de Nagios y lo han evolucionado. Aunque tienes muchos perfiles en el mercado, estos deben tener un conocimiento muy técnico y tu instalación dependerá de ellos al cien por cien. La futura migración podrá ser complicada.

2.3.2. Zabbix

La solución más adecuada para aquellos que necesitan una solución de monitorización de código abierto potente con una interfaz de usuario con configuración basada en la red y paneles de control incorporados.

Zabbix dispone de una interfaz web de fácil uso con autenticación de usuario segura. Además, cuenta con varias opciones de visualización entre las que se encuentran informaciones generales, gráficos, mapas, pantallas, y, adicionalmente, dispone de varios métodos flexibles para el análisis de datos al igual que para crear alertas.

De fácil configuración, su rendimiento empieza a descender cuando se empiezan a monitorizar muchos nodos. Zabbix es compatible con sondeo y captura. Destaca el servicio de monitorización sin necesidad de instalar agentes al igual que agentes de rendimiento nativos para recopilar datos de los sistemas operativos más comunes.

Ventajas

- Su comunidad es bastante activa.
- Es potente a bajo nivel.

Desventajas

- Aunque se ha utilizado en grandes instalaciones, a partir de 1.000 nodos puede disminuir su rendimiento.
- Difícil crear y definir plantillas de informes y alertas. Las configuraciones pueden requerir muchos pasos para completarlas.
- No posee informes en tiempo real.
- Es difícil de depurar cuando hay errores.
- Pobre tratamiento de traps.

La problemática que tiene es su escalado para grandes CPDs. Se debe tener mucho cuidado si la instalación tiene varios elementos del mismo tipo (por ejemplo, bases de datos) sus configuraciones van a ser complicadas.

El software es gratuito, aunque Zabbix ofrece servicio técnico a cambio de una suscripción, con el cual asegura implementaciones a prueba de sorpresas. El servicio se ofrece en cinco niveles diferentes que van desde servicio técnico por incidencia, a servicio técnico complejo que puede comprender asesoramiento y formación in situ, actualizaciones de sistema, resolución remota de problemas, con lo que queda garantizando que cada compañía u organización encontrará la asistencia que mejor se adapte a sus necesidades.

2.3.3. Observium

Herramienta de monitorización basada en PHP/MySQL/SNMP que permite descubrir automáticamente dispositivos e incluye soporte para una amplia gama de estos tanto a nivel de hardware como de software: Cisco, Linux, FreeBSD, Juniper, Brocade, Foundry, HP y muchos más.

Observium ha surgido de la falta de plataformas de gestión de red fáciles de configurar. Su objetivo es proporcionar una interfaz más navegable para la salud y el rendimiento de su red. Sus objetivos de diseño incluyen recopilar tantos datos históricos sobre dispositivos como sea posible, ser completamente auto descubiertos con poca o ninguna intervención manual y tener una interfaz muy intuitiva.

Entre las métricas que dispone se encuentra el uso de CPU y memoria, así como las estadísticas de almacenamiento. También cuantifica el tráfico de las interfaces, los paquetes que se están transmitiendo y genera estadísticos de errores detallados. Es capaz de crear gráficos del tráfico en tiempo real. Hace un listado con el inventario de dispositivos, en el que identifica sus direcciones MAC e IP. Genera estadísticas detalladas de la pila IPv4, IPv6, así como de TCP y UDP. Temperatura, velocidad del ventilador, voltaje, amperaje, potencia, humedad y sensores de frecuencia son otros datos de medidas que la herramienta ofrece.

Ventajas

- Las gráficas de Observium destacan por su gran detalle y diseño. Interesante para mostrar cuadros de mando a niveles gerenciales.
- Fácil interfaz y muy usable.
- Capaz de monitorizar grandes instalaciones.

Desventajas

- No se pueden configurar alertas en la versión libre.
- Es una buena herramienta, pero carece de funcionalidades básicas que en ocasiones son recomendables suplir con Nagios o Cacti.

2.3.4. LibreNMS

LibreNMS muestra su robustez en la respuesta rápida de su software, que debe su eficiencia a un API de código abierto. Este tipo de monitorización en tiempo real no es inusual para las más modernas herramientas de monitorización de red de Linux, pero hace que LibreNMS sea particularmente útil para proporcionar actualizaciones automáticas sobre el rendimiento de la red a un sistema de alerta multimedia expansivo. LibreNMS combina esta útil comunicación API con un sistema de red escalable horizontalmente, que permite a los usuarios ampliar rápidamente el número de nodos monitorizados por el control central sin demasiado esfuerzo. Debido a que es un software relativamente nuevo, fundado en 2013, LibreNMS tiene muchas otras capacidades útiles del siglo XXI, como la integración con aplicaciones Android e iOS, además de la compatibilidad con máquinas virtuales.

Se trata de un proyecto comunitario liberado bajo licencia GPLv3 (GNU General Public License v3.0, por lo que es software libre), es una bifurcación basada en la comunidad de la última versión de Observium. Escrito mayoritariamente en PHP, permite monitorizar los servicios y el hardware que se tengan desplegados en la infraestructura de red pudiendo detectar de forma automática una amplia gama de equipos de red y sistemas operativos.

Con esta solución se puede realizar autodescubrimiento de los equipos de red mediante el uso de diversos protocolos (CDP, FDP, LLDP, OSPF, BGP, SNMP y ARP), configurar de forma personalizada diversas alertas (dispone de un sistema de alerta altamente flexible, incluyendo la notificación por correo electrónico o IRC (indicadores clave de rendimiento), entre otros.), asimismo, dispone de generación de facturas para el cobro por ancho de banda utilizado, para puertos en la red según el uso o la transferencia, característica beneficiosa para ofrecer servicios a través de la red dependientes del uso de la misma.

Por otro lado, LibreNMS brinda la posibilidad de añadir otras funcionalidades como son la gestión de configuración, usando Oxidized, es un fork o bifurcación de RANCID. Oxidized extrae información del dispositivo a través de una API REST para hacer copias de seguridad. Dispone de un mayor soporte a los fabricantes menos populares, donde RANCID es más exclusivo. También se puede agregar Smokeping, una herramienta RRDtool, cuya finalidad principal es el tratamiento de datos temporales y datos seriales como temperaturas, transferencias en redes, cargas del procesador, etc., o NFSen, interfaz gráfica basada en web para nfdump herramientas de netflow, es una variante libre NetFlow propietaria de CISCO. Permite múltiples maneras de autentificación, ya sea base de datos con MySQL/MariaDB, HTTP, LDAP, Radius o Microsoft Active Directory.

3. Implementación en el laboratorio

3.1. Topología de la red

En este punto se desarrollará una explicación en detalle del entorno de trabajo empleado para la monitorización de los equipos que conforman el Laboratorio de Telemática de la Universidad de Cantabria.

En la *Figura 11 se* puede observar cómo está desplegada la red tanto del Laboratorio de Telemática como la del Laboratorio de Aplicaciones Telemáticas. En la práctica e implementación de la herramienta, se ha obviado el segundo de los laboratorios dado su inactividad. Asimismo, se ha descartado la monitorización de los nodos de la red X.25, 192.168.200.0/24, como del router Cisco C2500 y los que integran la red, 128.100.100.0/24, dado que están obsoletos.

El router Cisco C2600 hace de puente en la interconexión entre los laboratorios citados varias líneas arriba. Este mismo, posee un agente SNMP propietario de Cisco que remite o informa acerca de los datos de gestión del propio equipo. Una de sus interfaces, la FastEthernet0/0 o Fa0/0, cuya dirección IP es la 192.168.110.1, está conectada al Switch SMC por su puerto número 14.

Por otro lado, el switch Ethernet del fabricante SMC de 24 puertos tiene conectados los equipos locales del laboratorio, del 1 al 10, que tienen asignados una IP privada de la red 192.168.110.0/24. Así el PC denominado 'local2' posee la 192.168.110.2, hasta el 'local10' que tiene la 192.168.110.10, a excepción del 'local1' que cuenta con la IP 192.168.110.12. A parte de las direcciones destinadas a esos equipos en esa red, el switch SMC cuenta con la suya propia para su gestión, la 192.168.110.20. A través del puerto 19 se comunica con el otro switch del Laboratorio de Telemática.

Al otro lado del enlace, en su puerto 24, se encuentra este switch Ethernet de 24 puertos también, siendo su fabricante Ubiquiti Networks. Es el equipo del que "cuelgan" los otros 10 PCs locales, desde 'local11' con dirección de red 192.168.110.41, hasta 'local20' con la IP 192.168.110.50. Estos equipos se han renovado recientemente y ya cuentan con un sistema operativo Windows 10. La dirección que dispone el switch de Ubiquiti para realizar tareas de gestión es la 192.168.110.55.

Retomando la descripción del switch SMC, otro de los enlaces que tiene operativos es con el equipo Atlas. Esta unión física a través de un cable Ethernet se hace conectando el puerto 13 del switch SMC con la interfaz Ethernet1, o eth1, de Atlas.

Atlas es el servidor que da salida a Internet a los equipos de este laboratorio. Una de sus funciones es la de NAT (Network Address Translation) entre la red privada e Internet. En la *Figura 11*, se aprecia como su interfaz eth1 tiene una IP privada, 192.168.110.11, mientras que la eth0 cuenta con una pública, 193.144.186.15.

Otro equipo al que se tiene acceso por Internet es el que tiene como dominio lisitea.tlmat.unican.es. Se trata de un router Juniper J2320 que se encuentra situado en el Laboratorio 106 y su dirección de red es la 193.144.186.41.

Para el desarrollo de este trabajo en el laboratorio se ha asignado una IP dentro del rango de direccionamiento de la red 192.168.110.0/24 para el equipo nativo Windows 10, 192.168.110.54, y se ha conectado al switch SMC por su puerto número 1 mediante un cable Ethernet. Mientras que para la máquina virtual CentOS 8, se ha configurado la dirección 192.168.110.56. Esta última es la que utiliza el gestor de la herramienta LibreNMS para sus propias tareas de monitorización. Ambas tienen como gateway la IP 192.168.110.11, es decir, la correspondiente al interfaz eth1 de Atlas, como se ha mencionado anteriormente, es el equipo que da acceso a Internet.

3.2. Arquitectura de gestión

La arquitectura que se ha empleado en el laboratorio se basa en un modelo de gestión centralizado en el que sólo hay un único gestor que es el encargado de gestionar toda la de red. Una visión detallada de esta arquitectura se puede apreciar en la imagen de la *Figura 12*. Este gestor es el propio software de la plataforma LibreNMS, que se ejecuta en la estación encargada de monitorizar la red, es decir, en la máquina virtual, y su tarea consiste en consultar a los diferentes agentes SNMP que se encuentran en los nodos de la red los datos que han obtenido a través de sus MIBs.

En este sentido, los nodos gestionados van a ser consultados por el mismo LibreNMS mediante una monitorización *agentless* o sin agentes, por lo que no existe la necesidad de instalar un agente software en los hosts que se están supervisando. Por tanto, estos nodos gestionados van a ser los dispositivos del laboratorio que dispongan de agentes SNMP implementados como Atlas, el router C2600, los switches SMC y Ubiquiti, y, por último, el router Lisitea.

Entre los elementos de las MIB principales que LibreNMS consulta vía SNMP a los agentes, pueden ser la identificación de los propios nodos (sysDescr), que disponen de información acerca de su nombre (sysName), su localización (sysLocation), la persona de contacto (sysContact), el sistema operativo que utilizan o del tiempo que llevan encendidos (sysUpTime), la tabla de direccionamiento (ipRouteTable) o la tabla de interfaces (ifTable), las VLAN a las que pertenece, incluso del inventario hardware del que están compuestos.

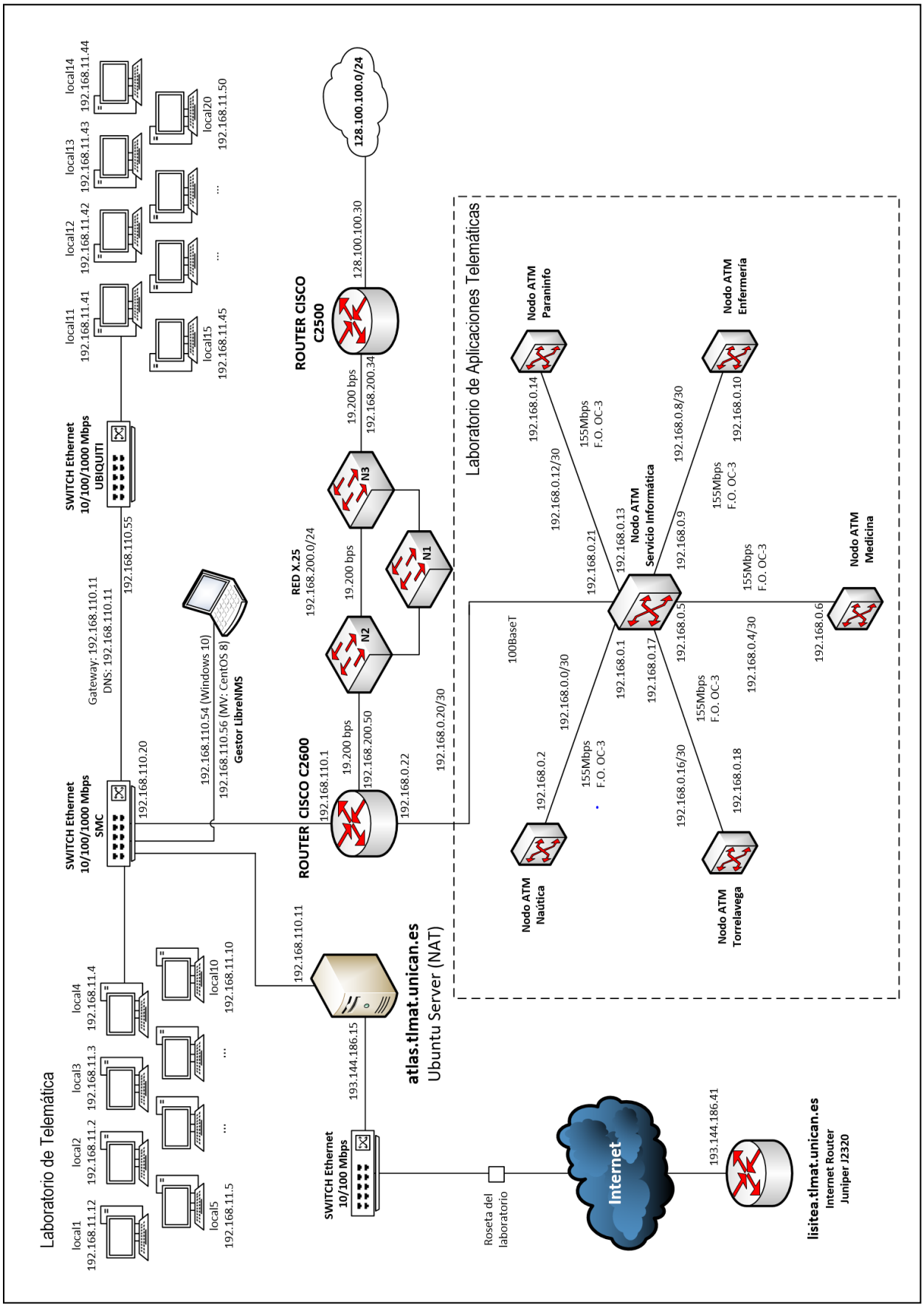


Figura 11. Esquema de red de los laboratorios de Telemática y de Aplicaciones Telemáticas de la UC.

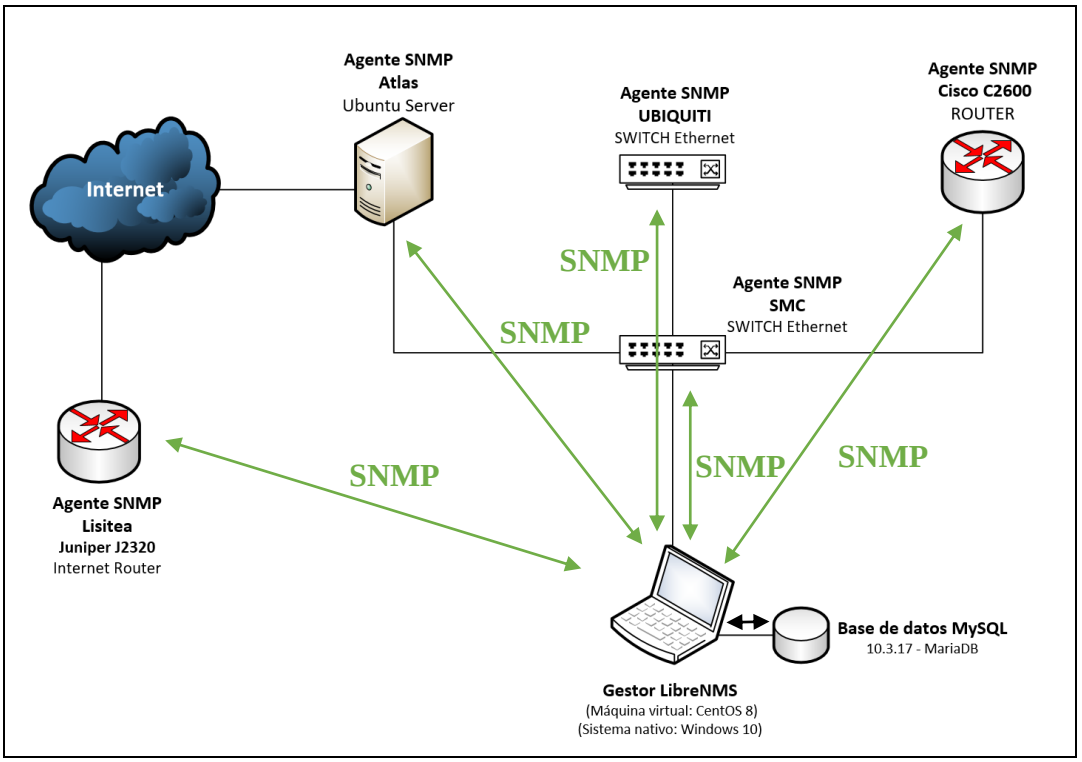


Figura 12. Arquitectura de gestión del laboratorio de Telemática.

Tal y como se ha mencionado al final del apartado anterior, la plataforma LibreNMS se implementa sobre una máquina virtual con un sistema operativo CentOS 8 sobre un portátil con un sistema operativo anfitrión Windows 10. Uno de los motivos de la elección de esta distribución de Linux frente a la disponible de Ubuntu, se debe a que CentOS se considera más estable que este última. Principalmente porque las actualizaciones de paquetes son menos frecuentes y posiblemente sea más seguro que Ubuntu. Respecto a la decisión acerca de la instalación del servidor web, se ha optado por Nginx frente a Apache, dado que es muy rápido, utiliza una arquitectura de subproceso asíncrono evitando crear nuevos procesos cuando se generan solicitudes de los clientes, procesa muy bien el tráfico que recibe, y hace un consumo más eficiente de recursos informáticos, es decir, de la RAM.

El software de virtualización que se emplea para ejecutar la máquina virtual es VirtualBox. En él se han tenido que configurar algunos parámetros como son la memoria base de la propia máquina, a la que se le han asignado 5048 MB, la memoria de almacenamiento de 12,09 GB, ya que es suficiente para todo el código que se necesita instalar, y también el adaptador de red en modo Bridge de manera que la máquina virtual pase a ser un equipo más dentro de la red local. Por este motivo, a fin de que se comunique con cualquier otro dispositivo presente en la red del laboratorio, se le ha asignado una dirección IP dentro del rango que disponen los equipos conectados al switch, para que pueda ser capaz de identificarles y añadirles a la plataforma sin inconvenientes, y estos podrán utilizar los recursos compartidos de la máquina virtual.

Una vez se ha concluido con éxito la instalación del software de gestión LibreNMS sobre la máquina virtual, configurando los archivos y añadiendo las dependencias necesarias como se indican en los pasos a seguir del Anexo 1. Instalación y configuración de la plataforma de gestión LibreNMS, se procede a incluir los equipos de red para que puedan ser monitorizados por la herramienta. Por este motivo, se tienen que indicar las comunidades que tienen definidas.

En primer lugar, se va a editar el archivo *config.php* situado en el directorio /opt/librenms. En sus líneas de código, hay un apartado *Default community* en el que está definida la comunidad SNMP que emplea la herramienta por defecto, '*public'*. Entonces, copiando el formato de esa línea únicamente se modifica lo que está dentro del paréntesis de la palabra array, tal y como se puede apreciar en la *Figura 13*:

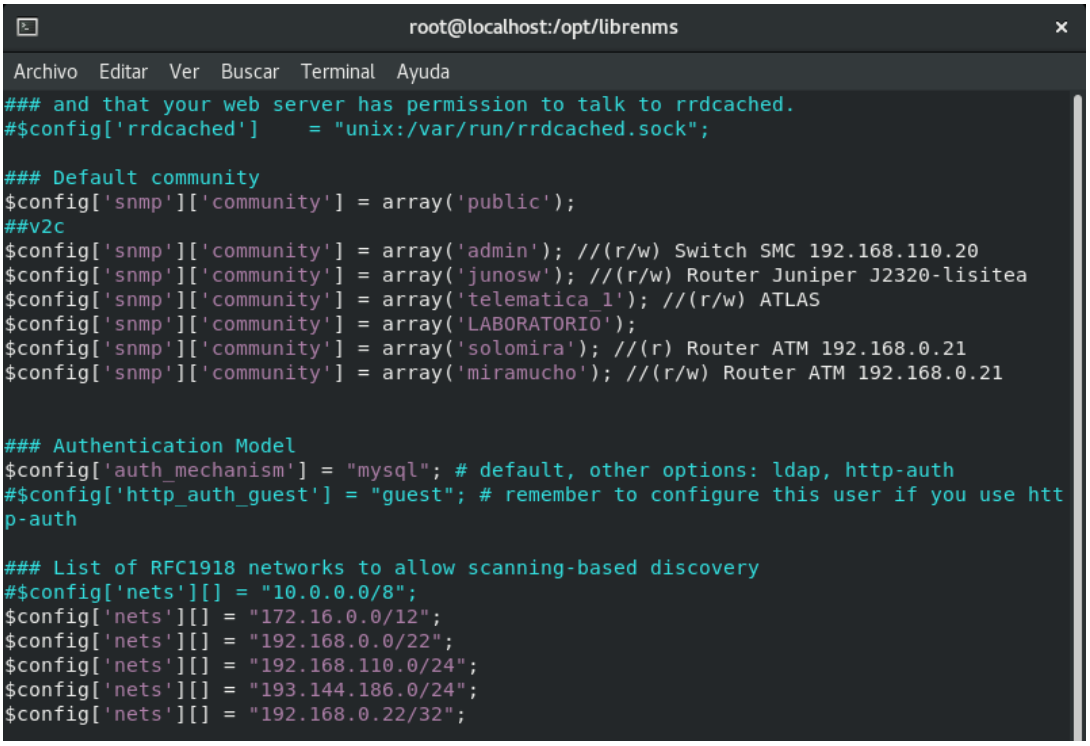


Figura 13. Captura de pantalla del fichero config.php

La mayoría de los equipos tienen definidos una comunidad de sólo lectura 'public'. Por el contrario, dado que los equipos son de topologías y fabricantes diferentes, sus comunidades de lectura y escritura también varían, limitando la modificación de sus datos de gestión al administrador por seguridad.

Lo siguiente que se debe determinar en ese mismo archivo de configuración, en el que ya se han definido las comunidades, son las redes o subredes que LibreNMS tiene que escanear automáticamente, para evitar que escanee redes fuera del escenario definido (véase *figura 13* – líneas que contienen `$config['nets'][]`). De este modo, empleando el protocolo de detección de capa de enlace LLDP o el CPD, propietario de Cisco, la herramienta solicita la lista de vecinos que los equipos de red ven en la red.

Puede suceder que el mismo dispositivo sea visto varias veces por LibreNMS, una vez usando LLDP/CDP, y otra vez a través de OSPF. En ese caso, LibreNMS termina agregándole dos veces. Por ejemplo, puede ver dos dispositivos llamados *atlas.tlmat.unican.es* y *_gateway*, y esto no es deseable. Dado que ambos dispositivos son de hecho el mismo, su SNMP sysName será idéntico, por lo que se puede indicar a LibreNMS que no añada dispositivos si ya existe uno con el mismo sysName. Si esto sucede, se podrá ver en la interfaz web en el Registro de eventos, entrada del menú 'Overview', sección 'Event Log'. Y para evitarlo, se añade la siguiente línea al fichero /opt/librenms/config.php:

```
$config['allow_duplicate_sysName'] = false;
```

El objetivo de todo lo anterior es permitir que SNMP descubra automáticamente los equipos únicamente por IP, y no intente realizar una búsqueda DNS inversa. Primero, LibreNMS descubre cada dispositivo de la red que se haya indicado. Esto significa que examina detalladamente cada host agregado previamente y determina qué información debe sondear. El script *discover.php* se encarga de esto y se ejecuta de la siguiente manera en la CLI:

```
# cd /opt/librenms
# sudo -u librenms php discovery.php -h all
```

Una vez termine de ejecutarse este script, se puede sondear a los hosts. La herramienta ya conoce la información que debe solicitar a cada host, pero debe añadir los valores iniciales para cada dispositivo a su base de datos. Para que se lleve a cabo, se escribirá lo siguiente:

```
# sudo -u librenms php poller.php -h all
```

Adicionalmente, se han tenido que realizar otras configuraciones en diferentes para permitir la integración de LibreNMS con otras herramientas como son Weathermap y Smokeping. Sus respectivas instalaciones se detallan en el Anexo 2. Instalación de Weathermap en LibreNMS y en el Anexo 3. Instalación de Smokeping en LibreNMS, que se sitúan al final de este documento.

Por un lado, el plugin Weathermap permite construir una red de mapas para visualizar las tasas del tráfico de la red. Weathermap es de código abierto y necesita php pear, que es un entorno de desarrollo y sistema de distribución para componentes de código *PHP*, para funcionar. Esta herramienta recoge los datos en los que se basan sus gráficos a través de plugins como son RRDtool, MRTG (RRD y log-format antiguo), archivos de texto delimitados por tabulaciones, SNMP, fping, scripts externos y datos específicos de Cacti. El plugin de RRDtool proporciona el acceso a datos de una amplia gama de herramientas de monitorización de código abierto.

Mientras que Smokeping está diseñado para mantener un registro histórico de los tiempos de retardo en una red, de hecho, es considerada una de las mejores herramientas existentes para la visualización de éstos y cuenta con un elevado número de plugins. Posee además un sistema de alertas sumamente configurable y una ventana gráfica en tiempo real con el retardo y medidas de pérdidas de paquetes. No se limita a medir y generar las gráficas de retardos como ICMP, sino también es capaz de generar gráficos a partir de tiempos de demoras para otros servicios (HTTP, DNS, SMTP, SSH, LDAP, etc). Otra característica importante de este sistema es que permite definir rangos estadísticos para generar alarmas enviadas vía correo electrónico.

4. Resultados de la monitorización

Este capítulo se destinará a la demostración de los resultados que se han obtenido utilizando la plataforma, a fin de dar cumplimiento a los objetivos definidos para este trabajo.

4.1. Listado de dispositivos

Si se accede desde el navegador web a la url: http://localhost/login y se introducen las credenciales de usuario creadas durante el proceso de instalación, se podrá empezar a explorar la información que se ha recopilado para los dispositivos supervisados.

Dentro de la sección 'Devices' del menú principal llamada 'All Devices', permite mostrar el listado de los equipos de red que se han incluido en la plataforma mediante el 'discovery' automático que se ha configurado previamente en el archivo *config.php*. Además, esta sección permite filtrar el listado atendiendo al tipo de dispositivo y servicio que ofrecen, 'Firewall', 'Network' o 'Server'. Se puede demostrar cómo la implementación ha resultado satisfactoria y que la información de los equipos es correcta (véase *Figura 15*).

En la misma pantalla (*Figura 15)*, en la columna del listado 'Actions' aparece un icono de la bola del mundo que, pinchando con el ratón sobre él, abre una nueva pestaña en el navegador con la interfaz web del software propietario de gestión del dispositivo. Es necesario autenticarse con un usuario y una contraseña para acceder a su contenido.

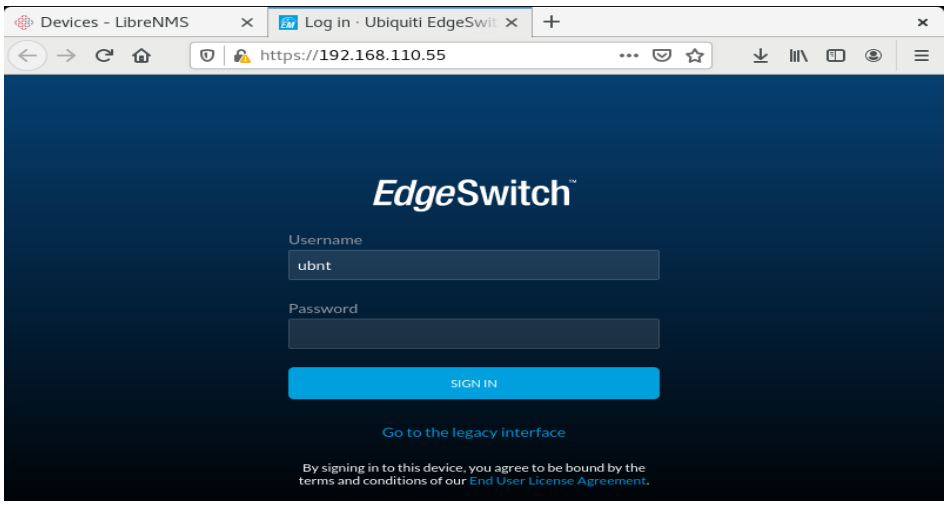


Figura 14. Interfaz web de gestión del switch de Ubiquiti.

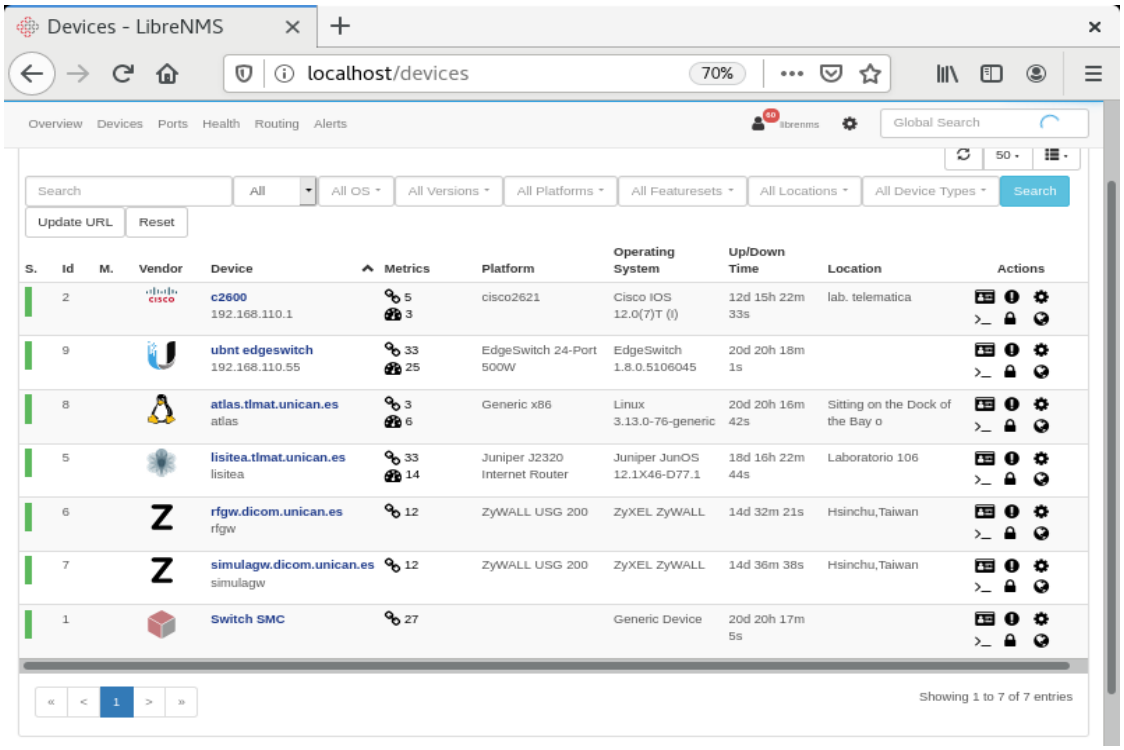


Figura 15. Listado de equipos monitorizados por LibreNMS.

La interfaz web de LibreNMS también permite añadir nuevos dispositivos en la opción del menú superior 'Devices', 'Add Device', al igual que eliminar alguno de los existentes en 'Devices', 'Delete Device'. Esta opción resulta más sencilla de realizar que empleando comandos. En la imagen de la *Figura 16* se muestra qué información hay que cubrir cuando se desea agregar un nuevo equipo:

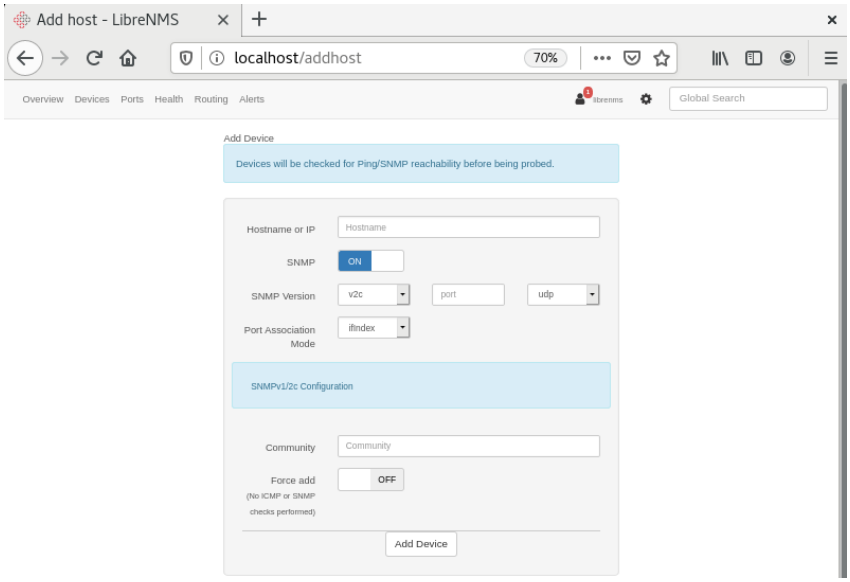


Figura 16. Sección para añadir nuevos dispositivos desde la interfaz web de LibreNMS.

4.2. Mapa de red

Puede que el administrador de red o el usuario de LibreNMS esté interesado en ver un mapa de red de los equipos a modo de obtener una visión global del entorno y conocer su estado. Pues bien, esto es posible, y además se pueden visualizar los enlaces creados entre ellos en los que se indica la interfaz o puerto que emplean. Incluso, situando el cursor encima de la línea del enlace, la herramienta facilitará un gráfico del tráfico de red en *bps* que está teniendo lugar por esa interfaz o puerto (véase *Figura 17*). Adicionalmente, posicionando el cursor encima de cada dispositivo, LibreNMS generará las gráficas correspondientes a la información sobre el tráfico, el uso de CPU y el uso de memoria en tiempo real.

Una de las veces en las que se accedió a este mapa de red en el laboratorio, el recuadro del switch SMC aparecía en color rojo. Al clicar el mismo, se mostraba la pantalla de información general sobre ese switch. Dentro del menú disponible que tienen los dispositivos, en el apartado 'Logs' se había creado una nueva entrada en la lista de eventos (*Figura 18*) registrados en la que se notificaba que el estado del switch había sido modificado a 'down' o caído por la información obtenida gracias al protocolo ICMP, es decir, de un ping realizado desde el propio gestor al switch.

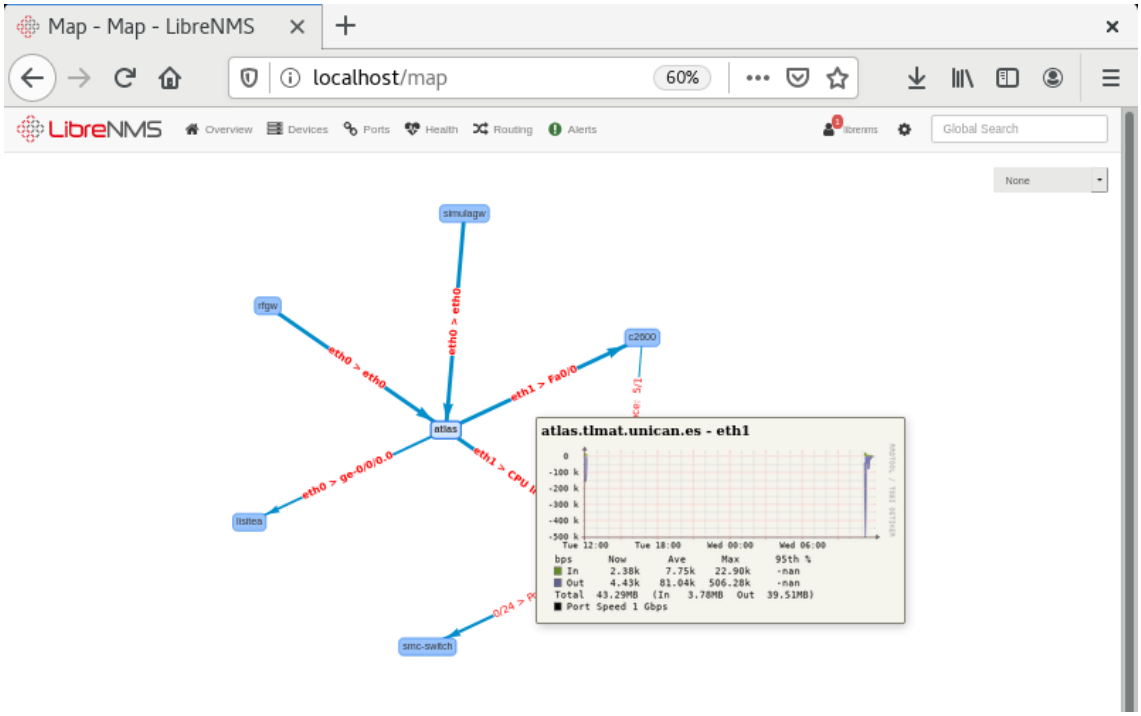


Figura 17. Mapa de red de los dispositivos monitorizados por LibreNMS.

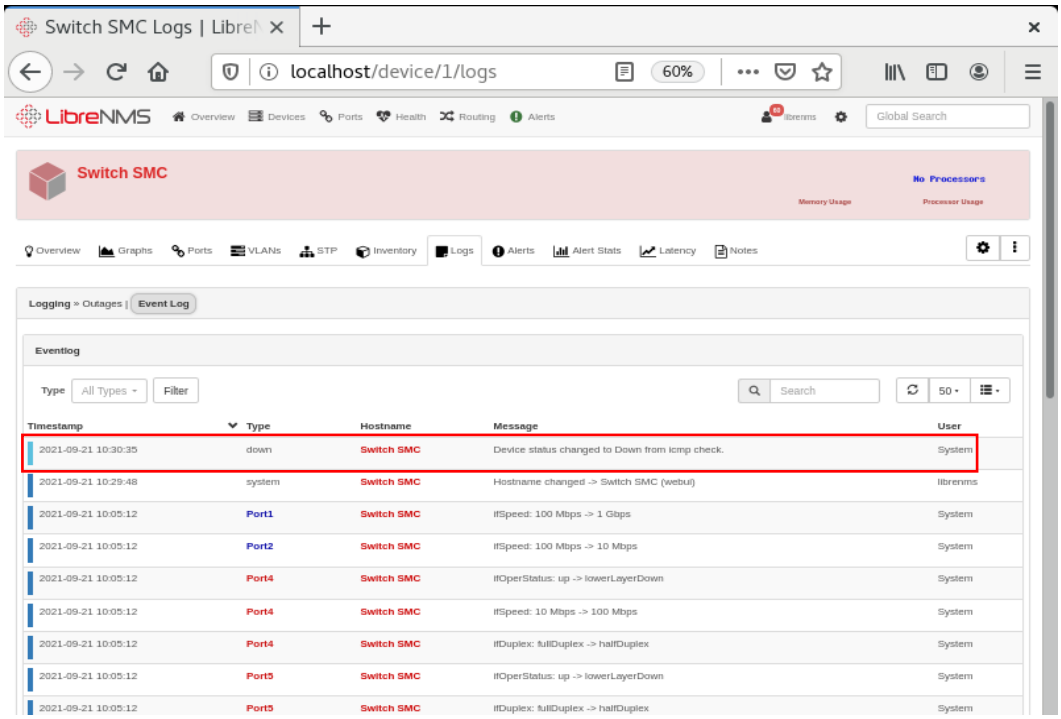


Figura 18. Listado del registro de eventos del Switch SMC.

4.3. Dashboard

Otra de las opciones que se pueden personalizar es el tablero o Dashboard. Cuando se inicia sesión por primera vez en LibreNMS, la pantalla predeterminada está vacía. Esta pantalla se puede editar usando la herramienta de dashboards para que se visualicen los widgets que el usuario crea convenientes, y asignarle permisos como privado, compartido o compartido de lectura. Incluso se pueden añadir varios dashboards para cada perfil de usuario o por información mostrada visualmente. En este trabajo, se ha considerado crear un tablero genérico que muestre información de todos los equipos del laboratorio como se puede ver en la siguiente imagen:

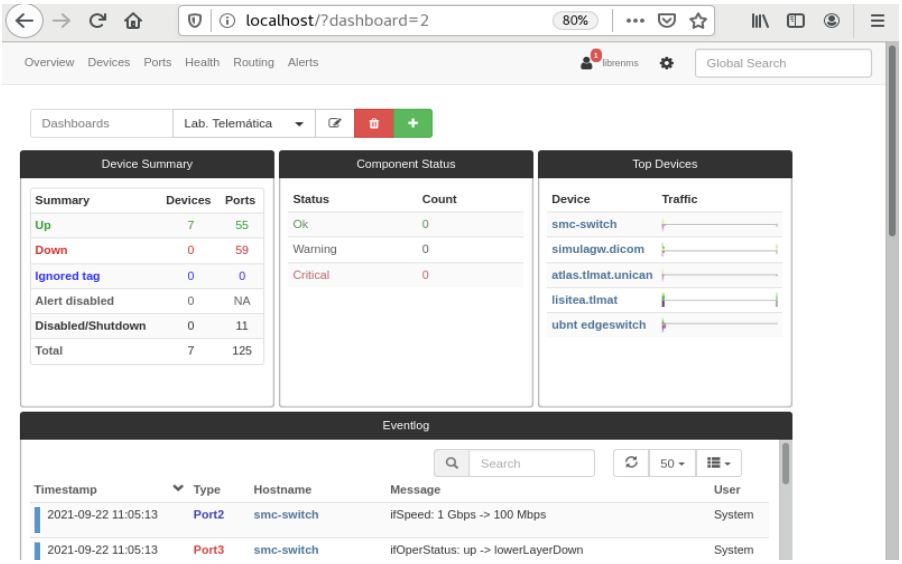


Figura 19. Dashboard personalizado para el Laboratorio de Telemática.

Adicionalmente, se ha añadido otro para mostrar la información en detalle del servidor Atlas, con gráficos de los datos obtenidos durante una hora sobre la carga de la CPU, su temperatura, los estadísticos sobre paquetes SNMP enviados y recibidos, o de Traps y de mensajes GetRequest. En la parte inferior de la *Figura 20*, el widget muestra gráficos semicirculares para datos estadísticos del propio servidor.

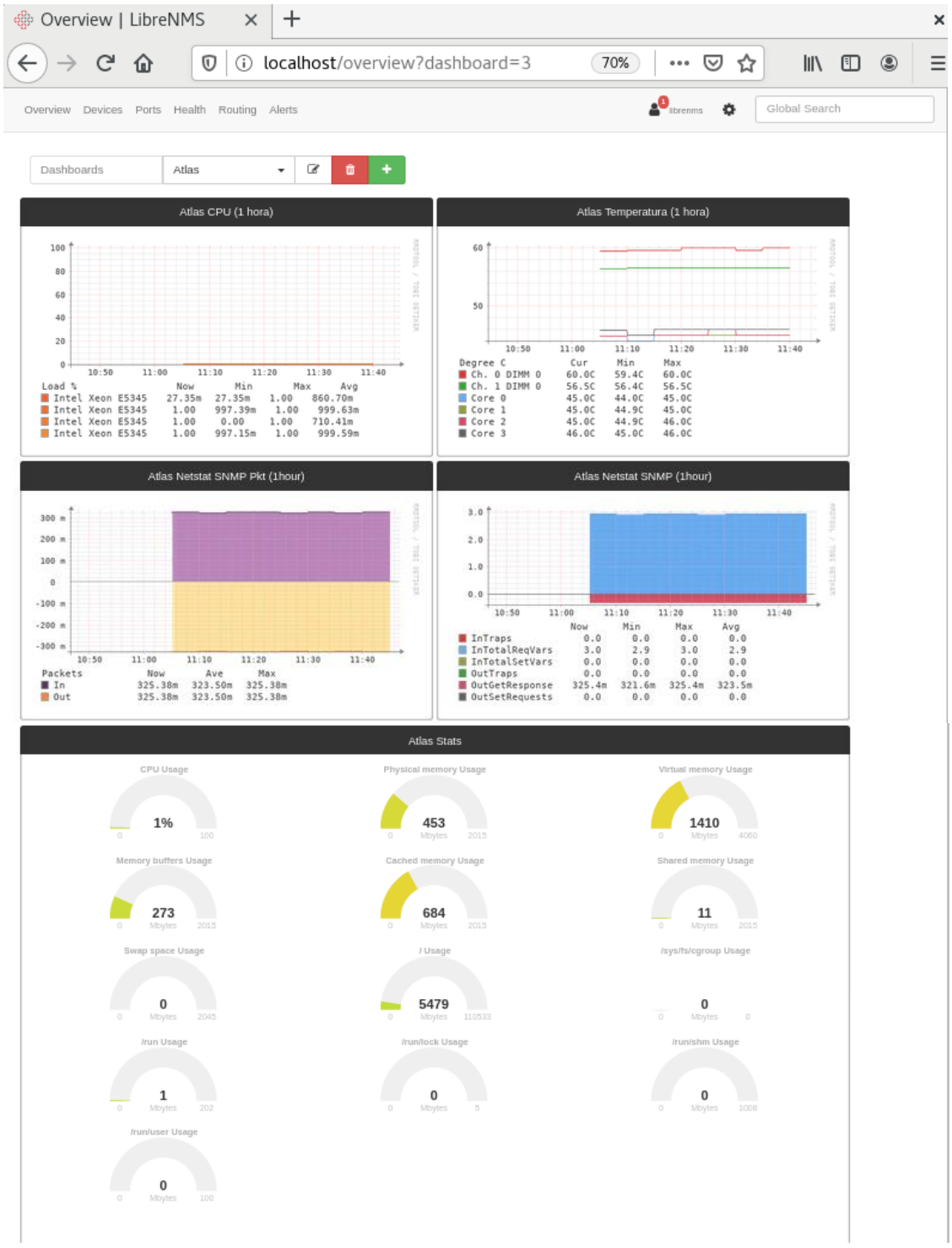


Figura 20. Dashboard personalizado con la información del servidor Atlas.

4.4. Detalles de un equipo

Del mismo modo que para el dashboard, se pueden obtener datos sobre un equipo en particular. LibreNMS facilita una pantalla o escritorio con un menú de pestañas con diferente información más detallada acerca de ese dispositivo registrado en su base de datos como se muestra en la captura de la *Figura 22*.

En la pestaña 'Ports' del switch Ubiquiti, identifica los puertos del equipo (véase *Figura 21*), por número de slot y puerto, mostrando en color azul los que están activos y tienen tráfico. La herramienta informa del tipo de conexión y capacidad de transmisión que tiene el puerto, además de su dirección MAC y el tamaño de la MTU. Estos datos son muy importantes dado que un puerto esté configurado por debajo de su capacidad, provocando así un cuello de botella en el enlace y, haciendo que se decrezca el rendimiento de la red.

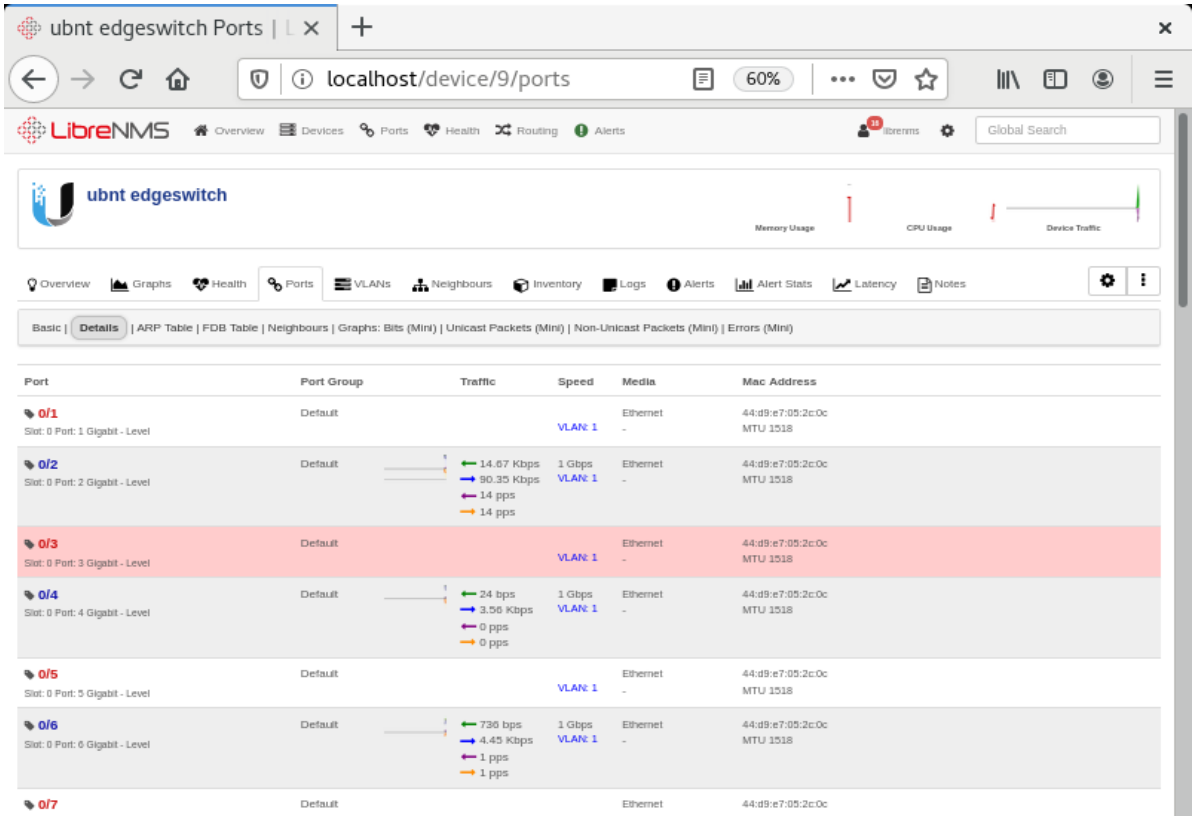


Figura 21. Listado de puertos del switch Ubiquiti.

Un ejemplo de gráfico que proporciona LibreNMS de los equipos es la latencia o 'Latency'. Esta medición permite saber si algún componente interno del equipo no está funcionando como debiera y tomar las acciones oportunas al respecto. Dentro de los componentes internos hay latencias que se producen no sólo al funcionar, sino también, cuando los componentes se comunican entre ellos. Las latencias son necesarias para que el equipo pueda recibir e interpretar correctamente la información con la que está

trabajando. En el gráfico de la *Figura 23*, las líneas dibujan de diferentes colores la información obtenida en ese periodo de tiempo y día, los valores de mínima y máxima latencia, su media y las pérdidas.

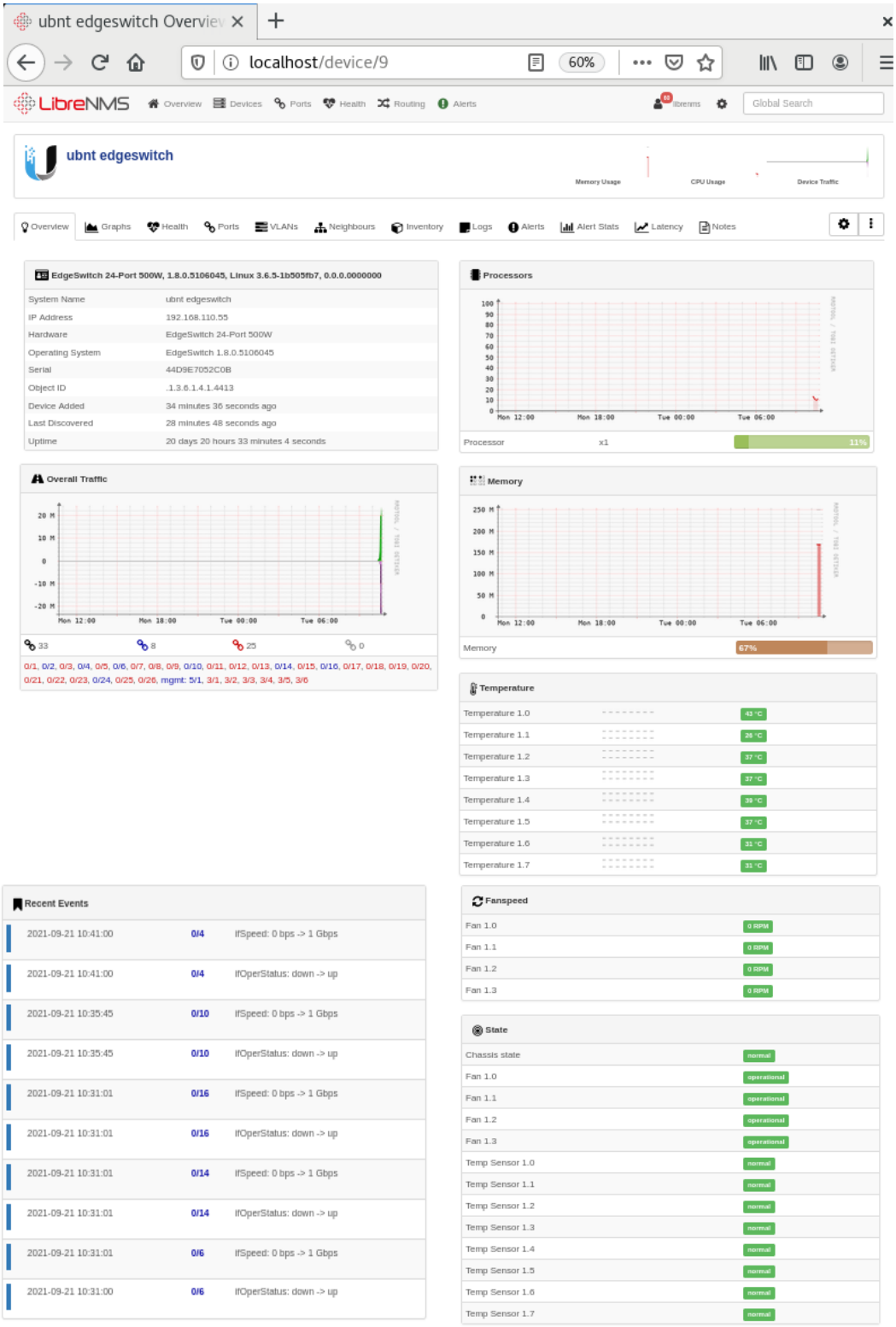


Figura 22. Escritorio principal del Switch Ubiquiti.

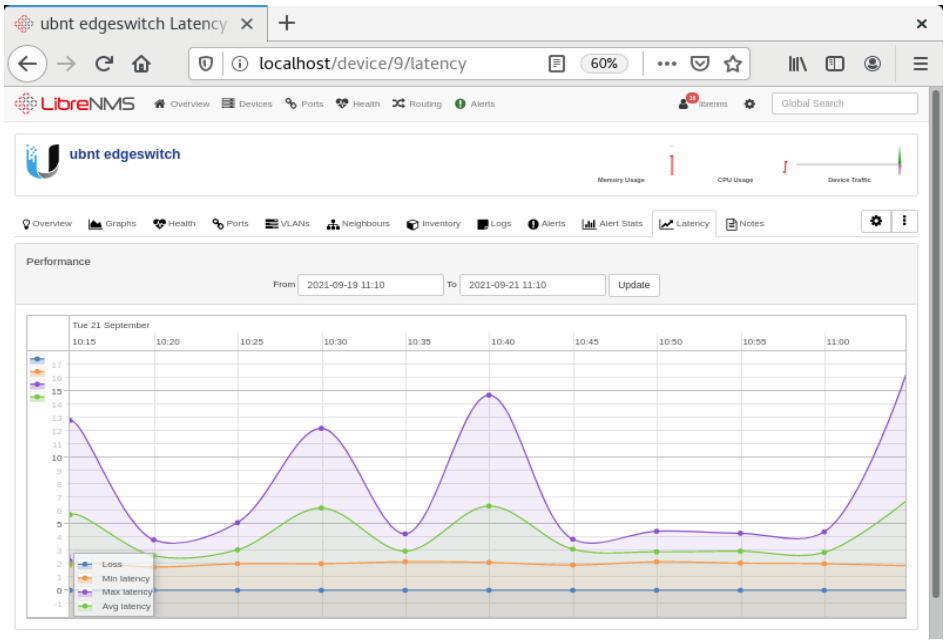


Figura 23. Gráfica de la latencia del switch Ubiquiti.

4.5. Opciones de autenticación

Se puede destacar que LibreNMS admite la habilitación de un módulo de autenticación en particular, o incluso cuenta con la identificación en dos pasos, como puede ser MySQL, Active Directory, LDAP, Radius, HTTP, o Single Sign-on. El sistema de autenticación deberá gestionar los usuarios por niveles y tipos de cuentas, y cada uno de ellos contará con unos permisos y privilegios diferentes para el uso de la herramienta. Por otra parte, el administrador puede ser capaz de ver un registro completo del histórico de usuarios logueados a la plataforma de gestión tal y como aparece en la siguiente imagen:

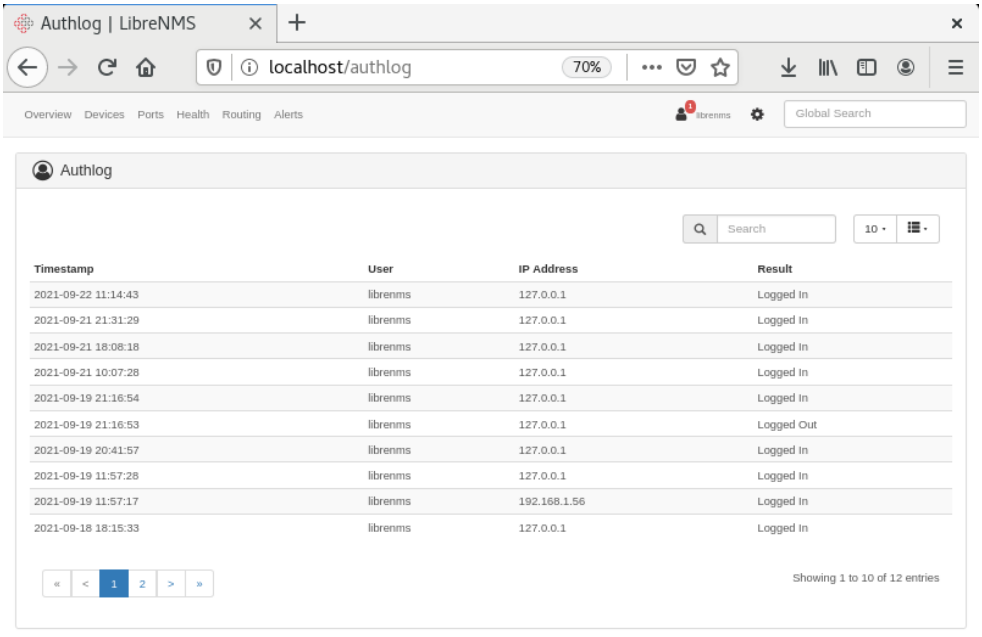


Figura 24. Histórico de registro de usuarios logueados en LibreNMS.

4.6. Plugins para servicios de red: Weathermap

En este trabajo se ha empleado este plugin para confeccionar un mapa de red personalizado con el fin de observar el tráfico existente entre los equipos principales del laboratorio. Por un lado, se ha creado la conexión entre Atlas y el switch SMC. Y por otro, desde el switch SMC se han desarrollado otras dos conexiones, una de ellas con el switch de Ubiquiti y, la otra, con el router C2600. El acceso al mapa en la interfaz web de LibreNMS es a través del menú principal, en la sección 'Overview', se selecciona la entrada de 'Plugins', se desplegará la opción de 'Weathermap' y mostrará el nombre del mapa recién creado.

Weathermap permite ver en cada momento como están de cargadas las interfaces de los equipos de la red, y al situarse encima de cada interfaz muestra el tráfico de la gráfica de cacti con un pop-up como se puede ver en la *Figura 25*, donde las líneas verdes hacen referencia al tráfico de entrada y las de color lila o azul indican el tráfico de salida.

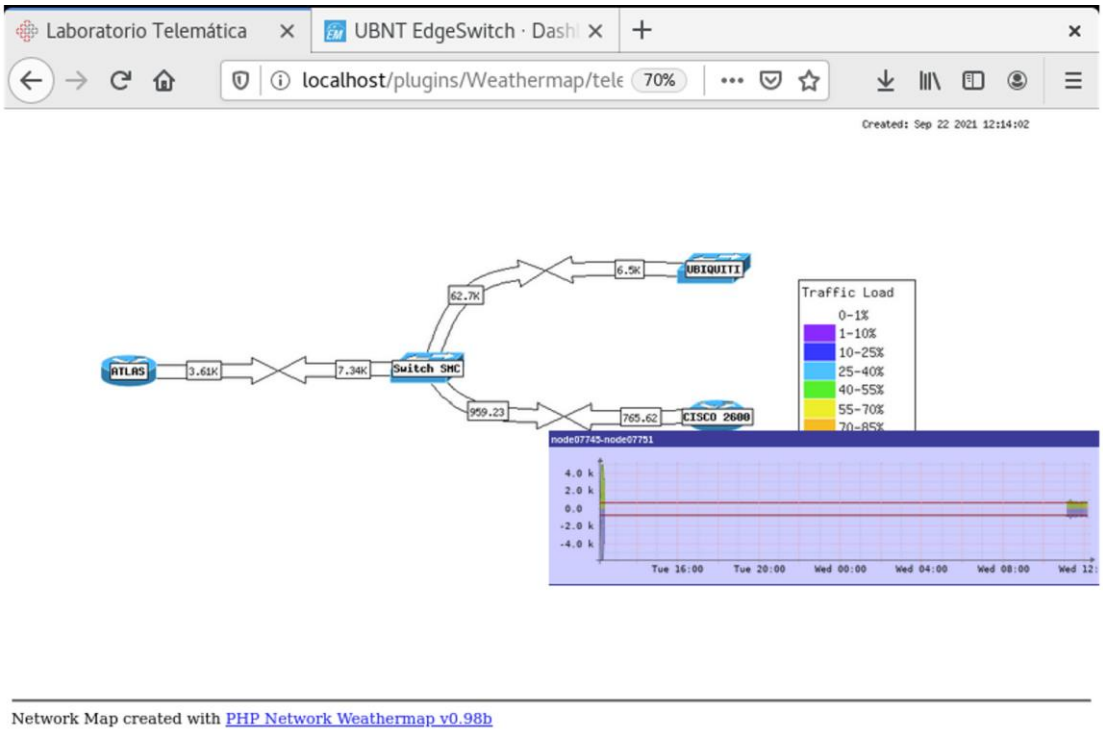


Figura 25. Pop-up del tráfico de red entre el interfaz Fa0/0 del router C2600 y el switch SMC.

No sólo se puede demostrar como Weathermap puede ser una herramienta de gran utilidad para comprobar los niveles de tráfico entre los diferentes nodos dentro de una red de menor tamaño, sino que también se hace uso de esta en grandes redes, como, por ejemplo, RedIris [6].

Se debe mencionar que a pesar de que Smokeping se ha instalado en la plataforma, no se han podido obtener resultados debido a errores con el servidor web.

5. Conclusiones y líneas futuras

Este capítulo sirve como colofón a todo el contenido previo expuesto. Se comenzará recopilando una serie de conclusiones, que se han podido entrever según se realizaba este trabajo. Se concluirá finalmente con la mención de las posibles líneas futuras a realizar.

5.1. Conclusiones

La realización de este proyecto ha supuesto un aprendizaje en el sentido de cómo se debe implementar un sistema de monitorización informático en el entorno de red del Laboratorio de Telemática situado en la Escuela de Ingenieros Industriales y de Telecomunicación de la Universidad de Cantabria para conseguir un correcto aprovechamiento del mismo.

Previo a la elección de la herramienta adecuada para cumplir los objetivos definidos en la fase inicial, se ha estudiado lo que supone el hecho de gestionar las redes y, cómo el protocolo SNMP sirve de base para poder llevar a cabo esta tarea.

A continuación, se ha desarrollado en qué consisten las plataformas de gestión, cuáles son sus componentes principales, su estructuración, y las finalidades que persiguen con su utilización. Adicionalmente, se destacan los factores claves que se deben tener en cuenta para su correcta elección. En este capítulo, se ha hecho un análisis comparativo con varias de las aplicaciones de código libre más valoradas en el mercado reflejando algunas de sus ventajas e inconvenientes.

En cuanto a la implementación de la herramienta LibreNMS, se ha podido demostrar que es un completo sistema de monitorización de redes, capaz de recopilar y mostrar un alto volumen de información de gestión al usuario, como también permite administrar y controlar los dispositivos desde una interfaz gráfica manejable e intuitiva, pese a que su configuración e instalación resultan más complejos de abordar.

Finalmente, se podría poner en valor cómo empleando esta solución se simplifican las labores del administrador de red, siendo fundamental no sólo para detectar un error, sino también para anticiparse a posibles fallos que puedan ocurrir, y es válida tanto para implementarla en un entorno de gestión sencillo con pocos equipos, como es el laboratorio, pero puede extenderse sin problemas a otros entornos de gestión más amplios siento estos públicos o privados.

5.2. Líneas futuras

En este apartado se van a mostrar las posibles mejoras que se pueden cometer en un futuro en la plataforma.

Por un lado, se puede dotar al servidor Nginx de una dirección IP estática para poder acceder a la interfaz web de LibreNMS en remoto. De este modo, se podrán visualizar datos de la gestión de la red desde cualquier ubicación, evitándose el desplazamiento hasta el laboratorio y permitiendo automatizar muchas tareas rutinarias diarias, liberando el tiempo de quienes se encargan de administrar y supervisar la red.

Por otro, para evitar fallos y ahorrar tiempo en la instalación de la plataforma, o quizá pensando en instalarla en otro equipo o host, existe la opción de emplear la imagen Docker que LibreNMS dispone en su web de github. Docker se utiliza para crear, ejecutar e implementar aplicaciones en contenedores. Una imagen contiene código de aplicación, bibliotecas, archivos de configuración, herramientas, dependencias y otros archivos necesarios para ejecutar una aplicación. Simplificando mucho su definición, se podría decir que Docker es una especie de máquina virtual, pero muy ligera. Algunos de sus puntos fuertes son la portabilidad debido a que sus contenedores comparten los sistemas operativos, o el aislamiento en sus procesos independientemente del sistema operativo de la máquina. Además, los contenedores se pueden ejecutar dentro de plataformas multi-cloud. Otro punto que destacar es su seguridad, ya que ningún contenedor Docker puede entrar a ver los procesos que se están ejecutando dentro de otro contenedor, y el ahorro de tiempo, por lo que facilita mucho el desarrollo y testeo de aplicaciones.

Por último, cabe la posibilidad de descargarse desde la web de la plataforma de gestión una imagen de Ubuntu y CentOS con LibreNMS ya instalado. De esta forma, se podrá probar en VMware o VirtualBox todas las opciones de monitorización y modificar lo que se desee sin tener que instalarla manualmente en un equipo real. Esta solución podría ser interesante a modo de realizar algún desarrollo práctico en el laboratorio, sirviendo como complemento a la teoría de alguna asignatura como, por ejemplo, Gestión y Operación de Redes. Así los alumnos podrán ir familiarizándose con una la plataforma de gestión, y les resulte útil para adquirir una base de conocimiento para su futuro laboral, dado que este campo es uno de los más demandados en la actualidad.

Bibliografía

[1] W. Stallings. "SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards". Addison-Wesley, Reading, MA, USA, 1993.

[2] LibreNMS, "LibreNMS Docs". [Online]. Disponible en: http://www.librenms.org/

[3] J. Á. Irastorza Teja. "Gestión SNMP v1, v2, v3". Apuntes de la Asignatura de Gestión y Operación de Redes, Universidad de Cantabria, 2017.

[4] "Simple Network Management Protocol". Departamento de Sistemas Telemáticos y Computación (GSyC), Universidad Rey Juan Carlos, diciembre 2013. [Online]. Disponible en: https://gsyc.urjc.es/~mortuno/lagrs/07-snmp.pdf

[5] Antoni Barba Martí. "Gestión de red". Edicions de la Universitat Politècnica de Catalunya, 1999. ISBN : 84-8301-212-X.

[6] Anónimo. Network Operation Center, "Estadísticas de Tráfico en España". RedIris. [Online]. Disponible en: https://www.rediris.es/conectividad/weathermap/

[7] James Cox. "Best Open Source Network Monitoring Tools and Software (Linux/Windows)", iTT Systems, Agosto 2021. [Online] Disponible en: https://www.ittsystems.com/best-open-source-network-monitoring-tools/

[8] UIT-T. Recomendación X.701. (01/1992). "Tecnología de la información – Interconexión de sistemas abiertos – Visión general de la gestión de sistemas". [Online]. Disponible en: https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=3052&lang=es

[9] Anna Pérez, responsable de Contenidos de OBS Business School. "Monitorización de redes, todo lo que hay que tener en cuenta". OBS Business School, febrero 2018. [Online]. Disponible en : https://www.obsbusiness.school/blog/monitorizacion-de-redes-todo-lo-que-hay-que-tener-en-cuenta

[10] NSRC, Network Startup Resource Center. Gestión y Monitoreo de Redes. "Observium : 'Todo en uno' monitoreo y gráfico de red ". [Online]. Disponible en: https://nsrc.org/workshops/2014/walc/raw-attachment/wiki/Agenda/observium.pdf

[11] Josphat Mutai. "How to Install LibreNMS on CentOS 8 / RHEL 8". Computing for geeks, julio 2021. [Online] Disponible en: https://computingforgeeks.com/how-to-install-librenms-on-rhel-centos-8/

[12] Mª del Carmen Romero. "Gestión de Redes". Sistemas Avanzados de Comunicaciones. E.T.S. Ingeniería Informática. Universidad de Sevilla. [Online]. Disponible en: http://www.dte.us.es/personal/mcromero/docs/sac/sac-gestionderedes.pdf

[13] Manuel Ramos Cabrer. "Tema 3: Modelos de gestión de red". Apuntes asignatura Gestión y Planificación de Redes con Sistemas Inteligentes. Universidad de Vigo. [Online].Disponible en: http://gssi.det.uvigo.es/users/mramos/public_html/gprsi/gprsi3.pdf

[14] Santiago Felici Castell y Ricardo Olanda. Apuntes de la Asignatura de Arquitectura de Redes y Servicios "SNMP. Simple Network Management Protocol". Universidad de Valencia. [Online]. Disponible en: http://informatica.uv.es/it3guia/ARS/apuntes/snmp.ppt

[15] Oetiker, T. Smokeping Documentation. 2007. [Online]. Disponible en: https://oss.oetiker.ch/smokeping/doc/index.en.html

[16] Anónimo. “Contenedores de software. ¿Qué es Docker?”. [Online]. Disponible en: https://www.redhat.com/es/topics/containers/what-is-docker?sc_cid=7013a000002wLwAAAU&gclid=EAIaIQobChMIjY2w466h8wIVGMHVCh2xxQRpEAAYASAAEgIca_D_BwE&gclsrc=aw.ds

[17] Gustavo B. “CentOS vs Ubuntu: ¿Cuál elegir para tu servidor web?”. Hostinger Tutoriales, julio 2021. [Online]. Disponible en: https://www.hostinger.es/tutoriales/centos-vs-ubuntu-elegir-servidor-web

[18] Deyimar A. “Qué usar – Nginx vs Apache”. Hostinger Tutoriales, julio 2021. [Online]. Disponible en: https://www.hostinger.es/tutoriales/que-usar-nginx-vs-apache

Anexo 1. Instalación y configuración de la plataforma de gestión LibreNMS

Preparación del servidor Linux

LibreNMS cuenta con una extensa documentación en su web, donde se pueden encontrar diversas maneras de instalar el software. Para este trabajo, la opción elegida ha sido la instalación de una distribución de sistema operativo CentOS 8, y un servidor web Nginx ya que es la recomendada. Es imprescindible ser el usuario *root* para ser capaz de ejecutarlos en la línea de comandos.

Siguiendo estos pasos, se procede a introducir los siguientes comandos para la instalación de los paquetes necesarios:

Instalación de los paquetes necesarios

```
dnf -y install epel-release
dnf install bash-completion composer cronie fping git ImageMagick
mariadb-server mtr net-snmp net-snmp-utils nginx nmap php-fpm php-
cli php-common php-curl php-gd php-json php-mbstring php-process
php-snmp php-xml php-zip php-mysqlnd python3 python3-PyMySQL
python3-redis python3-memcached python3-pip rrdtool unzip
```

Añadir al usuario librenms

Se añade el usuario 'librenms' al sistema:

```
useradd librenms -d /opt/librenms -M -r -s /usr/bin/bash
```

Descargar LibreNMS

LibreNMS está instalado con git. La instalación inicial desde github.com usa un "git clone". Las actualizaciones posteriores se realizan a través de "git pull".
Se procede a su descarga mediante:

```
cd /opt
git clone https://github.com/librenms/librenms.git
```

Dar permisos

A continuación, se tiene que crear y cambiar el dueño del directorio para preparar la interfaz web, y así poder agregar los dispositivos de la forma más rápida.

```
chown -R librenms:librenms /opt/librenms
chmod 771 /opt/librenms
setfacl -d -m g::rwx /opt/librenms/rrd /opt/librenms/logs
/opt/librenms/bootstrap/cache/ /opt/librenms/storage/
setfacl -R -m g::rwx /opt/librenms/rrd /opt/librenms/logs
/opt/librenms/bootstrap/cache/ /opt/librenms/storage/
```

Instalar las dependencias PHP

```
su - librenms
./scripts/composer_wrapper.php install --no-dev
exit
```

Fijar la zona horaria

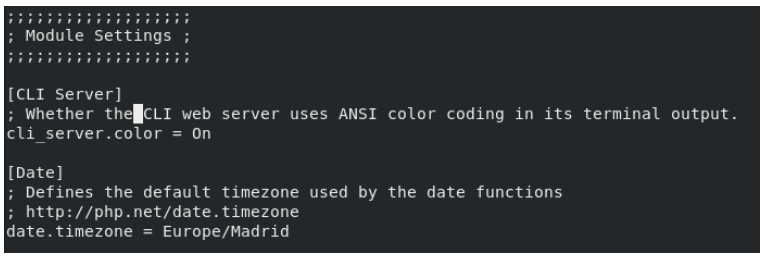
Se configura la zona horaria según la ubicación, editando el fichero "/etc/php.ini".

```
vi /etc/php.ini
```

Para este caso:

```
;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;

[Date]
;Defines the default timezone used by the date functions
;http://php.net/date.timezone
date.timezone = Europe/Madrid
```



También se debe fijar la zona horaria del sistema:

```
timedatectl set-timezone Etc/UTC
```

Configuración del servidor de la base de datos

Seguidamente, se configura MySQL/MariaDB:

```
vi /etc/my.cnf.d/mariadb-server.cnf
```

Dentro de la sección `[mysqld]` hay que añadir:

```
innodb_file_per_table=1
lower_case_table_names=0
```

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/run/mariadb/mariadb.pid
innodb_file_per_table=1
lower_case_table_names=0
```

Se habilita y reinicia el motor de la base de datos:

```
systemctl enable mariadb
systemctl restart mariadb
```

```
mysql -u root
```

Se crea la base de datos y se añade la contraseña de acceso 'librenms' para el usuario *librenms*:

```
CREATE DATABASE librenms CHARACTER SET utf8 COLLATE utf8_unicode_ci;
CREATE USER 'librenms'@'localhost' IDENTIFIED BY librenms;
GRANT ALL PRIVILEGES ON librenms.* TO 'librenms'@'localhost';
FLUSH PRIVILEGES;
exit
```

Configurar PHP-FPM

PHP-FPM (*FastCGI Process Manager*) es la implementación alternativa más popular de PHP FastCGI.

```
cp /etc/php-fpm.d/www.conf /etc/php-fpm.d/librenms.conf
vi /etc/php-fpm.d/librenms.conf
```

```
# Change "www" to "librenms"
[librenms]

# Change user and group to "librenms"
user = librenms
group = librenms

# Change listen to a unique name
listen = /run/php-fpm-librenms.sock
```

Sino existe otra aplicación web PHP en este servidor, se puede eliminar www.conf para ahorrar recursos.

Configuración del servidor web

Nginx como servidor web de alto rendimiento estable, y con un consumo de recursos muy bajo, es el compañero ideal de PHP-FPM. Nginx tiene una arquitectura asíncrona que es mucho más escalable, basada en eventos. Además, al usar Nginx con PHP-FPM se mejora la eficiencia a nivel de consumo de memoria.
PHP funciona como un servicio separado al usar PHP-FPM. Al usar esta versión de PHP como intérprete del lenguaje, las peticiones se procesan a través de un socket TCP/IP; de modo que el servidor web Nginx solo maneja las peticiones HTTP y PHP-FPM interpreta el código PHP. El hecho de tener dos servicios separados es clave para ganar en eficiencia.
Se procede a configurar el "virtual host" para NGINX:

```
vi /etc/nginx/conf.d/librenms.conf
```

Hay que añadir la siguiente configuración, editando el `server_name`:

SELinux

SELinux, se trata de un módulo de seguridad para el kernel Linux. En inglés "Security-Enhanced Linux", Seguridad-Mejorada de Linux. Dicho módulo proporciona el mecanismo para utilizar políticas de seguridad en el control de acceso más allá de los permisos tradicionales de propiedad, incluyendo controles de acceso obligatorios, como los implementados por el departamento de defensa de Estados Unidos en sistemas de alta seguridad.

Se instala la herramienta de política de seguridad para SELinux:

```
dnf install policycoreutils-python-utils
```

Configuración del contexto necesario para LibreNMS:

Para que funcione LibreNMS, debemos configurar el contexto:

```
semanage fcontext -a -t httpd_sys_content_t
'/opt/librenms/html(/.*)?'
semanage fcontext -a -t httpd_sys_rw_content_t
'/opt/librenms/(logs|rrd|storage)(/.*)?'
restorecon -RFvv /opt/librenms
setsebool -P httpd_can_sendmail=1
setsebool -P httpd_execmem 1
chcon -t httpd_sys_rw_content_t /opt/librenms/.env
```

```
server {
 listen      80;
 server_name localhost;
 root        /opt/librenms/html;
 index       index.php;

 charset utf-8;
 gzip on;
 gzip_types text/css application/javascript text/javascript
application/x-javascript image/svg+xml text/plain text/xsd text/xsl
text/xml image/x-icon;
 location / {
  try_files $uri $uri/ /index.php?$query_string;
 }
 location ~ [^/]\.php(/|$) {
  fastcgi_pass unix:/run/php-fpm-librenms.sock;
  fastcgi_split_path_info ^(.+\.php)(/.+)$;
  include fastcgi.conf;
 }
 location ~ /\.(?!well-known).* {
  deny all;
 }
}
```

Se tiene que eliminar la sección del `server` de `/etc/nginx/nginx.conf`. Se habilita el servidor web y php-fpm:

```
systemctl enable --now nginx
systemctl enable --now php-fpm
```

Habilitar fping dentro de SELinux

Se debe crear el fichero "http_fping.tt" con el siguiente contenido. Se puede crear el fichero en cualquier lugar, ya que se trata de un fichero desechable. En el último paso de este procedimiento de instalación, instalará el módulo en la ubicación correcta.

```
module http_fping 1.0;

require {
type httpd_t;
class capability net_raw;
class rawip_socket { getopt create setopt write read };
}

#============= httpd_t ==============
allow httpd_t self:capability net_raw;
allow httpd_t self:rawip_socket { getopt create setopt write read };
```

Una vez guardado el fichero se tienen que ejecutar los siguientes comandos:

```
checkmodule -M -m -o http_fping.mod http_fping.tt
semodule_package -o http_fping.pp -m http_fping.mod
semodule -i http_fping.pp
```

Se pueden encontrar problemas adicionales de SELinux ejecutando el siguiente comando:

```
audit2why < /var/log/audit/audit.log
```

Permitir el acceso a través del firewall

Habilitar acceso en el cortafuegos:

```
firewall-cmd --zone public --add-service http --add-service https
firewall-cmd --permanent --zone public --add-service http --add-service
```

Habilitar el comando lnms

Esta característica concede la oportunidad de utilizar el tabulador para completar los comandos lnms como lo haría para los comandos Linux normales.

```
ln -s /opt/librenms/lnms /usr/local/bin/lnms
cp /opt/librenms/misc/lnms-completion.bash /etc/bash_completion.d/
```

Configurar snmpd

Se tiene que copiar el fichero de configuración de LibreNMS.

```
cp /opt/librenms/snmpd.conf.example /etc/snmp/snmpd.conf
```

```
vi /etc/snmp/snmpd.conf
```

Se edita el texto que dice `RANDOMSTRINGGOESHERE` y se añade la comunidad oportuna. Para esta instalación se utilizará la comunidad "LABORATORIO":

```
# Change RANDOMSTRINGGOESHERE to your preferred SNMP community string
com2sec readonly  default         LABORATORIO
```

```
curl -o /usr/bin/distro
https://raw.githubusercontent.com/librenms/librenms-
agent/master/snmp/distro
chmod +x /usr/bin/distro
systemctl enable snmpd
systemctl restart snmpd
```

Trabajo del cron

Creación de tarea programada (Cron job):

```
cp /opt/librenms/librenms.nonroot.cron /etc/cron.d/librenms
```

NOTA: se debe tener en cuenta que cron, por defecto, únicamente utiliza un conjunto muy limitado de variables de entorno. Es necesario configurar las variables proxy para la invocación de cron. Alternativamente, también es posible agregar la configuración de proxy en config.php. El archivo config.php será creado en los pasos siguientes. Se puede revisar la siguiente dirección URL después de finalizar los pasos de instalación de librenms: https://docs.librenms.org/Support/Configuration/#proxy-support

Copiar la configuración del logrotate

LibreNMS guarda los registros de actividad en "`/opt/librenms/logs`", con el tiempo, estos pueden volverse grandes y rotar. Para poder rotar los registros antiguos, se puede utilizar el fichero de configuración para "logrotate" proporcionado:

```
cp /opt/librenms/misc/librenms.logrotate /etc/logrotate.d/librenms
```

Instalación web

Se deben seguir las instrucciones en el navegador web, escribiendo la URL: http://localhost/install

La primera pantalla que aparece indica si la instalación cumple con los requisitos:

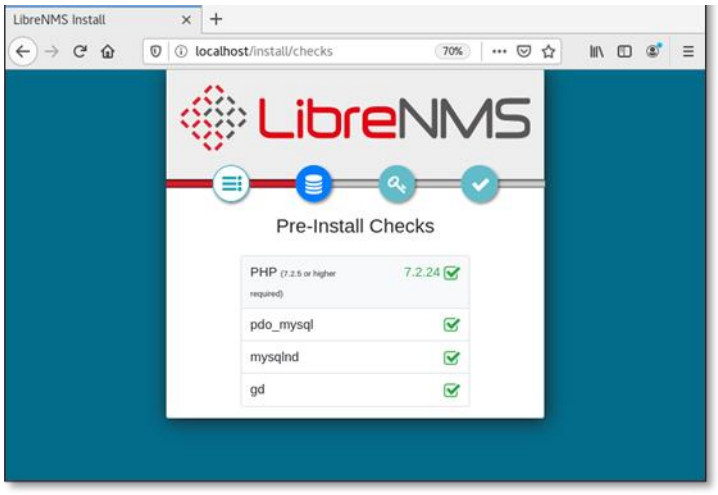


Figura 26. Lista de requisitos exigidos por LibreNMS.

A continuación, se tienen que indicar las credenciales de la base de datos que hemos configurado:

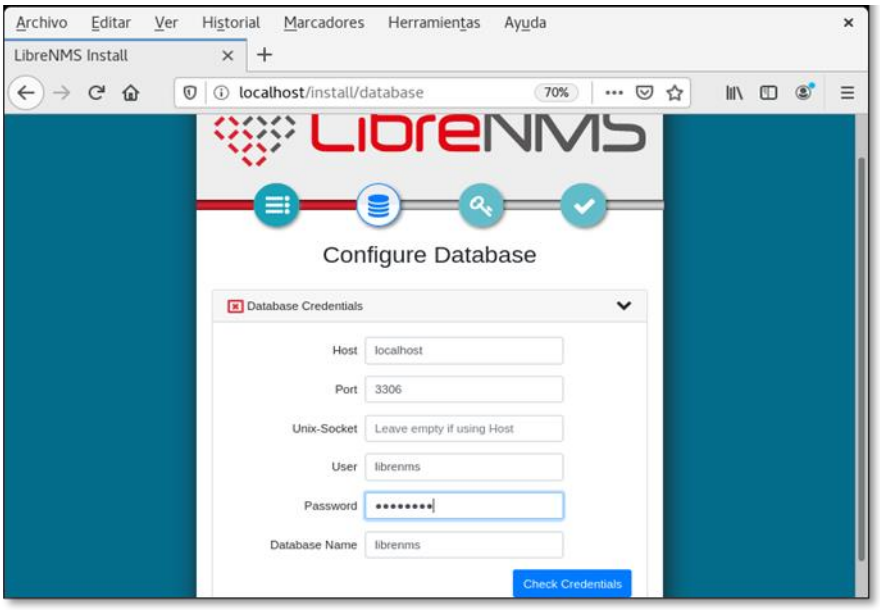


Figura 27. Credenciales de base de datos para la instalación de LibreNMS.

Se importa la base de datos:

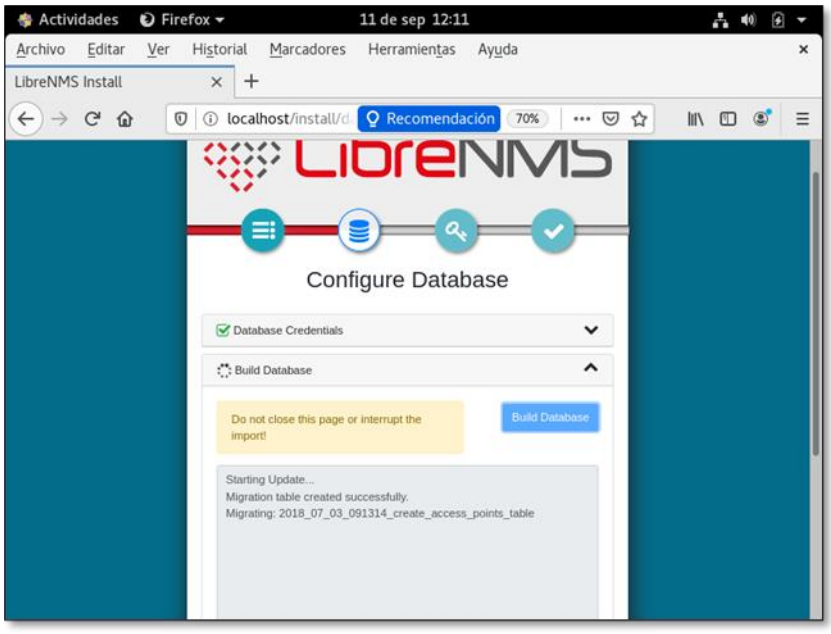


Figura 28. Proceso de importación de la base de datos.

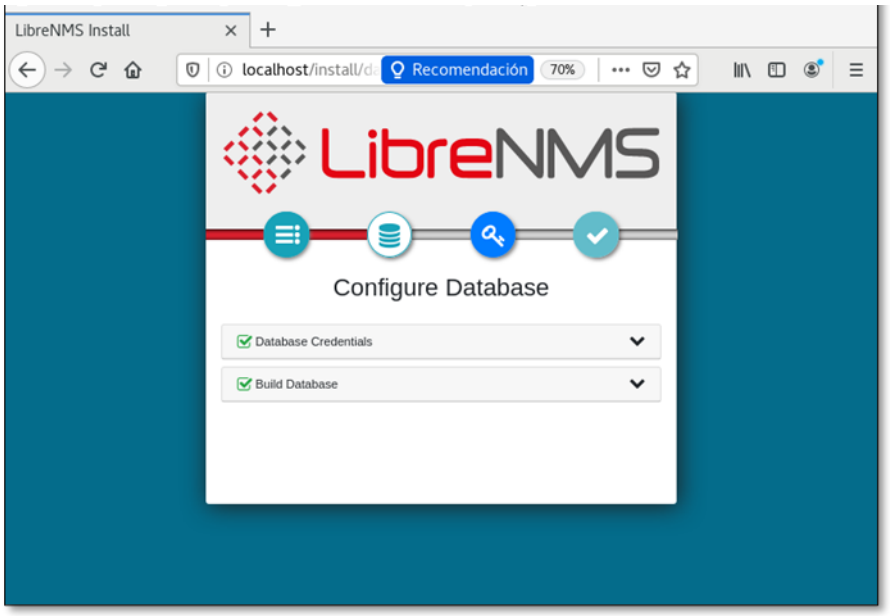


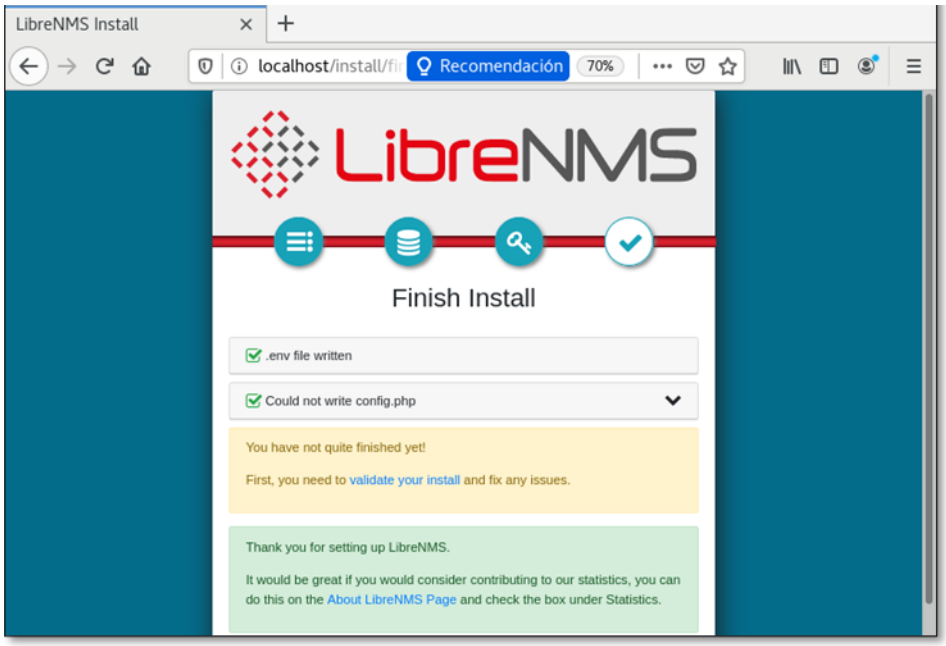
Figura 29. Proceso de creación de la base de datos finalizado.

Finalmente, se añade un usuario administrador, sus credenciales y una dirección de correo:

Figura 30. Creación del usuario administrador.

Figura 31. Finalización del proceso de creación del usuario administrador.

Una vez finalizada la instalación se visualizará la siguiente pantalla:



El instalador web puede solicitar la creación de un fichero llamado "config.php" en la ubicación de la instalación. Copiando el contenido que se muestra en pantalla al fichero. Si se lleva a cabo esto se debe recordar establecer los permisos del fichero, después de copiar los contenidos en pantalla:

```
chown librenms:librenms /opt/librenms/config.php
```

Pantalla de acceso a la plataforma:

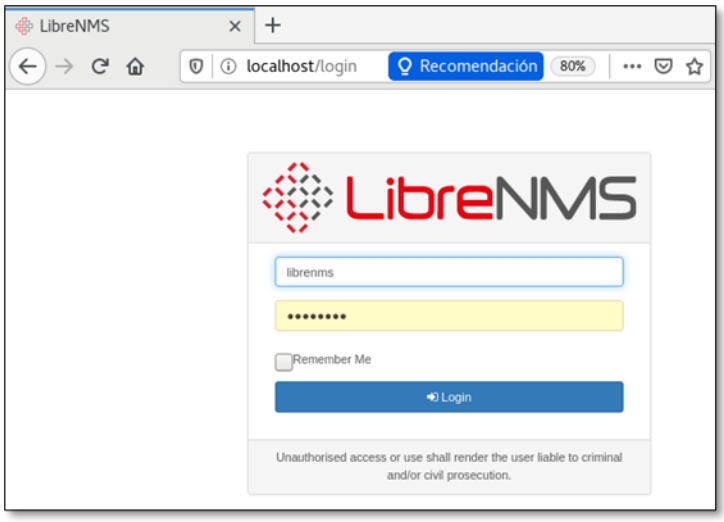


Figura 32. Pantalla de acceso a la plataforma.

Pasos finales: Añadir el primer dispositivo

Una vez instalada la plataforma, el escritorio está vacío, por lo que se debe crear uno al gusto con las vistas que interesen:

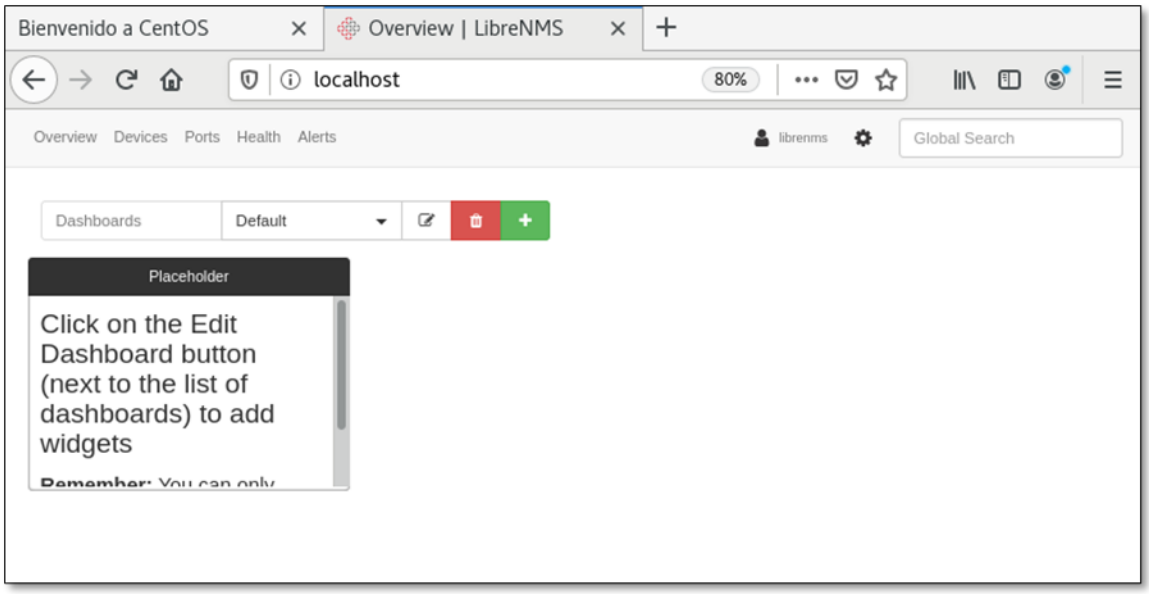


Figura 33. Pantalla de inicio o escritorio de LibreNMS.

Anexo 2. Instalación de Weathermap en LibreNMS

Paso 1

Hay que dirigirse al directorio de plugins de LibreNMS, `/opt/librenms/html/plugins`, en el que se encontrará el directorio: `/opt/librenms/html/plugins/Weathermap/`. La mejor manera es hacerlo via git. Situarse en el directorio donde se va a proceder la instalación y, después, `/opt/librenms/html/plugins`, se debe ejecutar:

```
git clone https://github.com/librenms-plugins/Weathermap.git
```

Paso 2

Se deben cambiar los permisos dentro del directorio html/plugins, introduciendo:

```
chown -R librenms:librenms Weathermap/
```

Después, dar permisos de escritura al directorio de configuración:

```
chmod 775 /opt/librenms/html/plugins/Weathermap/configs
```

Si se está utilizando SELinux, se debe introducir el siguiente comando:

```
chcon -R -t httpd_cache_t Weathermap/
```

Paso 3

Se debe habilitar el proceso cron editando el archivo cron actual de LibreNMS (normalmente /etc/cron.d/librenms) y se debe añadir lo siguiente:

```
*/5 * * * * librenms /opt/librenms/html/plugins/Weathermap/map-
poller.php >> /dev/null 2>&1
```

Paso 4

Se debe habilitar el complemento desde la interfaz de usuario web de LibreNMS en:

OverView -> Complementos -> Menú de administración de complementos.

Paso 5

Ahora se debería poder visualizar:

Weathermap Overview -> Plugins -> Weathermap Create your maps.

Cuando se quiera crear un mapa, hay que seleccionar *Map Style*, y asegurarse de que Overlib esté seleccionado para HTML y hacer clic en *submit*. Además, hay que establecer un nombre de archivo de imagen de salida y un nombre de archivo HTML de salida en *Map Properties*. Es recomendable utilizar la carpeta `output` ya que está excluida de las actualizaciones de git (es decir, usar `output/mymap.png` y `output/mymap.html`).

WeatherMapper

Genera automáticamente Weathermaps desde una base de datos de LibreNMS utilizando WeatherMapper.

Agregar Weathermaps de la red al panel principal

Una vez creado el 'Weather Map' de la red, se puede añadir a la pantalla del panel siguiendo estos pasos:

Paso 1

Una vez creado el Weathermap hay que asegurarse de exportarlo como HTML y PNG ya que se necesitará para salir del panel.

En la página del plugin Weathermap se podrán ver los mapas de salida. Se debe seleccionar uno de ellos con el clic derecho y hacer clic en `copy image address`.

URL de ejemplo:
`http://yourlibrenms.org/plugins/Weathermap/output/yourmap.html`

Paso 2

Después, hay que volver al panel principal, crear un nuevo panel y darle un nombre. Se debe seleccionar el widget como *Imágenes externas*. Se debe dar un nombre al widget.

La URL destino de la imagen deberá ser la dirección copiada anteriormente, pero quitándole el `.html` y reemplazándolo por `.png`:

`http://yourlibrenms.org/plugins/Weathermap/output/yourmap.png`

Seguidamente, se debe hacer clic en Set.

Ahora se debería poder ver el Weathermap que se acaba de crear en la lista de paneles de control. También es posible añadirlo a los que hay.

Anexo 3. Instalación de Smokeping en LibreNMS

Smokeping es una herramienta que permite rastrear la latencia de la red, y visualizarla a través de gráficos RRD.

LibreNMS puede soportar tanto instalaciones de SmokePing nuevas como mantener las ya existentes. Para instalaciones nuevas, se puede utilizar el comando `lnms` para generar el archivo de configuración de Smokeping.

Nueva instalación de SmokePing

Instalación e integración de Smokeping Backend - RHEL, CentOS y similares

Smokeping está disponible via EPEL, el cual ya se debe disponer si ya se está utilizando LibreNMS. Si se quiere ejecutar en un host diferente y enviar datos por medio de RRCached, se debe ejecutar este comando de instalación:

```
sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-
latest-7.noarch.rpm
sudo yum install smokeping
```

Una vez instalado, se necesita un script cron instalado para asegurarse de que el archivo de configuración está actualizado. Se puede encontrar un ejemplo en `misc/librenms-smokeping-rhel.example`. Hay que añadirlo al directorio /etc/cron.d/hourly y marcarlo como ejecutable:

```
sudo cp /opt/librenms/misc/smokeping-rhel.example
/etc/cron.hourly/librenms-smokeping
sudo chmod +x /etc/cron.hourly/librenms-smokeping
```

Finalmente, se debe actualizar la configuración predeterminada. Quite todo de las estrofas *** Probes *** y *** Targets ***, y hay que reemplazarlo por:

```
*** Probes ***

@include /etc/smokeping/librenms-probes.conf
```

```
*** Targets ***

probe = FPing

menu = Top
title = Network Latency Grapher
remark = Welcome to the SmokePing website of <b>Insert Company Name
Here</b>. \
         Here you will learn all about the latency of our network.

@include /etc/smokeping/librenms-targets.conf
```

Se debe tener en cuenta que puede haber otras estrofas (posiblemente `*** Slaves ***`). Entre los `*** Probes ***`y `*** Targets ***`estrofa – se deben dejar intactos.

No se debe modificar nada más. Si es necesario agregar otra configuración, se debe hacer después de la configuración de LibreNMS, y hay que tener en cuenta que Smokeping no permite módulos duplicados y se preocupa por la secuencia de configuración del archivo.

Una vez configurado todo, hay que iniciar manualmente el cron, habilitando e iniciando Smokeping:

```
sudo /etc/cron.hourly/librenms-smokeping
sudo systemctl enable --now smokeping
```

Instalación de Smokeping ya existente

Cuando ya existe un servidor de Smokeping, se deben seguir los siguientes pasos:

Se debe editar el archivo `/opt/librenms/config.php` y añadir lo siguiente:

```
$config['smokeping']['dir'] = '/var/lib/smokeping';
$config['smokeping']['pings'] = 20;
$config['smokeping']['probes'] = 2;
$config['smokeping']['integration'] = true;
$config['smokeping']['url'] = 'smokeping/';  // If you have a
specific URL or path for smokeping
```

`dir` debe coincidir con la ubicación en la que Smokeping escribe RRD a los `pings` que deben de coincidir con el valor de Smokeping predeterminado, los 20 `probes` deben ser el número de procesos para distribuir pings, que por defecto son 2.
Esta configuración también se puede establecer en la interfaz de usuario web.

Configurar la interfaz de usuario web de Smokeping - Opcional

Esta sección cubre la configuración requerida para su servidor web elegido. Esto incluye la configuración requerida para Nginx.

LibreNMS no necesita la interfaz de usuario web; se pueden encontrar los gráficos en la pestaña de *lantencia* de LibreNMS.

Se debe de tener en cuenta que debe instalar fcgiwrap para que el contenedor CGI interactúe con Nginx.

```
apt install fcgiwrap
```

Luego se tiene que configurar Nginx con la configuración por defecto:

```
cp /usr/share/doc/fcgiwrap/examples/nginx.conf
/etc/nginx/fcgiwrap.conf
```

Hay que añadir la siguiente configuración al archivo de configuración /etc/nginx/conf.d/librenms.

```
# Browsing to `http://yourlibrenms/smokeping/` should bring up the
smokeping web interface

location = /smokeping/ {
        fastcgi_intercept_errors on;

        fastcgi_param   SCRIPT_FILENAME         /usr/lib/cgi-
bin/smokeping.cgi;
        fastcgi_param   QUERY_STRING            $query_string;
        fastcgi_param   REQUEST_METHOD          $request_method;
        fastcgi_param   CONTENT_TYPE            $content_type;
        fastcgi_param   CONTENT_LENGTH          $content_length;
        fastcgi_param   REQUEST_URI             $request_uri;
        fastcgi_param   DOCUMENT_URI            $document_uri;
        fastcgi_param   DOCUMENT_ROOT           $document_root;
        fastcgi_param   SERVER_PROTOCOL         $server_protocol;
        fastcgi_param   GATEWAY_INTERFACE       CGI/1.1;
        fastcgi_param   SERVER_SOFTWARE
nginx/$nginx_version;
        fastcgi_param   REMOTE_ADDR             $remote_addr;
        fastcgi_param   REMOTE_PORT             $remote_port;
        fastcgi_param   SERVER_ADDR             $server_addr;
        fastcgi_param   SERVER_PORT             $server_port;
        fastcgi_param   SERVER_NAME             $server_name;
        fastcgi_param   HTTPS                   $https if_not_empty;

        fastcgi_pass unix:/var/run/fcgiwrap.socket;
}

location ^~ /smokeping/ {
        alias /usr/share/smokeping/www/;
        index smokeping.cgi;
        gzip off;
}
```

Después de guardar el archivo de configuración, se debe verificar si la sintaxis del archivo de configuración de Nginx está bien con `sudo nginx -t`, luego reiniciar Nginx con `sudo systemctl restart nginx`.

Se debería poder cargar la interfaz web de Smokeping en: `http://localhost/smokeping`