CAPÍTULO 3: Elección del sistema de gestión a utilizar

Dada la complejidad y dificultad de realizar una aplicación de gestión completa y la existencia de numerosas aplicaciones que implementan diferentes funcionalidades FCPAS, parece aconsejable la selección y parametrización de una herramienta existente para dar cumplimiento a los objetivos que se plantean en este proyecto. Se debe conseguir que la herramienta a implantar cumpla los siguientes requisitos:

- Que sea robusta para que esté operativa en todo momento.
- Que ofrezca un coste bajo o nulo.
- Que sea sencilla de manejar.
- Que permita recopilar información de cada dispositivo sean cuales sean sus características. Asimismo, debe ofrecer la posibilidad de programar alertas con distintas criticidades y acciones a realizar cuando salte alguna de estas.
- Que disponga de un inventario de todos los componentes de la red.
- Que no consuma demasiados recursos.
- Que muestre la información de la forma más sencilla posible para que sea fácil de interpretar.
- Que permita distintos niveles de acceso.
- Que incorpore una base de datos robusta para almacenar y disponer de todos los valores recogidos.

Por ello, a continuación se van a presentar distintos sistemas de gestión de redes de los que se destacarán sus principales funcionalidades, sus ventajas y sus inconvenientes. Para finalizar el capítulo se escogerá uno y se argumentará el por qué de su elección.

En el apéndice A [5], se muestra un cuadro comparativo de distintos sistemas de monitorización. En este apartado sólo se presentarán cuatro de estos.

3.1 - JFFNMS (*JUST FOR FUN NETWORK MANAGEMENT SYSTEM*)

JFFNMS [6] es un sistema de gestión de red designado para monitorizar una red IP. Se trata de software libre y esta bajo licencia GPL (*General Public License*).

JFFNMS usa PHP para recoger y mostrar la información de varias herramientas que pueden ser encontradas en una red de ordenadores. El principal módulo de JFFNMS está formado por las interfaces recogidas por una máquina. El concepto de interfaz engloba tanto a los puertos físicos de una red, como a algo que tiene un estado, uno o varios valores, o ambos. Puede informar sobre la situación de los interfaces mediante eventos, alarmas, gráficos o acciones. Por ejemplo, cuando existe un cambio de estado en un interfaz, se puede realizar una acción como el envío de un email.

JFFNMS recopila los datos de los elementos de la red de dos formas distintas:

- Por peticiones del servidor hacia el dispositivo. Conocido como *polling*, para lo que se utiliza SNMP sobre conexiones TCP.
- Por notificaciones que podrán ser traps SNMP o mensajes *syslog*.

Syslog es un estándar que se utiliza para el envío de mensajes de registro en una red IP. Por ejemplo, se utilizará para registrar un intento de acceso con contraseña equivocada, un acceso correcto al sistema, anomalías (variaciones en el funcionamiento normal del sistema), etc.

El nivel más bajo de NMS *(Network Management Systems)* es el elemento que supervisa un dispositivo específico y el más alto consigue mostrar una visión general del comportamiento de todo el sistema.

3.1.1 - CARACTERÍSTICAS PRINCIPALES

De sus características destacan:

- Alarmas de *syslog* y eventos SNMP trap.
- Representación gráfica de datos estadísticos de interfaces de red.
- Notificaciones vía email basadas en la configuración de alarmas.
- Permite monitorizar una red IP mediante SNMP, Syslog y TACACS+ (*Terminal Access Controller Access Control System*).

- Puede ser utilizado para monitorizar cualquier dispositivo SNMP, servidor, router, puerto TCP o cualquier elemento que se desee, siempre que se programe una extensión adecuada a dicho elemento para JFFNMS.

- Dispone de características orientadas al manejo de dispositivos Cisco.

- Está escrito en PHP y funciona en entornos GNU/Linux, FreeBSD y Windows 2000/XP.

- La Consola de Eventos muestra todos los tipos de eventos de manera ordenada en el mismo *display*.

- Genera gráficas para todos los dispositivos de la red de datos, mostrando parámetros como el tráfico de la red, la utilización de CPU, los errores, etc.

- Tiene soporte de base de datos (MySQL o PostgreSQL), integra log de Syslog y autenticación e informes de TACACS+. Un *log* se define como un fichero que recoge todos los pasos que se producen en el sistema. Es un registro oficial de eventos durante un periodo de tiempo en particular.

- Es muy modular y extensible, lo que significa que se pueden programar extensiones en caso de que no se disponga de soporte para los elementos específicos de la red.

- Se basa en las tecnologías Apache, Cron, MySQL, PHP, RRDTool (*Round Robin Database tool*) y SNMP. Señalar que *Cron* es una utilidad de Linux que permite programar en el tiempo algunas funciones como backup y procedimientos de manutención.

- Dispone de un Mapa de Estado que permite visualizar la red de una manera sencilla.

- La lista de dispositivos, o tipos de interfaz que JFFNMS puede monitorizar es extensa y gracias a sus desarrolladores y usuarios está creciendo.

- La comunidad de usuarios de JFFNMS posee una lista de distribución que compensa la carencia de documentación sobre la herramienta.

- Necesita la instalación y configuración del complemento (agente) SNMP en los clientes.

- Fácil instalación y configuración, sobre todo cuando funciona sobre Windows XP/2000. Todas las tareas se configuran siguiendo los mismos pasos.

- Ofrece el servicio de descubrimiento de interfaces y además se puede programar el periodo de repetición. Se puede observar un ejemplo de configuración en la figura 3.1.

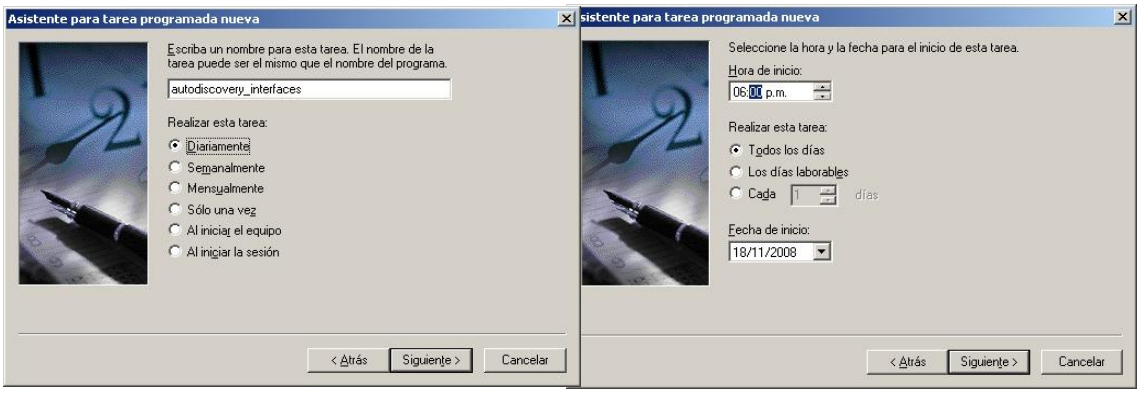


Figura 3.1 - Configuración para la realización de un descubrimiento diario a las 6:00.

3.1.2 - RECOGIDA Y CLASIFICACIÓN DE DATOS

Dichas tareas se consiguen programando entre otros, procesos como *poller.bat* o *consolidate.bat*, lo cual es bastante sencillo como se muestra en el ejemplo de la figura 3.2.

La forma en que se recoge y se procesa la información dentro de JFFNMS es lo que se pasa a describir en esta sección.

Figura 3.2 - Ejemplo de programación de tareas para la recogida y el almacenamiento de datos.

3.1.2.1 - Recogida de datos

Existen tres formas principales de que la información llegue a un elemento de red tal como un *router*, un *switch* o un servidor. Estas son el sondeo JFFNMS conocido como *poller*, los mensajes *syslog* y los *trap* SNMP.

Cada 5 minutos *cron* inicia la secuencia de sondeo. La información recogida se almacena en un fichero RRD, (fichero de RRDTool), como un evento o como un conjunto de datos. Esta operación es la conocida como *poller*.

Por otro lado, un elemento de red puede decidir enviar un *trap* SNMP o un mensaje *syslog* a JFFNMS. En este caso, la información se obtendrá mediante un programa externo, *snmptrapd()* o *syslogd()*. Dichos programas se ejecutan en el gestor a la espera de recibir

las notificaciones enviadas. Si se ha configurado correctamente, entonces el *trap* o mensaje *syslog* se almacenará en la base de datos como un mensaje de ese tipo.

3.1.2.2 - Fase de clasificación de datos recogidos

Una vez más, el *cron* inicia su tarea cada instante llamando al *consolidator*. Este proceso está dotado de varios tipos de clasificación, que generalmente trabajan en tres conjuntos de tablas: una de entrada, una de norma adaptada y una de salida.

En esta fase, se toman nuevas filas de la tabla de entrada, se intenta hacerlas coincidir con las entradas de la tabla de normas adaptadas y si hay una coincidencia, se modifica el dato y se añade a la tabla de salida.

Esta clasificación se realiza mediante 3 procesos:

- El de *trap* y *syslog* que examina la tabla intentando encontrar *trap* o mensajes *syslog* conocidos y si se corresponden, creará nuevos eventos en base a la regla a la que se refiere el tipo de *trap* o *syslog*.

- El evento *consolidator* cuyo criterio es más bien fijo. Toma como entrada la tabla de eventos y como salida la de alarmas. Examina cada evento nuevo y si puede encontrarlo en la interfaz referida y el tipo de evento genera una alarma, este activará la alarma correspondiente en la tabla de salida.

- El de alarma. Este se usa para lanzar los *triggers* que activan las acciones. Examina la tabla de *triggers* y la de alarmas para encontrar los *triggers* habilitados y las alarmas activadas. Es el responsable de generar emails cuando ocurre un evento o salta una alarma.

Además de los eventos *syslog* y SNMP, los procesos *pollers*, pueden directa o indirectamente crear eventos.

El método directo significa que el *poller* está usando la alarma suministrada para situaciones como el cambio de estado del interfaz.

El método indirecto es el analizador SLA (*Service Level Agreement*). Este examina las interfaces especificadas en los ficheros RRD e intenta ajustar los datos al criterio SLA. Si existe una coincidencia SLA, se crea un evento para esa interfaz. El *cron* finaliza al analizador SLA transcurridos 30 minutos.

Un esquema de todos estos pasos se muestra en la figura 3.3.

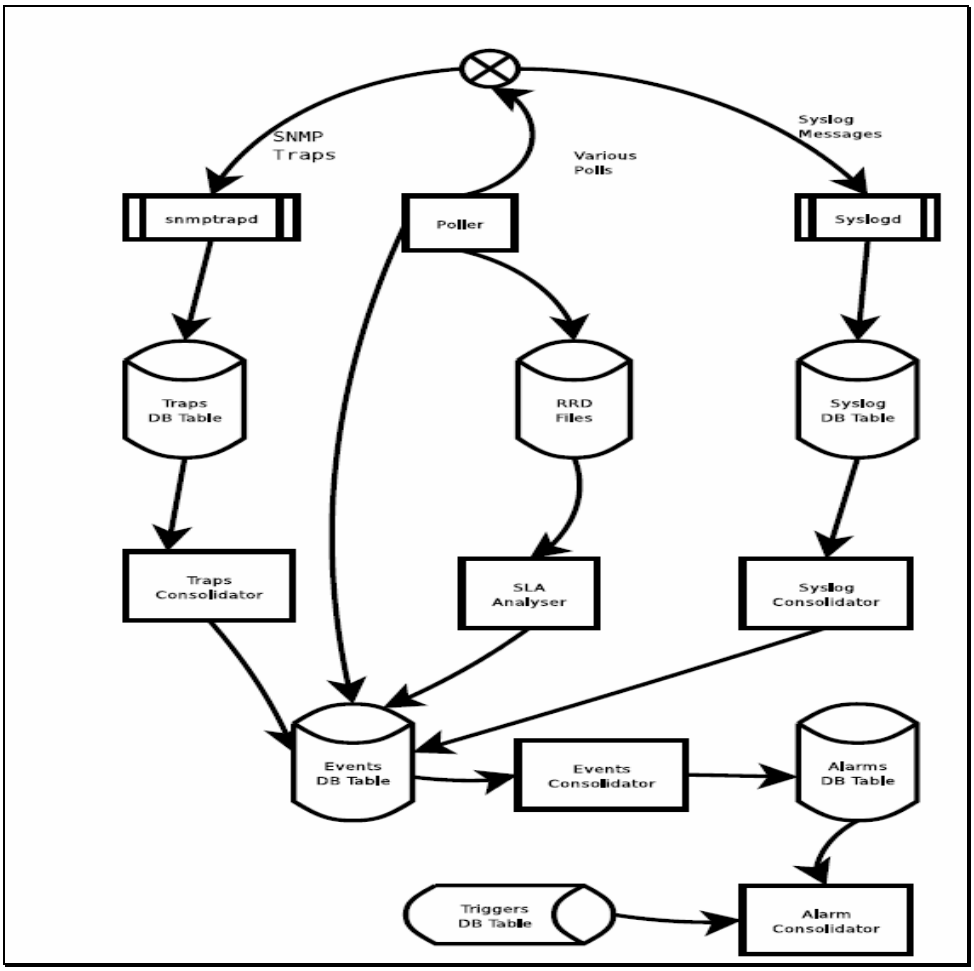


Figura 3.3 - Diagrama de flujo que siguen los datos en JFFNMS.

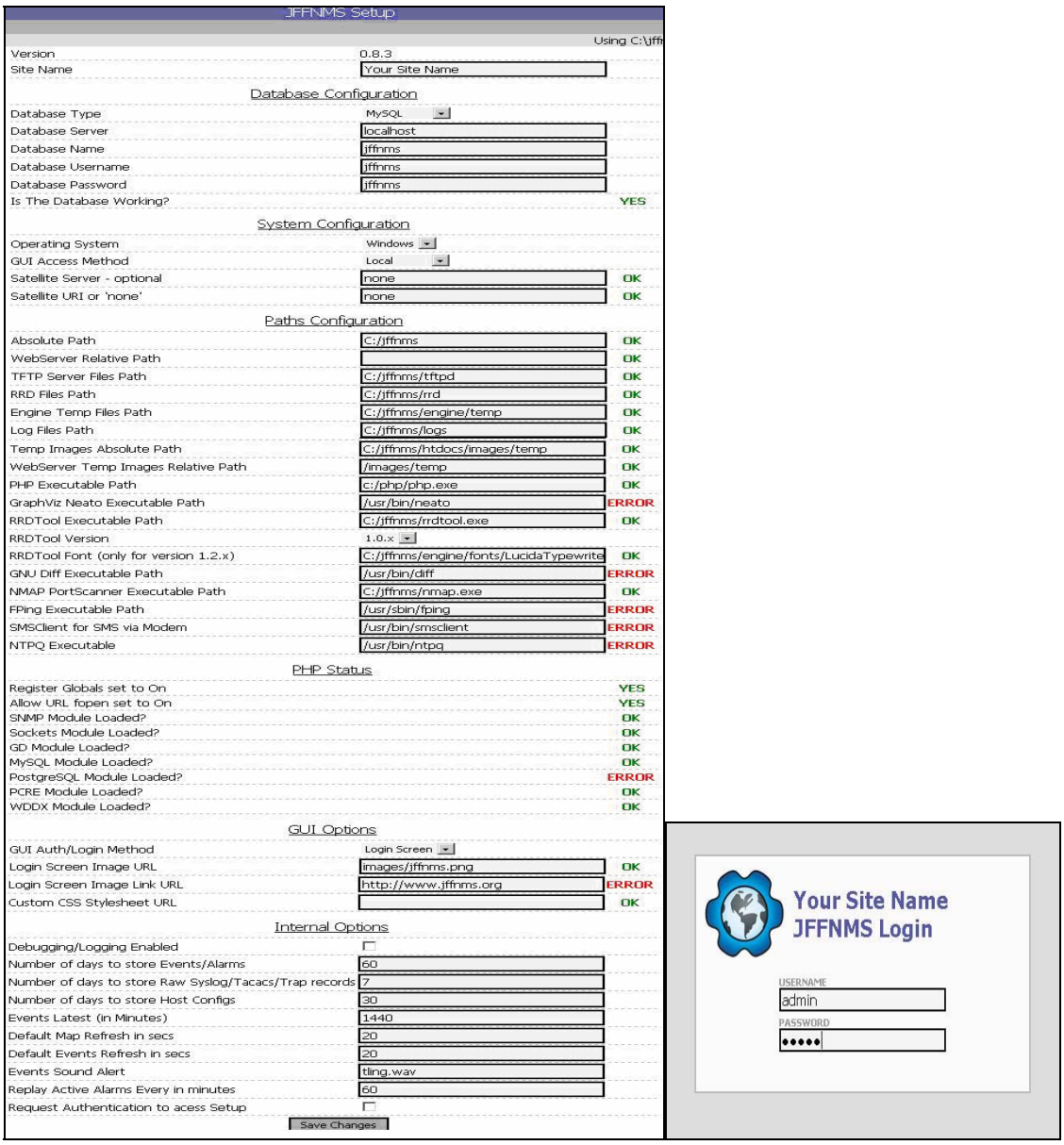
3.1.3 - INTERFAZ WEB

Una vez que se ha escrito la dirección correspondiente a la máquina JFFNMS en el navegador, se obtiene una pantalla parecida a la que aparece en la figura 3.4-a. Si se desea acceder como administrador, se debe proceder a registrarse pulsando la opción *Main*. Entonces aparecerá una pantalla que pedirá *login* y *password*, figura 3.4-b.

Con este tipo de acceso se pueden configurar todo tipo de acciones como las ya mencionadas anteriormente y además se pueden visualizar gráficas de los datos recogidos.

Por ejemplo se puede realizar:

- La monitorización de los servicios DHCP, DNS, FTP, WEB y E-MAIL.
- La monitorización de CPU, memoria RAM, memoria SWAP, procesos, cantidad de usuarios, tarjeta de red.
- La monitorización de la administración del sistema de gestión.



(a) **(b)**

Figura 3.4 - Interfaz web de JFFNMS: (a) Pantalla inicial; (b) Pantalla de registro.

De los informes de monitorización, una gestión clara donde se puede deducir si el equipo monitorizado tiene mucha carga de proceso, si su disco duro está por llenarse, si hay mucho tráfico de red en los dispositivos activos y tarjetas de red, etc.

3.2 - NETCRUNCH

Adrem es una compañía especializada en el software para administrar y monitorizar las redes corporativas. NetCrunch [7] es uno de sus desarrollos, el cual está especializado en las infraestructuras, el funcionamiento y la disponibilidad de los sistemas, optimizando la utilización de activos y reduciendo gastos indirectos de mantenimiento. NetCrunch es una

poderosa herramienta fácil de usar, que contiene diversas opciones para la monitorización de redes.

3.2.1 - COSTOS Y LICENCIAS

NetCrunch tiene licencia OEM (*Original Equipment Manufacturer*) lo que significa que el producto solo puede ser instalado en una estación de trabajo que se encargará de la monitorización de toda la red.

Entre los costes de dicha herramienta pueden citarse varios ejemplos. La versión NetCrunch 3x Premium con 1 año de asesoría profesional (12 meses de soporte vía telefónica) tiene un coste aproximado de $2390 USD. El paquete NetCrunch 3x Estándar con 1 año de asesoría profesional (12 meses de soporte vía telefónica) está valorado en $1690 USD. Las diferencias existentes entre las versiones de NetCrunch pueden visualizarse en la figura 3.5.

Una sola licencia de NetCrunch permite monitorizar alrededor de 1000 host, aunque también va a depender de la capacidad de la máquina donde fue instalado el programa.

3.2.2 - FUNCIONALIDAD DE NETCRUNCH

NetCrunch en su versión demo, maneja un cierto número de funcionalidades que permiten una evaluación del programa.

Feature	NetCrunch Standard	NetCrunch Premium
Network discovery		
Logical topology maps (i.e. subnets)	☑	☑
Routing topology maps	☑	☑
SNMP (v1, v2c)	☑	☑
Windows Domains & Workgroup views	☑	☑
Automatic discovery of newly found devices	☑	☑
Physical topology maps		☑
eDirectory Tree views		☑
Management		
300 precompiled SNMP mibs	☑	☑
Viewing SNMP parameters	☑	☑
Setting SNMP parameters		☑
MIB Compiler		☑
User profiles for Alerting, Reporting, and Web Access policies		☑
Web Access		
Built in Web server	☑	☑
ITools	☑	☑
Evert Log view	☑	☑
Generate/view reports	☑	☑
Enable SSL certificates for connection encryption		☑

Figura 3.5 - Comparativa de versiones NetCrunch.

Figura 3.6 - Atlas.

En monitorización de redes, puede realizar un descubrimiento automático de la red o las redes desde la estación de trabajo donde está instalado, hacer una representación gráfica con los distintos nodos y dispositivos encontrados, así como identificar a cada uno de ellos y a sus diferentes configuraciones, monitorizar diferentes protocolos o servicios, analizar el tráfico y limitarlo en ciertos nodos, etc. Todo esto se logra con una subfunción denominada Atlas, la cual viene integrada dentro del área de trabajo y cuya interfaz gráfica se puede observar en la figura 3.6.

Netcrunch cuenta con tres secciones principales:

1. *Ip Networks*: contiene un mapa de red y se puede observar en detalle los servicios que se están monitorizando, así como que dispositivo está activo o está desconectado.

2. *Custom View*: en esta opción se pueden configurar diferentes vistas de la red. Se puede personalizar de acuerdo a nuestras preferencias: colores, iconos de los dispositivos, nombres a los dispositivos, etc.

3. *Performance Charts*: Permite monitorizar el rendimiento de dispositivos conectados.

NetCrunch maneja una ventana emergente de eventos suscitados constantemente que a su vez, son alertas que el programa manda al administrador. Pueden ser configurables para enviarlos vía mail, pager (comúnmente conocido como busca), SMS, etc. También se pueden configurar acciones a tomar ante determinadas alertas que se presenten. NetCrunch trae consigo una lista de las posibles alertas que se pueden encontrar al monitorizar la red, pero también se pueden configurar nuevas acciones, figura 3.7, que el programa ejecutará ante las mismas alertas u otro tipo de alertas o eventos que ocurran.

NetCrunch tienen la opción de generar reportes diarios, tanto de nuestra red como de un nodo o un dispositivo determinado. La representación de estos reportes es gráfica y muestran en detalle, el tráfico de la red y el estado de los dispositivos (figura 3.8).

Tal como se muestra en la figura 3.9, NetCrunch maneja accesos vía Web para la monitorización remota y se configura automáticamente en el momento de la instalación del programa. En este entorno, se manejan las mismas opciones que en la estación de trabajo donde fue instalado. Se pueden generar distintos usuarios con distintos permisos para este tipo de acceso.

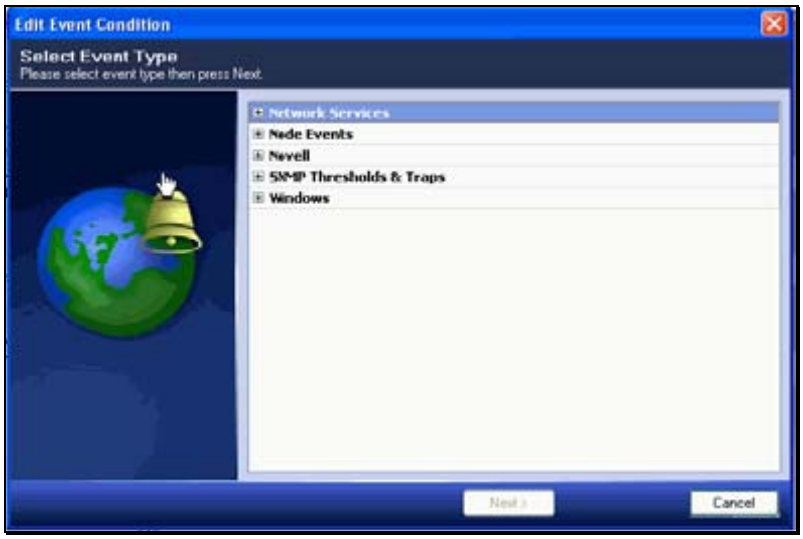


Figura 3.7 - Ventana de configuración de eventos.

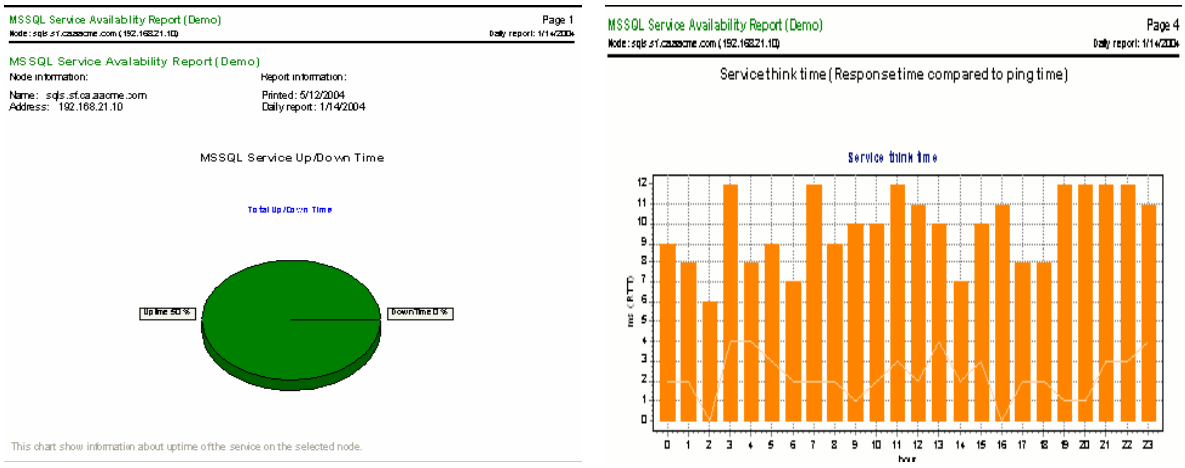


Figura 3.8 - Generación de reportes.

Figura 3.9 - Acceso vía Web.

3.3 - OPENNMS (*NMS: NETWORK MANAGEMENT SYSTEMS*)

Es un software [8] que sigue el modelo FCAPS, pero principalmente se pensó para cubrir la gestión de fallos y el rendimiento. Como ya se explicó en el apartado anterior, el objetivo de la gestión de fallos es reconocer, aislar, corregir y registrar averías que ocurren en las redes de telecomunicaciones. Además, utiliza análisis de tendencias para predecir

errores de tal manera que la red siempre está disponible. Esto puede ser establecido monitorizando cosas diferentes para un comportamiento anormal.

Cuando ocurre un fallo o un evento, frecuentemente un componente de la red, enviará una notificación al operador, utilizando un protocolo propietario o abierto como SNMP o al menos escribiendo un mensaje en su consola, para un servidor de consola, para capturar un *log* o página. Esta notificación se supone que se lanza automáticamente o mediante actividades manuales como por ejemplo, la recopilación de más datos para identificar la naturaleza y la gravedad del problema o poner el equipo en backup on-line. Los *log* de fallos, son entradas utilizadas para recopilar estadísticas que determinan el nivel de servicio proporcionado de elementos individuales de la red, así como de subredes o de la red entera. También se utilizan para determinar componentes de red, aparentemente frágiles, que necesitan más atención.

La gestión del rendimiento por su parte, permite al gestor preparar la red para el futuro y también determinar la eficiencia de la actual. El rendimiento de la red se puede medir con parámetros como el *throughput*, el porcentaje de utilización, las tasas de error y los tiempos de respuesta. Recolectando y analizando los datos de rendimiento, el estado de la red puede ser monitorizado.

Los umbrales de rendimiento pueden ser establecidos para lanzar una alarma. La alarma la manejará el proceso de gestión de fallos habitual. Las alarmas varían dependiendo de la severidad.

Para conocer mejor las características y el funcionamiento de esta herramienta, se pasan a comentar sus tres características principales.

3.3.1 - ESCANEO DE SERVICIOS

Se utiliza para saber si un servicio está o no disponible en la red. Los servicios disponibles por defecto son:

- ICMP
- SNMP
- FTP
- HTTP
- SMTP
- DNS
- Informix (TCP)
- Sysbase (TCP)
- MySQL
- Postgres
- Oracle (TCP)
- DHCP

- Citrix
- SSH
- SQLServer (TCP)
- Notes IIOP
- IMAP
- POP3

3.3.2 - RECOLECCIÓN DE DATOS MEDIANTE SNMP

Utiliza RRDtool para almacenar y representar los datos recogidos por SNMP. Se puede configurar el intervalo de peticiones, los datos a recoger y como almacenarlos. Se pueden hacer algunos informes con RRDtool dentro de OpenNMS.

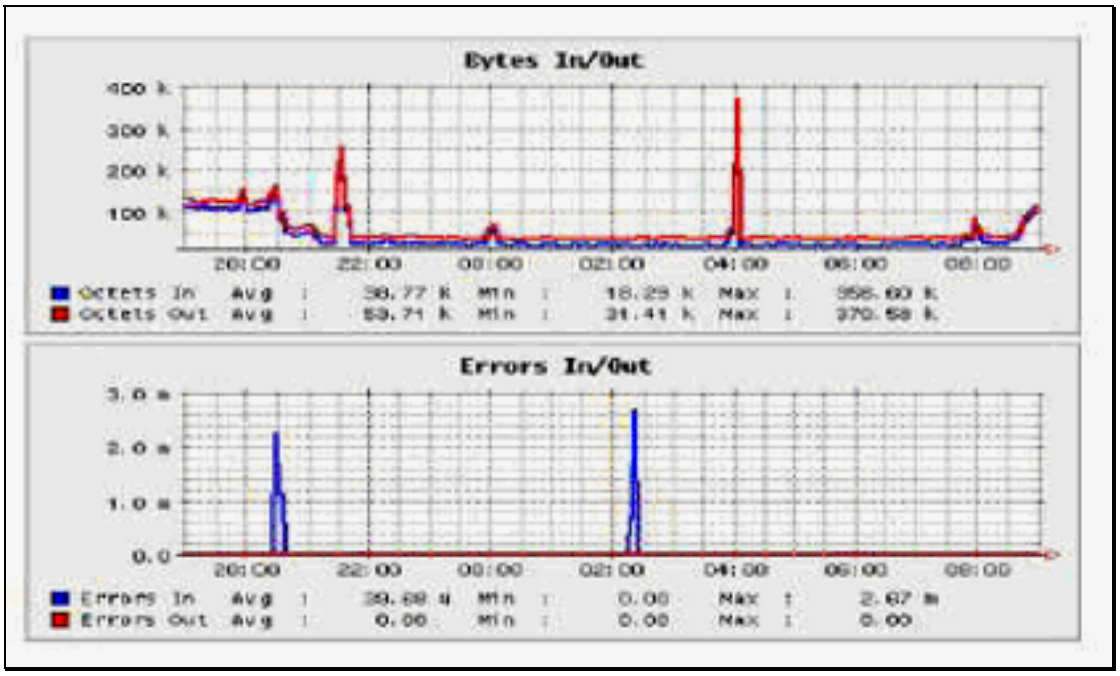


Figura 3.10 - Representación de datos con OPENNMS.

Por otro lado, utiliza MRTG (*Multi Router Traffic Grapher*), una herramienta escrita en C y Perl por Tobias Oetiker y Dave Rand, que se utiliza para supervisar la carga de tráfico de las interfaces de red. MRTG genera páginas HTML con gráficos que proveen una representación visual de este tráfico. También utiliza SNMP para recolectar los datos de tráfico de un determinado dispositivo, por tanto, es requisito contar con al menos un sistema con SNMP funcionando y correctamente configurado. SNMP manda peticiones con dos objetos identificadores (OID) al equipo. Una base de control de información (MIB) controla las especificaciones de los OID. Después de recoger la información la manda sin procesar mediante el protocolo SNMP. MTRG graba la información en un diario del cliente. El software crea un documento HTML de los diarios. Estos tienen una lista de gráficas detallando el tráfico del dispositivo. El software viene configurado para que se recopilen datos cada 5 minutos, pero el tiempo puede modificarse.

La aplicación de MRTG, consiste en una serie de *scripts* escritos en lenguaje PERL que usan el protocolo de red SNMP para leer los contadores de tráfico. Estos se encuentran ubicados en los conmutadores (*switch*) o los encaminadotes (*routers*) y mediante sencillos y rápidos programas escritos en lenguaje C, se crean imágenes en formato PNG, que representan el estado del tráfico de nuestra red. Estos gráficos, los inserta en una página web, que podemos consultar mediante cualquier navegador. Un ejemplo se muestra en la figura 3.10. Hemos de hacer notar que para poder ver los resultados, tenemos que tener instalado un servidor WEB en la máquina que tenga instalado el MRTG.

Como se puede observar en la tabla 3.1, la configuración para la recogida de datos de OpenNMS no es muy cómoda.

```
<group name = "signalQuality_DOWN" ifType = "128">
<mibObj oid=".1.3.6.1.2.1.10.127.1.1.4.1.2" instance="ifIndex"
alias="sigQUnerroreds_DN" type="counter"/>
<mibObj oid=".1.3.6.1.2.1.10.127.1.1.4.1.3" instance="ifIndex"
alias="sigQCorrecteds_DN" type="counter"/>
<mibObj oid=".1.3.6.1.2.1.10.127.1.1.4.1.4" instance="ifIndex"
alias="sigQUncorrect_DN" type="counter"/>
<mibObj oid=".1.3.6.1.2.1.10.127.1.1.4.1.5" instance="ifIndex"
alias="sigQSignalNoise_DN" type="integer"/>
<mibObj oid=".1.3.6.1.2.1.10.127.1.1.4.1.6" instance="ifIndex"
alias="sigQMicroreflec_DN" type="integer"/>
</group>
```

Tabla 3.1 - Ejemplo de configuración para la recogida de datos.

3.3.3 - GESTIÓN DE EVENTOS Y NOTIFICACIONES

A continuación se muestra en la tabla 3.2 los procesos concurrentes de gestión que permiten realizar las tareas a OpenNMS.

Entre las características de OpenNMS para los eventos y las notificaciones se encuentran las siguientes:

- No tiene mapas.
- Puede recibir eventos internos o externos y se pueden construir acciones para que se ejecuten en la recepción de un evento.
- Los eventos pueden producir notificaciones.
- Las notificaciones andan "un camino de destino" o *path*, que debe ser el correcto para asegurar que las alarmas alcanzan a la gente apropiada.

En las figura 3.11 y 3.12 se muestran ejemplos de la compleja configuración de eventos y umbrales.

Proceso concurrente	**Nombre de demonio**	**Descripción**
Demonio acción	actiond	Facilidad de autoejecución de una acción cuando ocurre un evento.
Demonio de recogida	collectd	Recoge datos desde los nodos administrados.
Demonio de capacidad	capsd	Permite el chequeo de capacidad en los nodos descubiertos para comprobar los servicios soportados.
Demonio de DHCP	dhcpd	Cliente DHCP para OpenNMS.
Demonio de descubrimiento	discovery	Realiza un descubrimiento de nodos de la red.
Demonio de eventos de administrador	eventd	Administra y almacena eventos originados por los procesos concurrentes.
Demonio de notificación	notifd	Realiza notificaciones externas a usuarios.
Demonio de recogida de información	outaged	Consolida eventos para proveer de un historial a cada nodo o servicio administrado.
Demonio de consulta	pollerd	Realiza un sondeo que comprueba regularmente el estado operacional de los nodos y servicios.
Demonio de RTC de administrador	rtcd	Recoge datos en tiempo real para conseguir información útil para categorías de usuarios definidos de nodos y servicios administrados.
Demonio de trap SNMP	trapd	Se encarga de recoger los traps SNMP.
Demonio de servicio de umbral	threshd	Monitoriza nodos y servicios administrados basándose en los valores de los atributos y dentro de los umbrales especificados.

Tabla 3.2 - Lista de procesos de OpenNMS.

```
<event>
<mask>
<maskelement>
<mename>id</mename>
<mevalue>.1.3.6.1.4.1.9.9.33.2</mevalue>
</maskelement>
<maskelement>
<mename>generic</mename>
<mevalue>6</mevalue>
</maskelement>
<maskelement>
<mename>specific</mename>
<mevalue>2</mevalue>
</maskelement>
</mask>
<uei>uei.opennms.org/vendor/Cisco/traps/cipCs
naLlc2ConnectionLimitExceeded</uei>
<event-label>CISCO-CIPCSNA-MIB defined
trap event:
cipCsnaLlc2ConnectionLimitExceeded</event-
label>
<descr>This trap indicates that a connection .....
</descr>
<logmsg dest='logndisplay'>&lt;p&gt;Cisco
Event: Connection Limit
Exceeded.&lt;/p&gt;</logmsg>
<severity>Warning</severity>
</event>
```

Figura 3.11 - Ejemplo de configuración de un evento.

```
<threshold type="high"
dsname="
cpuPercentBusy" ds-
type="node"
value="90" rearm="50"
trigger="3"/>
```

Figura 3.12 - Ejemplo de configuración de umbral.

3.4 - ZABBIX

Zabbix fue creado por Alexei Vladishev y actualmente es desarrollado y soportado por ZABBIX SIA. Se trata de una solución software de código abierto que permite la monitorización distribuida de una red.

Monitoriza numerosos parámetros asegurando la integridad y la salud de los servidores. Utiliza un mecanismo flexible de monitorización, que permite a los usuarios configurar el envío de email o sms cuando salta una determinada alerta provocada por un evento. Esto permite una rápida reacción para resolver los problemas que puedan ocurrir en los distintos dispositivos que conforman la red.

ZABBIX permite visualizar fácilmente los datos recogidos y almacenados con varias herramientas. Soporta tanto *polling* como *trapping* y además ofrece una interfaz web basada en frontend**,** parte software que interactúa con el usuario, que permite el acceso y la modificación de configuración, estadísticas e informes. También la utilización de este

interfaz permite visualizar mapas de la red que muestran en todo momento el estado de todos los componentes de la misma y permite localizar los errores que se puedan dar en un determinado instante.

Apropiadamente configurado, ZABBIX puede jugar un importante papel en la monitorización de infraestructuras de telecomunicaciones, tanto para pequeños organismos con unos cuantos servidores, como para grandes compañías con multitud de estos.

ZABBIX es software libre y por tanto, es gratis. Está escrito y distribuido bajo GPL. ZABBIX Company se encarga de la libre distribución y el código fuente está disponible para todo el público en general.

3.4.1 - CARACTERÍSTICAS PRINCIPALES

Se pueden resaltar las siguientes:

- Auto-descubrimiento de servidores y dispositivos de red. El módulo de monitorización distribuida de ZABBIX, permite encontrar en un rango de IP sistemas que soporten unos determinados servicios o que tengan instalados agentes ZABBIX o SNMP.

- Monitorización distribuida con administración WEB centralizada. Este tipo de monitorización se consigue mediante la habilitación de distintos gestores ZABBIX en distintos nodos de la red; sin embargo, es más sencilla la configuración centralizada que permite fácilmente la configuración de todos los nodos de la red desde un único gestor (una sola localización).

  La supervisión distribuida de ZABBIX está pensada para ambientes complejos que se encuentran en ubicaciones diferentes. ZABBIX puede supervisar un número ilimitado de nodos.

- Soporta mecanismos de *polling* y *trapping*.

- Ofrece software para gestor en distintas plataformas: Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X.

- Presenta agentes de alto rendimiento para distintos sistemas: Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X, Tru64/OSF1, Windows NT4.0, Windows 2000, Windows 2003, Windows XP, Windows Vista.

- Monitorización de dispositivos sin necesidad de agentes.

- Autentificación de usuarios para asegurar la integridad del sistema. En la figura 3.13 se muestra la pantalla de inicio de acceso al sistema ZABBIX. El primer paso es registrarse y según el tipo de usuario se podrá o no modificar los datos del sistema o visualizar información importante.

- Interfaz web. Permite el acceso al sistema tanto para la visualización de datos como para la configuración (figuras 3.13 y 3.14).

- Monitorización de aplicaciones WEB. El módulo de monitorización WEB permite la supervisión flexible y fácil de disponibilidad y funcionamiento de sitios y aplicaciones WEB. Utiliza el intercambio de GET y POST y soporta tanto HTTP como HTTPS. La información recogida será el tiempo de respuesta, la velocidad de bajada y el código de respuesta de la aplicación. También se pueden programar alarmas o notificaciones sobre estos datos.

- Monitorización del rendimiento. Es una de las características más importantes. Puede monitorizar numerosos parámetros del sistema tan importantes como: carga de procesador, número de procesos activos, número total de procesos, utilización de disco, espacio *swap* y espacio en memoria. Todos los valores de los parámetros monitorizados son almacenados en una base de datos. En la figura 3.14 se visualizan algunos de estos parámetros.

- Comprobación de integridad. Es capaz de supervisar la integridad de un servidor: todos sus archivos de configuración críticos, binarios, escrituras, las páginas HTML,... si estos sufren modificaciones el administrador es alertado.

- Notificación mediante email de eventos predefinidos. Para que las alarmas que se configuran sean efectivas, se debe disponer de un mecanismo de notificación poderoso que avise a los administradores. Con ZABBIX, un administrador puede definir mediante expresiones flexibles, prácticamente cualquier condición posible para una alerta. En cualquier momento, estas expresiones pueden cumplirse y entonces, una alarma será enviada por correo electrónico a cualquier dirección definida por el administrador. Se pueden utilizar programas externos para métodos de notificación definidos por el usuario como SMS, notificaciones telefónicas, etc. Ofrece además, soporte para mensajería instantánea Jabber.

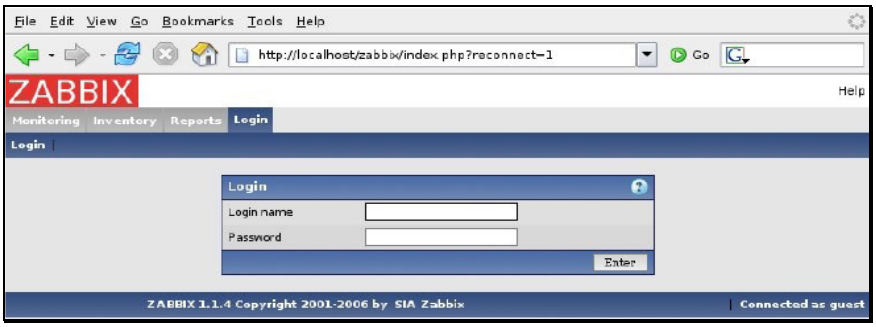


Figura 3.13 - Acceso Web de Zabbix.

ZABBIX también puede predecir el comportamiento futuro de parámetros supervisados utilizando el *Least Square Algorithm*. Esto permite al usuario ser notificado, antes de que el estado del sistema alcance el nivel crítico.

- Soporte eléctrico. Utiliza diferentes expresiones en *triggers* para detectar las máquinas activas o inactivas (ON/OFF).

- Servicio de almacenamiento de información recogida mediante *log*. ZABBIX puede utilizarse para la supervisión centralizada de este tipo de ficheros. El parámetro *LogFileSize* controla el tamaño del fichero tanto en el servidor como en el agente.

- Planificación de capacidad. La posibilidad de visualizar algunas tendencias como la carga de proceso, el uso de disco, la actividad de base de datos u otra métrica importante, permite a un administrador de sistema decidir claramente cuando se necesita una mejora del hardware.

- Alto nivel de recursos y servicios de información tecnológica. Permite la posibilidad de crear dependencias entre servicios, de forma que si un servicio falla se evite comprobar el estado de aquellos que dependen de él.

- Solución de código abierto.

- XML importación/exportación de datos. Esta nueva funcionalidad permite compartir plantillas, configuración de máquinas y parámetros o *triggers* de información relacionada.

- Bajo coste de propiedad.

- Sistema de monitorización centralizada. Toda la información (configuración, datos de funcionamiento) es almacenada en una base de datos relacional.

- Procedimiento programado de la recogida de datos. Permite programar cada cuanto tiempo se borran los datos recogidos.

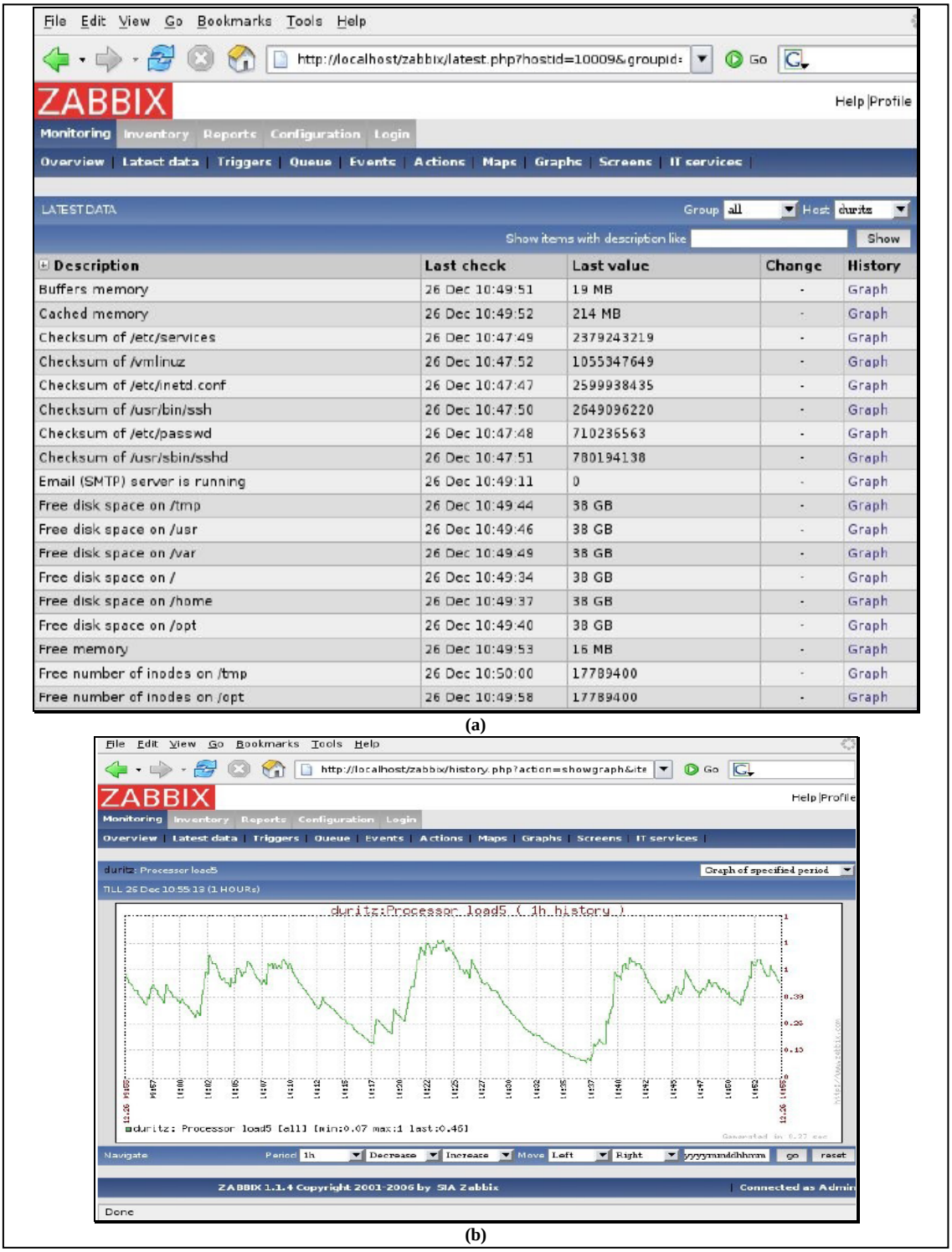


Figura 3.14 - Monitorización de datos mediante interfaz Web: (a) directamente los valores; (b) gráficamente.

– Supervisión de SLA: está capacitado para monitorizar el SLA (Service *Level Agreements*).Esto también guarda *SLArelated*, un historial de datos que ayuda a identificar y mejorar las zonas críticas de una infraestructura de telecomunicaciones.

- Soporte SQlite. SQlite es un sistema de gestión de bases de datos relacional compatible con ACID, y que está contenida en una relativamente pequeña (~500KB) biblioteca en C. A diferencia de los sistemas de gestión de base de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos) son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción. Esta propiedad permite utilizar Zabbix en ambientes integrados.

- Filtración de *log* en el lado del agente.

- Alto nivel de recursos de visualización. Existen distintas opciones a la hora de visualizar la información:

  - Gráficas superpuestas de distintos parámetros de una o varias máquinas.
  - Distintos colores para gráficos.
  - Visualización de componentes de la red mediante mapas (figura 3.15).
  - Opción de insertar imágenes de fondo con un tamaño que oscila entre 1.5 y 2 MB y que se puede ajustar en php.ini.
  - Asignación de iconos cuando las máquinas tienen un estado desconocido. Estos pueden visualizarse en mapas.
  - Uso de diapositivas: varias pantallas pueden ser agrupadas en una exposición de diapositivas para conseguir mejor presentación.
  - Distintos colores para los distintos niveles de criticidad de eventos.

- Fácil de utilizar y de configurar. Ofrece:

  - Acoplamiento de plantilla " muchos a muchos": La mayor flexibilidad en el acoplamiento máquina-plantilla hace ahorrar tiempo y hace que la configuración de máquinas sea más flexible y simple.
  - Nuevas plantillas por defecto: El programa ofrece plantillas que nos ahorran tener que hacer las nuestras.

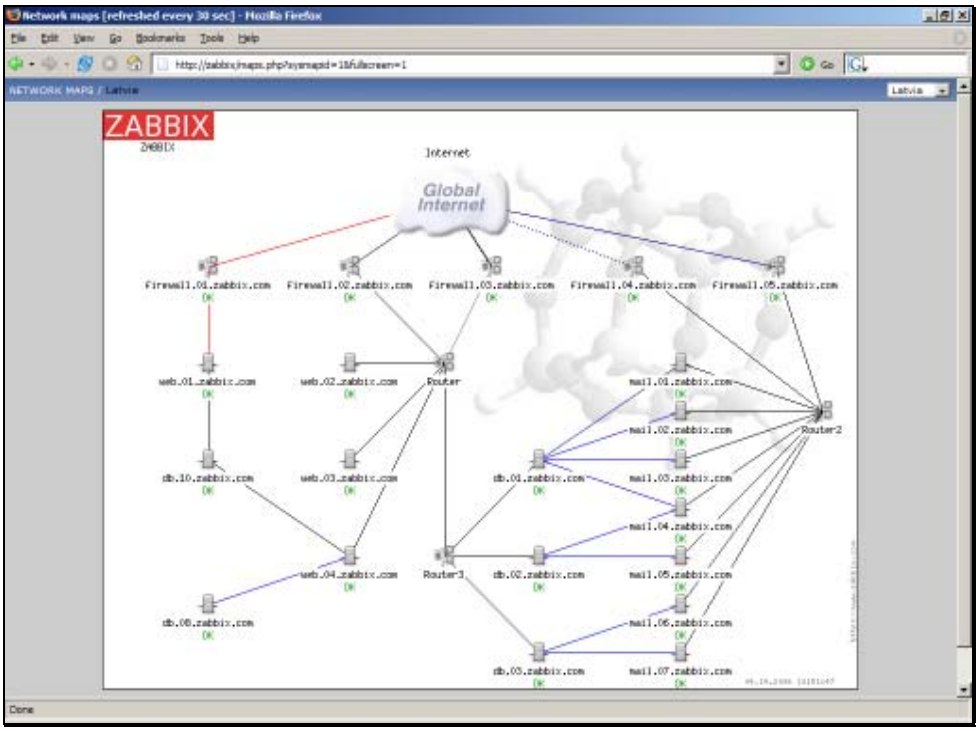


Figura 3.15 - Ejemplo de mapa de red visualizado.

- Soporte de enlace estático (link). Enlace estático entre agente y servidor binario.

- Intervalos de refresco flexibles. Apoya diferentes intervalos de refresco según el parámetro a controlar.

- Comprobaciones externas del lado del servidor. Esto puede ser usado para introducir comprobaciones de encargo, ejecutadas sobre el lado del servidor ZABBIX.

- El servidor ZABBIX puede extender la carga de trabajo a través de varios servidores: los grupos de procesos ofrecidos por el servidor (descubrimientos, poller, HTTP poller, trapper, etc.), pueden ser repartidos sobre distintos servidores físicos para mejorar el funcionamiento y la disponibilidad.

- Fácil instalación: Instalación Wizard. Este tipo de instalación chequea automáticamente prerrequisitos, conecta con la base de datos y genera un fichero de configuración para el WEB front-end.

Se ha implementado el filtrado de *log* por medio de expresiones regulares que hacen más eficiente la monitorización de los ficheros de este tipo mediante estilo Posix. Este estilo pertenece a una familia de estándares de llamadas al sistema operativo, cuya finalidad es generalizar las interfaces de los sistemas operativos para que una misma aplicación pueda ejecutarse en distintas plataformas.

- *Perro guardián* de base de datos. El servidor de ZABBIX advierte automáticamente al grupo de usuarios si la base de datos cae y continúa con las operaciones normales mientras se recupera. Puesto en práctica sólo para MySQL.
- Acciones más flexibles. Soporta múltiples operaciones por acción (notificaciones, la ejecución de escritura) y se puede introducir un algoritmo de cálculo de algún parámetro.

3.5 - ELECCIÓN DE ZABBIX COMO SISTEMA A UTILIZAR

Al comparar Zabbix con los distintos sistemas presentados, no se duda en elegirlo como mejor opción. Frente a NetCrunch presenta la ventaja de ser gratis y respecto a OpenNMS, Zabbix es mucho más sencillo de manejar y representa la información de tal forma que es más fácil de interpretar. Con el que se encuentra un menor número de diferencias es con JFFNMS, pero una serie de ventajas que se pasan a comentar, hicieron que se optara por utilizar Zabbix en este proyecto de gestión de redes. Zabbix respecto a JFFNMS ofrece:

- Mayor documentación y soporte mediante sus foros.
- Ejecución desde el gestor de comandos remotos en algunos agentes.
- Instalación en un mayor número de plataformas.
- Monitorización sin necesidad de agentes.
- Un mayor número de opciones para alertar.
- Configuración distribuida para la posible ampliación de la red.

Además de los aspectos comentados, hay que añadir las ventajas de utilizar Zabbix en el escenario real. La finalidad de este proyecto de gestión, es ofrecer una interfaz gráfica, fácil de entender por trabajadores sin formación alguna en el campo de las redes de telecomunicaciones. Para esta acción, Zabbix es el sistema que ofrece un mayor número de opciones con mapas, muestra unos iconos cuando todo funciona correctamente, iconos distintos cuando no, representación gráfica de datos, distintos colores según el nivel de gravedad del evento... De fácil instalación y sin suponer coste de software gracias a su licencia GPL, es un sistema fácil de instalar y configurar lo que permite que un trabajador de la empresa pueda ponerlo en funcionamiento con el respectivo ahorro que conlleva para la empresa. Asimismo, ofrece la opción de crear grupos de usuarios con sus correspondientes permisos, lo cual permite desarrollar la jerarquía existente en RTVA; los

trabajadores del primer nivel, sólo interpretarán que se dan errores a través de las distintas pantallas y diapositivas, pero no podrán acceder a todos los datos del sistema, ni modificar ninguna opción. Su cometido será el de alertar al técnico de nivel superior, que son los operadores de red. Dentro de este grupo, existirán distintas cuentas con diferentes permisos y el encargado del sistema, dispondrá de la de super-administrador, que además de visualizarlo todo, podrá configurar tanto permisos como todo lo demás a través de la interfaz web.

Si estas ventajas las unimos a todas las funciones explicadas anteriormente, se llega a la conclusión de que este es el sistema que mejor se adapta a las necesidades del proyecto.