

Portable Device for Easy Management and Automatic Recovery of Networking Systems

J. Morillo, and T. Mateo

Abstract—Configure communication equipment is a critical task that typically requires handling advanced concepts aimed at managing and troubleshooting networking issues, thus demanding high analytical and problem-solving skills to ICT technicians. To reduce this gap, this paper presents a portable hardware and software solution devoted to unify the administration of operating systems, configuration files and network services in communication devices. This includes the automatic recovery of firmware under failure conditions concerning the loss of boot image, corruption of Flash memory and forgotten passwords, among others. The prototype, mainly tested on routers and switches from Cisco and MikroTik, can be also applied to other equipment such as APs, firewalls or embedded devices since it looks for universality and independence from devices and manufacturers. As advantage, this facilitates the easy maintenance and quick deployment of network infrastructures, thus saving cost and time to engineers. So the experimentation carried out shows the significant advantage concerning the time taken to perform some automated functions rather than running manually. This stands for a powerful tool that, as a result, has been successfully patented and extended through an international cooperation treaty (PCT).

Index Terms—Automatic recovery, Cisco, Communication equipment, IOS, Raspberry Pi, Networking.

I. INTRODUCCIÓN

LOS equipos de comunicación utilizados para proporcionar conectividad a redes de datos, tales como conmutadores o enrutadores, poseen programas propios de instrucciones de máquina (i.e., firmware) creados para gestionar los servicios de la capa física, enlace y/o red del modelo de comunicaciones estándar OSI [1]. Las funciones —dependiendo del equipo donde se realicen— incluyen la activación de puertos de comunicación, tablas de enrutamiento, configuración de redes de área virtual (VPN), listas de control de acceso (ACL), traducción de direcciones de red (NAT) o configuración dinámica de clientes (DHCP), entre otras [2]. La información relacionada con estas funciones se encuentra almacenada habitualmente en forma de archivos ubicados en memorias no volátiles que, junto al firmware ubicado en la unidad de

almacenamiento físico interna del equipo, forman el sistema operativo (SO) y el perfil que se cargan durante el proceso de arranque de dicho equipo. En ocasiones, el sistema de arranque o los archivos de configuración se ven comprometidos por un funcionamiento defectuoso del equipo. En estos casos, la mayoría de fabricantes proporciona procedimientos manuales de recuperación, los cuales requieren un conocimiento profesional sobre los comandos de línea y conexiones necesarias a emplear [3].

Para dar solución al problema, en este artículo se propone DRACSC, acrónimo de dispositivo de recuperación automática y configuración de sistemas de comunicación. Esta solución, que surge como parte de un proyecto educativo ampliamente descrito en [4]-[8], ha sido probada en un laboratorio de redes de la Universidad de Huelva (España). Su cometido ha sido facilitar la gestión de un laboratorio que permita realizar prácticas a distancia con hardware real. De forma resumida, las contribuciones que persigue este trabajo son las siguientes: (i) presentar una solución novedosa para la administración centralizada de equipos de comunicación; (ii) examinar su aplicabilidad y ventajas para gestionar no solo configuraciones de equipo sino también servicios de red; y (iii) proporcionar a ingenieros de red una herramienta que simplifique y acelere la gestión de infraestructuras de datos. Para ello, se ha desarrollado un sistema hardware y software completo, portable y universal basado en Raspberry Pi 2 y aplicaciones web. Dicha solución ha sido patentada [9] y posteriormente extendida mediante tratado de cooperación internacional de patentes [10].

A continuación, el artículo está estructurado como sigue. En la Sección II se presenta el estado del arte respecto a soluciones software y hardware para la gestión de dispositivos de comunicación. La Sección III describe la arquitectura completa del sistema. La Sección IV explica la experimentación llevada a cabo y analiza el impacto de la solución sobre la gestión de equipos y servicios de red. Finalmente, el artículo aporta los resultados alcanzados y una discusión sobre trabajos futuros.

II. ESTADO DEL ARTE

La gestión o recuperación de equipos de comunicación se ha resuelto mediante soluciones divididas en aplicaciones software y equipos hardware [11]-[12]. Por un lado, las aplicaciones software contienen habitualmente una interfaz gráfica de usuario (GUI) para administrar remotamente los equipos de comunicación. La GUI reduce tiempo de gestión y facilita al usuario la ejecución de tareas básicas sin necesidad de conocer el repertorio completo de comandos. Si bien estas aplicaciones son ampliamente utilizadas, presentan el inconveniente de que

J.D. Morillo Reina, Universidad de Huelva, Dpto. Ingeniería Electrónica, Sistemas Informáticos y Automática, Ctra. Huelva-La Rábida S/N, 21819 Palos de la Frontera, Huelva, España, morreijuan@gmail.com.

T.J. Mateo Sanguino, Universidad de Huelva, Dpto. Ingeniería Electrónica, Sistemas Informáticos y Automática, Ctra. Huelva-La Rábida S/N, 21819 Palos de la Frontera, Huelva, España, tomas.mateo@diesia.uhu.es.

suelen percibirse por parte del usuario como limitantes del control que se tiene sobre el equipo debido a las menores posibilidades de configuración respecto a una interfaz basada en línea de comandos. Por otro lado, los equipos hardware — conocidos también como terminales o servidores de consola— son capaces de comunicarse con conmutadores, enrutadores y otros dispositivos a través del puerto de consola. Aunque con similar potencial de configuración, dichos equipos tienen el inconveniente de su elevado coste.

El sistema propuesto, además de combinar las facilidades de los sistemas hardware y software del estado del arte, aporta dos contribuciones principales como se describe a lo largo del artículo: portabilidad y estructura basada en plantillas para crear funciones MACRO de alto nivel.

A. Software para Gestión de Dispositivos de Red

Dentro de este tipo de soluciones se pueden citar Cisco Web Browser User Interface, una interfaz web suministrada con los propios conmutadores y enrutadores de la compañía que requiere una configuración adicional para trabajar con ellos [2]; Linksys Smart Wi-Fi, un conjunto de herramientas con funciones similares a Cisco Connect Express y Cisco Connect Cloud para administrar redes domésticas formadas por puntos de acceso y enrutadores Wi-Fi de forma remota y sencilla [11]; Colibri NetManager, antiguamente TeldaGES de la empresa Teldat, es una plataforma de gestión de enrutadores que aúna la auditoría, visión de red, acceso a equipos, recuperación de históricos de configuraciones y firmware a través de la nube [13]; Network Configuration Manager de la empresa ManageEngine, anteriormente llamado DeviceExpert, se trata de una solución muy potente que centraliza la administración, configuración, automatización de tareas y monitorización de elementos de una red (e.g., conmutadores, enrutadores, cortafuegos, etc.) desde una interfaz web [14]; y Network

Configuration Manager de la empresa SolarWinds, un referente en este campo que aúna la monitorización y configuración de elementos de red [15].

Con objeto de comparar las capacidades y características de las soluciones anteriores respecto al DRACSC se presenta la Tabla I. Mientras el software de Cisco o Teldat son solo compatibles con sus marcas, tanto el de ManageEngine como el de SolarWinds han sido diseñados para ser compatibles con el mayor número posible de equipos (e.g., Cisco, Juniper, HP, Dell, Brocade, F5, Aruba, Ruckus, etc.). Ello supone una ventaja sobre los demás. Respecto a las funciones, todos los sistemas se rigen por lo que puede llamarse como “camino lógico” dirigido a simplificar la administración de elementos de red, compartiendo por tanto características similares. Al respecto, en la tabla se ha establecido el nivel medio cuando el dispositivo implementa automatización de funciones y alto cuando además incluye monitorización de red o copias de seguridad entre otros. Respecto al coste, cabe mencionar que las licencias de ManageEngine se otorgan por dispositivo administrado y el paquete básico de 25 dispositivos supone \$1995 + \$399 de soporte al año. De forma similar, la adquisición del producto básico de SolarWinds comienza desde 2320 €. Para finalizar, se puede concluir que además de otras ventajas, solo el DRACSC puede manejar dispositivos de red a través del puerto de consola.

B. Hardware para Gestión de Dispositivos de Red

Respecto a las soluciones hardware que funcionan a modo de terminales o servidores de consola podemos encontrar Cisco Terminal Server [16], el cual puede adquirirse a partir de 2500 € para el modelo 2610XM-16TS. Entre otras se citan IOLAN SCS48 DAC, un dispositivo del fabricante Perle con precio alrededor de 4540 € que permite administrar distintos equipos utilizando conexión por consola, Telnet y SSH [17]; Dominion®

TABLA I
CARACTERÍSTICAS DE SOLUCIONES SOFTWARE PARA GESTIÓN DE DISPOSITIVOS DE RED

Nombre	Gestión de equipos en paralelo	Fabricantes admitidos	Grado de funcionalidad	Control por consola	Control por Telnet/SSH	Función de recuperación del sistema	Coste
CISCO Web Browser User Interface	Uno	Cisco	Medio	No	No	No	Gratuito
Linksys Smart Wi-Fi	Varios	Cisco	Medio	No	No	No	Gratuito
Colibri NetManager	Muchos	Teldat	Alto	No	✓	✓	-
Solarwinds Network Configuration Manager	Muchos	Muchos	Alto	No	✓	No	Alto
Network Configuration Manager	Muchos	Muchos	Alto	No	✓	No	Alto
DRACSC	Varios	Varios	Medio	✓	✓	✓	-

TABLA II
CARACTERÍSTICAS DE SOLUCIONES HARDWARE PARA GESTIÓN DE DISPOSITIVOS DE RED

Nombre	Gestión de equipos en paralelo	Fabricantes admitidos	Portabilidad	Grado de funcionalidad	Control por consola	Control por Telnet/SSH	Función de recuperación del sistema	Coste
CISCO Terminal Server	Varios	Cisco	No	Media	✓	No	No	Alto
IOLAN SCS48 DAC	Varios	Varios	No	Media	✓	✓	No	Alto
Dominion® SX	Varios	Varios	No	Alta	✓	No	No	Alto
Opengear IM7200	Varios	Varios	No	Alta	✓	✓	No	Alto
Raspisco	Uno	Varios	✓	Baja	✓	No	No	Gratuito
Servidor de consola basado en Raspberry Pi	Varios	Varios	No	Baja	✓	No	No	Gratuito
DRACSC	Varios	Varios	✓	Media	✓	✓	✓	Bajo

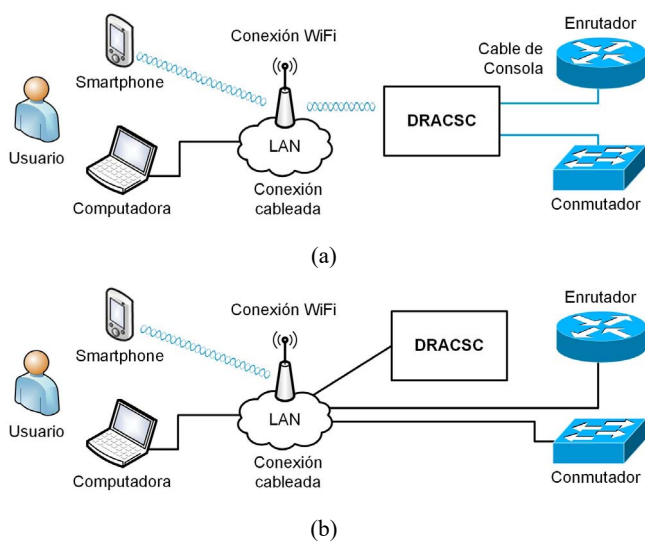


Fig. 1. Interacción del sistema DRACSC: a) dispositivos gestionados sin conexión Telnet/SSH y b) dispositivos gestionados con conexión Telnet/SSH.

SX, un dispositivo de Raritan similar al anterior que proporciona acceso, monitorización y control a través del puerto serie [18]; Opengear IM7200, un dispositivo más enfocado a servidor de consola que ofrece mayor número de funciones como cortafuegos, servidor DHCP, VPN, etc. con administración Telnet/SSH sobre puerto serie [19]. Con el objetivo de reducir el coste, nacen algunas soluciones que tratan de emular dicho hardware como Raspisco, un servidor de consola basado en Raspberry Pi, modem 3G, adaptador SerialToUSB y cable de consola para administración remota o local de dispositivos mediante concentrador USB y adaptadores [20]. También en este sentido, pero enfocado al ámbito docente, se usa Raspberry Pi como dispositivo de bajo coste que emula un servidor de consola por software [21].

La Tabla 2 presenta una comparativa de las capacidades y características de las soluciones anteriores respecto a la basada en DRACSC. Cabe mencionar que Dominion[®] SX es capaz de interactuar con distintos fabricantes como HP, Dell, Cisco o IBM. Por otro lado, salvo Raspisco que es un prototipo aún en desarrollo, la mayoría de soluciones se comunican a través del servicio Shell-in-a-box. Esto permite encapsular cualquier comunicación a través de un túnel SSH y tener acceso a cualquier dispositivo. Respecto a la solución DRACSC, cabe destacar la portabilidad y capacidad de recuperación de desastres más allá de la simple restauración de la configuración. Otra de las características que dispone el DRACSC es el uso de un espacio de almacenamiento para guardar datos de conmutadores y enrutadores cuyo protocolo de intercambio de ficheros es TFTP (RFC1350) y FTP (RFC1123).

III. ARQUITECTURA DEL SISTEMA

El escenario de funcionamiento habitual del sistema DRACSC concurre dentro de una LAN. Sin embargo, dado que dispone de un servidor de aplicaciones web al que el usuario se conecta mediante un navegador, podría extenderse a la WAN a través de NAT. Existen dos modos de funcionamiento dependiendo del estado en el que se encuentre el dispositivo a

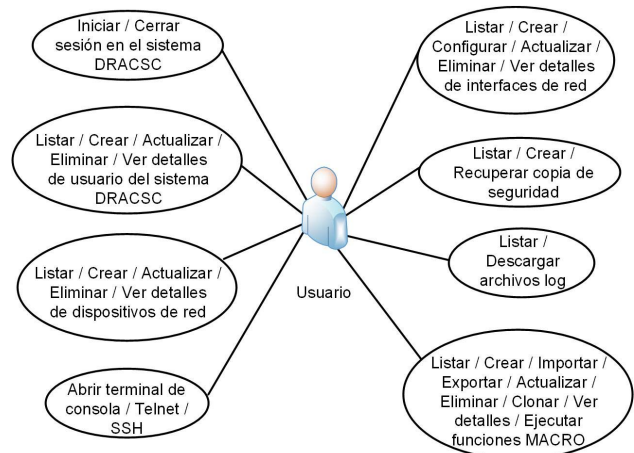


Fig. 2. Diagrama de casos de uso del sistema DRACSC.

gestionar. Esto es, dispositivos sin configuración Telnet/SSH o con ella (Fig. 1). En el primer caso, el conmutador o enrutador no puede comunicarse con el DRACSC a través de la red TCP/IP. Por tanto, no existe otra forma de trabajar con él más que conectado directamente a través del puerto de consola. Para acciones puntuales —como cargar una imagen IOS cuando el dispositivo se encuentra en modo ROMMON— se requiere configurar una red adicional. El segundo caso sería el escenario ideal ya que, al disponer el sistema DRACSC de conexión a la LAN —cableada o inalámbrica— no sería necesario portarlo. Así, podría permanecer conectado a una fuente de alimentación eléctrica y ampliar la autonomía de la batería interna. Con objeto de abundar en las acciones que pueden llevarse a cabo con el sistema DRACSC incluidas en la Tabla I y II, se incluye un diagrama de casos de uso cuyas funciones más importantes son explicadas a lo largo de las distintas secciones del artículo (Fig. 2).

A. Implementación Hardware y Software

Con objeto de alcanzar el mínimo coste posible, el sistema desarrollado ha sido implementado mediante Raspberry Pi, un hardware que viene siendo comúnmente utilizado en otros proyectos que no necesitan elevados recursos [22]. En concreto, el dispositivo utilizado ha sido Raspberry Pi 2 Model B que utiliza un procesador ARM Cortex A7 Quad-Core de 900 MHz y 1 GB de SDRAM con un rendimiento aceptable en aplicaciones similares [23]. Además, su funcionalidad ha sido extendida mediante una pantalla TFT resistiva modelo Adafruit PiTFT de 2,8 pulgadas (320 x 240 píxeles, 16-bits de resolución); una funda protectora a modo de carcasa; una memoria SanDisk SDHC de 32 GB (clase 10, 40 MB/s) cuya función es almacenar imágenes del sistema operativo de los equipos; un adaptador USB nano TP-Link TL-WN725N para interactuar con los equipos de forma inalámbrica mediante el estándar IEEE 802.11n; un cable de consola de tipo serial a RJ45 para gestionar los dispositivos a través del puerto de consola; un adaptador USBtoRS232; y una batería modelo Apokin Lw-106 de 2600 mAh que permite una autonomía de funcionamiento mínima de 2,88h con un consumo de 900 mAh en el peor de los casos. En total, el coste del sistema hardware es de 120,71 €.

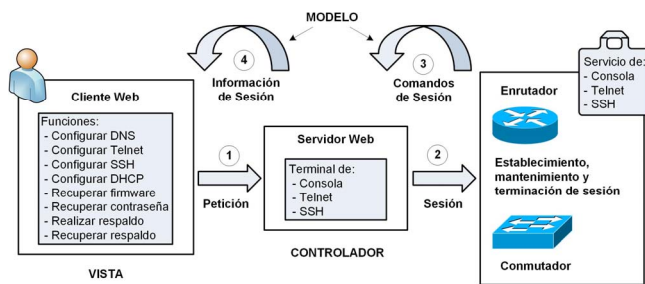


Fig. 3. Esquema del modelo-vista-controlador.

Uno de los objetivos de este trabajo consistió en desarrollar un prototipo basado en componentes de bajo coste y fácilmente disponibles en el mercado. Como resultado de aplicar estos criterios, se seleccionó Raspberry Pi que además fue elegido por la amplia comunidad que lo respalda. Otros dispositivos de tipo SBC (Single Board Computer) como Raspberry Pi —pero con especificaciones hardware más potentes— no disponen de una comunidad tan activa, por lo que podría implicar mayores tiempos de desarrollo y problemas de compatibilidad con periféricos.

Respecto al sistema operativo, servicios y software de terceros se ha seleccionado Raspbian Jessie 4.4 por ser una versión de Debian optimizada para Raspberry Pi. Sobre ella se han desarrollado las aplicaciones del sistema mediante Ruby 2.2.4, un lenguaje de programación interpretado distribuido bajo software libre. Para ello, se ha utilizado el entorno de proyectos de Ruby denominado Bundler [24]. Esto ha permitido usar las librerías ‘atftpd’ y ‘ftpd’ de Ruby —denominadas gemas— para desplegar los servidores TFTP y FTP del sistema.

Por otra parte, se ha seleccionado Phusion Passenger como servidor de aplicaciones web. Se trata de un servicio de código abierto compatible con nuestro sistema desarrollado a través del entorno ‘Ruby on Rails’ (RoR). Este servidor basado en Nginx es capaz de manejar peticiones HTTP, administrar procesos y recursos con menor carga para el sistema que otros servidores web (e.g., GlassFish), así como permite la monitorización y diagnóstico de problemas [25]. Las aplicaciones web han sido programadas mediante AngularJS, un entorno MVC de código abierto basado en JavaScript y CSS. Junto con él se ha utilizado ‘Bootstrap’, un potente entorno para diseño de proyectos web que permite adaptar dinámicamente la vista de la interfaz en función del dispositivo de usuario empleado. Ello ha permitido implementar versiones de interfaz similares que corresponden a pantallas de escritorio y pantallas móviles.

Por otro lado, se ha diseñado una BBDD mediante SQLite debido a que soporta una estructura de datos relacional que consume pocos recursos y es compatible con el modelo ACID. Esto significa que controla la atomicidad, consistencia, aislamiento y durabilidad de los datos en una biblioteca escrita en lenguaje C que ocupa tan solo ~500 KB. Como ventaja adicional, SQLite se enlaza con el programa principal pasando a ser parte integral del mismo, reduciendo así la latencia de acceso al contrario que otros gestores de base de datos como PostgreSQL, MySQL o MariaDB [26].

Respecto a los terminales de conexión se ha utilizado el servicio Ser2net tanto para realizar conexiones Telnet propias

TABLA III
SENTENCIAS Y PARÁMETROS PARA CONFIGURAR SERVICIOS DE RED EN DISPOSITIVOS CISCO A TRAVÉS DE LÍNEA DE COMANDOS

Servicio a configurar en el equipo gestionado	Comandos totales	Caracteres totales	Variables a definir
Respaldo hacia servidor FTP	2	37	2
Respaldo desde servidor FTP	2	44	2
Interfaz IP	4	50	3
Interfaz Telnet	4	58	3
DNS	5	93	2
DHCP	6	119	6
SSH	10	235	7
Restaurar ROMMON	8	142	1
Resetear contraseña (router)	9	129	1
Resetear contraseña (switch)	12	209	1

como para encapsular las conexiones de consola por el puerto serie en Telnet.

B. Aplicación Web

El entorno Web ha sido desarrollado siguiendo el patrón modelo-vista-controlador (MVC), el cual separa los datos, lógica de negocio y mecanismos de persistencia de la aplicación respecto de la interacción y representación de los datos en la interfaz (Fig. 3). En base a este esquema, el cliente web muestra la interfaz de usuario siendo su función representar los datos intercambiados entre el DRACSC y el equipo gestionado (i.e., vista). El DRACSC a su vez contiene un servidor web (i.e., controlador). Por un lado, envía información al usuario con un formato determinado que resulte práctico para su uso. Por otro, es el máximo responsable de iniciar, mantener y terminar la sesión con el equipo gestionado. Por tanto, el cometido del controlador es recibir los comandos enviados desde la vista al equipo (1)-(2) y actualizar la información de la sesión desde el equipo a la vista (3)-(4). La comunicación puede producirse por distintos medios de networking según sea realizada por consola, Telnet o SSH. No obstante, funcionalmente produce el mismo resultado de cara al usuario. De esta manera, cuando el usuario realiza una petición, el controlador crea un cliente de consola, Telnet o SSH a través de un hilo que inicia un procedimiento sobre el equipo gestionado. Por último, el modelo (i.e., la capa del sistema responsable de implementar la lógica de negocio) ha sido realizado mediante un sistema de mapeo de objetos relacionales denominado ‘registro activo’. Las ventajas que brinda son principalmente la independencia respecto a la BBDD, un desarrollo más rápido y el aumento de la seguridad. Para ello se definieron cadenas de consulta mediante anotaciones, las cuales no pueden ser alteradas por la aplicación en tiempo de ejecución y previene problemas de seguridad.

La interfaz web ha sido diseñada para permitir dos roles de usuario distintos, administradores y usuarios básicos, teniendo estos últimos funciones restringidas. Estas están disponibles tras acceder a una pantalla de identificación mediante nombre de usuario y contraseña. Después, un menú de opciones muestra la información y acciones que pueden llevarse a cabo sobre los dispositivos gestionados, la edición de acciones propias (i.e., macros), la configuración de consola e interfaces del sistema DRACSC anfitrión, la administración de roles de usuario y la ayuda general referente a la aplicación web. Estas funciones se corresponden con las opciones ‘login’, ‘devices’, ‘system’, ‘users’ y ‘help’ respectivamente.

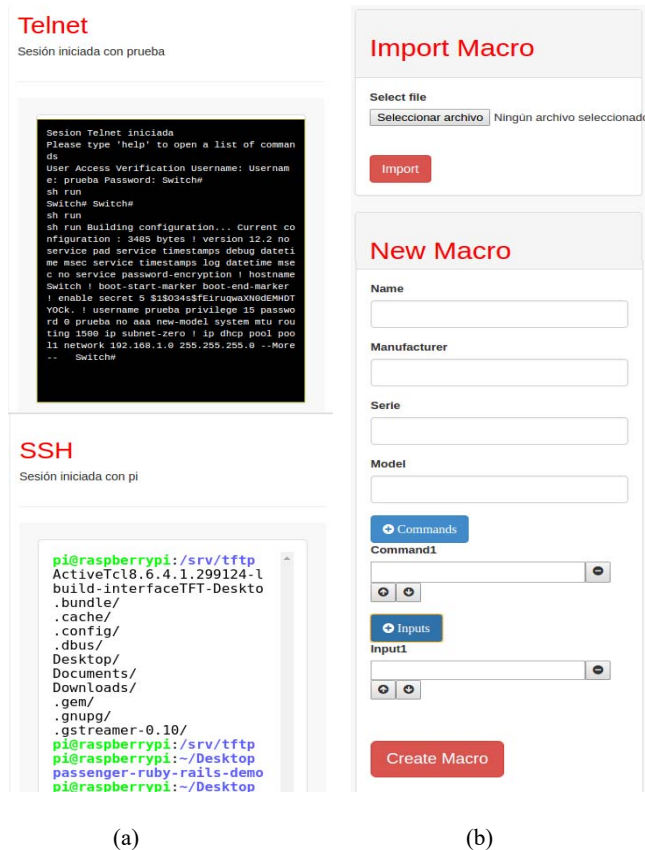


Fig. 4. Vista de la interfaz web: a) configuración rápida de equipo gestionado mediante conexión a través de Telnet y SSH, b) creación de funciones MACRO por el usuario.

C. Funciones MACRO

Todo procedimiento de configuración de un equipo gestionado —ya sea mediante conexión a través de consola, Telnet o SSH— requiere un envío secuencial de comandos. Las funciones disponibles por defecto en el sistema DRACSC facilitan la configuración rápida de los equipos gestionados sin necesidad de conocer los comandos precisos; estas son las representadas en la vista en la Fig. 3. Sin embargo, existen otras acciones avanzadas que pueden requerir mayor número de sentencias y parámetros introducidos por el usuario (ver Tabla III). Para agilizar y facilitar los procesos de configuración de los equipos se ha implementado una opción que permite a los usuarios editar funciones propias acordes con su experiencia y forma de trabajar (Fig. 4). Esta opción está disponible en la interfaz web y está soportada por una estructura basada en el uso de etiquetas en lenguaje XML (Fig. 5), permitiendo a los profesionales un mayor rango de configuración y optimización de tareas.

El principal objetivo de las funciones MACRO es la creación y aplicación de funciones de alto nivel que permitan un mayor nivel de abstracción. De esta forma, una MACRO permite al usuario poner el foco en la funcionalidad del dispositivo, evitando la necesidad de memorizar la sintaxis de los comandos y la posibilidad de cometer errores. Unido a la independencia respecto a modelos y fabricantes, podría servir como base para

```

<Macro>
  <name> Nombre de la MACRO </name>
  <!-- La etiqueta "device" define el dispositivo, esto implicará el
  poder aplicarse o no a un dispositivo u otro --!>
  <device>
    <manufacturer> Fabricante del dispositivo </manufacturer>
    <serie> Serie del dispositivo </serie>
    <model> Modelo del dispositivo </model>
  </device>
  <!-- Dentro de la etiqueta "inputs" se encuentran los distintos
  campos que pueden ser variables en el uso de una MACRO --!>
  <inputs>
    <input> </input>
    ...
    <input> </input>
  </inputs>
  <!-- En la etiqueta "commands" se incluyen los comandos que se
  lanzarán al dispositivo para su configuración --!>
  <commands>
    <command> command
    ...
    <command> </command>
  </commands>
</Macro>

```

Fig. 5. Estructura XML de una función MACRO.

el desarrollo de un lenguaje procedimental que permita la configuración armonizada de equipos de red.

Como ejemplo, el modo de operación Monitor de la ROM (ROMmon) de Cisco se trata de un estado en el que un conmutador o enrutador arranca directamente para ejecutar diversas acciones sobre la imagen del IOS. Esto se requiere cuando la imagen del IOS ha sido borrada o dañada, se quiere iniciar el equipo desde la ROM, ignorar el contenido de la NVRAM para resetear la contraseña en modo 'enable', establecer la velocidad de la consola o habilitar los mensajes de diagnóstico entre otros [27]. Para acceder al modo ROMMON de un enrutador debe enviarse un comando 'control+break' a través de consola o Telnet durante los primeros 60 segundos del arranque. En el caso de un conmutador, se requiere apagar manualmente el equipo y pulsar el botón MODE durante 5 segundos. Tras el reinicio es importante conocer el valor del registro de configuración, que es el encargado de establecer el arranque del dispositivo. El valor hexadecimal configurado por defecto de fábrica para enrutadores suele ser 0x2102. Esto significa que debe intentar cargar una imagen desde la memoria Flash, ignorar los break (excepto en los primeros 60 segundos), establecer una velocidad de 9600 baudios para la consola y, si falla el arranque, cargar la IOS desde la ROM. Como se aprecia, esta tarea resulta ciertamente compleja por lo que generalmente está reservada a técnicos especialistas verdaderamente experimentados. Como dato, las funciones de recuperación de contraseña necesitan entre 9 y 12 comandos más un parámetro de usuario, mientras recuperar un dispositivo en modo ROMMON requiere además una imagen de respaldo de la IOS almacenada en una ubicación externa.

IV. EXPERIMENTACIÓN

Para llevar a cabo las pruebas se ha utilizado como entorno de experimentación el laboratorio de redes de la Universidad de

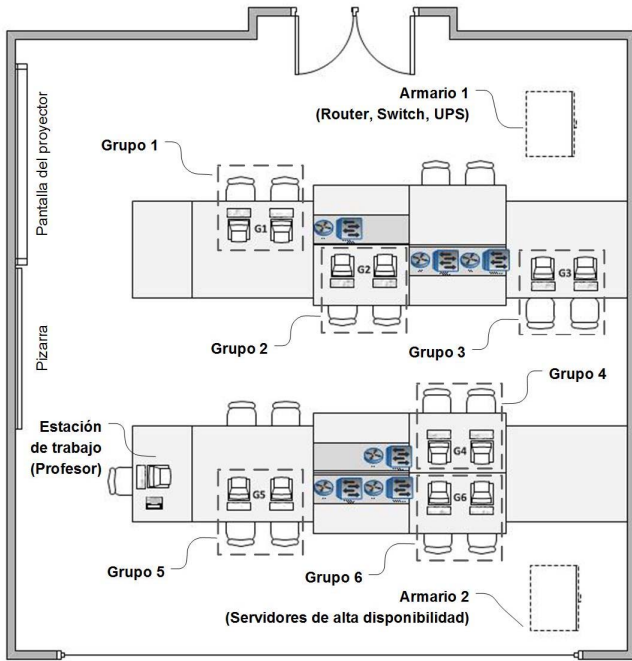
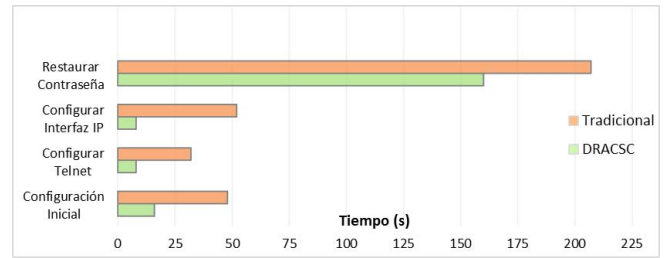


Fig. 6. Esquema de los sistemas de comunicación del laboratorio docente utilizado como entorno de experimentación.

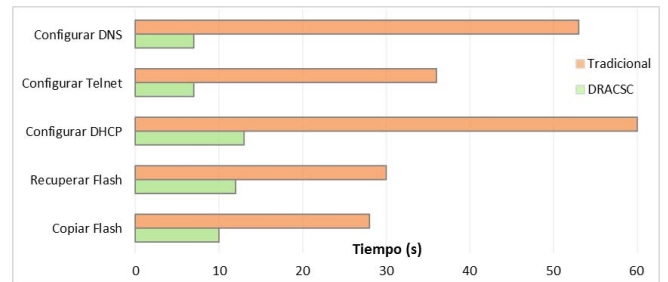
Huelva, cuyo diagrama puede verse en la Fig. 6. La arquitectura del laboratorio docente consta de 13 computadoras ACER Verition 5700GX con 160 GB HD en RAID, 1 Gb RAM, 2 tarjetas Gigabit Ethernet y arranque PXE. Las áreas de trabajo de los estudiantes se dividen en seis grupos, cada uno de ellos formado por dos PCs, dos enrutadores y un conmutador. El mantenimiento del laboratorio es administrado por dos servidores ubicados en armarios de comunicación, protegidos por UPS, que se han configurado con Ubuntu Server 10.04.1 y OpenGnSys 1.0.4a. Dicho laboratorio forma parte de un proyecto educativo más amplio cuyo objetivo es construir un aula para prácticas telemáticas en redes informáticas [28].

El sistema DRACSC se ha utilizado para la gestión de la docencia en el escenario descrito anteriormente. La solución permite al profesor preparar configuraciones personalizadas con los dispositivos de red con los que va a trabajar y copiarlas a los puestos de trabajo de los alumnos haciendo solo unos clics desde el sistema DRACSC. Adicionalmente, la solución implementada permite solventar problemas tales como configuraciones sobre escritas o bloqueadas por los alumnos en clases previas mediante una contraseña desconocida. El sistema DRACSC solventa dichos errores y automatiza las tareas de gestión, suponiendo un ahorro de tiempo significativo. Por estos motivos se ha utilizado DRACSC como solución para preparar un aula con el mínimo mantenimiento y maximizar las garantías de funcionamiento.

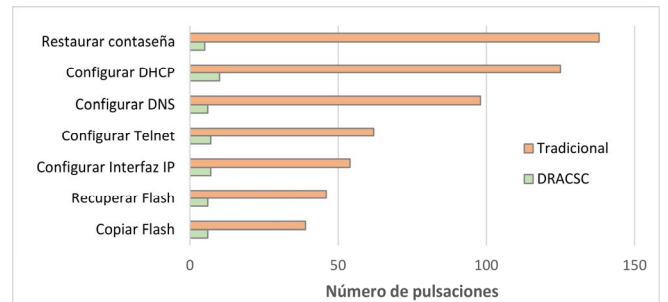
La viabilidad del prototipo DRACSC ha sido comprobada por un técnico de laboratorio con experiencia en administración de infraestructuras de red y certificación profesional CCNA. Para computar las acciones se ha considerado un escenario en el que dicho profesional se encuentra delante del equipo gestionado, encendido, con las conexiones establecidas y conoce de antemano los comandos requeridos. Por tanto, se ha



(a)



(b)



(c)

Fig. 7. Comparativa de distintas acciones de administración de servicios de red: a) tiempo de configuración en un conmutador a través de terminal de consola, b) tiempo de configuración en un enrutador a través de sesión Telnet, y c) pulsaciones necesarias para la configuración de los servicios anteriores.

excluido el tiempo de arranque de computadoras, conexión de cables o la búsqueda de comandos para ejecutar los procesos de configuración.

Con objeto de reflejar las ventajas del sistema DRACSC se han llevado a cabo dos casos de estudio consistentes en medir el tiempo que supone la administración de equipos utilizando el procedimiento automatizado y el manual tradicional. El primer caso de estudio ha consistido en gestionar un conmutador aplicando un procedimiento avanzado de recuperación de contraseña y tres de configuración básicos (i.e., interfaz IP, servicio Telnet y configuración inicial de servicios de red). Para el segundo caso se han elegido tres servicios de red básicos (i.e., DNS, Telnet y DHCP) y dos procedimientos de configuración avanzados (i.e., crear y restaurar memoria Flash en servidor remoto) sobre un enrutador.

En la Fig. 7a se muestra el procedimiento de configuración del conmutador utilizando una conexión mediante terminal de consola. Como se puede ver en todos los casos, es más rápido el uso de funciones automatizadas. El mayor tiempo del modo manual lo encontramos en el tiempo que necesitamos para escribir los comandos completos y asimilar los resultados

obtenidos. En la Fig. 7b se muestra el procedimiento de configuración del enrutador utilizando una conexión remota mediante Telnet, siendo necesario el servicio habilitado en el dispositivo gestionado. Como se observa, en general los tiempos son algo mayores que utilizando la sesión de consola. De forma análoga que en el caso anterior, los tiempos son mejores con el sistema DRACSC que con el método manual debido a la necesidad de escribir los comandos y asimilar los resultados. Como norma general, mientras más líneas de comandos necesite una función, mayor será la ventaja de usar las funciones automáticas.

En la Fig. 7c se incluyen los caracteres necesarios para llevar a cabo las operaciones de gestión mostradas en la Tabla III. En comparación a las operaciones de gestión tradicionales basadas en líneas de comando, la solución DRACSC requiere una pulsación de ratón para seleccionar la función MACRO, otra para definir cada variable de la función, otra para seleccionar la interfaz de comunicación y una pulsación final para ejecutarla.

V. DISCUSIÓN Y DESARROLLOS FUTUROS

La administración de equipos de comunicación es una tarea compleja —generalmente relegada a técnicos expertos en redes e Internet— que demanda grandes dotes para el análisis y resolución de problemas con el objetivo de ofrecer el mejor rendimiento y servicio de red. Mientras la metodología tradicional basada en la administración manual de equipos es poco efectiva debido al tiempo y coste —mayor cuanto más grande es la infraestructura a gestionar—, las soluciones centralizadas para administrar equipos y servicios de red deben proporcionar un alto grado de configuración y funcionalidad para ser realmente útiles.

Con este objetivo, se presenta un dispositivo portable que automatiza la gestión de configuraciones y sistemas operativos de equipos de red. El prototipo —denominado DRACSC— ha sido implementado mediante una solución hardware/software basada en Raspberry Pi 2, el cual ha sido probado sobre sistemas embebidos incluyendo conmutadores y enrutadores de los fabricantes Cisco y MikroTik, extensibles a otros. Esta solución se torna de especial interés en un entorno donde el número de elementos de red alcanza cierto volumen, pudiendo actuar sobre ellos sin necesidad de recordar qué credencial de acceso corresponde a cada equipo o hacer uso de varias aplicaciones de red para llegar al mismo resultado.

Con respecto a las soluciones basadas en software (Tabla I), el sistema DRACSC propuesto es el único que actúa a través del puerto de consola y, junto con Colibri NetManager, una de las soluciones que utiliza funciones de recuperación de errores. Con respecto a las soluciones basadas en hardware (Tabla II), el sistema DRACSC es el único que utiliza funciones de recuperación de errores y, junto con Raspisco, una de las soluciones que presenta portabilidad. Además, aporta otra contribución novedosa como es la edición de funciones de alto nivel basadas en plantillas estructuradas —denominadas MACROS— que permiten clonar configuraciones y desplegarlas en una infraestructura de red. En este sentido, las pruebas de rendimiento han mostrado que cuanto más compleja sea la tarea de administración de enrutadores y conmutadores

—contabilizada por el número de sentencias y caracteres a teclear— mayor es la ventaja de usar las funciones automatizadas del sistema DRACSC. Así, el número de pulsaciones promedio se redujo en un $92,17 \pm 4,24\%$. Ello resultó en un ahorro de tiempo promedio de $68,72 \pm 19,58\%$ con tiempos mínimos de 7s (p.ej., configuración de Telnet/DNS) y tiempos máximos de 160s (p.ej., recuperación de errores de la imagen del IOS).

Con respecto al trabajo futuro, los desarrollos actuales están centrados en obtener mayor difusión y aceptación del sistema DRACSC por parte de los usuarios. Para ello, se ampliará la muestra y se analizará un cuestionario sobre la opinión de los usuarios que permita obtener datos estadísticos significativos. Esto incluye alumnos, docentes y profesionales del sector. Por otro lado, se está trabajando en implementar un repositorio público de funciones MACRO alrededor de una comunidad que apoye y posibilite el crecimiento del proyecto. En este sentido —gracias a la flexibilidad de la estructura software del dispositivo— los esfuerzos están siendo dirigidos a concebir DRACSC como una herramienta con opciones mejoradas para configurar y administrar redes, extendiéndose a diferentes fabricantes. En este sentido, se propone desarrollar una plataforma en la nube que se comunicará con dispositivos DRACSC, permitiendo la realización de acciones adicionales que requieren un mayor consumo de recursos y que, por limitaciones de hardware, no podrían llevarse a cabo en la solución aportada en este trabajo. Entre ellas se incluye la administración de múltiples redes, el mantenimiento de una BBDD con históricos de configuraciones, recuperación de errores y la reparación automática de elementos de red mediante reglas definidas a través de conocimiento experto.

REFERENCIAS

- [1] Glen E. Clarke, “*CompTIA Network+ Certification Study Guide*”, McGraw Hill Professional, 2018.
- [2] Mohammed M. Alani, “*Guide to Cisco Routers Configuration: Becoming a Router Geek*”, Springer, 2017.
- [3] E. Ariganello, “*REDES CISCO. Guía de estudio para la certificación CCNA Routing y Switching*”, Ra-Ma, 2017.
- [4] T.J. Mateo Sanguino, I.J. Fernández de Viana González, J. Espejo Fernández, and A. García Domínguez, “Using Identity Provider and Automatic Resource Management to Improve a Remote Networking Lab,” *IEEE Latin American Transactions*, vol. 16(5), pp. 1547-1556, 2018.
- [5] T.J. Mateo Sanguino, et al., “OpenGnSys: a Novel System toward Centralized Deployment and Management of Computer Laboratories,” *Computers & Education*, 75, pp. 30-43, 2014.
- [6] D. Pardo Garrido, et al., “OpenGnSys: un Sistema de Gestión Centralizada y Despliegue de Sistemas Operativos en el Aula,” *Proc. 8th Iberian Conf. Information Systems and Technologies*, pp. 295-301, 2013.
- [7] T.J. Mateo Sanguino, et al., “The Role of Telematic Practices in Computer Engineering: a Low-cost Remote Power Control in a Network Lab,” *Intern. J. Online Engineering*, 8(2), pp. 15-22, 2012.
- [8] J.M. Andújar Márquez, and T.J. Mateo Sanguino, “Design of Virtual and/or Remote Laboratories. A Practical Case,” *Revista Iberoamericana de Automática e Informática Industrial*, 7(1), pp. 64-72, 2010.
- [9] T.J. Mateo Sanguino, and J.D. Morillo Reina, “Dispositivo y Sistema para la Recuperación de Equipos de Comunicación,” ES 2569414, 2014.
- [10] T.J. Mateo Sanguino, and J.D. Morillo Reina, “Device and System for the Recovery of Communication Equipment,” WO/2016/055682, 2016.
- [11] T. Bakhshi, “State of the Art and Recent Research Advances in Software Defined Networking,” *Wireless Communications and Mobile Computing*, pp. 1-35, 2017.

- [12] K. Elmansor, "Towards Automated Network Configuration Management," *Master's Thesis*, College of Computing and Digital Media, 2013.
- [13] Cisco Networking Academy, *Scaling Networks Companion Guide*, Cisco Press, 2014.
- [14] It Digital Media Group, "Las Claves de una Adecuada WLAN en la Empresa," *It - User Tech & Business*, pp. 33-38, 2016.
- [15] D. Curtis, and J. Kowall, "MarketScope for Network Configuration and Change Management," *Tech. Rep.*, Gartner, Inc., pp. 1-18, 2011.
- [16] EMA, "Essential IT Monitoring: Seven Priorities for Network Management," *Tech. Rep.*, 2013. [Online] <http://content.solarwinds.com/>
- [17] Perle Systems Ltd., "IOLAN SDS/SCS/STS User's Guide," *Tech. Rep.*, 2018. [Online] <https://www.perle.com>.
- [18] Raritan, Inc., "Dominion SX User Guide 3.5.0," *Tech. Rep.*, 2014. [Online] <http://support.raritan.com>
- [19] R. Shepherd "Converged Infrastructure Products and Systems," *Inside Networks*, pp. 50, 2013.
- [20] Belunix, "Raspisco — remote access to Cisco through Raspberry Pi," *Tech. Rep.*, 2013. [Online] <http://developers-club.com/posts/192188/>
- [21] D. Kyuchukova, G. Hristov, P. Zahariev and S. Borisov, "A study on the possibility to use Raspberry Pi as a console server for remote access to devices in virtual learning environments", *IEEE 2015 International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2015.
- [22] L.F. D Haro, R. Cordoba, J.I. Rojo Rivero, J. Diez de la Fuente, D. Avendano Peccas and J.M. Bermudo Mera, "Low-Cost Speaker and Language Recognition Systems Running on a Raspberry Pi", *IEEE Latin American Transactions*, vol. 12(4), pp. 755-763, 2014.
- [23] E. Moreno, F. Anjo Lima, W.R. Azevedo Dias, "Performance Analysis of a Low Cost Cluster with Parallel Applications and ARM Processors", *IEEE Latin American Transactions*, vol. 14(11), pp. 4591-4596, 2016.
- [24] P. Cooper, "Beginning Ruby: From Novice to Professional", Apress, 2016.
- [25] I. Alshansky, "Nginx Tricks for PHP Developers," *International PHP Conference*, Munich, Germany, 2015.
- [26] M. Forde, "Using the Raspberry Pi for Data Collection, Display and Dissemination," *Technical Conf. Meteorological and Environmental Instruments and Methods of Observation*, 2016.
- [27] M. Larsson, "Sanitization of embedded network devices," *Master's Thesis*, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.
- [28] T.J. Mateo Sanguino, I.J. Fernández de Viana González, E. Cortés Ancos, J. Espejo Fernández, "Exploring Strengths and Weaknesses: a Case Study After Developing a Remote Network Lab," *Computer Applications in Engineering Education*, vol. 26, pp. 1422-1434, 2018.



Tomás de J. Mateo Sanguino recibió en 2010 el doctorado en Ingeniería Electrónica por la Universidad de Huelva (UHU), España. Ha sido Ingeniero Electrónico y Máster en Enseñanza Universitaria en 2001 y 2008, respectivamente. Desde 2004 ha trabajado como profesor colaborador y contratado doctor en el Dpto. de Ingeniería

Electrónica, Sistemas Informáticos y Automática de la UHU. Actualmente trabaja también como instructor en el programa CCNA de Cisco Networking Academy. Ha participado como investigador en 20 proyectos, siendo autor y coautor de más de 75 publicaciones en diferentes revistas y congresos. Sus intereses de investigación actuales incluyen la educación de ingeniería.



Juan Diego Morillo Reina es Ingeniero Informático por la Universidad de Huelva. Comenzó profesionalmente en el mundo de la informática en 2010 como Técnico Superior en Administración de Sistemas Informáticos. Posteriormente estudió el Grado en Ingeniería Informática con especialidad en Ingeniería de Computadores (2014) y el Máster en

Ingeniería Informática (2016). Actualmente trabaja como CTO, entre sus intereses están las redes y sistemas informáticos en los que actualmente realiza su Tesis Doctoral.