ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ANÁLISIS DEL IMPACTO ECONÓMICO DE LA IMPLEMENTACIÓN DEL PROTOCOLO TR-069 EN LOS CPE'S DE LA EMPRESA DE TELECOMUNICACIONES PUNTONET

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

JUAN FERNANDO LÓPEZ TITO
johnferlo6689@gmail.com

DIRECTOR: ING. RICARDO XAVIER LLUGSI CAÑAR
ricardo.llugsi@epn.edu.ec

CODIRECTOR: Dr. DIEGO JAVIER REINOSO CHISAGUANO
diego.reinoso@epn.edu.ec

Quito, junio 2019

AVAL

Certificamos que el presente trabajo fue desarrollado por Juan Fernando López Tito, bajo nuestra supervisión.

Ing. RICARDO XAVIER LLUGSI
DIRECTOR DEL TRABAJO DE TITULACIÓN

Dr. DIEGO JAVIER REINOSO
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo Juan Fernando López Tito, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

Juan Fernando López Tito

DEDICATORIA

Dedico este trabajo a mi familia, sin ustedes todo sería complicado, gracias por existir, tú eres yo y yo soy tú, yo existo porque tú existes.

.

AGRADECIMIENTO

Quiero agradecer a todas aquellas personas que me acompañaron durante todo este proceso, ya que cada uno de ustedes han sido y son parte fundamental en la culminación de esta etapa, pero quiero agradecer de manera especial a mis padres, por su apoyo incondicional en cada uno de los momentos a lo largo de mi etapa de estudios, a mis hermanas por ser esa fuente de inspiración.

También quiero agradecer a mi futura esposa y madre de mi hijo, que, con sus palabras, retroalimentación, y apoyo me permitió rediseñarme en cada uno de esos momentos que las circunstancias parecían ser más grande.

Y por último quiero agradecer a mis amigos de trabajo, al equipo de Inspira y Trasciende que estuvieron apoyando y aportando con ideas para el desarrollo de este trabajo.

.

ÍNDICE DE CONTENIDOS

AVAL.....I

DECLARACIÓN DE AUTORÍA II

DEDICATORIA III

AGRADECIMIENTO IV

ÍNDICE DE CONTENIDOS V

RESUMEN VII

ABSTRACT VIII

1. INTRODUCCIÓN.....1

1.1 OBJETIVOS.....1

1.2 ALCANCE.....2

1.3 MARCO TEÓRICO 3

TR-069 PROTOCOLO DE ADMINISTRACIÓN WAN CPE DENTRO DE PUNTONET [1] 3

1.3.1 INTRODUCCIÓN.....3

1.3.2 ARQUITECTURA 4

1.3.3 MÉTODOS RPC.....8

1.3.4 GATEWAY ASOCIADO 9

1.3.5 MANEJO DE PROXY 10

1.3.6 TR-181 MODELO DE DATOS DEL DISPOSITIVO PARA TR-069 [4] 10

1.3.7 SITUACIÓN ACTUAL DE LA EMPRESA PUNTONET 18

1.3.8 RADIO ENLACES RESIDENCIALES 21

1.3.9 PROCESOS DE SOPORTE TÉCNICO TELEFÓNICO [9].....22

2. METODOLOGÍA.....29

2.1 INTERACCIÓN ENTRE EL SERVIDOR ACS Y CPE'S.....30

2.1.1 SISTEMA DE ADMINISTRACIÓN WAN CPE FREEACS [11] 30
2.1.2 INSTALACIÓN DEL SERVIDOR ACS 58
2.1.3 PROCESO DE APROVISIONAMIENTO.....68
3. RESULTADOS Y DISCUSIÓN 84
3.1 RESULTADOS OBTENIDOS EN LAS PRUEBAS.....84
3.2 ANÁLISIS DE COSTOS.....88
3.2.1 Número de Visitas Técnicas 88
3.2.2 Costos de una Visita Técnica 89
3.3 COMPARACIÓN DE RESULTADOS 90
CAPITULO IV.....93
4. CONCLUSIONES Y RECOMENDACIONES 93
4.1 CONCLUSIONES 93
4.2 RECOMENDACIONES 94
5. REFERENCIAS BIBLIOGRÁFICAS.....95
6. ANEXOS 97
ORDEN DE EMPASTADO.....98

RESUMEN

Con la realización del proyecto denominado “ANÁLISIS DEL IMPACTO ECONÓMICO DE LA IMPLEMENTACIÓN DEL PROTOCOLO TR-069 EN LOS CPES DE LA EMPRESA DE TELECOMUNICACIONES PUNTONET” se analizaron los procesos de soporte, se validó la interacción entre los equipos y el servidor y se presentan los resultados obtenidos a través de pruebas prácticas, que permitieron analizar y comparar los procesos de actualización de firmwares de los CPE’s además de los costos que se generan al realizar visitas técnicas por fallas en la actualización.

En el Capítulo I, se describirá el protocolo TR-069 tomando en cuenta su arquitectura, métodos de llamada, modelo de datos, adicional. Se mencionará la situación actual de la empresa de servicios PUNTONET, señalando el tipo de servicios que se ofrecen, la tecnología que se usa y los procesos de soporte.

En el Capítulo II, se describe el Servidor ACS utilizado, donde se detalla el funcionamiento, arquitectura e instalación. Se mencionará el funcionamiento de los módulos que conforman el servidor. Se muestra cómo interactúan los equipos aprovisionados y el servidor ACS.

En el Capítulo III, se hace un análisis de los resultados obtenidos, determinando el porcentaje de efectividad del proceso de actualización de firmwares al usar el ACS, y con el cual se puede hacer una comparativa entre los gastos generados por visitas técnicas debido a actualizaciones fallidas con el proceso actual y al usar el ACS.

En el Capítulo IV, se presentarán conclusiones y recomendaciones en base a las diferentes etapas desarrolladas a lo largo de este proyecto.

PALABRAS CLAVE: TR069, CPE, FreeACS, Unittype, profile, job.

ABSTRACT

The realization of the project "ANALYSIS OF ECONOMIC IMPACT ABOUT IMPLEMENTATION OF PROTOCOL TR-069 IN CPES FROM PUNTONET TELECOMMUNICATIONS COMPANY" got results through practical tests.

These results allowed to analyze and compare the updating device firmware process and the costs generated by technical visits that are made by actualization fails.

Chapter I, TR-069 protocol will be described about its architecture, Remote Procedure Call and data model. Also the current situation of PUNTONET will be mentioned. Indicating the type of services offered, technology that is used and the support processes.

Chapter II, ACS Server will be described detailing its modules operation, architecture and installation. Also how provisioned equipment and ACS server interact will be mentioned.

Chapter III, With the results obtained in different tests. It can determining a percentage of effectiveness of firmware update process when ACS Server is used. With this a comparison can be made between expenses generated by technical visits due to failed updates with the current process and when ACS Server is used.

Chapter IV, conclusions will be presented based on different stages developed throughout this project and a series of recommendations will be developed.

KEYWORDS: TR069, CPE, FreeACS, Unittype, profile, job.

1. INTRODUCCIÓN

El crecimiento continuo de la Empresa de Telecomunicaciones PUNTONET, conlleva a enfrentar un enorme reto en lo que concierne a la administración a gran escala, de los equipos de puerta de enlace (CPE), los cuales requieren ser actualizados y monitoreados de manera periódica para que su funcionamiento sea óptimo.

Actualmente estos requerimientos son atendidos uno a uno y sobre un cierto grupo de clientes, que establecen contacto con el soporte técnico. Lo que hace que el resto clientes, presenten inconvenientes en su servicio. En consecuencia, esto se ve reflejado en las numerosas solicitudes de visitas técnicas que se generan, debido a incidencias relacionadas con temas de actualización de firmware de los CPE's, aumentando así, el uso de los recursos logísticos.

Todo esto conlleva a buscar herramientas que permitan optimizar los procedimientos de gestión, administración y diagnóstico en los equipos de enlace. Este tipo de herramientas deben presentar un alto grado de seguridad, flexibilidad y escalabilidad. Esta solución debe permitir encontrar un porcentaje de ahorro en los costos que se generan en la gestión de visitas técnicas.

En la actualidad la administración y gestión remota, a gran escala y de una manera segura de los CPE's, es una prioridad a ser atendida dentro de la empresa, para poder solventar requerimientos periódicos tales como: la auto-configuración, actualizaciones, diagnóstico y optimización de los dispositivos, de tal manera que para el usuario final todas estas tareas sean totalmente transparentes. Esto reduciría considerablemente la cantidad de actividades de gestión que realizan los operadores de red, y simultáneamente se irá reduciendo las incidencias de visitas técnicas presentadas por eventos antes mencionados.

1.1 OBJETIVOS

El Objetivo General de este trabajo es: Analizar el impacto económico de la implementación de un sistema de administración y gestión remoto, usando el protocolo TR-069, en los CPE's de la Empresa de Telecomunicaciones PUNTONET.
Los Objetivos Específicos de este trabajo son:

- Analizar el funcionamiento, requerimientos, arquitectura, procedimientos y las diferentes funciones del protocolo TR-069, para la administración de CPE's y correcto despliegue dentro de una infraestructura.
- Analizar las especificaciones de la norma TR-181 y las especificaciones del fabricante de los equipos Mikrotik, para establecer parámetros de configuración que permitan la gestión remota de los CPE's a través del servidor de configuración automática ACS.
- Establecer diferencias entre el modelo de gestión actual y el modelo de gestión usando el protocolo TR-069.
- Elaborar informes, donde se presenten los resultados de las pruebas realizadas y a través de los cuales se pueda evaluar los beneficios que otorga este modelo gestión y estimar un porcentaje de reducción en los costos que se generan al gestionar incidencias de visitas técnicas.

1.2 ALCANCE

Con la realización de este proyecto, se busca brindar una alternativa para la gestión y administración, a gran escala, de dispositivos de última milla que se usan para proveer del servicio de internet a los diferentes suscriptores con tecnologías de radio, en la ciudad de Quito, pertenecientes a la cartera de clientes con planes Home de la empresa PUNTONET.

El desarrollo de este trabajo está basado en el análisis de los resultados obtenidos en las pruebas de laboratorio con los equipos de la marca Mikrotik, antenas SXT y Metal y routers RB750 y RB2011, en los cuales se implementa el protocolo TR-069 y en loa que se considera las recomendaciones del Broadband Forum que es la entidad que define el Protocolo TR-069 y al mismo tiempo se toma en cuenta lo indicado de la norma TR-181 en la que se basa Mikrotik para el soporte del protocolo TR.069.

Con los resultados obtenidos se pudo realizar una comparación con la actual metodología de administración de los CPE's, y se establecieron conclusiones de factibilidad técnico-económicas, que permitirán a PUNTONET, analizar y determinar si es o no una solución para optimizar las labores de gestión, monitoreo y diagnóstico, a través de un servidor de configuración automática (ACS), de los equipos de última milla.

1.3 MARCO TEÓRICO

TR-069 PROTOCOLO DE ADMINISTRACIÓN WAN CPE DENTRO DE PUNTONET

1.3.1 INTRODUCCIÓN [1]

El Reporte Técnico, TR-069 describe el CWMP (CPE *WAN Manage Protocol*, Protocolo de Administración WAN CPE); el mismo hace referencia, al Informe Técnico publicado por el Broadband Forum, desarrollado para permitir a los proveedores de servicios de Internet, la implementación y administración de los equipos en las instalaciones del cliente, ya sean en redes domésticas y comerciales.

TR-069 está destinado a la comunicación entre un CPE (*Costumer Premises Equipment*, Equipo en las instalaciones del cliente), y un servidor ACS (*Auto Configuration Server*, Servidor de Configuración Automática). El CWMP define un mecanismo que abarca la auto configuración segura de un CPE y está diseñado para admitir una variedad de funcionalidades para administración de los equipos, incluyendo las siguientes capacidades principales:

- Configuración automática y provisión dinámica de servicios.
- Actualización de paquetes de software.
- Gestión de módulos de software.
- Monitoreo de estado y desempeño.
- Diagnóstico.

Se basa en una arquitectura punto a punto donde el ACS es un servidor que se ubica en la red corporativa y administra dispositivos dentro de las instalaciones de los clientes. Este protocolo se puede utilizar para administrar diferentes tipos de equipos, incluidos enrutadores y dispositivos de LAN. Es indiferente al medio de acceso específico utilizado por el ISP, aunque depende de que la conectividad de capa IP haya sido establecida por el dispositivo.

El CWMP está diseñado para proporcionar un alto grado de seguridad y escalabilidad. Tiene la intención de proporcionar flexibilidad en el modelo de conectividad; además está destinado al descubrimiento y asociación entre ACS y CPE. En la figura 1.1 se puede observar un ejemplo de la arquitectura punto a punto.

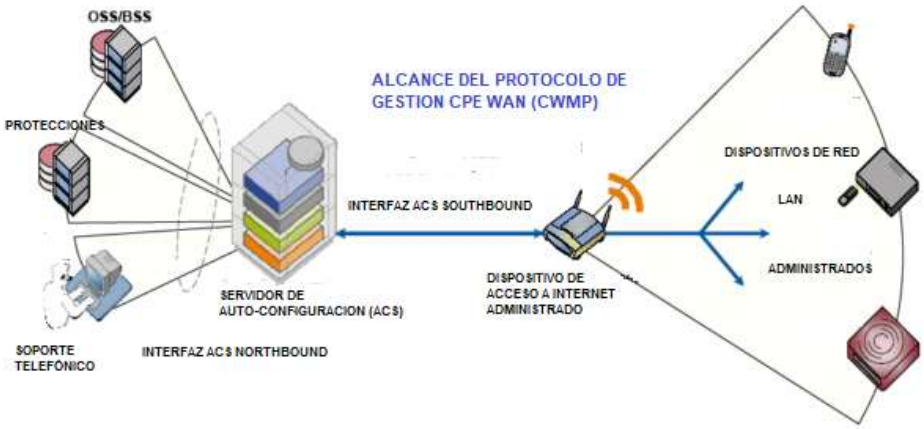

Figura 1.1. Arquitectura punto a punto [2]

1.3.2 ARQUITECTURA

1.3.2.1 Componentes del protocolo

El stack de protocolos definido por el CWMP, comprende varios componentes que son exclusivos de este protocolo, y a su vez, se complementan con protocolos estándar. En la Tabla 1.1 se hace una breve descripción de cada componente del stack de protocolos.

Tabla 1.1. Descripción del stack de protocolos

Protocolo	Descripción
Aplicación CPE/ACS	Usa el protocolo de administración CPE WAN tanto en el CPE como en el ACS.
Métodos RPC	Procesos específicos definidos por el protocolo de administración CPE WAN
SOAP	Es un protocolo basado en XML utilizado para codificar llamadas a procedimientos remotos.
HTTP	Hypertext Transfer Protocol - HTTP / 1.1
TLS	El protocolo estándar de seguridad de la capa de transporte de Internet.
TCP/IP	Estándar TCP / IP.

1.3.2.2 Componentes Arquitectónicos

1.3.2.2.1 Parámetros

TR-069 utiliza un mecanismo genérico mediante el cual un ACS puede leer o escribir parámetros para configurar, revisar estadísticas, verificar el estado de un CPE, estos parámetros están organizados en una estructura jerárquica bien definida la cual es

compatible con la mayoría de dispositivos y fabricantes. Cada parámetro consiste en un par que contiene nombre y valor. El nombre identifica el parámetro en particular. El valor de un parámetro puede ser uno de varios tipos de datos definidos.

1.3.2.2.2 Transferencias de archivos

Los métodos RPC (*Remote Procedure Call*, Llamado a Proceso Remoto) proporcionan mecanismos para facilitar las transferencias de archivos para una variedad de propósitos, como descargar actualizaciones de firmware o archivos de configuración específicos del proveedor. Las transferencias de archivos se pueden realizar mediante unicast o protocolos de transporte multicast. Los protocolos unicast incluyen HTTP / HTTPS, FTP, SFTP y TFTP. Los protocolos de multidifusión incluyen FLUTE y DSM-CC.

1.3.2.2.3 Sesiones iniciadas CPE

Los métodos RPC brindan un mecanismo que permite a un CPE informar a un ACS sobre diversas condiciones y asegurar que la comunicación CPE-ACS ocurra con una frecuencia mínima.

1.3.2.3 Procedimientos

1.3.2.3.1 Descubrimiento de ACS [1]

En el CWMP existen tres métodos a través de los cuales un CPE puede descubrir la dirección de su ACS asociado:

- Configuración de la URL del ACS para cada CPE.
- Uso de un servidor DHCP en la red de acceso para incluir la URL de ACS como una opción DHCP.
- Uso de una dirección URL predeterminada de ACS, que es utilizada en caso de que no se proporcione una diferente.

El puerto 7547 ha sido asignado por IANA (*Internet Assigned Numbers Authority*, Autoridad de Números Asignados en Internet) para el protocolo de administración WAN CPE, y el ACS puede utilizar este puerto en su URL.

1.3.2.3.2 Establecimiento de conexión

El CPE puede iniciar en cualquier momento una conexión al ACS utilizando la dirección predeterminada de ACS. Es necesario que un CPE establezca una conexión con la ACS en los siguientes casos:

- La primera vez que el CPE establece una conexión a la red de acceso en la instalación inicial.
- Al encender o reiniciar el CPE.
- Siempre que en el CPE se complete su temporizador de conexión.
- Siempre que el CPE reciba una Solicitud de Conexión válida de un ACS.
- Siempre que la URL del ACS cambie.
- Siempre que se modifique un parámetro.

El ACS puede en cualquier momento solicitar que un CPE inicie una conexión con él utilizando el mecanismo de solicitud de conexión. [1]

1.3.2.3.3 Uso de TLS y TCP

El uso de TLS (*Transport Layer Security*, *Seguridad en la Capa de Transporte)* para transportar el CWMP es recomendado, aunque el protocolo puede ser utilizado directamente sobre una conexión TCP (*Transmission Control Protocol*, Protocolo de Control de Transporte). TLS proporciona confidencialidad e integridad de datos, y permite la autenticación basada en certificados [1]en lugar de la autenticación basada en secreto compartido.

1.3.2.3.4 Uso de HTTP

Los mensajes SOAP (*Simple Object Access Protocol*, Protocolo de Acceso Simple a Objetos) se transfieren entre un CPE y un ACS usando HTTP 1.1, donde el CPE actúa

[1] La autenticación en redes basada en certificados: consiste en el uso de un certificado digital (credencial) para identificar a un usuario y, con frecuencia, también a un dispositivo (o dispositivos) utilizado por un usuario conocido de la red. [21]

como el cliente HTTP y el ACS actúa como el servidor HTTP. La capa HTTP proporciona un medio alternativo de autenticación CPE y ACS basada en secretos compartidos.[2]

1.3.2.3.5 Uso de SOAP

El protocolo de gestión WAN CPE define el SOAP 1.1 como la sintaxis de codificación para transportar las llamadas al método RPC.

SOAP es un protocolo para el intercambio de información en un entorno descentralizado y distribuido, basado en XML3 (*Extensible Markup Language*, Lenguaje de Marcas Extensible) que consta de tres partes: un envoltura que define un marco para la descripción de lo que está en un mensaje y cómo procesarlo, un conjunto de reglas de codificación para expresar instancias de tipos de datos definidos por la aplicación, y una convención para representar las llamadas a procedimiento remoto y respuestas. [3]

1.3.2.3.6 Procedimientos de la Sesión

Todas las sesiones deben comenzar con un mensaje de información del CPE contenido en el HTTP POST inicial. Esto sirve para inicializar el conjunto de transacciones y comunicar las limitaciones del CPE con respecto a la codificación de mensajes. En la Figura 1.2 se muestra un ejemplo de la transacción entre CPE y ACS.

La sesión termina cuando tanto la ACS como la CPE no tienen más solicitudes que enviar y no quedan respuestas de la ACS o de la CPE. En ese momento, el CPE debe cerrar la conexión. No puede haber más de una sesión entre un CPE y su ACS asociado a la vez.

---

[2] Autenticación basada en secretos compartidos: Este protocolo consiste en que una parte envía un número aleatorio a la otra, quien a continuación lo transforma en una forma especial y después regresa al resultado. [22]

[3] XML: es una especificación para diseñar lenguajes de marcado, que permite definir etiquetas personalizadas para descripción y organización de datos. [24]

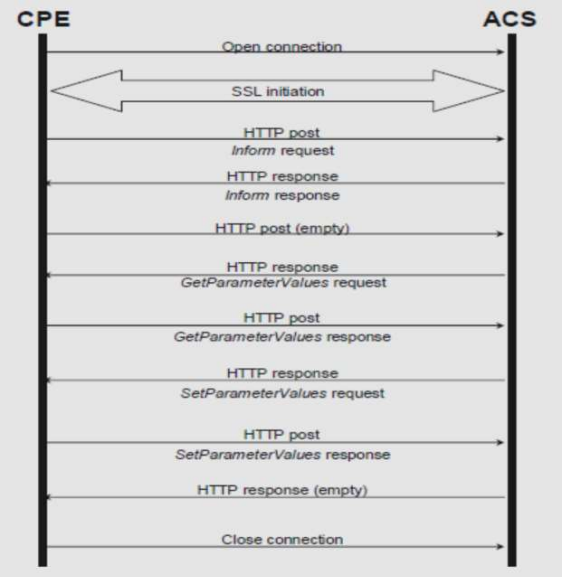

Figura 1.2. Ejemplo de una sesión entre un ACS y CPE [1]

1.3.3 MÉTODOS RPC

Se utiliza para la comunicación bidireccional entre el CPE y el ACS.

1.3.3.1 Mensajes RPC de línea de base

- **GetRPCMethods:** Este método puede ser utilizado por un CPE o ACS para descubrir el conjunto de métodos soportados por el ACS o CPE que está en comunicación.

1.3.3.2 Métodos CPE

Estos métodos deben ser soportados por el CPE, y solo un ACS puede solicitar cada uno de estos.

- **SetParameterValues:** permite al ACS modificar uno o varios de los valores de los parámetros de in CPE.
- **GetParameterValues:** permite al ACS obtener el valor de uno o más parámetros CPE.
- **GetParameterNames:** permite al ACS descubrir los parámetros accesibles en un CPE particular.

- **SetParameterAttributes:** permite al ACS modificar atributos asociados con uno o más parámetros CPE.
- **GetParameterAttributes:** permite al ACS leer los atributos asociados con uno o más parámetros CPE.
- **AddObject:** Este método puede ser utilizado por el ACS para crear una nueva instancia de un objeto de múltiples instancias.
- **DeleteObject:** Este método se utiliza para eliminar una instancia particular de un objeto.
- **Download:** Este método puede ser utilizado por el ACS para hacer que el CPE descargue un archivo especificado desde la ubicación designada.
- **Reboot:** Este método hace que el CPE se reinicie.

1.3.3.3 Métodos ACS

Estos métodos deben ser soportados por el ACS, y solo un CPE puede solicitar cada uno de estos.

- **Inform:** permite al CPE iniciar una secuencia de transacción cada vez que se establece una sesión con un ACS.
- **TransferComplete:** Este método informa a la ACS de la finalización, sea exitosa o no, de una transferencia de archivos iniciada por una llamada anterior de descarga, ScheduleDownload o Upload.
- **AutonomousTransferComplete:** Este método informa al ACS de la finalización de una transferencia de archivos que no fue específicamente solicitada por la ACS.

1.3.4 GATEWAY ASOCIADO

El protocolo de administración WAN CPE se puede utilizar para administrar remotamente los dispositivos CPE que están conectados a través de una LAN a través de un gateway. Cuando un ACS administra tanto un dispositivo como el gateway a través del cual está conectado el dispositivo, puede ser útil que el ACS pueda determinar la identidad de ese Gateway en particular.

1.3.5 MANEJO DE PROXY

CWMP se puede extender a dispositivos que no tienen un CPE incorporado, pero en su lugar admiten otro protocolo de gestión o Proxy Protocol. Un CPE Proxier es un CPE que admite un equipo final y también admite uno o más protocolos de proxy, por ejemplo, los servicios que incluyen UPnP DM, Z-Wave, entre otros. Un CPE Proxier utiliza estos Protocolos de Proxy para administrar los dispositivos conectados a él, es decir, los Dispositivos Proxiados.

La función del CPE Proxier es incorporar de manera transparente todos los elementos y mecanismos del Proxy Protocol. Independientemente de la implementación, el ACS administra el dispositivo Proxy a través de mecanismos CWMP y no tiene conocimiento de ningún comando Proxy Protocol que pueda utilizarse para completar las acciones solicitadas.

Con el fin de soportar una amplia gama de dispositivos, CWMP tiene dos formas de proporcionar funcionalidades de un dispositivo proxy; como un dispositivo virtual CWMP y utilizando un objeto embebido.

- Un dispositivo virtual CWMP se utiliza para modelar un dispositivo Proxied más complejo, como un puente, un enrutador, Set Top Box o dispositivos de tipo similar. El mecanismo virtual del dispositivo CWMP representa un dispositivo proxy con un punto final CWMP en el CPE Proxier.
- Un Objeto Embebido se utiliza para modelar un Dispositivo Proxied más simple, tal como un sensor binario, un interruptor de alimentación o dispositivos de tipo similar. El Mecanismo de Objetos Embebidos utiliza un Objeto Embebido u Objeto de Servicio dentro del propio Proxier CPE para representar el Dispositivo Proxy.

1.3.6 TR-181 MODELO DE DATOS DEL DISPOSITIVO PARA TR-069 [4]

El Broadband Forum define varios modelos de datos para su uso con el Protocolo de gestión WAN CPE. Estos modelos de datos contienen objetos y parámetros que describen las diferentes funciones y capacidades disponibles para los dispositivos y servicios que son manejables a través de CWMP. Los modelos de datos CWMP se dividen en dos tipos: Root y Service.

El modelo de datos Root, se utiliza para describir las funciones principales de un dispositivo compatible con la red, incluidas las interfaces, software o firmware, diagnósticos,

componentes comunes a CWMP y otros servicios, y la información básica del dispositivo necesaria para CWMP.

Los modelos de datos Service describen la funcionalidad modular que permite la extensión del modelo de datos root en un dispositivo para proporcionar servicios particulares, como un servicio de voz, un servicio de decodificador, almacenamiento conectado a la red.

Cada modelo de datos está definido por un nombre: sintaxis de versión. Un dispositivo define su modelo de datos definiendo un tipo de dispositivo, un documento XML que se correlaciona o importa objetos del modelo de datos oficiales de BBF (*Broad Band Forum*) y objetos específicos del proveedor.

Las primeras versiones del modelo de datos CWMP definían dos modelos de datos root, InternetGatewayDevice y Device. La versión 2 del modelo de datos root del dispositivo define el modelo de datos Device: 2. Esta nueva versión anula InternetGatewayDevice y Device: 1 a favor de un único modelo de datos root.

El modelo de datos Device:2, resuelve varias limitaciones presentadas por su antecesor, sin embargo, esta nueva generación del modelo de datos soporta las configuraciones de modelos anteriores, el cual se aplica a todos los tipos de dispositivos habilitados para TR-069, incluidos dispositivos finales, puertas de enlace residenciales y otros dispositivos de infraestructura de red.

Device:2, consiste en un conjunto de objetos de datos que abarcan cosas como información básica del dispositivo, configuración de hora del día, configuración de la interfaz de red y stack de protocolos, gestión de enrutamiento y bridging; y, estadísticas de rendimiento y pruebas de diagnóstico.

Las interfaces de red y las capas de protocolo se modelan como objetos de datos independientes que se pueden apilar, uno encima del otro, en cualquier configuración que un dispositivo pueda soportar.

La Figura 1.3 ilustra la estructura de modelo de datos Device: 2 de nivel superior. La Figura 1.4, la Figura 1.5 y la Figura 1.6 ilustran la estructura del modelo de datos con mayor detalle. [4]

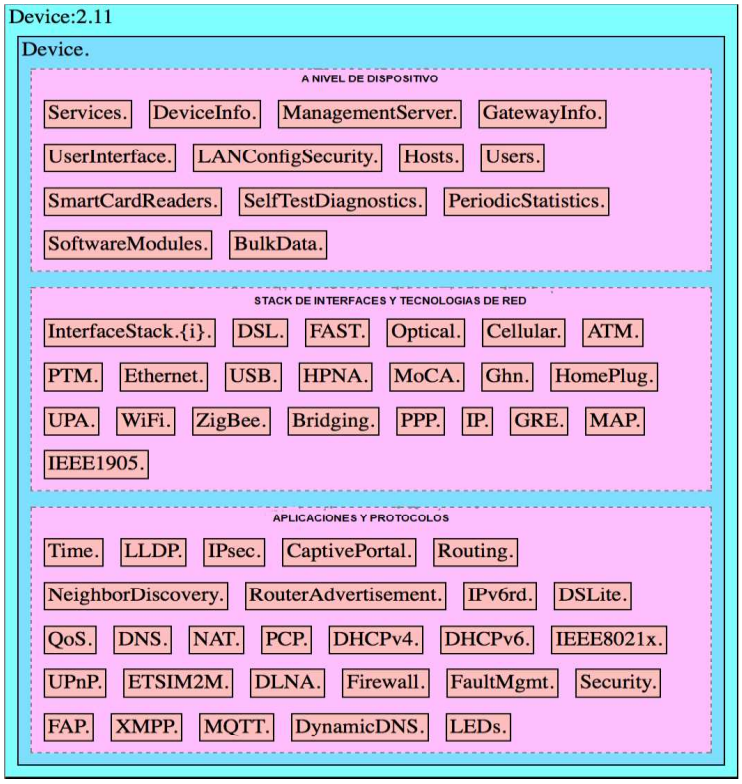

Figura 1.3. Estructura del modelo de datos-Resumen General

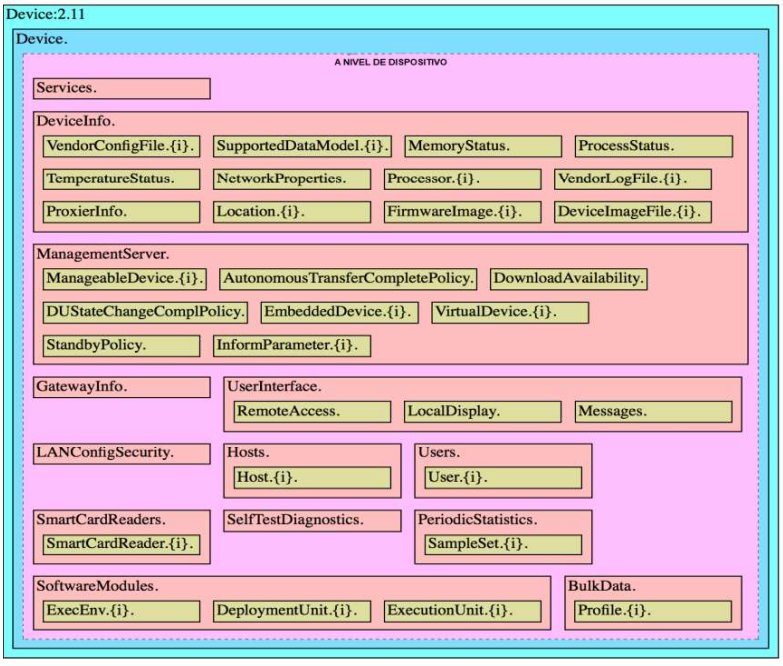

Figura 1.4. Estructura del modelo de datos-Nivel de Dispositivo

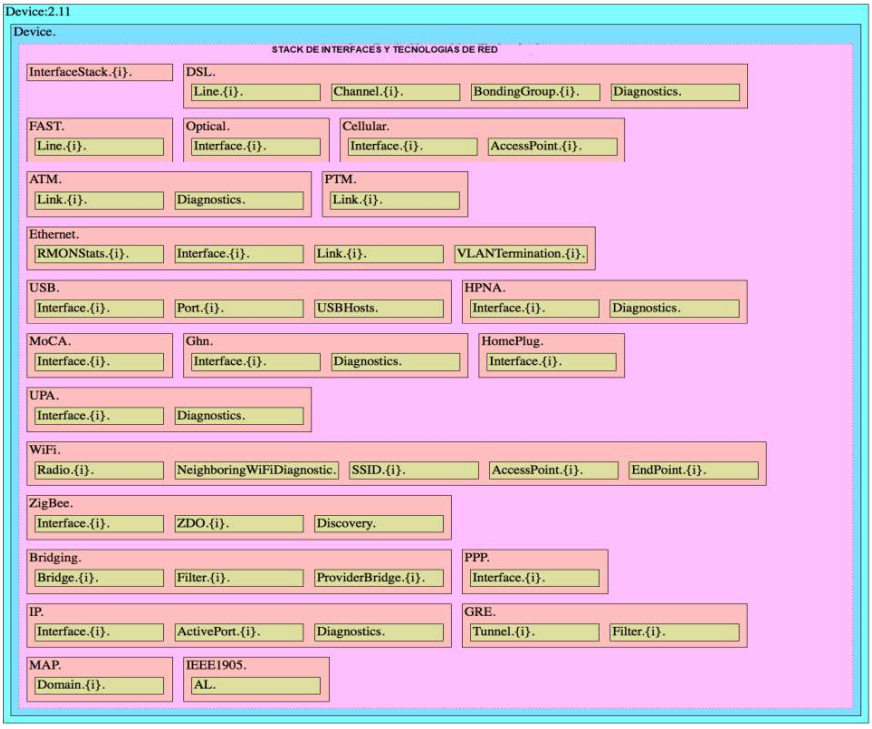

Figura 1.5. Estructura del modelo de datos-Pila de Interfaces y Tecnologías de Red

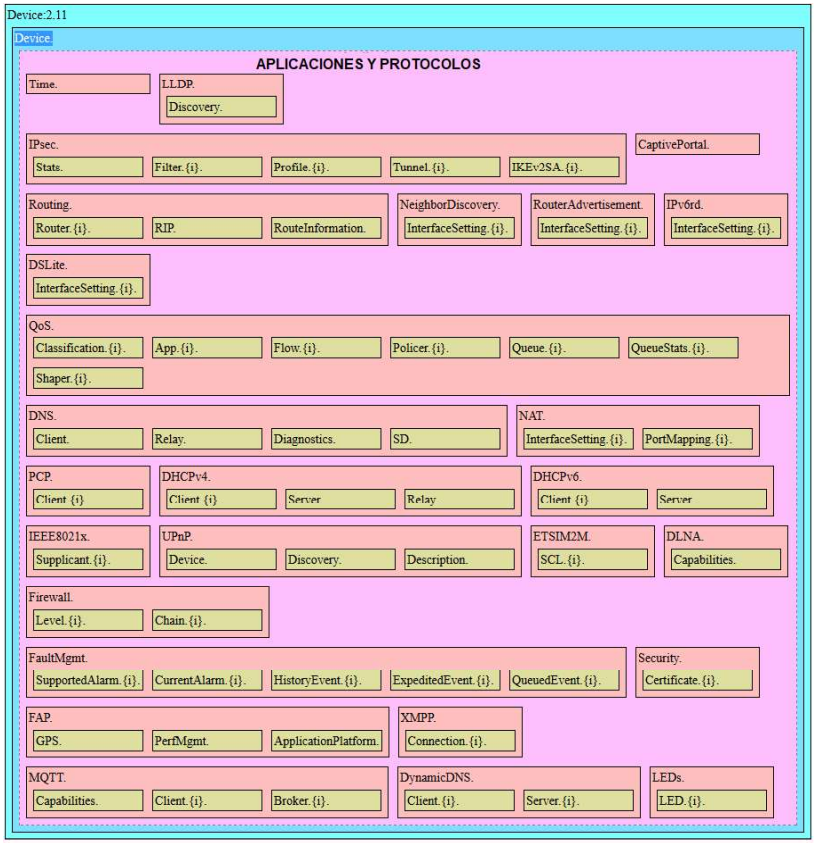

Figura 1.6. Estructura del modelo de datos-Aplicaciones y Protocolos

1.3.6.1 Arquitectura

1.3.6.1.1 Capa de Interfaces

Los objetos de interfaz o también conocidos como interfaces se pueden apilar, uno encima del otro, utilizando referencias de ruta para definir dinámicamente las relaciones entre sí.

Los objetos de interfaz están arbitrariamente restringidos a definiciones que operan, en o por debajo, de la capa de red IP. Sin embargo, pueden definirse objetos de interfaz específicos del proveedor que quedan fuera de este ámbito restringido. La capa OSI sirve como guía para poder ilustrar a qué nivel de la pila se espera que aparezca un objeto de interfaz. Sin embargo, un CPE no necesita soportar ni utilizar todas las interfaces. La Figura 1.7 enumera las interfaces definidas en el modelo de datos Device: 2.

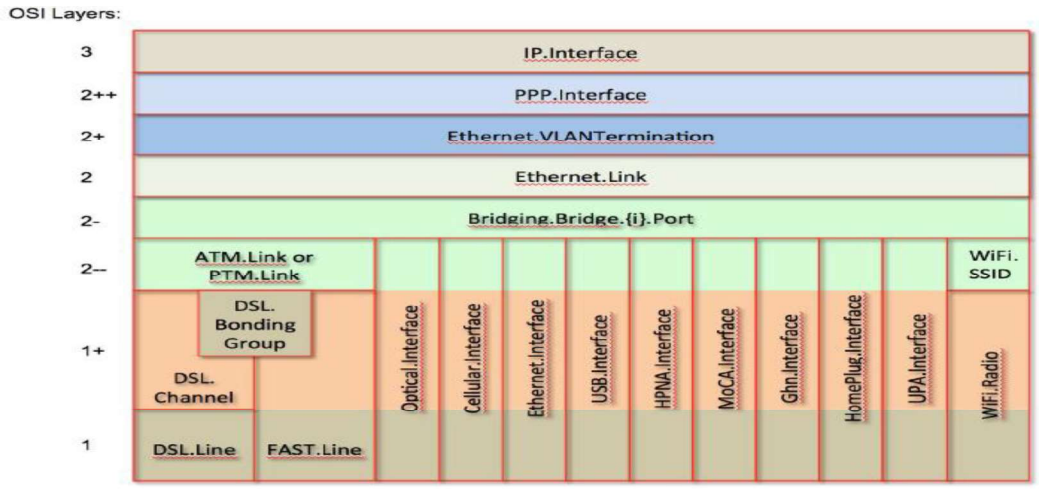

Figura 1.7. Capas Modelo OSI E Interfaces [4]

1.3.6.1.2 Objetos de Interfaz

Un objeto en un modelo de datos CPE es un elemento de funcionalidad que puede ser configurado por un ACS. Un Objeto contiene un conjunto básico de parámetros, que se configuran utilizando el RPC SetParameterValues. [5]

1.3.6.1.3 Capas inferiores

El parámetro LowerLayers es una lista separada por comas, de referencias de ruta a objetos de interfaz. Cada elemento de la lista representa un objeto de interfaz que se apila inmediatamente debajo de la interfaz de referencia, así se construye una jerarquía lógica de todas las relaciones de interfaz.

La capa más baja de una pila completamente configurada y operativa es generalmente la interfaz física. Dentro de estos objetos de interfaz física, el parámetro LowerLayers será una lista vacía, a menos que algunos objetos de interfaz específicos del proveedor de capa inferior estén definidos y presentes.

1.3.6.1.4 Status Administrativo y Operacional

Los parámetros Enable y Status de un objeto de interfaz especifican el estado administrativo y operacional actual de la interfaz, respectivamente. Los valores válidos para el parámetro Status son: Up, Down, Unknown, Dormant, NotPresent, LowerLayerDown y Error.

1.3.6.1.5 Objetos de interfaz específicos del proveedor

Si el ACS encuentra un objeto de interfaz desconocido para el proveedor dentro de la pila de interfaz de un CPE, en lugar de responder con un error, procede como si las interfaces de capa superior de este objeto estuvieran directamente vinculadas a sus interfaces de capa inferior. La Figura 1.8 ilustra el comportamiento descrito.

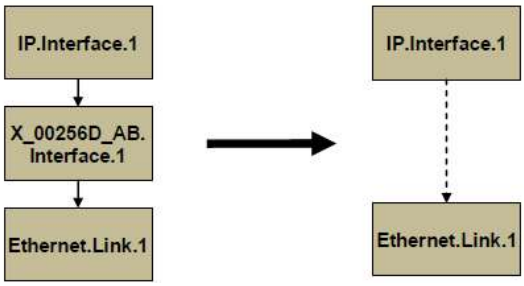

Figura 1.8. Objeto de interfaz apilado específico del proveedor que está siendo anulado por el ACS [4]

1.3.6.1.6 Tabla de InterfaceStack

La tabla InterfaceStack es un objeto del modelo de datos Device: 2. Esta es una tabla cuyas filas son generadas automáticamente por el CPE en función de los enlaces actuales que se configuran entre los objetos. Cada fila de la tabla representa un enlace entre un objeto

de interfaz de capa superior y un objeto de interfaz de capa inferior. La Tabla 1.2 muestra un ejemplo de una configuración de tabla de InterfaceStack.

Tabla 1.2. Ejemplo de una tabla de InterfaceStack

Row/Instance	Higher Layer Interface	Lower Layer Interface
1	IP.Interface.1	PPP.Interface.1
2	PPP.Interface.1	Ethernet.Link.1
3	Ethernet.Link.1	ATM.Link.1
4	ATM.Link.1	DSL.Channel.1
5	DSL.Channel.1	DSL.Line.1
6	IP.Interface.2	Ethernet.Link.2
7	Ethernet.Link.2	ATM.Link.2
8	ATM.Link.2	DSL.Channel.1
9	IP.Interface.3	Ethernet.Link.3
10	Ethernet.Link.3	Bridging.Bridge.1.Port.1
11	Bridging.Bridge.1.Port.1	Bridging.Bridge.1.Port.2
12	Bridging.Bridge.1.Port.2	Ethernet.Interface.1
13	Bridging.Bridge.1.Port.1	Bridging.Bridge.1.Port.3
14	Bridging.Bridge.1.Port.3	Ethernet.Interface.2
15	Bridging.Bridge.1.Port.1	Bridging.Bridge.1.Port.4
16	Bridging.Bridge.1.Port.4	WiFi.SSID.1
17	WiFi.SSID.1	WiFi.Radio.1

La Figura 1.9 muestra cómo se puede representar esta información. Las instancias de interfaz se representan mediante cuadros de color, mientras que las instancias de InterfaceStack se representan mediante círculos numerados.

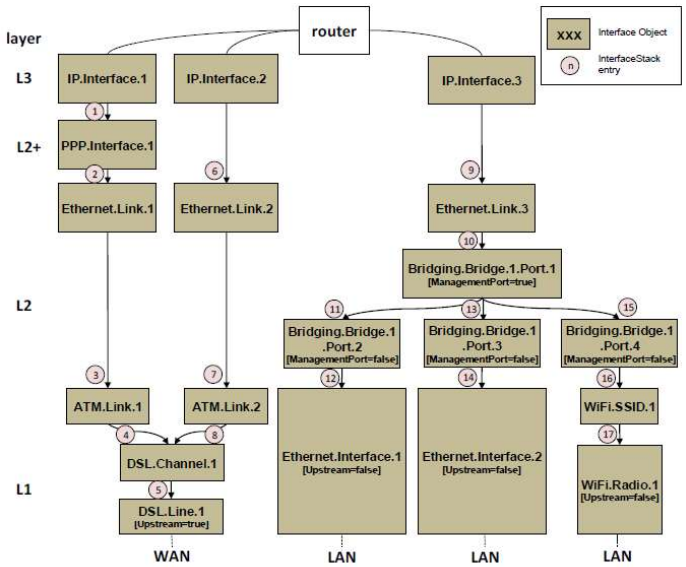

Figura 1.9. Representación de la información de la tabla InterfaceStack [4]

La Tabla 1.3 enumera cada instancia de interfaz y su valor de parámetro LowerLayers correspondiente.

Tabla 1.3. Instancia de interfaz y su valor de parámetro de LoweLayer

Interface	LowerLayers value
IP.Interface.1	PPP.Interface.1
IP.Interface.2	Ethernet.Link.2
IP.Interface.3	Ethernet.Link.3
PPP.Interface.1	Ethernet.Link.1
Ethernet.Link.1	ATM.Link.1
Ethernet.Link.2	ATM.Link.2
Ethernet.Link.3	Bridging.Bridge.1.Port.1
Bridging.Bridge.1.Port.1	Bridging.Bridge.1.Port.2, Bridging.Bridge.1.Port.3, Bridging.Bridge.1.Port.4
Bridging.Bridge.1.Port.2	Ethernet.Interface.1
Bridging.Bridge.1.Port.3	Ethernet.Interface.2
Bridging.Bridge.1.Port.4	WiFi.SSID.1

1.3.6.1.7 Operación Wi-Fi en el Modelo de Datos Device: 2

El modelo de datos Device: 2, puede admitir tecnologías multiradio, multibanda, haciendo uso del WiFi.Radio objects. Este objeto modela una radio inalámbrica 802.11 en un dispositivo.

Es importante tomar en cuenta que Wi-Fi puede operar en dos bandas de frecuencia diferentes:

- Tecnologías Wi-Fi basadas en el estándar IEEE 802.11b/g que operan en la banda de frecuencia de 2.4 GHz.
- Tecnologías Wi-Fi basadas en el estándar IEEE 802.11a/ac que operan en la banda de frecuencia de 5 GHz.
- Tecnologías Wi-Fi basadas en el estándar IEEE 802.11n que opera en la banda de frecuencia de 2.4 y 5 GHz.

Los dispositivos que únicamente operan en una sola frecuencia son conocidos como radios de banda simple, mientras que los dispositivos que operan en ambas frecuencias reciben el nombre de radios de doble banda.

Considerando que la banda de frecuencia es un parámetro importante, se define el SupportedFrequencyBands que es una lista de parámetros que contiene un elemento para las radios de banda simple (2.4GHz ó 5GHz), y dos elementos para las radios de banda doble (2.4GHz y 5GHz). También se define el OperatingFrequencyBand que indica que

banda de frecuencia está siendo utilizada. Esto se complementa con el SupportedStadards, que es una lista de los modos de la capa física soportados por un dispositivo, y también por el OperatingStandardas, que permite limitar la operación de un sub conjunto de modos de capa física.

1.3.7 SITUACIÓN ACTUAL DE LA EMPRESA PUNTONET

1.3.7.1 EMPRESA PORTADORA DE SERVICIOS PUNTONET [6]

La empresa PUNTONET es una organización que nació en el Ecuador en el año 2000 en respuesta a la demanda de los ecuatorianos de un servicio de Internet rápido y confiable, libre de errores e interferencias, y que asegure la confidencialidad de la información de los Usuarios.

La empresa provee un servicio de telecomunicaciones, que entrega soluciones de Internet a clientes personales y corporativos, pudiendo acceder desde su casa, negocio o empresa a todos los servicios y beneficios que brinda una conexión rápida y eficiente.

En la actualidad PUNTONET se encuentra entre los principales proveedores de telecomunicaciones a nivel nacional, cuenta con sucursales en: Quito, Cuenca, Guayaquil, Santo Domingo, Ibarra, Loja, Machala, Manta, Coca, Ambato, Esmeraldas, Riobamba y Portoviejo.

1.3.7.1.1 Servicios

La empresa cuenta con una de las redes físicas más robustas, la cual es administrada por personal técnico certificado de primer nivel, lo que permite contar con un amplio portafolio de productos diseñados para soportar las necesidades de conectividad de hoy y del mañana.

Gracias a esto la empresa brinda gran variedad de servicios tanto a clientes corporativos como a clientes de hogar.

1.3.7.1.1.1 Tecnología Corporativa

1.3.7.1.1.1.1 PuntoCloud

Cloud Computing es un modelo tecnológico que facilita las funciones de auto-aprovisionamiento, permitiendo entregar recursos de cómputo, red y software, dentro de una Infraestructura de alta tecnología, que puede ser accedida desde cualquier lugar del mundo a través de Internet.

El Centro de Datos de PUNTONET cumple con estándares internacionales que permiten ofrecer soluciones, altamente disponibles y confiables, entregando a sus clientes servicios de Cloud como IaaS (Infraestructure as a Service), SaaS (Software as a Service) y BaaS (Backup as a Service).

Algunas de las características de este servicio son:

- Estricta seguridad tanto física como lógica en el acceso.
- Solución FlexPod Datacenter para integración de componentes de mercado tales como son: NetApp, Cisco y VMware.
- Energía y temperatura regulada.
- Personal propio altamente calificado y certificado, en el manejo y operación de la Plataforma.
- Soporte técnico en línea.
- Asesoría en la implementación de nuevas soluciones y aplicaciones.

1.3.7.1.1.1.2 PuntoSat

La plataforma de PUNTONET está diseñada para adaptarse a las necesidades de cada empresa tanto en servicios de Transmisión de Datos como de Acceso a Internet. Los principales beneficios de la plataforma satelital son:

- Cobertura en localizaciones de difícil acceso en el territorio continental e insular.
- Conexión permanente 24 horas al día.
- Flota Satelital con tecnología de vanguardia.
- Equipamiento con los más altos estándares en la industria satelital.
- Soporte y monitoreo en línea.
- Telepuerto integrado a redes terrestres sobre plataforma MPLS.

- Priorización de tráfico y manejo de políticas de QoS.

1.3.7.1.1.1.3 PuntoEmpresarial

1.3.7.1.1.1.3.1 Transmisión de datos

El servicio de Transmisión de Datos ofrece el intercambio de información entre oficinas, locales y sitios remotos.

PUNTONET cuenta con una sólida plataforma tecnológica, así como la amplia experiencia de personal técnico en implementación de soluciones TI, garantizando los siguientes beneficios:

- Tecnología GPON que permite una mayor eficiencia y eficacia en el transporte de la información, garantizando la más alta velocidad.
- Red MPLS que ofrece una mayor escalabilidad y flexibilidad para adaptarse a los esquemas de tráfico que demanda el crecimiento tecnológico, ofreciendo servicios MPLS en capa 2 y en capa 3.
- Convergencia de aplicaciones (voz, datos y video) en una sola plataforma.
- Se cuenta con tecnologías de última generación que permiten una amplia interoperabilidad y compatibilidad, garantizando altos niveles de tráfico de datos y contenidos multimedia de manera confiable y segura.
- Centralización de operaciones y control de información.

1.3.7.1.1.1.3.2 Internet Corporativo

PUNTONET cuenta con Enlaces Redundantes de Alta Capacidad a los principales NAP's de Interconexión Local e Internacional, con lo cual garantiza a través de sus últimas millas de Fibra Óptica, Microonda y Satelital, los siguientes beneficios:

- Alta disponibilidad.
- Transferencia ilimitada de información.
- Ancho de Banda garantizado.
- Escalabilidad inmediata.
- Soporte para aplicaciones IPv4, IPv6 y NAT 64.
- Seguridad y confiabilidad en la transmisión de datos y recepción de la información.
- Soporte para creación y configuración de VPN´s.
- Baja latencia.
- Enlaces basados en tecnología de última generación que posibilita la integración a redes multiservicios.

1.3.7.2 Tecnología Hogares

En la actualidad, uno de los servicios que más crecimiento ha tenido debido a su gran utilidad es el servicio de internet. PUNTONET ofrece gran cobertura no solo en la ciudad de Quito, sino también en sus valles aledaños, con velocidades acorde a las necesidades de clientes hogar. Posee también un nivel de compartición de 6 a 1, nivel muy competitivo en el actual mercado.

1.3.8 RADIO ENLACES RESIDENCIALES

Los enlaces para hogares de PUNTONET, en su gran mayoría utilizan como última milla enlaces de radio con equipos de la marca Mikrotik, la topología utilizada en este tipo de servicios, son enlaces punto multipunto, donde se crean AP's virtuales que permiten la creación de canales exclusivos para los clientes, dichos canales son establecidos dependiendo de la frecuencia que se esté utilizando en el rango desde los 5150 hasta los 5350 MHz. En el domicilio del suscriptor se instala una antena y el Internet dentro de del domicilio es direccionado por medio de un router. En la figura 1.10 se muestra la operación de los enlaces inalámbricos de banda ancha de PUNTONET

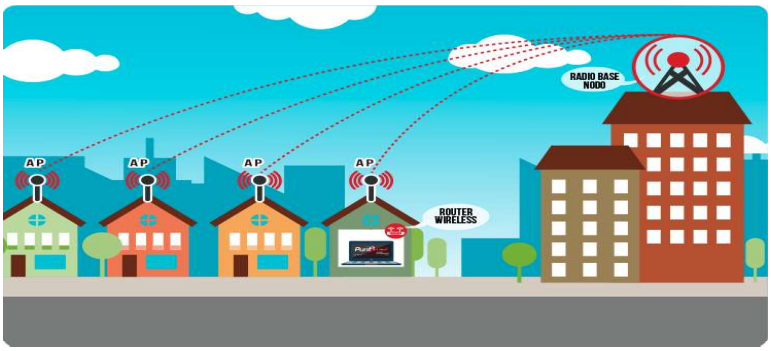

Figura 1.10. Enlaces Inalámbricos de Banda Ancha [6]

1.3.8.1 Enlaces sobre la plataforma Mikrotik [7]

Los enlaces sobre plataforma Mikrotik se basan en la familia de normas 802.11x en las bandas no licenciadas de 2,4 y 5,8 GHz. Lo que distingue a estos dispositivos es el software que los controla llamado RouterOS.

Los dispositivos se configuran, revisan y diagnostican mediante la herramienta Winbox, esta ofrece una sofisticada interfaz gráfica para el sistema operativo RouterOS. El software también permite conexiones a través de FTP y Telnet, SSH y acceso Web para su administración, donde se puede observar datos importantes como: niveles de señal, CCQ, tráfico, logs del sistema entre otros.

Las capacidades de los enlaces se ven afectadas debido a la interferencia y a la distancia del entre los puntos, dependiendo de esos factores, y ya que se trata una banda de frecuencia libre, se han alcanzado capacidades de 28 Mbps en enlaces que no superan los 15 Km de distancia y de entre 12 y 20 Mbps en enlaces que superan los 40 Km.

1.3.8.1.1 Mikrotik RouterOS [8]

Mikrotik RouterOS es el sistema operativo del hardware Mikrotik RouterBOARD, que tiene las características necesarias para un ISP: firewall, Router, MPLS, VPN entre otras. Es un sistema operativo independiente basado en el Kernel de Linux v3.3.5, el cual puede ser instalado en PC´s y otros dispositivos de hardware compatibles con x86.

1.3.8.2 Mikrotik RouterBOARD

Es una familia de soluciones de hardware con circuitos diseñados por Mikrotik, que se caracterizan por incluir su sistema operativo RouterOS y actualizaciones. Las arquitecturas que soporta son: mipsbe, smips, tile, ppc, arm, x86, mipsle.

1.3.9 PROCESOS DE SOPORTE TÉCNICO TELEFÓNICO [9]

El soporte telefónico está orientado a ofrecer soluciones y asesorías técnicas de alto nivel que satisfagan las necesidades de clientes con servicio de internet para hogares, con la máxima cantidad de clientes atendidos telefónicamente, dando seguimiento a requerimientos técnicos y de atención al cliente.

1.3.9.1 Responsabilidades de los miembros del área de Soporte Técnico Telefónico.

Dentro del área de soporte técnico telefónico para hogares, existen varios roles que son importantes, para el desarrollo de las actividades de soporte y resolución de requerimientos, estos roles son:

- Supervisor.
- Líder.
- Agentes de Soporte Técnico Telefónico.

1.3.9.1.1 Es responsabilidad del Supervisor

- Generar reportes de llamadas y visitas técnicas.
- Generar reportes de llamadas por nodo y sucursales.
- Generar reportes de notificaciones Facebook.
- Generar reportes de indicadores por agente.
- Capacitación sobre aspectos encontrados en la supervisión, evaluar y realizar seguimiento.
- Retroalimentar al personal en soporte técnico y procesos, dar tips y proponer mejoras.
- Realizar controles a manera de monitoreo de llamadas para detectar mejoras en el trato al cliente y soporte técnico.
- Resolver los problemas del área.
- Asegurar que todos los elementos de trabajo se encuentren disponibles y funcionando.

1.3.9.1.2 Es responsabilidad de Líder

- Generar reporte de porcentaje de llamadas por agente.
- Generar reporte por agente en fin de semana.
- Monitorear llamadas del personal a su cargo.
- Retroalimentar al personal sobre procedimientos y temas técnicos.
- Realizar un continuo seguimiento de los casos escalados a otros niveles y verificar que se brinde solución.
- Atender y solucionar los casos reportados por diferentes canales como: Facebook, correos electrónicos.
- Enviar un reporte de novedades al final del turno.

1.3.9.1.3 Es responsabilidad de Agente de Soporte Técnico Telefónico

- Brindar un soporte técnico adecuado a los clientes de internet masivo.
- Cumplir con los protocolos establecidos en el área de call center masivo.
- Registrar la información en los sistemas manejados por la empresa.
- Cumplir con el proceso de devolución de llamadas por correo y requerimientos escalados.

1.3.9.2 Descripción del Soporte Técnico Telefónico

En la figura 1.11 se muestra el diagrama de flujo del soporte técnico telefónico para los clientes con servicio de internet para hogares.

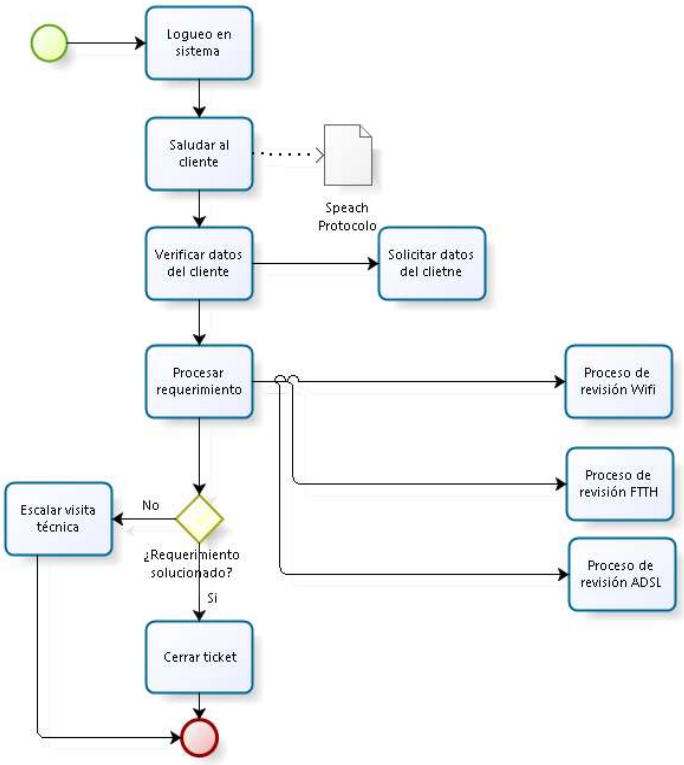

Figura 1.11. Diagrama de flujo del soporte técnico telefónico para los clientes con servicio de internet para hogares. [9]

1.3.9.2.1 Logueo

El agente de soporte técnico telefónico debe registrar su disponibilidad de atención en el sistema de monitoreo de llamadas, con el número de agente y la extensión telefónica asignadas.

1.3.9.2.2 Saludar al cliente

El agente debe brindar un saludo amable, con tono de voz firme, e identificarse antes de empezar con el soporte de acuerdo a la guía proporcionada a cada miembro del área.

1.3.9.2.3 Verificar Datos

Ubicar la información del cliente utilizando filtros como: número de cedula del titular, nombres completos, código del contrato.

1.3.9.2.4 Procesar Requerimiento

Una vez que se identifica al cliente se procede con las revisiones.

1.3.9.2.4.1 Identificar Tecnología

Una vez identificado el problema se debe verificar que tipo de tecnología tiene instalado el cliente, para realizas las pruebas correspondientes a cada tecnología. Existen tres tipos de tecnologías a través de las cuales se proporciona el servicio de internet, las cuales son: Inalámbrica, fibra óptica y línea telefónica.

1.3.9.2.5 Proceso de devolución de llamadas

Esto ocurre cuando la solución a un requerimiento, requiere un mayor tiempo de gestión, cuando existen solicitudes de soporte a través de correos electrónicos provenientes de la misma área de soporte técnico telefónico, o provenientes de otros departamentos.

1.3.9.2.6 Solución

Una vez terminado el proceso de revisión se procede a cerrar o escalar el ticket dependiendo qué tipo de solución requiera el cliente.

1.3.9.2.7 Cerrar Ticket

Se procede a cerrar el ticket una vez que se dio solución al requerimiento del cliente en el soporte telefónico, indicando el problema que reportó, el detalle de las pruebas y la solución que se dio al caso.

1.3.9.2.8 Escalar Visita Técnica

Se procede a escalar una visita técnica, después de haber realizado las pruebas vía telefónica y no haber podido dar una solución al requerimiento del cliente.

1.3.9.3 Soporte para clientes de hogar

A continuación, se detalla el proceso que debe seguir un agente de soporte técnico telefónico para clientes de hogar, al momento de realizar las revisiones en los enlaces de internet, cuando un cliente reporta problemas con el servicio.

1.3.9.3.1 Revisar el estatus del servicio del cliente.

- Ubicar la IP del cliente a través de la MAC o código en la base de datos.
- Hacer pruebas de conectividad hacia la IP del cliente.
- Ingresar al AP o al OLT, al cual se engancha el cliente.
- Hacer pruebas de primer nivel con el cliente.

1.3.9.3.1.1 Pruebas de Primer nivel

- Confirmar que los equipos estén correctamente conectados a la energía eléctrica y encendidos.
- Verificar que el cableado esté conectado en los puertos adecuados.
- Después de las pruebas no se tiene respuesta del enlace se gestiona una visita técnica.
- Verificar si existen cortes de última milla con un ping extendido a la IP del cliente.

Todas estas revisiones se realizan únicamente cuando el cliente se pone en contacto con el soporte técnico. Con la información obtenida de las revisiones realizadas se puede determinar si un cliente requiere o no una visita técnica

1.3.9.4 DATOS ESTADÍSTICOS DE INSTALACIONES Y VISITAS TÉCNICAS

En la tabla 1.4 se muestra el número total y el porcentaje de las instalaciones realizadas en cada sucursal, donde se incluyen traslados, migraciones, reactivaciones, reinstalaciones e instalaciones nuevas. Se observa que el 60% corresponde al aporte de la sucursal de Quito, donde se enfocara el análisis para el desarrollo de este proyecto. Dicho porcentaje es el mismo en los diferentes puntos que se tomaran en cuenta en este apartado.

Tabla 1.4. Instalaciones a nivel nacional [10]

SUCURSAL	ACTIVACIONES	%
Quito	14610	60%
Guayaquil	2783	11%
Cuenca	2233	9%
Santo Domingo	3289	13%
Manta	1519	6%
TOTAL	24434	100%

A lo largo del año se ha conseguido la instalación del servicio de internet para hogares a nuevos clientes con las diferentes tecnologías proporcionadas por PUNTONET.

La tabla 1.5 muestra el número total de instalaciones realizadas a lo largo del año 2017 a nivel nacional a nuevos clientes.

Tabla 1.5. Instalaciones de nuevos clientes [10]

MES	ADSL HOME	Fibra Home	Internet	Wifi	TOTAL	%
ENERO	1	91	1	1262	1355	7%
FEBRERO	1	156	5	1238	1400	7%
MARZO	3	134	6	1562	1705	8%
ABRIL		194	2	1331	1527	7%
MAYO	3	237	4	1461	1705	8%
JUNIO	2	294	3	1945	2244	11%
JULIO	2	328	2	1694	2026	10%
AGOSTO		251		1090	1341	7%
SEPTIEMBRE	1	313	2	1574	1890	9%
OCTUBRE	3	361		1747	2111	10%
NOVIEMBRE		296		1432	1728	8%
DICIEMBRE		185		1360	1545	8%
TOTAL	16	2840	25	17696	20577	

En la tabla 1.6 se muestra el tiempo de gestión de vistas técnicas de clientes con tecnología de Radio Enlace (Wifi) a nivel nacional.

Tabla 1.6. Tiempo de gestión de vistas técnicas de Radio [10]

VISITAS POR DÍA NACIONAL-WIFI	
DÍAS	**VISITAS**
0 a 1 días	2794
2 a 3 días	6604
4 a 5 días	5429
6 a 7 días	2985
8 a 9 días	1630
10 a 11 días	1187
Más de 12 días	2959
TOTAL GENERAL	**23588**

En la figura 1.12 se muestra el gráfico correspondiente a la tabla 1.6

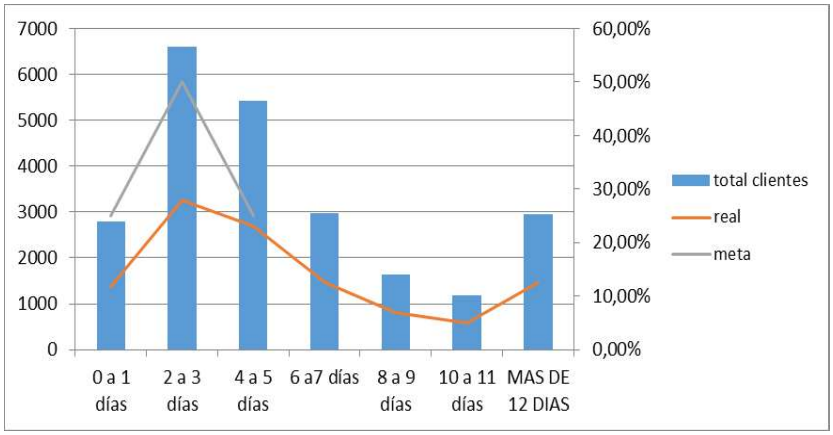

Figura 1.12. Tiempos de gestión de visitas técnicas de radio [10]

A lo largo de los procesos de instalación, soporte telefónico y visitas técnicas, no se tiene un procedimiento para mantener actualizados los firmwares de las radios a través de los cuales se proporciona el servicio de internet a los clientes home con tecnología inalámbrica. Por tal motivo este proyecto busca proporcionar una solución a esta problemática, de tal manera que las visitas que se generan por inestabilidad del enlace, a consecuencia de la falta de actualización de las versiones de los firmwares, se reduzcan, ya que por el tipo de causal son atendidas en un periodo de 4 a 7 días. Siendo este lapso de tiempo un motivo de insatisfacción en los clientes, que en algunos casos deciden suspender el servicio, generando pérdidas a la empresa.

El desarrollo de este proyecto permitirá determinar si es factible tanto técnica como económicamente el despliegue de un sistema de administración a través del enlace WAN, con el cual se pueda realizar, la descarga y actualización de las versiones de los firmwares de las antenas Mikrotik instaladas, haciendo uso de un servidor a través del cual se pueda realizar dicha actualización a gran escala, siendo transparente para los suscriptores.

2. METODOLOGÍA

En la fase teórica del proyecto, se recolectó toda la información sobre CPE WAN Management Protocol (CWMP), en el cual se definen varios modelos para el uso del protocolo TR-069, que contienen objetos y parámetros que describen las muchas funciones y capacidades disponibles para los dispositivos y servicios que son manejables a través de CWMP.
Dentro de la fase del desarrollo, se analizó la arquitectura del protocolo, que permitió conocer los componentes, mecanismos de seguridad y parámetros más relevantes. Así también se revisaron los requerimientos y procesos de los que depende dicho protocolo.

En la fase de análisis, se consideró que actualmente el proceso de revisión y diagnóstico de los equipos de última milla Mikrotik SXT y Metal se los realiza a través del sistema Winbox, únicamente cuando el cliente se pone en contacto con el soporte técnico, en este sistema se pueden observar los eventos que se producen en el equipo, y, además, se pueden realizar configuraciones y ajustes para solventar algún problema presentado.

La empresa puso a disposición los dispositivos con los que se trabajó, los mismos que fueron revisados para descartar problemas en su funcionamiento o para reemplazarlos en caso de existir algún desperfecto. Se actualizaron los sistemas operativos de los dispositivos para tener compatibilidad con el protocolo TR-069. De manera simultánea se seleccionó el software del servidor de configuración automática (ACS) con el cual se va a gestionar las diferentes funcionalidades que se configuran en los CPE. La instalación del ACS se llevó a cabo siguiendo las recomendaciones de los desarrolladores del software. Con el apoyo de la documentación de la norma TR-181, se establecieron parámetros de configuración de los equipos, tomando en cuenta las recomendaciones del fabricante y lo establecido en la norma.

La última fase, fue la fase validación, en donde se ensambló un entorno de red controlado, donde se pudo emular el comportamiento de los equipos, y adicionalmente se pudo evaluar la interacción entre CPE y ACS. Todo permitió que se puedan elaborar pruebas de funcionamiento, análisis de resultados, ajustes de configuración, para determinar si el sistema de gestión y administración remota es compatible. Con la información obtenida se establecieron conclusiones, las que darán un indicativo de cuán factible sería la implementación de este sistema, dentro de la red de PUNTONET.

Al finalizar el estudio se hizo una evaluación comparativa entre el modelo de gestión y administración de los CPE's que actualmente se utiliza y el modelo gestión con el protocolo TR-069. Esta comparación consistió en observar el número de visitas técnicas generadas, tiempos de respuesta, y tiempo de solución del problema, con el fin de contrastar los beneficios técnico-económicos de cada uno de los modelos. Además, se pudo estimar un porcentaje de reducción de los costos, tomando en cuenta los diferentes elementos involucrados al momento de enviar visitas técnicas y los cuales podrían ser optimizados, entre ellos: combustible, horas hombre, tiempo de atención de una incidencia, tiempo solución del problema.

2.1 INTERACCIÓN ENTRE EL SERVIDOR ACS Y CPE'S

2.1.1 SISTEMA DE ADMINISTRACIÓN WAN CPE FREEACS [11]

2.1.1.1 Introducción

Esta herramienta ofrece una solución para los proveedores que desean aumentar la satisfacción del cliente y disminuir los costos de mantenimiento, al mismo tiempo que se mantienen abiertos a las futuras posibilidades de servicio permitiendo a los proveedores controlar y monitorear de forma remota el equipo local de las instalaciones (CPE) y los servicios que brinda.

2.1.1.2 Despliegue

FreeACS se puede implementar dentro de una red corporativa, donde el proveedor asume los roles de Administrador y Operador. FreeACS se está implementando como parte de una solución hospedada ofrecida por Hosting Providers, esto significa que el proveedor de alojamiento asume el rol de ser el administrador, mientras que el proveedor virtual solo tiene la función de ser el operador. Esto reduce los requisitos tanto de habilidades técnicas, como las horas-hombre del Operador, ya que el Administrador de FreeACS maneja las tareas más complejas como escalabilidad, copias de seguridad de bases de datos, equilibrio de carga.

2.1.1.3 Aprovisionamiento

FreeACS fue construida como una solución a gran escala que permita ser escalable, proporcionando una funcionalidad de aprovisionamiento enriquecida en múltiples protocolos de administración de dispositivos como el estándar TR-069 y estándares como

HTTP (S) / TFTP. Esto lo convierte en un sistema que puede manejar una gran variedad de CPE's en el mercado.

2.1.1.3.1 TR-069

Esta es la columna vertebral del sistema, ya que la mayoría de los CPE's en la actualidad, son compatibles con TR-069, por tal motivo, es fundamental que el servidor sea flexible con los dispositivos. El servidor TR-069 ofrece algunas funcionalidades que se pueden activar o desactivar dependiendo del dispositivo que se conecta al sistema. El servidor en sí está diseñado para acomodarse con respecto a la entrada del dispositivo. En un entorno TR-069 se utilizan mensajes HTTP o HTTPS y SOAP, lo que permite que los mensajes pasen a través de firewalls y puertas de enlace NAT.

2.1.1.4 Monitoreo

2.1.1.4.1 Syslog

FreeACS ofrece un servidor syslog que le permite recopilar información de los dispositivos y almacenarlos en la base de datos. Esta información puede ser extremadamente útil si ocurre un error en el dispositivo.

2.1.1.4.1.1 Informe

El informe permite obtener una descripción general de la situación o puede ampliar los detalles para identificar que dispositivos son responsables de los mensajes.

2.1.1.5 Interfaces

2.1.1.5.1 Web

La interfaz Web permite al operador de red realizar varias acciones dentro de FreeACS, entre ellas:

- Visualizar informes.
- Validar de trabajos ejecutados.
- Almacenar de archivos en la base de datos.
- Configurar fácilmente tareas complejas y programas de aprovisionamiento.

2.1.1.5.2 Shell

La interfaz de Shell se asemeja en cierta medida a un Shell de Unix normal, viene incorporado con un sistema de ayuda para facilitar el aprendizaje mientras se escribe. Esta interfaz permite automatizar y realizar operaciones que requieren un mayor procesamiento.

2.1.1.5.3 Servicios web

Permite la integración de los dispositivos al servidor ACS.

2.1.1.5.4 Telnet

FreeACS ofrece la posibilidad de ejecutar scripts de Telnet en un conjunto de dispositivos. Con estos scripts, es posible ejecutar comandos, además de leer datos del dispositivo.

2.1.1.6 Arquitectura

FreeACS es una arquitectura a base de módulos, que tienen un funcionamiento independiente y comunican a través de una base de datos central. Todos los módulos se implementan en Java y se crean para ejecutarse en cualquier middleware compatible con JEE (*Java Plataform, Enterprise Edition*, Java Empresarial) Esto proporciona un sin número de soluciones, para implementaciones distribuidas en redes, plataformas de hardware y sistemas operativos. Estos módulos dependen de MySQL para la base de datos, que se carga, como parte del procedimiento de instalación.
FreeACS está constituido por tres tipos de módulos: de lado sur, de lado norte y de *Core* o módulos centrales.

2.1.1.6.1 Módulos de Lado Norte

Los módulos del lado norte generalmente solo se pueden acceder desde la red corporativa, sin embargo, la interfaz web y los servicios web de FreeACS, pueden ser de acceso público, siempre que estén debidamente protegidos. Los módulos de lado norte implementan interfaces de administración para FreeACS, que son los siguientes:

- FreeACS Web: una interfaz web para la gestión de CPE’s basada en navegador.

- FreeACS Web Services: una API[4] (*Application Programming Interface*, Interfaz de Programación de Aplicaciones), SOAP o WSDL[5] (*Web Services Description Language*, Lenguaje de Descripción de Servicios Web) para la integración con una infraestructura de terceros.
- FreeACS Shell: una interfaz de scripting de alto nivel para la gestión de CPE.

2.1.1.6.2 Módulos Core

Los módulos Core de FreeACS siempre deben ubicarse en la red corporativa, la cual es una red privada y segura en cada ISP, inaccesible desde Internet. Estos módulos interactúan solo con la base de datos de FreeACS y actúan de acuerdo con los cambios y las solicitudes insertadas allí por otros módulos. Los módulos Core manejan la automatización y la supervisión del progreso de las tareas, la vigilancia de los módulos del servidor y otras tareas de mantenimiento, además facilita a los operadores de red, la realización de cambios en la implementación de los CPE y en las tareas de monitoreo.
Los módulos Core son:

- FreeACS Core Server: control avanzado de trabajos, generación de informes, ajuste syslog.
- Servidor FreeACS Monitor: un monitor que verifica el funcionamiento correcto de otros módulos FreeACS.

2.1.1.6.3 Módulos de Lado Sur

Los módulos de lado sur, permanecen expuestos a Internet, y son responsables de la comunicación con los CPE. Estos módulos se implementan para brindar estabilidad, mejorar el rendimiento, y proporcionar escalabilidad, además aseguran que las características de FreeACS funcionen a través de protocolos. Los protocolos de gestión soportados actualmente son:

- Servidor FreeACS TR-069: protocolo de aprovisionamiento estándar de la industria.
- Servidor FreeACS SPP (*Simple Provisioning Protocol*, Protocolo de Aprovisionamiento Simple), maneja HTTP, HTTPS, TFTP y Telnet.

---

[4] API: son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. [23]

[5] Está pensando para desarrolladores que desean exponer sus propios servicios al uso de otros. [25]

Los módulos individuales de FreeACS, se comunican entre sí exclusivamente a través de la base de datos. En la figura 2.1 se muestra una ilustración de la arquitectura FreeACS:

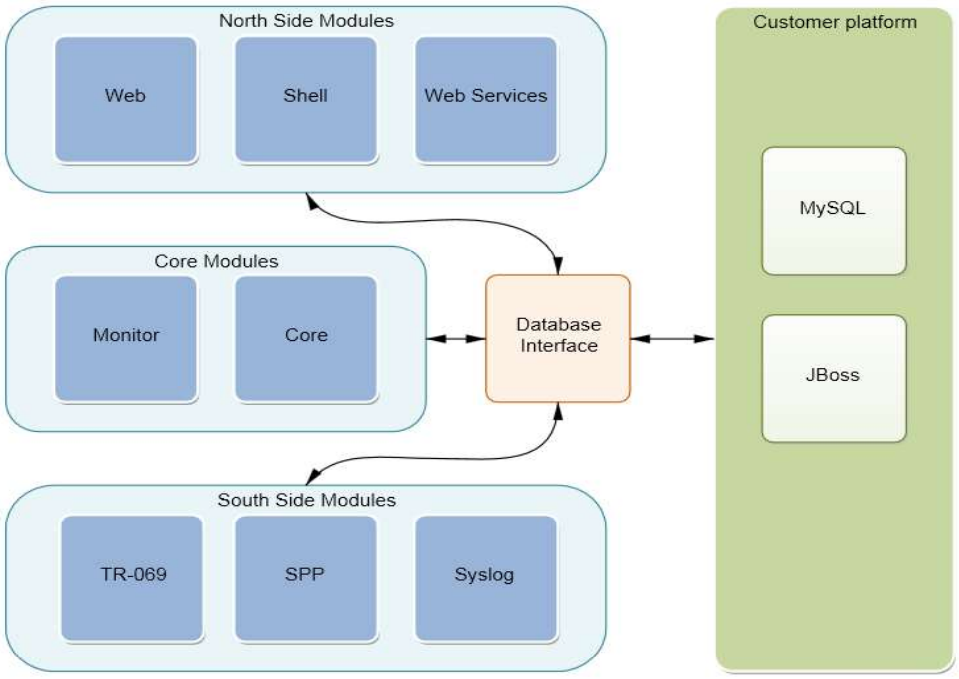

Figura 2.1. Plataformas de FreeACS [11]

2.1.1.7 Servidores

2.1.1.7.1 Base de datos FreeACS

Actualmente el FreeACS puede ejecutarse en una base de datos: MySQL. Este motor de base de datos ha funcionado muy bien, sin muchos ajustes.

2.1.1.7.2 FreeACS Web

En FreeACS Web puede crear, actualizar, eliminar y administrar un conjunto de Unidades. Todas las Unidades están asociadas con un Perfil y todos los Perfiles están asociados con un Tipo de Unidad. Estos conceptos son una parte crucial de cómo opera la web. Está dirigida tanto a la atención al cliente como a la administración del sistema. La GUI permite realizar una amplia gama de operaciones en FreeACS, tales como:

- Buscar unidades.
- Trabajar con unidades, perfiles y tipos de unidades.

- Trabajar con grupos y trabajos.
- Buscar y ver mensajes syslog.
- Gestión de software.
- Aprovisionamiento del modo de inspección.
- Asistentes para ayudar con operaciones comunes.

2.1.1.7.2.1 Creación de Unidades

Cuando se inicia sesión por primera vez no hay unidades, ni tipos de unidades ni perfiles, para aprisionar un dispositivo se debe realizar lo siguiente:

- Crear primero un nuevo Tipo de Unidad, donde se definen los parámetros para todos los Perfiles que serán creados.
- Crear un nuevo perfil, que contiene un conjunto de valores compartidos entre todas las unidades que serán creadas.
- Crear una nueva unidad dentro del perfil. Esta unidad es una representación lógica de un CPE físico, que permite a un operador cambiar la configuración de CPE.
- Agrupar varias unidades, en un grupo, con uno o varios parámetros en común.

2.1.1.7.2.2 Parámetros

La información que un dispositivo ofrece a un servidor de aprovisionamiento es un conjunto de parámetros que se pueden leer y escribir. Las siguientes reglas se aplican al nombre de un parámetro:

- El conjunto de parámetros en el dispositivo es descubierto por el servidor TR-069, para que pueda leerse o escribirse en el dispositivo.
- Todos los parámetros necesarios del sistema se crean o actualizan, una vez que se crea o modifica un tipo de unidad.
- No se debe insertar espacios ni caracteres extraños.
- Se usa puntos para separar palabras o jerarquía.

La tabla 2.1 muestra la descripción de parámetros.

Tabla 2.1. Descripción de parámetros

	Comentario	**Ejemplo**
Nombre	Un nombre que identifica de manera única este parámetro. El nombre del parámetro debe definirse en el tipo de unidad en FreeACS antes de utilizarse en cualquiera de estos servicios.	Device.DeviceInfo.SerialNumber
Valor	El valor podría ser una cadena de hasta 1024 caracteres.	Hola Mundo
Indicador	Los indicadores son información para indicar al servidor cómo usar el parámetro, pero también pueden ser utilizados por el cliente. Los indicadores en uso dependen del servicio, para: • addOrChangeUnitRequest • addOrChangeUnittypeRequest • addOrChangeProfileRequest los indicadores pueden ser AC (AddChange) o D (Delete). para: • getUnitsRequest los indicadores pueden ser EQ (Equal) o NE (Not Equal), LT (Menor que), LE (Menor o igual), GE (Mayor o Igual) o GT (Mayor que). para: • getUnitsResponse Los indicadores pueden ser P (Perfil) o U (Unidad).	AC

2.1.1.7.2.2.1 Indicadores obligatorios

- Si un parámetro es del dispositivo, debe tener el indicador R o RW.
- Para todos los parámetros del sistema, debe especificar el indicador X.

2.1.1.7.2.2.2 Indicadores opcionales

- El indicador seleccionable (S), se establece para permitir la búsqueda de un parámetro específico en la sección Avanzado de la página Búsqueda.
- El indicador desplegable (D), se establece siempre que desee visualizar este parámetro particular en la lista de unidades de la página Búsqueda.

- El indicador confidencial (C), se configura para ocultar los parámetros confidenciales, otro uso importante de este indicador permite al servidor TR-069 saber, cuándo esperar que vuelvan los valores en blanco desde el dispositivo.
- El indicador inspección (I), se especifica en los parámetros que desea inspeccionar en caso de una llamada de cliente. Estos parámetros siempre se borran de FreeACS DB tan pronto como se cierra el modo de inspección.
- El indicador Leer a Distancia(A), se especifica cuando el parámetro siempre se leerá desde el dispositivo.

La figura 2.2 muestra varios parámetros y sus correspondientes indicadores.

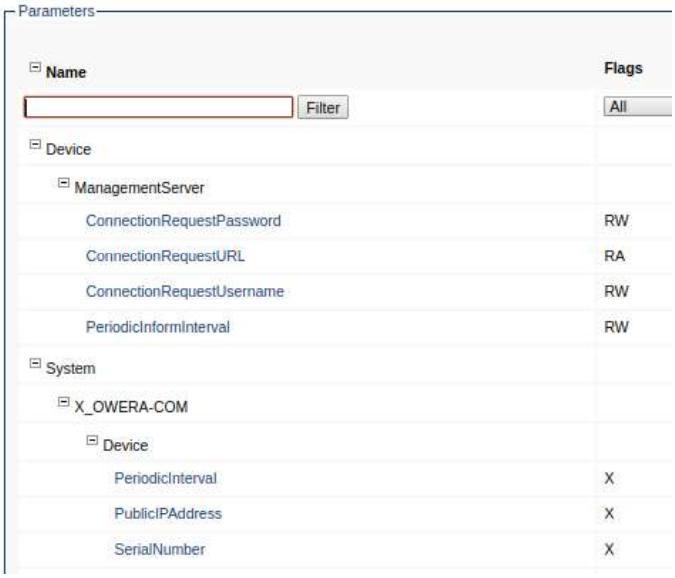

Figura 2.2. Parámetros y sus indicadores

2.1.1.7.2.3 La configuración rápida

Primero se debe asegurar que los dispositivos sean descubiertos, para que tenga un Tipo de Unidad. Luego, en el perfil predeterminado se agregan todas las unidades que desea aprovisionar. Se especifican los valores del parámetro de perfil para todos los valores comunes. Se conectan los dispositivos al servidor de suministro y el sistema debe proporcionar a los dispositivos la configuración correcta. La figura 2.3 muestra el resumen de una unidad aprovisionada

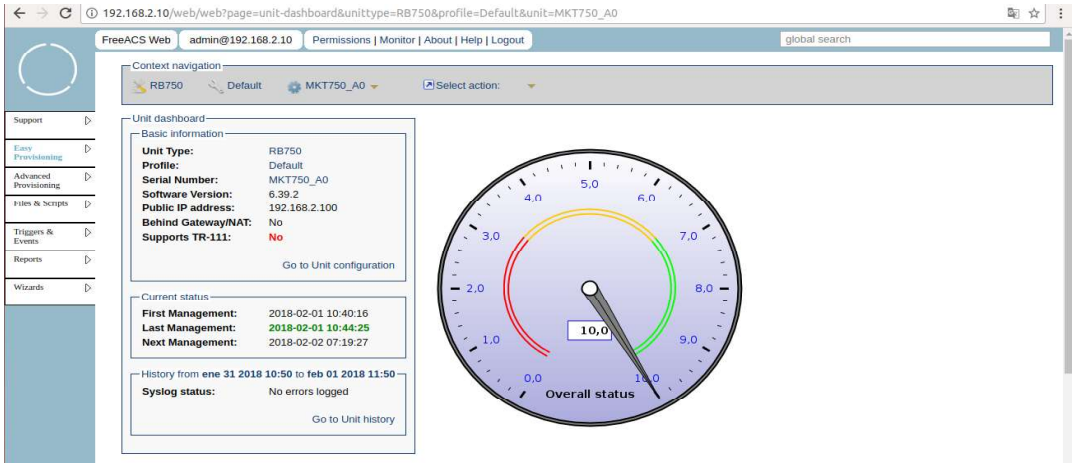

Figura 2.3. Resumen de la unidad aprovisionada

2.1.1.7.2.4 Actualización de software

Las unidades y perfiles se pueden actualizar a otra versión de software a través de un asistente de actualización. Este asistente de actualización no realiza la actualización en sí, sino que cambia el parámetro de versión de software deseado. Los dispositivos que se conectan a los servidores de aprovisionamiento FreeACS se actualizarán al contactarse. Por lo tanto, una actualización de software en un perfil puede tardar días o incluso semanas en completarse, ya que no todos los dispositivos se conectan regularmente. La figura 2.4 muestra la ventana de actualización de software.

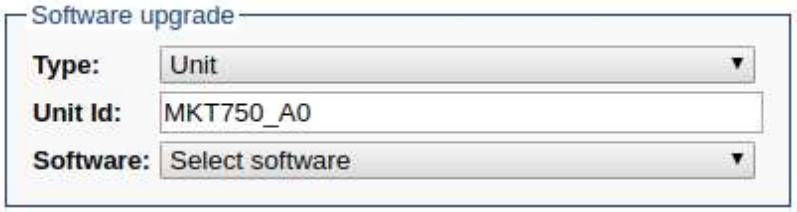

Figura 2.4. Ventana de actualización de software

2.1.1.7.2.5 Ventana de servicio y frecuencia de aprovisionamiento

Un SW (*Service Window*, Ventana de Servicio) define cuándo se puede aprovisionar una unidad. El ACS informará al CPE cuándo puede conectarse según esta ventana de servicio. FreeACS proporciona dos Service Windows:

- El SW regular se usa para todos los cambios que no interrumpen los servicios en el CPE, que se encuentran dentro del intervalo de tiempo especificado.

- El SW disruptivo es para todos los cambios que interrumpen los servicios en el CPE, como una actualización de software.

En la figura 2.5 muestra la configuración de un Service Window.

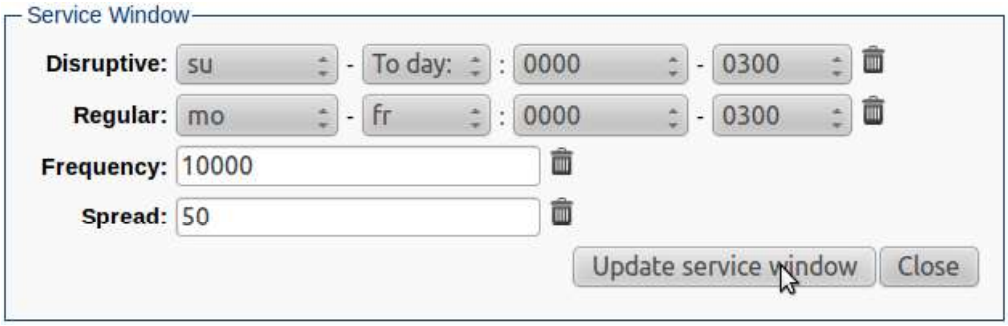

Figura 2.5. Configuración Service Window

2.1.1.7.2.6 Modos de aprovisionamiento

2.1.1.7.2.6.1 Modo periódico

El funcionamiento normal del aprovisionamiento de FreeACS es estar en modo periódico. Esto significa que el dispositivo se conectará al servidor tantas veces como lo solicite el servidor a través del parámetro denominado "*PeriodicInformInterval*". Al conectarse, el dispositivo pasará por una sesión de aprovisionamiento normal, verificando cambios y trabajos, leyendo y escribiendo datos.

2.1.1.7.2.6.2 Modo de inspección (Read All)

El modo de inspección es extremadamente útil para el aprovisionamiento de una Unidad individual. La idea es que se pueda inspeccionar los parámetros que no se visualizan con el aprovisionamiento estándar. En el modo de inspección, puede examinar más datos del dispositivo de lo que normalmente está disponible.

2.1.1.7.2.6.3 Modo de kick Provisión

FreeACS puede solicitar un aprovisionamiento inmediato, que se realiza en 30 segundos. Esto es útil cuando tiene algunos cambios en el servidor que debe aplicar en el CPE de manera rápida. La figura 2.6 muestra los botones para modo Kick e Inspección.

Figura 2.6. Botones para los diferentes tipos de aprovisionamientos

2.1.1.7.2.7 Jobs

Un Job realizará cambios controlados y seguros en varias unidades, ya sea una actualización de software o un cambio de configuración. Un Job es la acción que desea realizar en un grupo al que está conectado. Para crea un Job, se debe seleccionar el Tipo de Unidad apropiado, un grupo, un nombre y una descripción. Existen varios tipos de Job, los cuales se describen a continuación. La figura 2.7 Muestra la interfaz para crear un Job.

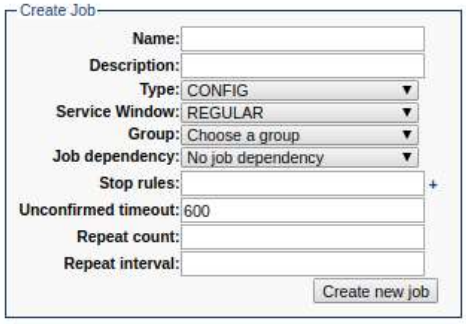

Figura 2.7. Creación de un Job

2.1.1.7.2.7.1 Config

Un Job de configuración es el requerimiento más típico, donde se especifica y aplica un conjunto de parámetros en el dispositivo a medida que el dispositivo se conecta al servidor.

2.1.1.7.2.7.2 Software

Un Job de software, actualiza el software del dispositivo al conectarse. Para lo cual se debe cargar el archivo con la nueva versión en la sección de almacenamiento y especificar el nombre del software en el Job.

2.1.1.7.2.7.3 Script

Un Job tipo script transmite una secuencia de comandos al dispositivo al conectarse. Esto generalmente es algo que se hace para superar las limitaciones del TR-069 en el dispositivo ya que algunos cambios podrían no ser posibles a través de los parámetros TR-069.

2.1.1.7.2.7.4 Reiniciar

Un Job de reinicio emitirá un comando Reiniciar al dispositivo.

2.1.1.7.2.7.5 Restablecer

Un Job de restablecimiento emitirá un comando de reseteo de Fabrica para el dispositivo.

2.1.1.7.2.7.6 Kick

Un Job tipo kick emite un comando de aprovisionamiento inmediato a todo un grupo de unidades, que activará los dispositivos para que se conecten al servidor de manera instantánea, lo que provocará un cambio de los dispositivos rápidamente. El intervalo de este Job de manera predeterminada es de 1 segundo entre cada conexión, por lo tanto, un grupo de 3600 dispositivos tardará aproximadamente 1 hora en arrancar.

2.1.1.7.2.7.7 Telnet

Un Job telnet emite un script telnet a los dispositivos del grupo. Por lo general, esta secuencia de comandos realizará cambios en el dispositivo que no fue posible lograr con el aprovisionamiento regular. Para que este Job funcione, se debe especificar un conjunto de parámetros que comparten el mismo prefijo "System.X_OWERA-COM.Telnet.". Los parámetros son:

- La dirección IP del dispositivo.
- Número de Puerto.
- Nombre de usuario.
- Contraseña.

2.1.1.7.3 FreeACS Shell

FreeACS Shell es la interfaz de línea de comando para FreeACS, que está dirigido especialmente a las operaciones que se repiten, de manera que los scripts creados permiten automatizar tareas grandes y complejas. Existen algunas funciones que solo están disponibles a través de FreeACS Shell, estas funcionalidades son:

- Exportar la base de datos de FreeACS o un tipo de unidad a los archivos.
- Importar la base de datos de FreeACS o un tipo de unidad desde archivos.
- Eliminar la base de datos FreeACS o un tipo de unidad en la base de datos.
- Eliminar los parámetros del sistema no utilizados.
- Agregar o cambiar los parámetros de trabajo específicos de la unidad.
- Crea scripts complejos con lógica if / else / loops.

2.1.1.7.4 Servicios Web de FreeACS [12]

La interfaz de Servicios web está destinada a la comunicación máquina a máquina, es decir, a la integración de cualquier sistema de terceros con FreeACS. Para la mayoría de los proveedores más grandes, el aprovisionamiento es solo una pequeña parte de lo que se necesita para proporcionar un servicio al usuario final. Cuando se crea un usuario final en los sistemas de Proveedores. Las funciones más importantes son:

- Agregar unidad.
- Cambiar unidad.
- Mover la unidad.
- Eliminar unidad.
- Unidades de búsqueda.
- Recuperar unidad.

FreeACS Web Services es parte del lado norte de FreeACS y su único propósito es ser una interfaz de integración para otros sistemas específicos del cliente. Hasta el momento hay nueve servicios en esta interfaz:

- AddOrChangeUnit.
- deleteUnit.
- getUnits.
- addOrChangeUnittype.
- deleteUnittype.
- getUnittype.

- addOrChangeProfile.
- deleteProfile.
- getProfile.

2.1.1.7.4.1 Casos de Uso

2.1.1.7.4.1.1 Add or Change Unit "Add"

Este es el caso donde la base de datos FreeACS no tiene información alguna sobre la Unidad. Para agregar una Unidad hay dos criterios de información que son necesarios:

- Identificación de la unidad.
- Secreto (conocido como contraseña ACS, contraseña de aprovisionamiento).

Con estos dos criterios de información en su lugar, FreeACS está listo para establecer contacto con los dispositivos. La tabla 2.2 muestra la ubicación de la contraseña dependiendo del protocolo.

Tabla 2.2. Parámetro de contraseña

Protocolo	Nombre del parámetro	Ejemplo
TR-069	< Keyroot > .X_OWERA-COM.Secret	ASecretStoredInTheDevice
HTTP / TFTP	< Keyroot > .X_OWERA-COM.Secret	ASecretStoredInTheDevice

2.1.1.7.4.1.2 Agregar Unidad con ID de Unidad conocida + secret

La tabla 2.3 muestra que se puede agregar tantos parámetros a la unidad como desee.

Tabla 2.3. Ingreso de varios parámetros

	Objeto	Valor de ejemplo	Comentario
Entrada	Login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unit.unittype.name	NPA201E	El tipo de unidad debe estar presente en FreeACS
	Unit.profile.name	Defecto	El perfil debe estar presente en FreeACS
	Unit.unitId	123456-Modelname-123456789012	Este valor debe ser único dentro de FreeACS y corresponder al protocolo.

	Unit.parameters[0] .name	InternetGatewayDevice.X_OWERA-COM.Secret	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[0] .value	ASecretStoredInTheDevice	El valor real del parámetro. Esto se almacenará como un parámetro de la unidad.
	Unit.parameters[1] .name	InternetGatewayDevice.DeviceInfo.SerialNumber	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[1] .value	123456789012	El valor real del parámetro. Esto se almacenará como un parámetro de la unidad.
Salida	Unit.unitId	123456-Modelname-123456789012	The Unit Id

2.1.1.7.4.1.3 Add Or Change Unit "Change"

Una unidad se puede cambiar de muchas maneras:

- Agregar un parámetro.
- Cambiar un parámetro.
- Eliminar un parámetro.
- Cambiar el perfil.

2.1.1.7.4.1.3.1 Cambiar una unidad, Id e unidad conocida

Cuando se cambia una unidad, usualmente se conoce la ID de la Unidad. La identificación de la unidad es una clave universal en FreeACS, por lo que en este caso no necesita más información para identificar la unidad.

El perfil se cambia simplemente configurando el nombre del perfil en algo que no sea el perfil existente. Sin embargo, el nuevo valor de Perfil debe definirse en FreeACS.

Los Parámetros pueden tomar varios valores de banderas. Si decide agregar un parámetro use "AC", en caso de borrado use "D". Omitir el valor de banderas se interpretará como "AC". Naturalmente, puede agregar tantos parámetros como desee.

La tabla 2.4 muestra el proceso para cambiar parámetros cuando se conoce la ID.

Tabla 2.4. Cambio de parámetros con ID conocida

	Objeto	Valor de ejemplo	Comentario
Entrada	Login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unit.unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	Esta unidad debe estar presente en FreeACS
	Unit.unittype.name	NPA201E	El tipo de unidad debe estar presente en FreeACS
	Unit.profile.name	NotDefault	Se supone que la Unidad estaba previamente en Perfil predeterminado. Este valor moverá la Unidad al Perfil "NotDefault ".
	Unit.parameters[0].name	Device.VoiceService.1.VoiceProfile.1.Line.1.SIP.AuthUserName	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[0].value	SIPUser	El valor real que se agregará / cambiará para esta Unidad
	Unit.parameters[1].name	Device.VoiceService.1.VoiceProfile.1.SIP.ProxyServer	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[1].flags	re	El parámetro se eliminará de esta unidad
Salida	Unit.unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	

2.1.1.7.4.1.3.2 Cambiar una unidad, ID de la unidad desconocida

La tabla 2.5 muestra una forma para cambiar parámetros cuando no se conoce la id.

Tabla 2.5. Cambio de parámetros con ID desconocida

	Objeto	Valor de ejemplo	Comentario
Entrada	Login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña

	Unit.serial Number	123456789012	Este Número de serie debe encontrarse en FreeACS y debe representar solo una Unidad.
	Unit.unittype.name	NPA201E	Se supune que el número de serie que se está buscando se define en una unidad dentro de este tipo de unidad.
	Unit.profile.name	Defecto	Se supone que la Unidad está en Perfil predeterminado, por lo tanto, no hay movimiento de perfil.
	Unit.parameters[0].name	Device.VoiceService.1.VoiceProfile.1.Line.1.SIP.AuthUserName	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[0].value	SIPUser	El valor real que se agregará / cambiará para esta Unidad
	Unit.parameters[1].name	Device.VoiceService.1.VoiceProfile.1.SIP.ProxyServer	El nombre del parámetro debe definirse dentro del tipo de unidad correspondiente en FreeACS.
	Unit.parameters[1].flags	re	El parámetro se eliminará de esta unidad
Salida	Unit.unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	

2.1.1.7.4.1.4 Delete Unit

Este servicio es relativamente sencillo: simplemente borra una Unidad de FreeACS. El servicio no es compatible con múltiples eliminaciones.

2.1.1.7.4.1.4.1 Eliminar una unidad, ID de unidad conocida

La tabla 2.6 muestra una forma para borrar una unidad con ID conocida

Tabla 2.6. Borrar unidad con ID conocida

	Objeto	**Valor de ejemplo**	**Comentario**
Entrada	Login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unit.unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	Esta unidad debe estar presente en FreeACS

Salida	borrado	cierto	Volverá verdadero si la unidad fue encontrada y eliminada. Si la unidad no se encontró, devolverá falso.

2.1.1.7.4.1.4.2 Eliminar una unidad, ID de unidad desconocida

La tabla 2.7 muestra una forma para borrar una unidad con ID conocida

Tabla 2.7. Borrar unidad con ID desconocida

	Objeto	**Valor de ejemplo**	**Comentario**
Entrada	Login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unit.serialNumber	123456789012	Este Número de serie debe encontrarse en FreeACS y debe representar solo una Unidad.
	Unit.unittype.name	NPA201E	Se supone que el número de serie que se está buscando se define en una unidad dentro de este tipo de unidad.
Salida	borrado	cierto	Volverá verdadero si la unidad fue encontrada y eliminada. Si la unidad no se encontró, devolverá falso.

2.1.1.7.4.2 Add Or Change Unit Type

Los métodos mencionados hasta ahora están destinados a la integración entre un cliente de FreeACS que requiere una integración limitada o estándar. Las operaciones de agregar, cambiar, eliminar en Unittype o Profile se lo realiza manualmente. La tabla 2.8 muestra la forma de crear un Unittype.

Tabla 2.8. creación de Unittype

	Objeto	**Valor de ejemplo**	**Comentario**
Entrada	Login.username	administración	Nombre de usuario
	Contraseña	xaps	Contraseña
	Unittype.name	Teléfono móvil	El nombre delunittype

	Unittype.matcherId	000000	El matcherId puede ser importante si el protocolo es OPP.
	Unittype.description	Un dispositivo móvil	Una descripción
	Unittype.vendor	Nokia	El fabricante del teléfono.
	Unittype.protocol	TR-069	Puede ser TR-069, OPP o N/A. Esta decisión tendrá un impacto en qué formato se acepta en las unitds dentro de este unittype .
	Unittype.parameters[0] .name	Device.DeviceInfo. SerialNumber	El nombre del parámetro unittype
	Unittype.parameters[0] .value	SAR	El indicador del parámetro unittype. Debe ser R, RW o X.
	Unittype.parameters[0] .flags	C.A.	Indica si el parámetro debe agregarse o modificarse (AC) o eliminarse (D).
Salida	Devolverá el objeto UnitType (pero incluirá todos los datos de FreeACS, no sólo los datos modificados (si se trata de una solicitud de cambio) en esta convocatoria)		

2.1.1.7.4.3 Delete Unit Type

Un tipo de unidad es el concepto más básico en FreeACS y muchos otros conceptos derivan o dependen del tipo de unidad. Por lo tanto, antes de que se pueda eliminar un Tipo de Unidad, todas las demás dependencias deben ser eliminadas. Algunas cosas se borran automáticamente por este servicio, como Eventos de Syslog, Grupos y Parámetros de Tipo de Unidad. Sin embargo, los Softwares, Trabajos y Permisos no se eliminarán automáticamente; estos deben eliminarse a través de FreeACS Shell o FreeACS Web. Los perfiles y las unidades también se deben eliminar. La tabla 2.9 muestra la forma de borrar un Unittype

Tabla 2.9. Eliminación de un Unittype

	Objeto	**Ejemplo valor**	**Comentario**
Entrada	login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	unittypeName	Teléfono móvil	El nombre de la UnitType
Salida	esta borrado	cierto	Devolverá false si la operación de eliminación (en SQL) no respondió de 1 fila eliminada.

2.1.1.7.4.4 Get Unit Types

Devolverá todos los tipos de unidades que se les permite ver. La tabla 2.10 muestra la forma de búsqueda de Unittypes

Tabla 2.10. Búsqueda de Unittypes

	Objeto	Ejemplo valor	Comentario
Entrada	login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unittype.name	Teléfono móvil	El nombre de la UnitType
Salida	Unittype.name	Teléfono móvil	El nombre de la UnitType
	Unittype.matcherId	000000	El matcherId puede ser importante si el protocolo es OPP.
	Unittype.description	Un dispositivo móvil	Una descripción
	Unittype.vendor	Nokia	El fabricante del teléfono.
	Unittype.protocol	TR-069	Puede ser TR-069, OPP o N / A. Esta decisión tendrá un impacto en la que se acepta el formato de los unitIds dentro de este UnitType .
	Unittype.parameters [0] .name	Device.DeviceInfo.SerialNumber	El nombre de la UnitType parámetro
	Unittype.parameters [0] .value	SAR	La bandera de la UnitType parámetro. Debe ser R, RW o X.
	Unittype.parameters [0] .FLAGS		

2.1.1.7.4.5 Add Or Change Profile

La tabla 2.11 muestra la forma de crear un perfil.

Tabla 2.11. Creación de perfiles

	Objeto	Ejemplo valor	Comentario
Entrada	login.username	administración	Nombre de usuario
	Contraseña de inicio de sesión	xaps	Contraseña
	Unittype.name	Teléfono móvil	El nombre de la UnitType
	Nombre de perfil	Defecto	El nombre del perfil

	Profile.parameters [0] .name	Device.ManagementServer.URL	El nombre del parámetro Tipo de la unidad (el parámetro ya debe existir en el tipo de unidad)
	Profile.parameters [0] .value	http://xaps.com/tr069	El valor del parámetro.
	Profile.parameters [0] .FLAGS	C.A.	Indica si el parámetro se va a añadir o cambiado (AC) o borrado (D).
Salida	Devolverá el objeto de perfil (pero incluirá todos los datos de FreeACS, no sólo los datos modificados (si se trata de una solicitud de cambio) en esta convocatoria)		

2.1.1.7.4.6 Borrar perfil

Para la eliminación de un perfil se debe tener en cuenta, que puede que se deba eliminar grupos, permisos y Unidades asociados con ese perfil en.

2.1.1.7.4.7 Get Profiles

Devolverá todo el perfil que se les permite ver.

2.1.1.7.4.8 Generalidades de FreeACS

2.1.1.7.4.8.1 Iniciar sesión

Para acceder a los servicios, se requiere un inicio de sesión. Los inicios de sesión se administran y crean utilizando Web o Shell. El único usuario disponible al inicio es 'admin' con la contraseña ' xaps ', y este usuario siempre tendrá acceso a todas las partes del sistema. Para cada inicio de sesión hay un conjunto de permisos. Estos permisos restringen a qué tipos de unidades y perfiles se puede acceder.

2.1.1.7.4.8.2 ID de la unidad

El ID de la unidad es el único identificador en FreeACS. El formato de la ID de unidad varía de acuerdo con el protocolo de provisión. La tabla 2.12 muestra el formato del ID de acuerdo al protocolo de provisión.

Tabla 2.12. Formato del ID de acuerdo al protocolo de provisión.

Protocolo	Formato	Comentario	Ejemplo
TR-069	<OUI> - <ProductClass - < SerialNumber >	OUI: es un número hexadecimal de 6 dígitos para identificar la organización ProductClass: es a menudo el nombre del modelo del dispositivo SerialNumber: es más a menudo la dirección MAC	123456-Modelname-123456789012
HTTP / TFTP	< SerialNumber >	SerialNumber.	123456HJK1234

2.1.1.7.4.8.3 Keyroot

Cada parámetro que representa un atributo o propiedad del dispositivo siempre inicia su nombre con un Keyroot. Esta Keyroot puede ser Device o InternetGatwayDevice, dependiendo si el dispositivo está conectado con una IP pública.

Ejemplos de nombres de parámetros:

- Device.DeviceInfo.SerialNumber.
- InternetGatewayDevice.DeviceInfo.SerialNumber.

2.1.1.7.4.8.4 Buscar una unidad individual

Puede recuperar información sobre una sola Unidad utilizando uno de estos métodos:

- UnitId -search: busca una sola unidad usando la ID de unidad completa.
- Búsqueda de valor único: busca una unidad única que use un número de serie completo o cualquier otro valor único para una unidad.

La tabla 2.13 muestra una forma para la búsqueda de una unidad única.

Tabla 2.13. Búsqueda de una unidad única

	Objeto	Valor de ejemplo	Comentario
Entrada	Login.username	administración	Nombre de usuario
	Login.password	xaps	Contraseña
	Unit.unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	Esta unidad debe estar presente en FreeACS
Salida	Unit[0].unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	La ID de la unidad

	Unit[0].serialNumber	123456789012	El número de serie de la unidad
	Unit[0].profile.name	Defecto	El perfil de la unidad
	Unit[0].unittype.name	NPA201E	Tipo de unidad de la unidad
	Unit[0].parameter[0].name	Device.VoiceService.1.VoiceProfile.1.Line.1.SIP.AuthUserName	Un nombre de parámetro de unidad
	Unit[0].parameter[0].value	SIPUser	Un valor de parámetro de la unidad
	Unit[0].parameter[0].flags	U	La bandera indica que este es un parámetro de la unidad
	Unit[0].parameter[1].name	Device.DeviceInfo.SerialNumber	Un nombre de parámetro de unidad
	Unit[0].parameter[1].value	123456789012	Un valor de parámetro de la unidad
	Unit[0].parameter[1].flags	U	La bandera indica que este es un parámetro de la unidad
	moreUnits	falso	No más unidades encontradas

2.1.1.7.4.8.5 Buscar varias unidades

Se puede usar uno de estos métodos.

- Búsqueda simple: Busca muchas Unidades usando el Tipo de Unidad y / o Perfil.
- Búsqueda de valor: Busca varias Unidades que usan ID parcial de la Unidad o un valor (parcial).
- Búsqueda de parámetros: Busca varias unidades que usan parámetros. Se puede especificar la búsqueda con uno de los seis tipos de operadores (especifique como indicadores):
  - ✓ EQ (Igual) (predeterminado).
  - ✓ NE (no es igual).
  - ✓ LT (menos que).
  - ✓ LE (menos o igual).
  - ✓ GE (mayor o igual).
  - ✓ GT (Mayor que).

La tabla 2.14 muestra el proceso para la búsqueda de varias unidades a través de parámetros.

Tabla 2.14. Búsqueda de varias unidades a través de parámetros.

	Objeto	Valor de ejemplo	Comentario
Entrada	Login.username	administración	Nombre de usuario
	Login.password	xaps	Contraseña
	Unit.serialNumber	123456789012	Utilizar cualquier tipo de valor único (-address ip,mac-address, nombre de usuario, contraseña, número de serie)
	Unit.unittype.name	NPA201E	El tipo de unidad debe estar presente en FreeACS
Salida	Unit[0].unitId	dd37cbd4-d4a3-3d66-8958-5d9cf30635f5	La ID de la unidad
	Unit[0].serialNumber	123456789012	El número de serie de la unidad
	Unit[0].profile.name	Defecto	El perfil de la unidad
	Unit[0].unittype.name	NPA201E	Tipo de unidad de la unidad
	Unit[0].parameter[0].name	Device.VoiceService.1.VoiceProfile.1.Line.1.SIP.AuthUserName	Un nombre de parámetro de unidad
	Unit[0].parameter[0].value	SIPUser	Un valor de parámetro de la unidad
	Unit[0].parameter[0].flags	U	La bandera indica que este es un parámetro de la unidad
	Unit[0].parameter[1].name	Device.DeviceInfo.SerialNumber	Un nombre de parámetro de unidad
	Unit[0].parameter[1].value	123456789012	Un valor de parámetro de la unidad
	moreUnits	falso	No más unidades encontradas

2.1.1.7.4.8.6 Preferencia

El tipo de unidad y la información del perfil siempre se tendrán en cuenta en todas las búsquedas. Si no se proporciona, la búsqueda se aplicará dentro de los Tipos de unidades y Perfiles permitidos.

La prioridad entre los otros criterios de búsqueda es la siguiente:

- Si se detecta Unit Id en la entrada, la búsqueda será una búsqueda con identificación de unidad.
- Si se detecta un número de serie en la entrada, la búsqueda será una búsqueda de valor.
- Si se detecta un parámetro en la entrada, la búsqueda será una búsqueda de parámetros.

2.1.1.7.4.8.7 Límite de devolución

Las búsquedas pueden mostrar de 0 a 50 resultados, este límite se especifica para optimizar el tiempo de respuesta y el CPU del servidor.

2.1.1.7.4.8.8 El conjunto de resultados

El servicio siempre devuelve una matriz de Unidades. Cada unidad de la matriz contiene una identificación de la unidad, un número de serie, un perfil, un tipo de unidad y un conjunto de parámetros. Por lo tanto, el Tipo de unidad como el Perfil se representan en el conjunto de resultados solo por sus nombres.

2.1.1.7.4.9 Redirección o Solución de problemas

El FreeACS WS Server ofrece una característica de redirección. Esto está destinado principalmente para la solución de problemas. La idea con re-direccionamiento es ser capaz de atrapar el XML hacia y desde el servidor y almacenarlo en un archivo. Esto es importante ya que el XML a menudo se pierde cuando se trabaja con tramas de servicios Web. Comparando diferentes salidas XML puede ser eficaz para localizar donde las cosas iban mal.

- El XML se registra en xaps-WS-xml.log, o si se ha configurado de otra manera en xaps-ws-logs.properties .

- Se debe configurar la dirección URL correcta para redirigir a la FreeACS WS servicio en xaps-ws.properties .

2.1.1.7.5 Servidor FreeACS Core

Se requiere que FreeACS Core Server use la funcionalidad Jobs en FreeACS. El servidor FreeACS Core supervisa el estado de los Jobs e inicia o detiene los trabajos en función de las reglas de detención. Este Módulo ha sido desarrollado para poder tratar con un gran número de Unidades sin un impacto negativo en el rendimiento de la base de datos de FreeACS. También maneja generación de informes y recorte syslog.
Este servidor es responsable de una serie de tareas que se ejecutan en segundo plano:

- Actualizar contadores de Jobs.
- Detener el Job automáticamente según las reglas del trabajo.
- Eliminar Jobs viejos y terminados (limpieza).
- Proceso de eliminación de syslog, elimina entradas antiguas de la base de datos syslog.
- La generación del informe.
- Ejecución del script de Shell para TR-069 / Web.

2.1.1.7.6 Servidor FreeACS de Monitoreo

FreeACS Monitor Server supervisa la disponibilidad de los otros módulos de FreeACS, proporcionando información de estado, presentada por el módulo FreeACS Web, además almacena información histórica de disponibilidad, como las marcas de tiempo de eventos, el tiempo de actividad y los mensajes de supervisión para los módulos de FreeACS.

2.1.1.7.7 Servidor FreeACS Syslog

El FreeACS Syslog Server es muy útil si se tiene dispositivos que envían mensajes a un syslog-server. En esos casos, puede supervisar los eventos en los dispositivos directamente en FreeACS Web.

2.1.1.7.8 Servidor FreeACS TR-069

El servidor FreeACS TR-069 es responsable de la comunicación entre los CPE habilitados TR-069 y FreeACS. Este Módulo se puede implementar en varios hosts para escalar el rendimiento en implementaciones grandes con dispositivos habilitados para TR-069.

Los CPE deben configurarse con una URL que apunta a este servidor. Cuando esto esté en su lugar, el servidor TR-069 cumplirá con su obligación: proporcionar parámetros a los CPE, leer los parámetros de los CPE, actualizar la base de datos con los datos del CPE y actualizar el firmware o software y la configuración.

2.1.1.7.8.1 Seguridad

2.1.1.7.8.1.1 Autenticación

El servidor puede ejecutarse sin autenticación, autenticación básica y autenticación resumida. El servidor podría ejecutar la autenticación sin usar certificados de seguridad web.

2.1.1.7.8.2 Descarga del Firmware o del Archivo de configuración

El servidor TR-069 funciona así: comprueba si un dispositivo tiene la versión correcta de software o archivos de configuración. De lo contrario, emite un método de descarga con el archivo de configuración o software correcto. El archivo se encuentra por defecto en el almacenamiento de archivos de FreeACS y se identifica según el número de versión y el tipo de archivo.

2.1.1.7.8.3 Descarga de firmware

Para el software, el servidor verifica el parámetro que se recupera del dispositivo, ya sea en la solicitud de *Inform-request* o en la *GetParameterValues*-response:

- *DeviceInfo.SoftwareVersion*

Y compara este valor de parámetro con el parámetro del sistema

- *System.X_OWERA-COM.DesiredSoftwareVersion*

No se emite ningún método de descarga si estos son iguales. Si estos parámetros difieren, se emitirá una descarga.

Si el firmware no establece el número de versión correctamente, en el parámetro *Device.SoftwareVersion*. FreeACS no comprenderá que este firmware ya se ha descargado, y emitirá una nueva secuencia de descarga del mismo firmware, una y otra vez.

FreeACS determina y crea la URL para recuperar el firmware utilizando esta información:

- Número de versión especificado en DesiredSoftwareVersion-parameter.
- El hecho de que el firmware es de SOFTWARE de tipo Archivo y se encuentra en el propio sistema de archivos de FreeACS.
- El tipo de unidad en el que se especifica el parámetro.

La construcción de la URL de descarga ocurre en segundo plano y de manera automática. Sin embargo, se puede especificar una URL en particular, configurando el siguiente parámetro:

- System.X_OWERA-COM.SoftwareURL.

2.1.1.7.8.4 Descarga del Archivo de configuración

Para los archivos de configuración, el servidor solicita lo siguiente:

- Device.VendorConfigFile.1.Version.
- Device.VendorConfigFile.1.Name.
- Device.VendorConfigFile.2.Version.
- Device.VendorConfigFile.2.Name.

Si el dispositivo no admite *VendorConfigFile*, no se podrá descargar archivos de configuración al dispositivo. Si el dispositivo admite el VendorConfigFile, se pueden especificar algunos parámetros del sistema para indicar al servidor que archivos de configuración deben estar presentes. Los parámetros del sistema deben coincidir con los parámetros de vendor-config-file:

- System.X_OWERA-COM.TR069Script. <Nombre> Version = <Versión>

Los archivos que se descargarán en el dispositivo, se encuentran en la base de datos de FreeACS, al igual que los firmwares, pero ahora el tipo de archivo es TR069_SCRIPT. Hay una adición a la URL en comparación con los firmwares: el nombre de archivo se agrega a

la URL. Esto podría ser útil si el CPE requiere que la URL finalice con un sufijo determinado como .cfg. Sin embargo, puede anular este comportamiento si se especifica la URL que se utilizará en el método de descarga.

- System.X_OWERA-COM.TR069Script. <Nombre> .URL = <URL>

2.1.1.7.8.5 Asistente de actualización de software (FreeACS Web)

Este asistente puede configurar una unidad o un perfil específicos para actualizar. También puede establecer una actualización para un Job en particular. Un Job permite elegir un conjunto de unidades y controlar el progreso de las actualizaciones.

2.1.1.7.9 Servidor FreeACS SPP

El servidor de FreeACS SPP es responsable de la comunicación entre los dispositivos HTTP / HTTPS / TFTP / Telnet y FreeACS. El Módulo se puede implementar en varios hosts para ampliar el rendimiento en implementaciones grandes.

La solicitud de un dispositivo generalmente contiene información vital e importante, FreeACS busca 4 tipos de información:

- Número de serie (imprescindible para el aprovisionamiento).
- Mac (una necesidad para syslog).
- Versión de software (una necesidad para la actualización de firmware).
- Modelname (una necesidad para el modo de aprovisionamiento y descubrimiento).

2.1.2 INSTALACIÓN DEL SERVIDOR ACS

Para la instalación del servidor ACS se ha seguido las recomendaciones del fabricante, por lo que se ha elegido instalar el sistema operativo Ubuntu Server 14.04 para la incorporación del sistema FreeACS. Se realizaron algunos procedimientos previos, entre los que se destacan la implementación de una interfaz gráfica para el sistema operativo, ya que de manera predeterminada el SO Ubuntu Server solo trabaja por líneas de comando, se configuraron las tarjetas de red de tal manera que se adapten a las necesidades del servidor.

De la página oficial de FreeACS se obtuvo documentación donde se detalla el proceso de instalación y de puesta en marcha del sistema de gestión de CPE. En primera instancia es necesario correr el script de instalación llamado install-or-update-FreeACS-ubuntu.sh, el cual permite instalar la mayor parte de los requerimientos que FreeACS demanda.

Para la instalación y modificación de algunos archivos se requiere tener permiso de superusuario. En la figura 2.8, se muestra el inicio del proceso de instalación usando el script proporcionado por el fabricante.

```
jflopez@Servidor:~$ chmod 755 install-or-update-freeacs-ubuntu.sh && sudo ./inst
all-or-update-freeacs-ubuntu.sh

              ****************************************
              * Fusion FreeACS Installation & Upgrade *
              ****************************************

This script does 90% of the installation of Fusion FreeACS. The last 10%
must be done manually (see Fusion Instalation.pdf) by changing the contents
of various files. Run the script again to download and install new releases
or if something didn't work out the first time you installed. If default
configuration changes in a new release, you will be duly notified.

The script is targeted at Ubuntu 14.04 64-bit, but may also work with other
Ubuntu versions, like Ubuntu 12.04 LTS.

Do you run this script with sudo (root) permission? (y/n)
```

Figura 2.8. Proceso de instalación

A continuación, se mencionarán los cambios más relevantes en los archivos de configuración tanto de la base de datos como de los diferentes módulos de FreeACS.

El bind-address permite que la base de datos sea accesible desde un servidor local externo. De lo contrario, la base de datos solo será accesible para las aplicaciones que se ejecutan en localhost.

El max_allowe_packet permite la adición de firmware hasta este tamaño en la base de datos de FreeACS.

El innob_buffer_pool_size es la configuración de memoria más importante, MySQL debería tener acceso tal vez al 50% de la memoria total en el servidor. Una vez que se realizaron estos cambios, se reinició el servicio mysql. Las figuras 2.9 y 2.10 muestran las configuraciones realizadas en el archivo /etc/mysql/my.cnf.

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
#
# * Fine Tuning
#
key_buffer              = 16M
max_allowed_packet      = 32M
thread_stack            = 192K
thread_cache_size       = 8
```

Figura 2.9. Seteo de bind-address y max allowed packet

```
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
innodb_buffer_pool_size = 1024M
innodb_log_file_size = 256M
```

Figura 2.10. Seteo de los tamaños de los de memoria de almacenamiento

La figura 2.11 muestra las configuraciones realizadas en el archivo /var/lib/tomcat7/conf/catalina.properties. Se adjuntan las sentencias seleccionadas en el archivo de configuración de tomcat, para que apunte a un directorio donde se colocarán las propiedades de FreeACS.

```
common.loader=${catalina.base}/common,${catalina.base}/common/*.properties,${cata
lina.base}/lib,${catalina.base}/lib/*.jar,${catalina.home}/lib,${catalina.home}/l
ib/*.jar,${catalina.home}/common/classes,${catalina.home}/common/*.jar
```

Figura 2.11. Apuntamiento propiedades FreeACS

Las figuras 2.12 y 2.13 muestra las configuraciones del archivo /etc/default/tomcat7.

```
# Use "-XX:+UseConcMarkSweepGC" to enable the CMS garbage collector (improved
# response time). If you use that option and you run Tomcat on a machine with
# exactly one CPU chip that contains one or two cores, you should also add
# the "-XX:+CMSIncrementalMode" option.
JAVA_OPTS="-Djava.awt.headless=true -Xmx768m -XX:+UseConcMarkSweepGC"
```

Figura 2.12. Aumento de memoria para cambiar la capacidad de aprovisionamiento

```
# If you run Tomcat on port numbers that are all higher than 1023, then you
# do not need authbind.  It is used for binding Tomcat to lower port numbers.
# NOTE: authbind works only with IPv4.  Do not enable it when using IPv6.
# (yes/no, default: no)
AUTHBIND=yes
```

Figura 2.13. Permiso para que Tomcat use los puertos 80 y 443

La figura 2.14 muestra las configuraciones del archivo /etc/passwd.

```
tomcat7:x:118:127::/var/lib/tomcat7:/bin/false
```

Figura 2.14. Cambio del directorio de Tomcat

La figura 2.15 muestra las configuraciones del archivo /var/lib/tomcat7/conf/server.xml

```
<!-- A "Connector" represents an endpoint by which requests are received
     and responses are returned. Documentation at :
     Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
     Java AJP  Connector: /docs/config/ajp.html
     APR (HTTP/AJP) Connector: /docs/apr.html
     Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="80" protocol="HTTP/1.1"
           connectionTimeout="20000"
           URIEncoding="UTF-8"
           redirectPort="8443" />
```

Figura 2.15. Cambio de puerto del 8080 al 80

La figura 2.16 muestra las configuraciones del archivo /etc/init.d/tomcat7

```
### BEGIN INIT INFO
# Provides:          tomcat7
# Required-Start:    $local_fs $remote_fs $network $mysql
# Required-Stop:     $local_fs $remote_fs $network
# Should-Start:      $named
# Should-Stop:       $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Tomcat.
# Description:       Start the Tomcat servlet engine.
### END INIT INFO
```

Figura 2.16. Tomcat depende de Mysql para el arranque

Se añade esa sentencia ya que algunos servicios de FreeACS tienen la costumbre de informar errores de correo electrónico si no se pueden conectar a la base de datos cuando comienzan. Para evitar esto, se hace que Tomcat dependa de MySQL para que comience en el arranque.

Las figuras 2.17 y 2.18 muestran las configuraciones del archivo /var/lib/tomcat7/common/xaps-monitor.properties.

```
# SMTP hostname - the below example shows how to use a gmail-account to send emai
ls
#alerts.smtp.host     = smtp.gmail.com
alerts.smtp.host     = smtp.puntonet.ec
# SMTP port
#alerts.smtp.port        = 465
alerts.smtp.port        = 25
# SSL? true/false
alerts.smtp.ssl      = true
# STMP username, may not be necessary for a "local" STMP host
#alerts.smtp.user        = your-gmail-user
alerts.smtp.user        = jflopez
# STMP password, may not be necessary for a "local" STMP host
#alerts.smtp.pass        = your-gmail-password
alerts.smtp.pass        = junlo$$2017
# Email-address to send monitor messages. Also trigger alarms can end up here.
#alerts.to                      = to-email@address.com
alerts.to                       = cccorporativo@puntonet.ec
# The from address
#alerts.from                    = from-email@address.com
alerts.from                     = jflopez@puntonet.ec
```

Figura 2.17. Configuración para obtener correo sobre eventos y errores en FreeACS

```
# Email URL
# The URL to reach web from outside Fusion (will be used in all emails regarding
Monitor
# Alarms og Trigger Alarms/Reports). Specify the correct hostname of your Fusion
server
# Default is "http://localhost/", which will not work well for most cases!
#fusion.urlbase         = http://localhost/
fusion.urlbase          = http://192.168.2.10/
```

Figura 2.18. URL para acceso remoto

Esta URL se usará en el correo que se le envíe; específica una base de URL que puede llegar al FreeACS desde afuera.

La figura 2.19 muestra las configuraciones del archivo /var/lib/tomcat7/common/xaps-stun.properties. Configuración de la dirección IP del servidor. El servidor intentará enlazarse a esta IP en el puerto 3478. Se devuelve una página web. De lo contrario, se debería cambiar la URL o verificar si el servidor de Monitor se está ejecutando realmente

```
# The STUN server needs 2 network interface to work

# 3478 is the default Stun server port
primary.port = 3478
secondary.port = 3479

# Primary ip. Specify the default public interface of your server
primary.ip = 192.168.2.10
```

Figura 2.19. Configuración del servicio stun

La figura 2.20 muestra las configuraciones del archivo /var/lib/tomcat7/common/xaps-web.properties.

```
# --- Monitor server ---
# where to retrieve monitor status (not including context root and other paramete
rs)
#monitor.location = http://localhost/
monitor.location = http://192.168.2.10/
```

Figura 2.20. Configuración para acceso Web

La figura 2.21 muestra las pruebas realizadas desde el terminal

```
jflopez@Servidor:~$ wget localhost/web
--2018-01-15 23:57:56--  http://localhost/web
Resolviendo localhost (localhost)... ::1, 127.0.0.1
Conectando con localhost (localhost)[::1]:80... conectado.
Petición HTTP enviada, esperando respuesta... 302 Movido temporalmente
Ubicación: http://localhost/web/ [siguiente]
--2018-01-15 23:57:56--  http://localhost/web/
Reutilizando la conexión existente a [localhost]:80.
Petición HTTP enviada, esperando respuesta... 302 Movido temporalmente
Ubicación: http://localhost/web/login [siguiente]
--2018-01-15 23:57:56--  http://localhost/web/login
Reutilizando la conexión existente a [localhost]:80.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
Grabando a: “web.1”

    [ <=>                                   ] 2.015       --.-K/s   en 0s

2018-01-15 23:57:56 (146 MB/s) - “web.1” guardado [2015]

jflopez@Servidor:~$ wget 192.168.2.10/web
--2018-01-15 23:58:07--  http://192.168.2.10/web
Conectando con 192.168.2.10:80... conectado.
Petición HTTP enviada, esperando respuesta... 302 Movido temporalmente
Ubicación: http://192.168.2.10/web/ [siguiente]
--2018-01-15 23:58:07--  http://192.168.2.10/web/
Reutilizando la conexión con 192.168.2.10:80.
Petición HTTP enviada, esperando respuesta... 302 Movido temporalmente
Ubicación: http://192.168.2.10/web/login [siguiente]
--2018-01-15 23:58:07--  http://192.168.2.10/web/login
Reutilizando la conexión con 192.168.2.10:80.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
Grabando a: “web.2”

    [ <=>                                   ] 2.015       --.-K/s   en 0s

2018-01-15 23:58:07 (131 MB/s) - “web.2” guardado [2015]
```

Figura 2.21. Pruebas de respuesta del URL

La figura 2.22 muestra la página de inicio del servidor FreeACS

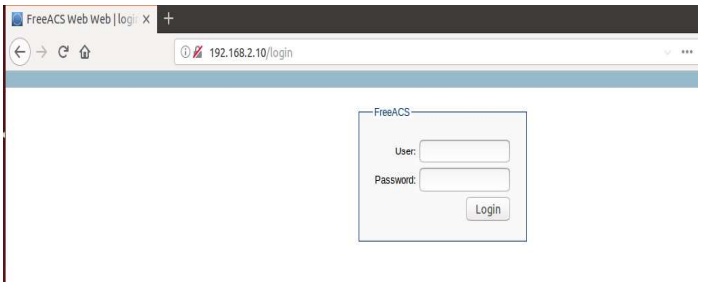

Figura 2.22. Ingreso a la interfaz Web

2.1.2.1 Servidor FreeACS Core [13]

La configuración del servidor Core es importante, porque si no se realiza correctamente, puede causar una carga muy pesada en la base de datos de FreeACS. Los dos principales culpables son la eliminación de syslog y la generación de informes, por lo que hay que tener especial cuidado con esos problemas. La figura 2.23 muestra cómo se especifica un límite de horas en el que los viejos trabajos completados se eliminen del sistema.

```
# --- Controls ---

# In general, read the doc about the groups/jobs and the core server.

# Old completed jobs must be removed from the job system, because it will
# decrease performance in the long run to keep them there. Hence this value.
# In hours. Default is 48.
completed.job.limit = 48
```

Figura 2.23. Límite de tiempo de Jobs antiguos

La tabla syslog no debe crecer demasiado, por lo que se debe eliminar las filas antiguas después de un tiempo. La figura 2.24 muestra la configuración del tiempo que se mantiene los eventos syslog.

```
# All values are stated as a number of days to save before deletion.
# Default 90 days for severity 0 (Emergency)
syslog.severity.0.limit = 90
# Default 90 days for severity 1 (Alert)
syslog.severity.1.limit = 90
# Default 90 days for severity 2 (Critical)
syslog.severity.2.limit = 90
# Default 90 days for severity 3 (Error)
syslog.severity.3.limit = 60
# Default 60 days for severity 4 (Warning)
syslog.severity.4.limit = 30
# Default 30 days for severity 5 (Notice)
syslog.severity.5.limit = 7
# Default 7 days for severity 6 (Info)
syslog.severity.6.limit = 4
# Default 4 days for severity 7 (Debug)
syslog.severity.7.limit = 2
```

Figura 2.24. Límite de tiempo eventos Syslog

La limpieza de Syslog es muy importante para mantener el tamaño de la base de datos bajo control. Si está activado, solo se ejecutará en la primera hora después de la medianoche. Hay tres modos:

- Normal: se eliminará según las reglas de eliminación definidas anteriormente, así como los límites de eliminación de eventos de syslog.
- Comprensivo: Eliminará usando las mismas reglas que en el modo normal. Sin embargo, el modo normal no verificará toda la base de datos, solo los días más recientes. Este método comprobará toda la base de datos, pero esto solo es necesario si sospecha que se ha recopilado una gran cantidad de basura en la base de datos syslog.
- Off: no ejecutará la rutina de eliminación. Nunca configure esto como producción para pruebas cortas de este módulo.

La figura 2.25 muestra la configuración del modo de limpieza.

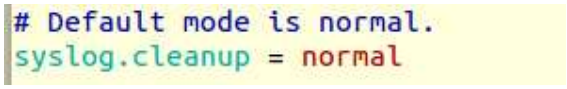

Figura 2.25. Seteo del modo de limpieza

2.1.2.2 Servidor FreeACS Monitor [14]

La página del monitor en FreeACS Web se usa para monitorear los módulos de FreeACS, y es una herramienta de depuración efectiva en los casos en que tiene la sensación de que algo no funciona como debería en el sistema. Por defecto, el servicio de monitor tiene un intervalo muy corto para verificar los módulos. Por lo tanto, de forma predeterminada podrá ver rápidamente si un sistema se ha desactivado.

Además de estar integrado en FreeACS Web, también puede verificar su estado directamente. Puede acceder a ella como una aplicación independiente:

El servicio de monitoreo actualmente solo está verificando que los módulos están activos y respondiendo, mas no si los módulos funcionan como deberían. Se puede depurar los módulos de forma individual al buscar en el syslog o archivos de registro.

Se puede acceder a la página del monitor desde FreeACS Web si se tiene los permisos adecuados y si se tiene habilitada la opción en xaps-web.properties. La figura 2.26 muestra la página de monitoreo.

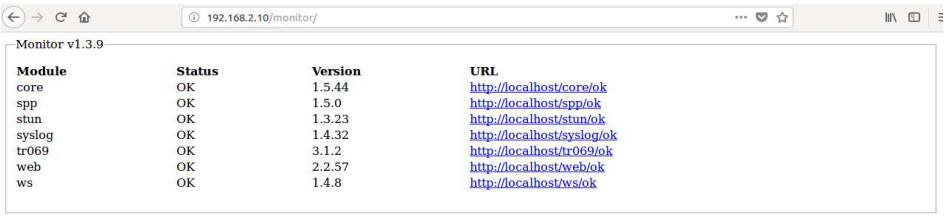

Figura 2.26. Interfaz del Servicio de Monitoreo

2.1.2.3 Servidor FreeACS SPP [15]

Discovery creará automáticamente la unidad en la base de datos de FreeACS. No se crearán parámetros. El dispositivo se colocará en el perfil predeterminado en el tipo de unidad correspondiente al nombre del modelo. En la figura 2.27 se muestra como se establece el modo de autodescubrimiento.

```
# Discovery will automatically create the unit in xAPS database. No parameters w
ill be created;
# we have no information about the device upon contact. The device will be place
d in the Default profile
# in the unittype corresponding to the modelname (if such can be found). Remembe
r to specify a
# modelname pattern above. Discovery can be true or false. Default is false.
discovery = true
```

Figura 2.27. Seteo de autodescubrimiento de dispositivos

La figura 2.28 muestra la configuración del puerto TFTP

```
# --- TFTP ---

# Decide the port to run the TFTP server. Default is 1069. (Default IANA standar
d is 69)
tftp.port = 1069
```

Figura 2.28. Habilitación del puerto 1069 para TFTP

2.1.2.4 Servidor FreeACS TR-069

2.1.2.4.1 Archivo Varios Controles

En este archivo se decide si quiere autenticación o no. La configuración recomendada es "digest", pero también es posible usar "basic" y "none". Para utilizar la autenticación, también debe agregar el parámetro "secret" en la base de datos.
El modo de descubrimiento normalmente debería estar deshabilitado. Si está activo, violará el modelo de seguridad, ya que se acepta el tráfico entrante incluso sin el parámetro secreto en la base de datos. El punto de este modo es permitirle conectar un CPE al ACS y luego llenar automáticamente la base de datos de FreeACS con toda la información necesaria para aprovisionar el dispositivo. En realidad, tanto el tipo de unidad, los parámetros de tipo de unidad, el perfil, la unidad y el parámetro de la unidad secreta se crean sobre la marcha. Si el dispositivo no es compatible con la autenticación básica, el modo de descubrimiento no funcionará, ya que no se tendría forma de obtener el secreto del CPE. Todo este procedimiento solo se ejecutará una vez; la próxima vez que se conecte el CPE realizará una conversación estándar, aunque siempre utilizando la autenticación básica.

Si usa la funcionalidad de Jobs de FreeACS, puede establecer un cierto límite de descargas simultáneas. Esto es útil cuando la cantidad de CPE para actualizar puede ser mayor de lo que su red puede manejar. La figura 2.29 muestra la configuración de los controles varios.

```
# *** Fusion TR-069 Server Configuration file ***

# --- Various controls ---

# Allowed values are "none", "basic" and "digest". Digest authentication
# is default, and it is the most secure way to communicate with the devices.
# Combining this with SSL-setup, will give you a very secure provisioning.
auth.method = digest

# Will require username/password to download a Firmware/Config file/etc
# using the TR-069 Download method. FreeACS will instruct the CPE to use
# the ACS-username/password in the HTTP basic/digest challenge. Default
# is false, since this is a change introduced in version 3.1.0. It will become
# default within a few releases.
file.auth.used = false

# Discovery Mode can be set to true if you want to automatically add a new
# unittype and unit. This mode is violating the security of the system,
# because it allows unknown units to connect and then changes will be performed
# in the database. So use this option with caution, preferably when you want to
# add a new unittype to the system. Default is false.
discovery.mode = true

# Commaseparated black-list (if discovery.mode is true) - units with
# ACS-username containing these strings will be blocked.
discovery.block =

# concurrent download limit will limit the number of concurrent downloads
# allowed from this provisioning server. This is done to conserve bandwidth.
# This will override jobs/servicewindows if necessary, thus postponing the
# download to later. Default is 1000000 (virtually no limit).
concurrent.download.limit = 1000000
```

Figura 2.29. Configuración de los controles varios

2.1.2.4.2 Depuración

Al configurar debug.test.mode como "verdadero" puede probar un CPE. Luego se debe ir a la URL: http://192.168.2.10/tr069/test y registrar el dispositivo que desea probar. Además,

debe copiar la carpeta "tests" que encuentre dentro de tr069.war en su carpeta de trabajo (/ var / lib / tomcat7 /) y dentro de esta carpeta, a su vez, crear la carpeta "resultados" y la carpeta "modificada". La figura 2.30 muestra la página de inicio del proceso de pruebas.

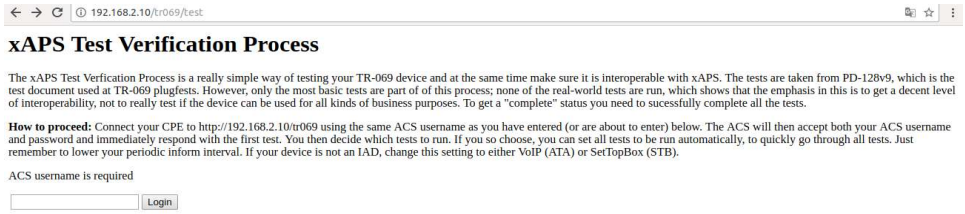

Figura 2.30. Interfaz del inicio del proceso de depuración.

La figura 2.31 muestra la interfaz de pruebas del sistema FreeACS.

Figura 2.31. Interfaz del proceso de depuración.

2.1.2.5 Servidor FreeACS Web [16]

Al iniciar sesión por primera vez, la base de datos de FreeACS no tiene usuarios pre configurados. En este caso, el sistema aceptará una combinación de nombre de usuario y contraseña predeterminada:

- username: admin
- password: xaps

La figura 2.32 muestra la interfaz de FreeACS Web una vez que se inició la sesión.

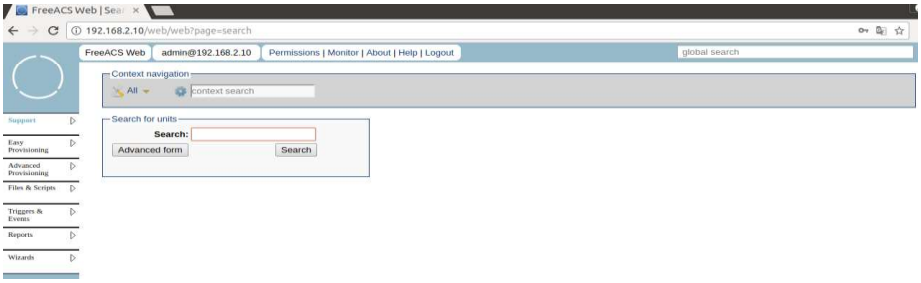

Figura 2.32. Interfaz de inicio de sesión en el Servidor FreeACS

2.1.3 PROCESO DE APROVISIONAMIENTO

2.1.3.1 Configuración del Mikrotik

Previamente a la instalación del paquete del tr069-client, se actualizó el firmware del Mikrotik para que permita la conexión con el servidor ACS, a partir de la versión de RouterOS 6.38 está disponible los recursos para tr069. Adicional se coloca el paquete tr069 correspondiente a la versión del firmware que está corriendo en el equipo, y se reinicia el Mikrotik. La figura 2.33 muestra los recursos del equipo de prueba.

```
[ admin@MikroTik ] /system resource > pr
                   uptime: 2h52m32s
                  version: 6.39.2 (stable)
               build-time: Jun/06/2017 08:01:04
             free-memory: 10.2MiB
            total-memory: 32.0MiB
                      cpu: MIPS 24Kc V7.4
                cpu-count: 1
            cpu-frequency: 400MHz
                 cpu-load: 12%
           free-hdd-space: 39.3MiB
          total-hdd-space: 64.0MiB
    write-sect-since-reboot: 1196
         write-sect-total: 1323426
              bad-blocks: 0%
        architecture-name: mipsbe
              board-name: RB750
                platform: MikroTik
[ admin@MikroTik ] /system resource >
```

Figura 2.33. Recursos del CPE

La figura 2.34 y 2.35 muestra la instalación del paquete TR-069 correspondiente a la versión utilizada.

Figura 2.34. Paquete TR-069.

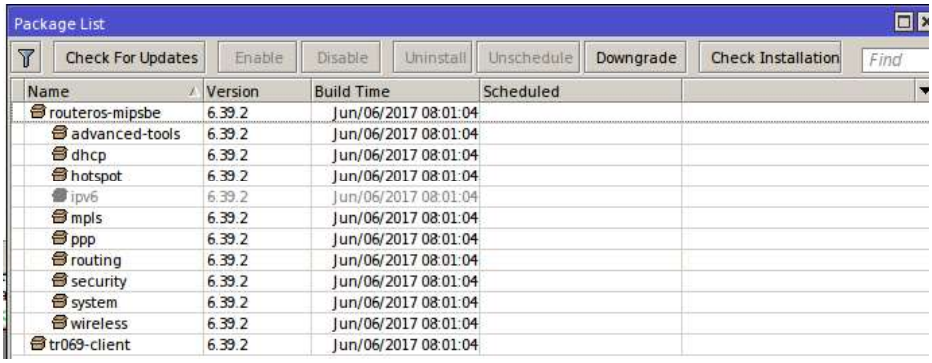

Figura 2.35. Revisión de los paquetes instalados.

La figura 2.36 muestra una secuencia de comandos que permitirá que Mikrotik se registre con el servidor FreeACS, siendo la dirección MAC la Id con la que se registra en el Servidor.

```
/system script
add name=identity owner=admin
policy=ftp,reboot,read,write,policy,test,password,sniff,sensitive source="delay 10;\
\n:local macaddress [/interface get ether1 mac-address]\
\n:local tidymac\
\n\
\n:for i from=0 to=([:len \$macaddress] - 1) do={ \
\n :local char [:pick \$macaddress \$i]\
\n :if (\$char = \":\") do={\
\n :set \$char \"\"\
\n }\
\n :set tidymac (\$tidymac . \$char) \
\n}\
\n:put \$tidymac; /tr069-client set acs-url=http://192.168.2.10/tr069 enabled=yes
password=xaps periodic-inform-enabled=yes periodic-inform-interval=640s username=\$tidymac"
```

Figura 2.36. Comandos para el registro en el Servidor

La figura 2.37 muestra los datos con los que se registrara el equipo después de realizar un reinicio y la figura 2.38 muestra los logs del equipo cuando se registra al servidor.

```
[ admin@MikroTik ] /tr069-client > pr
              enabled: yes
              acs-url: http://192.168.2.10/tr069
             username: 000C42E6F1A0
             password: xaps
   periodic-inform-enabled: yes
  periodic-inform-interval: 21h34m59s
connection-request-username:
connection-request-password:
       client-certificate: none
                   status: running
       last-session-error:
              retry-count: 0
[ admin@MikroTik ] /tr069-client >
```

Figura 2.37. Activación del protocolo TR-069

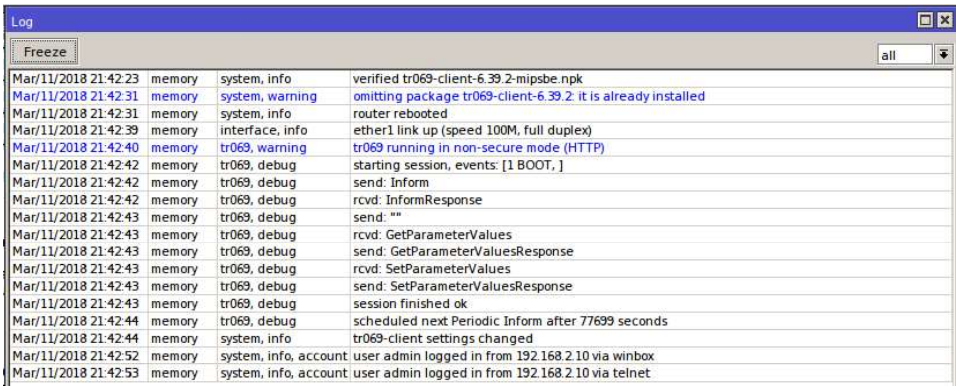

Figura 2.38. Logs de registro del CPE con el Servidor

2.1.3.2 Administración Enrutadores

Se inicia sesión en FreeACS, se hace clic en support> search y luego clic en el botón de búsqueda. El Mikrotik se mostrará con la dirección MAC como ID de la unidad. El dispositivo se puede administrar a través de TR-069. Una vez que se ha identificado el equipo se podrá realizar varios cambios como: el tiempo de monitoreo de un equipo, hacer una lectura de todos los parámetros configurados en el equipo. La figura 2.39 muestra que el equipo se encuentra aprovisionado en el Servidor.

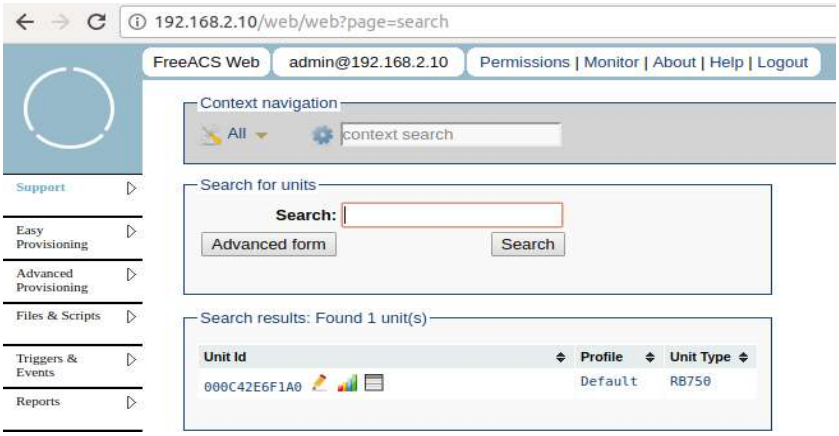

Figura 2.39. CPE aprovisionado

2.1.3.3 Comunicación entre el Servidor ACS Y El CPE [1]

Como se menciona en 1.2.3.2, el CPE inicia una sesión con el ACS en diferentes situaciones, una vez que se establece con éxito la conexión al ACS, el CPE inicia una

sesión enviando una solicitud de información inicial al ACS. Esto indica al ACS el estado actual del CPE y que el CPE está listo para aceptar solicitudes del ACS. La figura 2.40 muestra los logs en el CPE cuando inicia una nueva sesión.

Mar/14/2018 00:08:01	memory	system, info	tr069-client settings changed by admin
Mar/14/2018 00:08:02	memory	tr069, debug	starting session, events: [1 BOOT,]
Mar/14/2018 00:08:02	memory	tr069, debug	send: Inform
Mar/14/2018 00:08:03	memory	tr069, debug	rcvd: InformResponse
Mar/14/2018 00:08:03	memory	tr069, debug	send: ""
Mar/14/2018 00:08:03	memory	tr069, debug	rcvd: GetParameterValues
Mar/14/2018 00:08:03	memory	tr069, debug	send: GetParameterValuesResponse
Mar/14/2018 00:08:03	memory	tr069, debug	rcvd: SetParameterValues
Mar/14/2018 00:08:03	memory	system, info	tr069-client settings changed
Mar/14/2018 00:08:03	memory	tr069, debug	send: SetParameterValuesResponse
Mar/14/2018 00:08:03	memory	tr069, debug	session finished ok
Mar/14/2018 00:08:03	memory	tr069, debug	scheduled next Periodic Inform after 99477 seconds

Figura 2.40. Inicio de sesión del CPE con el ACS

2.1.3.3.1 Inform – InformResponse

Al recibir la solicitud de Inform inicial del CPE, si el ACS desea permitir el inicio de la Sesión, debe responder con una respuesta de InformResponse. El CPE debe considerar que la Sesión se inició con éxito si, y solo si, recibe una respuesta informada satisfactoria.

Desde el momento en que se inicia una sesión hasta que finaliza la sesión, el CPE debe garantizar la integridad transaccional de todos los parámetros accesibles a través del protocolo de gestión WAN CPE. Durante el transcurso de una sesión, todos los Parámetros configurables del CPE deben aparecer en el ACS como un conjunto consistente modificado solo por el ACS. La figura 2.41 y la figura 2.42 muestra el envió del Método Inform e InformResponse respectivamente.

```
eXtensible Markup Language
<soapenv:Envelope
   xmlns:soap='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:xsd='http://www.w3.org/2001/XMLSchema'
   xmlns:cwmp='urn:dslforum-org:cwmp-1-0'
   xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
 <soapenv:Body>
  <cwmp:Inform>
   <DeviceId>
   <Event
   <MaxEnvelopes>
   <CurrentTime>
   <RetryCount>
   <ParameterList
    </cwmp:Inform>
  </soapenv:Body>
 </soapenv:Envelope>
```

Figura 2.41. Envió del RPC Inform

Figura 2.42. RPC InformReponse

Durante una sesión, si el CPE tiene una o más solicitudes para enviar al ACS, el CPE debe enviar una de estas solicitudes en el próximo HTTP POST si y solo si se cumplen todas las condiciones siguientes:

- La respuesta HTTP recibida más recientemente del ACS no contenía una solicitud SOAP.
- El ACS ha indicado que HoldRequests es falso. Esta condición se cumple si y solo si la respuesta HTTP recibida más recientemente del ACS contenía uno de los siguientes:
  - ✓ Un sobre SOAP con el encabezado HoldRequests establecido en un valor de falso.
  - ✓ Un sobre SOAP sin encabezado HoldRequests.
  - ✓ Sin sobre SOAP (una respuesta HTTP vacía).

2.1.3.4 GetParemeterValues - GetParemeterValuesResponse

Este método puede ser utilizado por un ACS para obtener el valor de uno o más parámetros de CPE. La figura 2.43 muestra los argumentos de llamada para este método, los mismos que están contenidos en una cadena string.

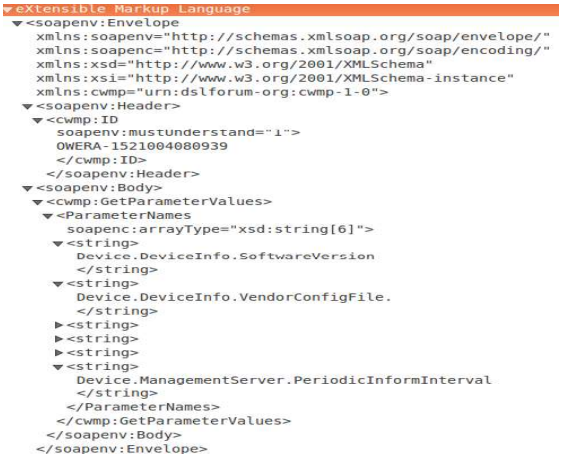

Figura 2.43. Argumentos RPC GetParemeterValues

La figura 2.44 muestra los argumentos en la respuesta para este método, los cuales están contenidos en la sentencia ParameterValueStruct.

Figura 2.44. Argumentos RPC GetParemeterValuesResponse

2.1.3.5 SetParameterValues - SetParameterValuesResponse

Este método puede ser utilizado por un ACS para modificar el valor de uno o más parámetros de CPE. La figura 2.45 muestra los argumentos de llamada para este método los que están dentro de las sentencias ParameterList y ParameterKey.

Figura 2.45. Argumentos RPC SetParemeterValues

Una respuesta exitosa a este método devuelve una enumeración de enteros definida de la siguiente manera:

- 0 = Todos los cambios de parámetros han sido validados y aplicados.
- 1 = Todos los cambios de parámetros han sido validados y confirmados, pero algunos o todos aún no se han aplicado.

La figura 2.46 muestra los argumentos en la respuesta para este método, los cuales están contenidos en la sentencia status.

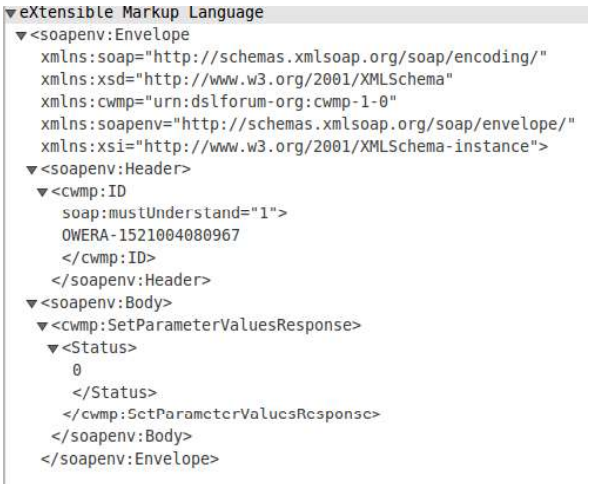

Figura 2.46. Argumentos RPC SetParemeterValuesResponse

2.1.3.6 Aprovisionamiento Forzado [17]

Si se requiere utilizar ConnectionRequestURL para activar los dispositivos, el dispositivo debe ser accesible directamente desde ACS al ConnectionRequestURL en particular. Además, la URL debe ser conocida por FreeACS de lo contrario, no podrá enviarle una solicitud. FreeACS podrá conocer este valor si y solo si especifica el indicador de parámetro unittype "AR" (lee "always-read"). Esto significa que siempre que el CPE cambie de dirección, se notificará a FreeACS sobre el cambio.

Dependiendo de la implementación del CPE, puede o no aceptar una solicitud a esta URL sin autenticación. Si su CPE requiere nombre de usuario y contraseña, entonces FreeACS también debe conocer estos parámetros. Sin embargo, los indicadores de parámetros, esos dos parámetros son "RW", lo que significa que estos parámetros deben ser especificados por FreeACS. Por lo tanto, debe especificar un nombre de usuario y contraseña en FreeACS para estos dos parámetros. No se debe cambiar la bandera de

"RW" a "RA". Después de que el CPE haya recibido su nombre de usuario y contraseña, en el próximo ciclo de provisiones y FreeACS tiene una URL correcta en su base de datos Y el CPE es accesible en esa URL en particular, luego el botón "Leer todo" y "Provisión" trabajaran de manera normal. Esto permite que se pueda realizar los cambios de manera inmediata.

2.1.3.6.1 Pruebas de comunicación

2.1.3.6.1.1 Cambio de IP de la interfaz LAN

La figura 2.47 muestra las configuraciones iniciales de las subredes del equipo router.

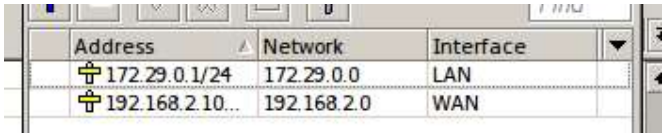

Figura 2.47. Subredes del Equipo router

La figura 2.48 muestra como a través del servidor Web de FreeACS en el campo IPV4Address.IPAddress se realiza el cambio.

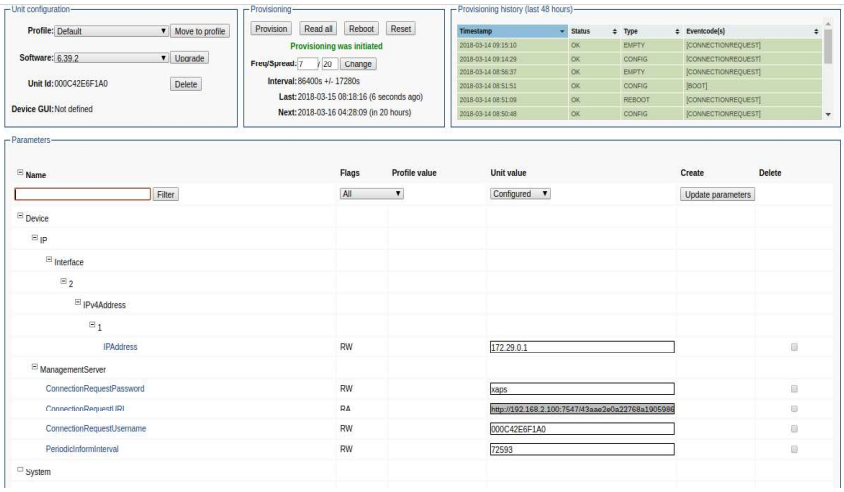

Figura 2.48. Cambio de Ip LAN en el FreeACS

Se realiza los cambios requeridos, se presiona Update Parameters y se da click en el botón provisioning, con estos el ACS envía una solicitud de inicio de sesión al CPE para realizar el cambio. La figura 2.49 muestra los logs del equipo router donde se observa que se realiza el cambio.

Mar/15/2018 08:18:17	memory	tr069, debug	session finished ok
Mar/15/2018 08:18:17	memory	tr069, debug	scheduled next Periodic Inform after 72593 seconds
Mar/15/2018 08:25:10	memory	tr069, debug	starting session, events: [6 CONNECTION REQUEST,]
Mar/15/2018 08:25:10	memory	tr069, debug	send: Inform
Mar/15/2018 08:25:10	memory	tr069, debug	rcvd: InformResponse
Mar/15/2018 08:25:10	memory	tr069, debug	send: ""
Mar/15/2018 08:25:10	memory	tr069, debug	rcvd: GetParameterValues
Mar/15/2018 08:25:10	memory	tr069, debug	send: GetParameterValuesResponse
Mar/15/2018 08:25:10	memory	tr069, debug	rcvd: SetParameterValues
Mar/15/2018 08:25:10	memory	system, info	address changed
Mar/15/2018 08:25:10	memory	system, info	tr069-client settings changed
Mar/15/2018 08:25:10	memory	tr069, debug	send: SetParameterValuesResponse
Mar/15/2018 08:25:10	memory	tr069, debug	session finished ok
Mar/15/2018 08:25:10	memory	tr069, debug	scheduled next Periodic Inform after 72552 seconds

Figura 2.49. Logs de Cambio de IP LAN del equipo

La figura 2.50 muestra el cambio realizado en la interfaz LAN del CPE

Address	Network	Interface
20.20.20.1/24	20.20.20.0	LAN
192.168.2.10...	192.168.2.0	WAN

Figura 2.50. Configuración final de la subred LAN

Analizando los paquetes con la herramienta wireshark se puede visualizar que el GetParametersValuesResponse enviado por el CPE informa al ACS que tiene configurada la IP 172.29.0.1 como Gateway de la red LAN, como se muestra en la figura 2.51. Posteriormente el SetParametersValues enviado por el ACS setea la nueva IP para la red seleccionada.

```
▼<soapenv:Envelope
   xmlns:soap='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:xsd='http://www.w3.org/2001/XMLSchema'
   xmlns:cwmp='urn:dslforum-org:cwmp-1-0'
   xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
   xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'>
 ▶<soapenv:Header>
 ▼<soapenv:Body>
  ▼<cwmp:GetParameterValuesResponse>
   ▼<ParameterList
      soap:arrayType='cwmp:ParameterValueStruct[9]'>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▼<ParameterValueStruct>
     ▼<Name>
        Device.IP.Interface.2.IPv4Address.1.IPAddress
        </Name>
     ▼<Value
        xsi:type='xsd:string'>
        172.29.0.1
        </Value>
      </ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
    ▶<ParameterValueStruct>
     </ParameterList>
   </cwmp:GetParameterValuesResponse>
  </soapenv:Body>
 </soapenv:Envelope>
```

Figura 2.51. GetParametersValuesResponse muestra el Gateway inicialmente configurado

2.1.3.6.1.2 Trasferencia de archivos de configuración [18]

RouterOS tiene muchos atributos de configuración y no todo se puede transportar a los parámetros de CWMP, por eso RouterOS ofrece la posibilidad de ejecutar su poderoso lenguaje de scripts para configurar cualquier atributo. Se puede realizar una alteración de configuración (que en realidad es una ejecución de script normal) usando Download RPC FileType = "3 Vendor Configuration File" con la extensión de archivo descargable ". alter". Esta potente función se puede utilizar para configurar cualquier atributo ROS que no esté disponible a través de los parámetros de CWMP.

2.1.3.6.1.2.1 Proceso para realizar Trasferencia.

- Crear un archivo de texto que contenga las configuraciones que se requieren, cuya extensión es. alter.
- Se lo almacena en la base de datos de FreeACS como un archivo TR069_SCRIPT.
- Se crea un grupo donde se contenga los usuarios a los que se requiere realizar los cambios.
- Se crea un trabajo a través del cual se pueda ejecutar el script ya almacenado.

La figura 2.52 muestra el proceso descrito.

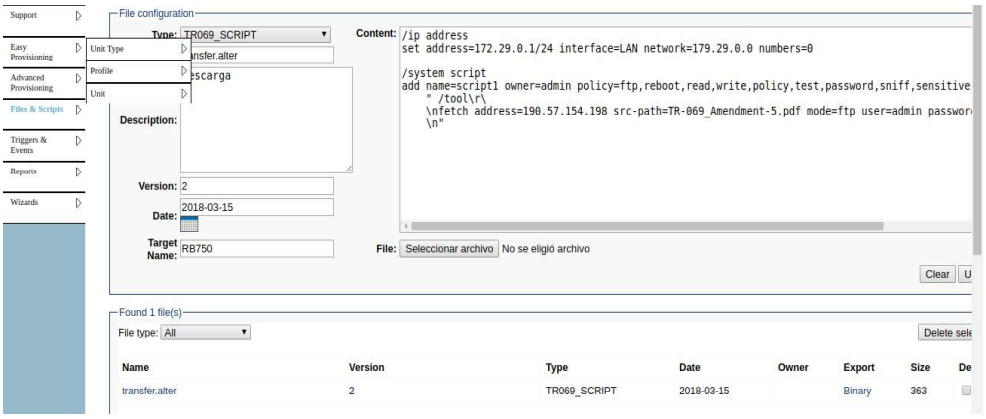

Figura 2.52. Creación de archivo ".alter" en el FreeACS

La figura 2.53 muestra el trabajo creado para realizar la transferencia.

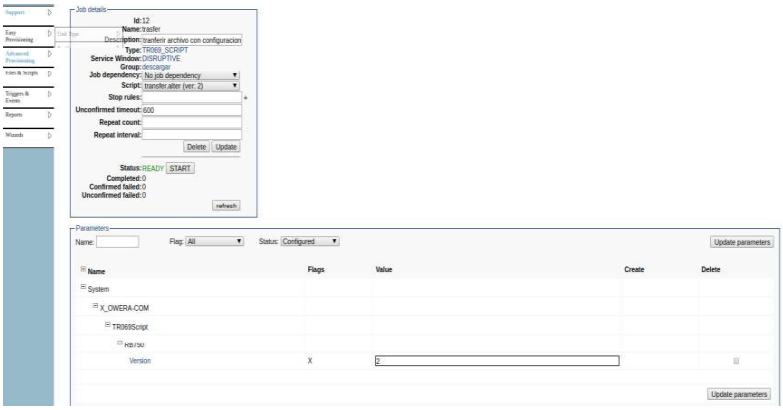

Figura 2.53. Ejecución del Trabajo de Transferencia

Para la prueba se ha creado un script con el cual se cambie la configuración de la interfaz LAN del equipo Mikrotik y adicional se introduce un script para la descarga de un archivo un servidor externo. La figura 2.54 muestra la configuración inicial del equipo Mikrotik.

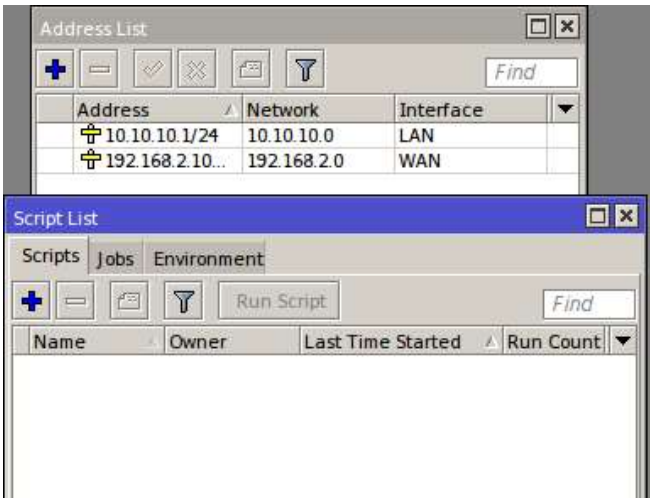

Figura 2.54. Configuración inicial para la prueba

Al revisar los logs del equipo se observa que se inicia la sesión y se realiza la descarga del archivo, y la confirmación de la transferencia, tal como se muestra en la figura 2.55.

Mar/19/2018 09:32:50	memory	tr069, debug	starting session, events: [6 CONNECTION REQUEST,]
Mar/19/2018 09:32:50	memory	tr069, debug	send: Inform
Mar/19/2018 09:32:50	memory	tr069, debug	rcvd: InformResponse
Mar/19/2018 09:32:50	memory	tr069, debug	send: ""
Mar/19/2018 09:32:50	memory	tr069, debug	rcvd: GetParameterValues
Mar/19/2018 09:32:50	memory	tr069, debug	send: GetParameterValuesResponse
Mar/19/2018 09:32:50	memory	tr069, debug	rcvd: Download
Mar/19/2018 09:32:50	memory	tr069, debug	send: DownloadResponse
Mar/19/2018 09:32:51	memory	tr069, debug	session finished ok
Mar/19/2018 09:32:51	memory	system, info	address changed
Mar/19/2018 09:32:51	memory	system, info	new script added new script added
Mar/19/2018 09:32:51	memory	tr069, debug	starting session, events: [7 TRANSFER COMPLETE, M Download (2),]
Mar/19/2018 09:32:51	memory	tr069, debug	send: Inform
Mar/19/2018 09:32:51	memory	tr069, debug	rcvd: InformResponse
Mar/19/2018 09:32:51	memory	tr069, debug	send: TransferComplete [""]
Mar/19/2018 09:32:51	memory	tr069, debug	rcvd: TransferCompleteResponse
Mar/19/2018 09:32:51	memory	tr069, debug	send: ""
Mar/19/2018 09:32:51	memory	tr069, debug	rcvd: GetParameterValues
Mar/19/2018 09:32:51	memory	tr069, debug	send: GetParameterValuesResponse
Mar/19/2018 09:32:51	memory	tr069, debug	session finished ok
Mar/19/2018 09:32:51	memory	tr069, debug	starting session, events: []
Mar/19/2018 09:32:51	memory	tr069, debug	send: Inform
Mar/19/2018 09:32:51	memory	tr069, debug	rcvd: InformResponse
Mar/19/2018 09:32:51	memory	tr069, debug	send: ""
Mar/19/2018 09:32:51	memory	tr069, debug	rcvd: GetParameterValues
Mar/19/2018 09:32:51	memory	tr069, debug	send: GetParameterValuesResponse
Mar/19/2018 09:32:51	memory	tr069, debug	session finished ok
Mar/19/2018 09:32:51	memory	tr069, debug	scheduled next Periodic Inform after 86908 seconds

Figura 2.55. Logs de Descarga y de Transferencia

Al analizar los paquetes XML correspondiente al RPC Download se verifica que el servidor FreeACS genera la URL de acceso al archivo. Además, se observa que lo detecta como FileType = "3 Vendor Configuration File", tal como se muestra en la figura 2.56.

```
eXtensible Markup Language
 <soapenv:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
  <soapenv:Header>
    <cwmp:ID
     </soapenv:Header>
  <soapenv:Body>
   <cwmp:Download
      xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
    <CommandKey>
       2
       </CommandKey>
    <FileType>
       3 Vendor Configuration File
       </FileType>
    <URL>
       http://192.168.2.10:80/tr069/file/TR069_SCRIPT/2/RB750/000C42E6F1A0/transfer.alter
       </URL>
    <Username>
    <Password>
    <FileSize>
    <TargetFileName>
    <DelaySeconds>
    <SuccessURL>
    <FailureURL>
       </FailureURL>
     </cwmp:Download>
   </soapenv:Body>
  </soapenv:Envelope>
```

Figura 2.56. Revisión RPC Download

La figura 2.57 muestra que se han configurado de manera correcta los comandos dentro del equipo Mikrotik.

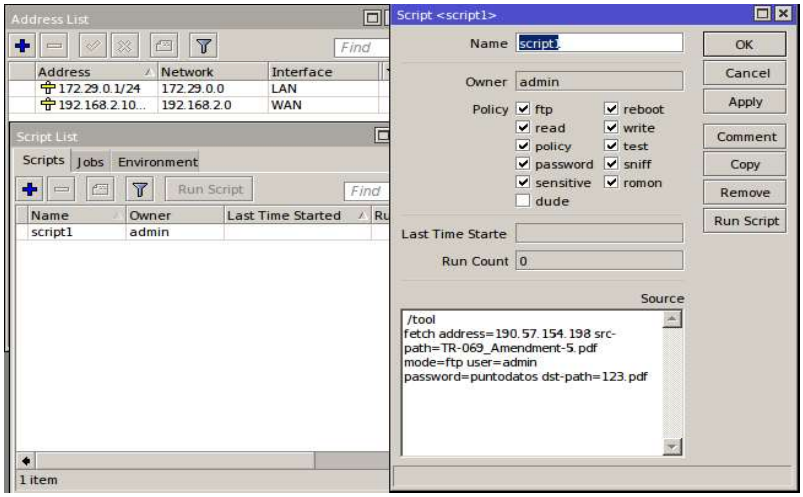

Figura 2.57. Nueva Configuración después de haber corrido el archivo ".alter"

2.1.3.6.1.3 Actualización de Firmware [18]

El estándar CWMP define que el firmware de CPE puede actualizarse mediante Download RPC con FileType="1 Firmware Upgrade Image" y una sola URL de un archivo descargable. El standard también establece que el archivo descargado puede ser de cualquier tipo y que el proceso específico del vendedor se puede aplicar para finalizar la actualización del firmware. Debido a que la actualización de Mikrotik está basada en paquetes, se usa un archivo XML para describir la actualización del firmware. La configuración XML admite el suministro de varias URL de archivos, que se descargarán y aplicarán de forma similar a la actualización regular de RouterOS mediante la carga de archivos de firmware y paquetes.

Para la actualización del paquete RouterOS y la actualización del paquete tr069-client. Se creó un archivo XML el cual se almacenó en la base de datos de FreeACS como SOFTWARE en el detalle del tipo de archivo, adicionalmente se colocó la versión deseada del RouterOS y del paquete TR-069. En el archivo XML se colocan los links de descarga de los paquetes de actualización, para que puedan ser extraídos por el CPE.

Como se muestra 2.58 se ha creado un archivo .xml que contiene las direcciones donde están alojadas las nuevas versiones de Software necesarios para el Mikrotik, las mismos que están almacenadas dentro del mismo Servidor FreeACS, tal como se muestra en la figura 2.59.

upgrade. xml ×

```
<upgrade version="1" type="links">
<config/>
<links>
<link>
<url>http://192.168.2.10/tr069/file/SOFTWARE/6.40/RB750/routeros-mipsbe-6.40.npk</url>
</link>
<link>
<url>http://192.168.2.10/tr069/file/SOFTWARE/6.40.0/RB750/t069-client-6.40-mipsbe.npk</url>
</link>
</links>
</upgrade>
```

Figura 2.58. Creación del archivo XML que contiene los links de descarga

File type: All — Delete selected files

Name	Version	Type	Date	Owner	Export	Size	Delete
routeros-mipsbe-6.40.npk	6.40	SOFTWARE	2018-03-19		Binary	10447478	
tr069-client-6.40-mipsbe.npk	6.40.0	SOFTWARE	2018-03-19		Binary	135249	
upgrade.xml	3	SOFTWARE	2018-03-19		Binary	283	

Figura 2.59. Nuevas Versiones del Software almacenadas en el FreeACS

Al igual que la transferencia de archivos, se crea un trabajo para poder realizar la descarga del archivo XML, al analizar los paquetes transferidos, se observa que los paquetes del Download RPC detectan el FileType="1 Firmware Upgrade Image" apuntando al archivo XML que contiene los links de los paquetes a actualizar, tal como se observa en la figura 2.60.

```
eXtensible Markup Language
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
 <soapenv:Header>
  <cwmp:ID
     soapenv:mustUnderstand="1">
     OWERA-1521475631975
     </cwmp:ID>
   </soapenv:Header>
 <soapenv:Body>
  <cwmp:Download
     xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
   <CommandKey>
      3
     </CommandKey>
   <FileType>
      1 Firmware Upgrade Image
     </FileType>
   <URL>
      http://192.168.2.10:80/tr069/file/SOFTWARE/3/RB750/000C42E6F1A0
     </URL>
   <Username>
   <Password>
   <FileSize>
   <TargetFileName>
   <DelaySeconds>
   <SuccessURL>
   <FailureURL>
    </cwmp:Download>
  </soapenv:Body>
</soapenv:Envelope>
```

Figura 2.60. RPC Download de Software

Una vez que el Mikrotik reconoce el archivo XML, descarga los paquetes y se reinicia el equipo de manera automática, una vez que se inicia el equipo nuevamente se habrá realizado la actualización, tal como se muestra en la figura 2.61 y 2.62

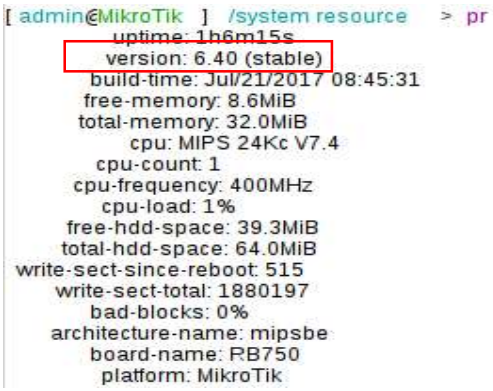

Figura 2.61. Versión Actualizada del Firmware

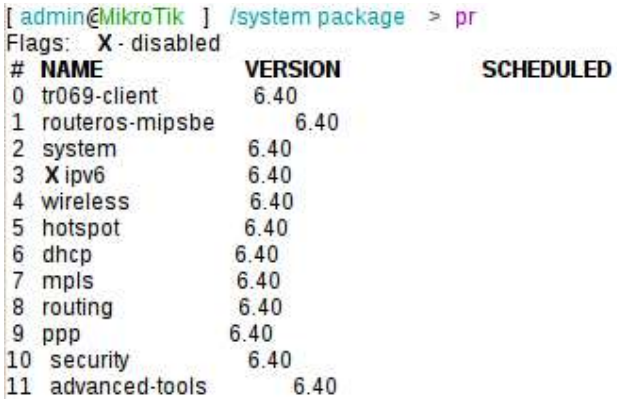

Figura 2.62. Paquetes Actualizados

2.1.3.7 Diagrama de Red de Equipos

Tal como se menciona en 1.8, los enlaces de radio residenciales utilizan una topología punto multipunto, esto permite que dentro de un nodo se puedan tener varias bases con diferentes ángulos de inclinación, con esto se puede abarcar grandes áreas de cobertura dentro de un determinado sector y con un amplio número de suscriptores. Actualmente se mantiene un promedio de 7 suscriptores en cada una de las bases de un nodo.

PUNTONET al ser dueño de su propia red MLPS puede enrutar los diferentes enlaces pertenecientes a cada nodo a través de un equipo de borde que es el que se conecta con la nube de internet. Esto permite que los diferentes concentradores a los que se asocia cada nodo, puedan ser visualizadas desde las áreas de soporte. A partir de esto se ha diseñado un diagrama de red donde se considera que el servidor ACS ya está dentro de la red corporativa, por lo que tiene comunicación con el equipo de borde que le permite la comunicación hacia la nube de internet y a la vez puede visualizar a los diferentes

concentradores de tal manera que puede tener conexión entre las bases y las radios de cada uno de los clientes. La figura 2.63 muestra el diagrama red del proyecto.

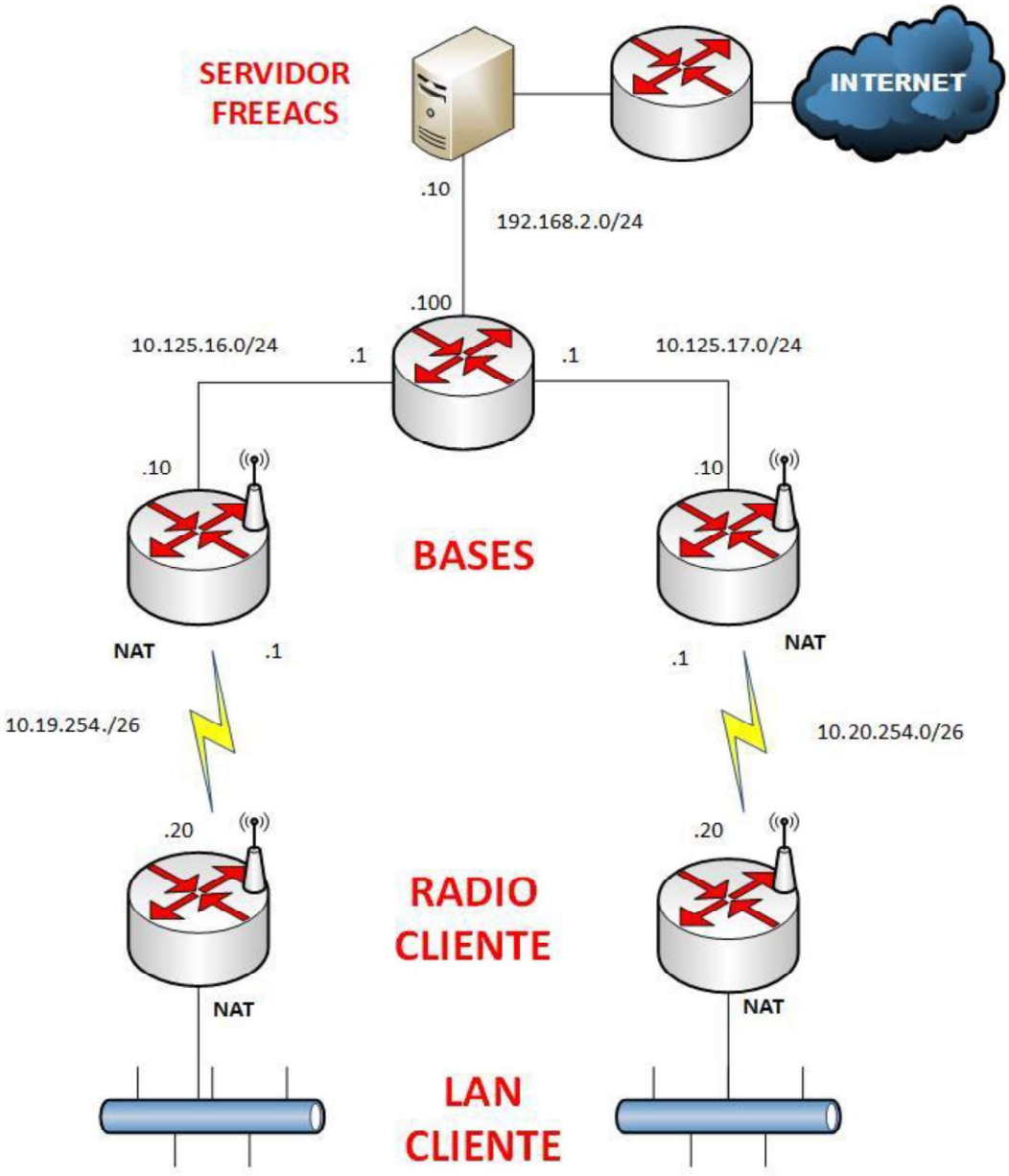

Figura 2.63. Diagrama de Red

3. RESULTADOS Y DISCUSIÓN

3.1 RESULTADOS OBTENIDOS EN LAS PRUEBAS

Las pruebas que se realizaron están basadas en el diagrama de red mostrado en 2.3.4. Se realizaron pruebas con una cantidad desde 4 hasta 7 equipos Mikrotik con los cuales se realizaron diez repeticiones de la prueba de descarga del archivo xml que contienen los links de los paquetes que permiten actualizar el software del equipo. Cada repetición de la prueba siguió el siguiente proceso:

- Instalación de una versión del RouterOS a partir de la 6.38.
- Instalación del paquete TR069 correspondiente a la versión actual del equipo.
- Configuración del TR069-client en el Mikrotik.
- Aprovisionamiento del equipo en el servidor FreeACS.
- Carga de los paquetes de la versión a la que se quiere actualizar los equipos en la base de datos del servidor.
- Creación del archivo xml que contiene los links de descarga de los paquetes de actualización.
- Carga del archivo xml en la base de datos del servidor.
- Creación de los Jobs necesarios para ejecutar la descarga del archivo xml para que sea identificado por el equipo Mikrotik.
- Validación de la descarga del archivo xml en el equipo Mikrotik.
- Reinicio del equipo.
- Validación si la actualización fue exitosa.

Dentro de las pruebas realizadas se usaron conexiones por radio y por cobre, donde no se identificaron diferencias durante el proceso de actualización del firmware de los equipos, por tal motivo las pruebas realizadas, en su gran mayoría, se realizaron con enlaces con conexiones por cable.

Las figuras 3.1, 3.2 y 3.3 muestran la ejecución de los Jobs de actualización, indicando las descargas que se van completando.

Figura 3.1. Actualizaciones completadas de equipos RB750

Figura 3.2. Actualizaciones completas de equipos SXT, durante la ejecución del Job

Figura 3.3. Actualizaciones completas de equipos SXT 5HPnD durante la ejecución del Job

La tabla 3.1 muestra los resultados obtenidos de las pruebas con 4 equipos.

Tabla 3.1. Resultados Obtenidos en las Pruebas con 4 equipos

	Número de equipos de prueba	Número de equipos actualizados
Prueba 1	4	3
Prueba 2	4	2
Prueba 3	4	4
Prueba 4	4	4
Prueba 5	4	3
Prueba 6	4	4
Prueba 7	4	2
Prueba 8	4	2
Prueba 9	4	4
Prueba 10	4	4

De las diez pruebas realizadas con cuatro equipos, solo en cinco se actualizaron todos los equipos, lo que arroja un 50% de efectividad del total de pruebas realizadas. En tres de las pruebas restantes, solo dos equipos lograron completar la actualización, lo que nos da un 15% de efectividad. En los otros dos momentos de prueba, se actualizaron tres equipos lo que nos da un 15% de efectividad. Así se consigue una efectividad del 30% en las pruebas, donde se actualizaron de dos a tres equipos. En total se obtuvo un 80% de efectividad al actualizarse los equipos. La tabla 3.2 muestra los resultados obtenidos de las pruebas con 5 equipos.

Tabla 3.2. Resultados Obtenidos en las Pruebas con 5 equipos

	Número de equipos de prueba	Número de equipos actualizados
Prueba 1	5	3
Prueba 2	5	5
Prueba 3	5	5
Prueba 4	5	4
Prueba 5	5	2
Prueba 6	5	5
Prueba 7	5	4
Prueba 8	5	4
Prueba 9	5	4
Prueba 10	5	3

De las diez pruebas realizadas con cinco equipos, solo en tres se actualizaron todos los equipos, lo que arroja un 30% de efectividad del total de pruebas realizadas. En cuatro de las pruebas restantes, cuatro equipos lograron completar la actualización, lo que nos da un 32% de efectividad. En dos momentos de prueba, se actualizaron tres equipos lo que nos da un 12% de efectividad. En una sola prueba, se consigue que dos equipos se actualicen,

lo que arroja un 4% de efectividad Así se consigue una efectividad del 48% en las pruebas, donde se actualizaron de dos a cuatro equipos. En total se obtuvo un 78% de efectividad al actualizarse los equipos. La tabla 3.3 muestra los resultados obtenidos de las pruebas con 6 equipos.

Tabla 3.3. Resultados Obtenidos en las Pruebas con 6 equipos

	Número de equipos de prueba	**Número de equipos actualizados**
Prueba 1	6	4
Prueba 2	6	4
Prueba 3	6	6
Prueba 4	6	6
Prueba 5	6	6
Prueba 6	6	4
Prueba 7	6	4
Prueba 8	6	5
Prueba 9	6	4
Prueba 10	6	6

De las diez pruebas realizadas con seis equipos, solo en cuatro se actualizaron todos los equipos, lo que arroja un 40% de efectividad del total de pruebas realizadas. En una de las pruebas restantes, cinco equipos lograron completar la actualización, lo que nos da un 8.33% de efectividad. En cinco momentos de prueba, se actualizaron cuatro equipos lo que nos da un 33.33% de efectividad. Así se consigue una efectividad del 41.66% en las pruebas, donde se actualizaron entre cuatro a cinco equipos. En total se obtuvo un 81.66% de efectividad al actualizarse los equipos. La tabla 3.4 muestra los resultados obtenidos de las pruebas con 7 equipos.

Tabla 3.4. Resultados Obtenidos en las Pruebas con 7 equipos

	Número de equipos de prueba	**Número de equipos actualizados**
Prueba 1	7	5
Prueba 2	7	7
Prueba 3	7	6
Prueba 4	7	5
Prueba 5	7	6
Prueba 6	7	7
Prueba 7	7	4
Prueba 8	7	7
Prueba 9	7	4
Prueba 10	7	5

De las diez pruebas realizadas con siete equipos, solo en tres se actualizaron todos los equipos, lo que arroja un 30% de efectividad del total de pruebas realizadas. En dos de las pruebas restantes, seis equipos lograron completar la actualización, lo que nos da un 17.14 % de efectividad. En cinco momentos de prueba, se actualizaron cuatro equipos lo que nos da un 33.33% de efectividad. En tres momentos de pruebas, cinco equipos completan la actualización, lo que arroja un 21.43% de efectividad. En dos momentos de prueba restantes, cuatro equipos logran completar la actualización, lo que nos arroja un 11.43% de efectividad. Así se consigue una efectividad del 50% en las pruebas, donde se actualizaron entre cuatro a seis equipos. En total se obtuvo un 80% de efectividad al actualizarse los equipos.

En base a los datos obtenidos se ha podido determinar el porcentaje de efectividad promedio del proceso de actualización, el cual alcanza el 80% de efectividad al actualizar los equipos Mikrotik. A partir de este porcentaje se podrá realizar una comparación entre los costos actuales y los costos que se tendría si se implementara el sistema de administración TR-069. La tabla 3.5 muestra el porcentaje de efectividad de las pruebas.

Tabla 3.5. Porcentajes de las pruebas.

Porcentaje 1	Porcentaje 2	Porcentaje 3	Porcentaje4	Promedio de Efectividad
80	78	81.66	80	80

El 20% restante representa la cantidad de equipos que no se pudieron actualizar debido a uno de los siguientes factores: mala conexión física, inhibición de los equipos, cables deteriorados, desconexión del equipo con el servidor.

3.2 ANÁLISIS DE COSTOS

3.2.1 Número de Visitas Técnicas

Dentro de la revisión de los enlaces inalámbricos, existen varias soluciones que se brinda a los clientes, entre ellas se destacan: el cambio de frecuencias del enlace de radio, el cambio de base de manera remota y la generación de una visita técnica. Las visitas técnicas pueden ser generadas por diferentes causales. La Tabla 3.6 muestra el número de visitas realizadas el primer tercio del año 2018 y los motivos de las visitas.

Tabla 3.6. Número de visitas técnicas y sus causales [19]

	Enero	Febrero	Marzo	Abril	Total	%
Reapuntamiento	62	67	78	72	279	46
Cambio de Equipo	29	28	36	36	129	22
Cambio de Poe	25	30	34	30	119	20
Actualización de Firmware	16	17	18	19	70	12
NUMERO DE VISTAS	131	142	167	157	597	100

De acuerdo a la tabla anterior se puede observar que las visitas por Actualización de Firmware representan el 12% del total de visitas realizas en el transcurso de los primeros cuatro meses del año, siendo este, un valor elevado, el cual es un factor que requiere ser optimizado.

3.2.2 Costos de una Visita Técnica

La Tabla 3.7 muestra los valores que se consideran dentro de una visita técnica de radio.

Tabla 3.7. Costo de una Visita/Hora en dólares [10]

Elementos de costo	Valor($)	Periodo	Valor/día($)	Valor/h.lab($)
Equipos y Herramientas	2200	mensual	73,33	9,17
Combustibles	30	semanal	6,00	0,75
Gastos Varios	15	semanal	3,00	0,38
Alimentación	30	semanal	6,00	0,75
Salario Técnico	520	mensual	17,33	2,17
Salario Ayudante	420	mensual	14,00	1,75
Equipos de Protección Personal	850	mensual	28,33	3,54
Indumentaria	200	anual	0,55	0,07
Alquiler de vehículos	350	mensual	11,67	1,46
Costo Visita/Hora($)	20,03			

La Taba 3.8 muestra el costo mensual que se tiene al realizar visitas técnicas por Actualización de Firmware, considerando que en promedio una visita técnica dura una hora y media.

Tabla 3.8. Costo mensual de Visitas por Actualización de Firmware [10]

	ENERO	FEBRERO	MARZO	ABRIL
Número de vistas	16	17	18	19
Duración promedio visita(horas)	1,5			
Costo visita/hora($)	20,03			

Costo mensual($)	**480,72**	**510,77**	**540,81**	**570,855**
Costo total($)	**2103,15**			

Los costos que se generan por las visitas a causa de no poder actualizar el software de los equipos de manera remota son elevados.

3.3 COMPARACIÓN DE RESULTADOS

Actualmente no existe un método para realizar las actualizaciones de los equipos CPE's, este proceso se realiza únicamente cuando un suscriptor se comunica con el soporte técnico, siendo el operador de CallCenter, el que considera si se debe o no efectuar un upgrade de la versión del equipo, por lo tanto, no se garantiza que al final de un soporte telefónico el cliente tenga la última versión estable del software, instalado en su equipo.

Como se observa en los resultados de las pruebas obtenidas, el uso del protocolo TR-069, tiene un 80% de efectividad al momento de realizar una actualización masiva de las versiones de los firmwares de los equipos, permitiendo reducir el número de visitas técnicas generadas debido a este causal. La tabla 3.9 muestra el porcentaje estimado de reducción de las visitas técnicas

Tabla 3.9. Porcentaje Estimado De Reducción De Visitas

	ACTUALIDAD	**TR 069 CON 20 % SIN ACTUALIZAR**
Vistas por actualización	70	14
Número total de visita	597	541
Porcentaje de visita	12	2,4
Porcentaje de reducción	**9,6**	

Figura 3.4. Número de visitas

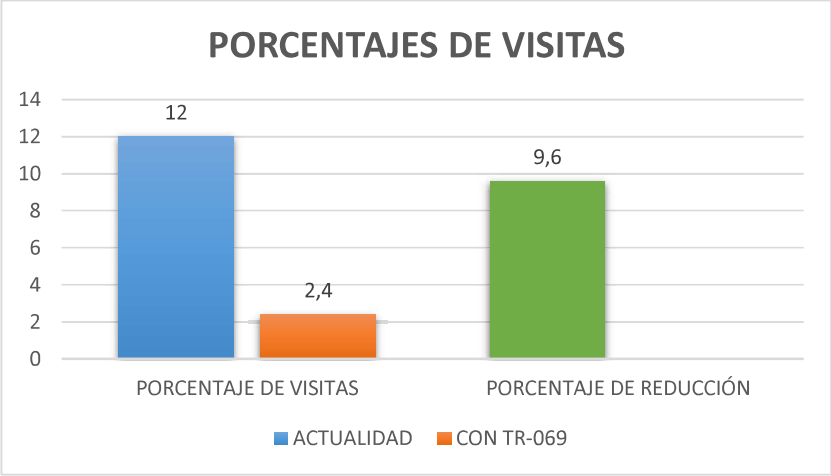

Figura 3.5. Porcentaje de visitas

Considerando que dentro de las pruebas realizadas se obtuvo un 20% de equipos que no se actualizaron, se podría asumir que ese sería el porcentaje de clientes que necesariamente requerirían una visita técnica para actualización del firmware. Por lo que se te tiene una reducción de 70 visita a 14 visitas, por actualización de firmware, manteniendo el número de visitas generadas por los otros causales, el nuevo valor estimado de visitas que se hubiesen tenido al ejecutar el protocolo TR-069 sería de 541 visitas en total, tal como se muestra en la figura 3.4. En porcentajes se tiene una reducción que representa el 9.6% del total de visitas técnicas dentro del periodo comprendido entre enero y abril del 2018, tal como se muestra en la figura 3.5.

La tabla 3.9 muestra el porcentaje de ahorro en los costos de las visitas por actualización de firmware.

Tabla 3.10. Porcentaje de Ahorro en Visitas por Actualizaciones.

	Actualidad	**Con TR-069**
Vistas por actualización	70	14
Duración promedio visita(horas)	1,5	
Costo visita/hora	20,03	
Costo	2103,15	420,63
Valor ahorrado	1682,52	
Porcentaje de ahorro	80%	

Figura 3.6. Costos y ahorro

Dentro del periodo de estudio que corresponde a cuatro meses, se estima que se ha generado un gasto 2103,15 dólares, en la realización de visitas por actualización de firmware, el porcentaje estimado de ahorro con la ejecución del protocolo TR-069 podría alcanzar el 80%. Se estima que se produciría un ahorro de 1682.52 dólares dentro del primer cuatrimestre, tal como se muestra en la figura 3.6. Si se considera que en los siguientes dos cuatrimestres se mantiene un igual valor de ahorro, al finalizar el año, podría representar un ahorro aproximado de 5047,56 dólares. Por lo que la implementación del protocolo TR-069 resulta una solución viable para reducción del número de visitas por actualizaciones de firmware, y así generar un gran porcentaje de ahorro.

CAPITULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El protocolo TR-069 es un sistema de administración a través del acceso WAN, con el que se puede gestionar diferentes equipos CPE's, independientemente del fabricante, ya que posee herramientas suficientemente útiles para poder realizar cualquier modificación de parámetros, que no estén dentro de los detectados por el protocolo como la generación de scripts. Esto permitiría a la empresa PUNTONET tener un mayor control sobre los equipos instalados, elevando los porcentajes de estabilidad de los enlaces.

- El protocolo TR-069 se complementa con algunos protocolos estándar para poder facilitar la comunicación entre el servidor ACS y el CPE, mediante archivos xml, los que permiten la lectura y escritura de parámetros trasferencia de archivos de configuración y actualización de firmwares.

- El Servidor ACS FreeACS es una plataforma fácil de instalar y con una interfaz muy amigable para el operador, que engloba una gran cantidad de parámetros, que pueden ser detectados a través de los equipos aprovisionados por lo que existe una amplia gama de dispositivos con los que puede establecer comunicación.

- El servidor FreeACS permite un alto grado de escalabilidad y flexibilidad, debido a que sus módulos son implementaciones Java, que pueden ejecutarse en un middleware JEE estándar, es decir, sobre cualquier sistema operativo con una implementación JEE ejecutándose en él.

- Por medio del uso del servidor FreeACS, se pudo analizar el funcionamiento, requerimientos, y las diferentes funciones que presta el protocolo TR-069, para la administración de CPE's de manera remota

- Dentro de las pruebas realizadas se pudo comprobar el comportamiento que tienen los equipos Mikrotik al ser administrados con el protocolo TR-069 de tal manera que

se pudo realizar cambios de diferentes parámetros de manera remota de los CPE's a través del servidor de configuración automática ACS.

- Dentro del análisis de resultados se observa que actualmente no se tiene un proceso definido para realizar las actualizaciones en los equipos del cliente, con la implementación del protocolo TR-069 se podrá crear un cronograma de actualización de firmware donde se pueda tomar en cuenta los sectores, periodos de menor uso, y número de suscriptores.

4.2 RECOMENDACIONES

- Dado el porcentaje de efectividad obtenido en las pruebas del sistema de administración WAN, sería recomendable que este sistema sea implementado dentro de la red de PUNTONET, como una herramienta para poder tener control sobre los periodos de actualización y las sobre las versiones de los firmwares con mayor estabilidad para evitar problemas en las comunicaciones y así extenderlo a los diferentes Proveedores de Servicios de Internet

- El Protocolo de Administración WAN, es una poderosa herramienta, que de ser adoptada por los diferentes ISP's, podría contribuir al aumento de la capacidad de escalabilidad y flexibilidad debido a su compatibilidad con los diferentes medios de transmisión como cobre, radio y fibra.

- Generar procesos de manejo y administración del sistema, donde se defina qué tipo de permisos tendrían los diferentes usuarios que se generen dentro del servidor.

- Definir políticas de seguridad para detallar un mejor protocolo de conexiones entre el servidor y los equipos de Core de la empresa, para así evitar posibles ataques de accesos no autorizados.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] BroadBand Forum, TR-069 CPE WAN MANAGEMET PROTOCOL, 2013.

[2] «CONECTRONICA,» 4 Oct 2012. [En línea]. Available: https://www.conectronica.com/wireless/redes-wireless/tr-069-y-la-gestion-de-miles-de-millones-de-dispositivos. [Último acceso: 2 Sep 2018].

[3] W3C, «Simple Object Access Protocol (SOAP) 1.1,» 8 mayo 2000. [En línea]. Available: https://www.w3.org/TR/2000/NOTE-SOAP-20000508. [Último acceso: 30 mayo 2017].

[4] BroadBand Forum, TR-181 DEVICE DATA MODEL FOR TR-069, 2016.

[5] «QA Cafe,» [En línea]. [Último acceso: sep 2018].

[6] Puntonet, «Puntonet,» [En línea]. Available: www.puntonet.ec. [Último acceso: 14 10 2017].

[7] Mikrotik, «Mikrotik,» [En línea]. Available: www.mikrotik.com. [Último acceso: 14 10 2017].

[8] ACADEMY XPERTS, CONCEPTOS FUNDAMENTALES DE MIKROTIK ROUTEROS, 2016.

[9] Puntonet, *PRO T-O-CCM 001 PROCEDIMIENTO SOPORTE TELEFONICO,* Quito: Puntonet, 2015.

[10] Puntonet, *CONVENCION OPERACIONES MASIVAS,* Quito, 2017.

[11] «Fusion FreeACS,» Marzo 2013. [En línea]. Available: https://github.com/freeacs/readme/blob/master/Fusion%20Product%20Description.pdf. [Último acceso: marzo 2018].

[12] «Fusion FreeACS,» febrero 2013. [En línea]. Available: https://github.com/freeacs/ws/tree/master/docs. [Último acceso: marzo 2018].

[13] «Fusion FreeACS,» febrero 2013. [En línea]. Available: https://github.com/freeacs/core/tree/master/docs. [Último acceso: marzo 2018].

[14] «Fusion FreeACS,» febrero 2013. [En línea]. Available: https://github.com/freeacs/monitor/tree/master/docs. [Último acceso: marzo 2018].

[15] «Fusion FreeACS,» febrero 2013. [En línea]. Available: https://github.com/freeacs/spp/tree/master/docs. [Último acceso: marzo 2018].

[16] «Fusion FreeACS,» febrero 2013. [En línea]. Available: https://github.com/freeacs/web/tree/master/docs. [Último acceso: marzo 2018].

[17] FreeACS, «Questions/issues about FreeACS,» [En línea]. Available: https://www.tapatalk.com/groups/freeacs/index.php.

[18] Mikrotik, «Mikrotik Wiki,» [En línea]. Available: https://wiki.mikrotik.com/wiki/Manual:TR069-client. [Último acceso: 2018].

[19] PUNTONET, *Agenda Puntonet,* Quito, 2018.

[20] IBM, «Estándares de servicio Web,» [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SS4JE2_7.5.5/org.eclipse.jst.ws.doc.user/concepts/cwsstandards.html. [Último acceso: Junio 2018].

[21] GLOBALSIGN, [En línea]. Available: http://documentos.computerworld.es/whitepapers/content-download/c0b09fbd-2a6d-40cd-81d3-251ad22f997b/white-paper-autenticacion-basada-en-certificadso-para-control-de-accesos....pdf. [Último acceso: Sep `2018].

[22] A. S. Tanenbaum, Redes de Computadoras, Mexico: Pearson, 2003.

[23] «ABC Consultorio,» Feb 2015. [En línea]. Available: https://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html. [Último acceso: Sep 2018].

[24] Areá de programacion y Desarrollo, «Manual de XML,» [En línea]. Available: http://www.mundolinux.info/que-es-xml.htm. [Último acceso: Sep 2018].

[25] IBM, «Web Services Description Language,» Ago 2011. [En línea]. Available: https://www.ibm.com/developerworks/ssa/webservices/tutorials/ws-understand-web-services2/index.html. [Último acceso: Sep 2018].

6. ANEXOS

ANEXO I

VIDEO DE FUNCIONAMIENTO DEL SISTEMA DENTRO DE LA RED DE PUNTONET

ORDEN DE EMPASTADO