CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Repositorio Universidad de Zaragoza

UNIVERSIDAD DE ZARAGOZA

CENTRO POLITÉCNICO SUPERIOR

PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

DESARROLLO DE UN AGENTE SNMPv3 PARA LA GESTIÓN TÉCNICA DE DISPOSITIVOS EN ESCENARIOS DE ATENCIÓN DOMICILIARIA

Andreas Muñoz Zuara

Director:

Nelia Lasierra Beamonte

Ponente:

Álvaro Alesanco Iglesias

Dpto. de Ingeniería Electrónica y Comunicaciones

Mayo de 2011

Agradecimientos

Este proyecto es el final de una etapa de mi vida llena de esfuerzo y dedicación, pero que sin la ayuda de diversa gente que he ido conociendo a los largo de estos años no habría sido posible.

Quiero dar las gracias a **Álvaro** *por haberme ofrecido la oportunidad de realizar un proyecto interesante y por haber conseguido gracias a sus asignaturas que me decidiera por telemática. Todavía más debo agradecer a* **Nelia** *su esfuerzo y dedicación para que yo consiguiera terminar el proyecto a tiempo para esta convocatoria, y su apoyo cuando más estresado me encontraba. Seguro que echa de menos el llegar a casa y encontrar una docena de correos de mi parte preguntándole dudas desesperado.*

A mis compañeros de laboratorio a lo largo de estos arduos meses. A **Gonzalo**, *que siempre nos devolvía al trabajo cuando estábamos vagos; a* **Carol**, *por contagiarnos su alegría; a* **Luis** *y* **Yasmina**, *por todos los buenos ratos que hemos pasado "frikeando" por internet.*

Tampoco debo olvidar a mis compañeros de clase. A **Rubén**, *por todas esas risas que nos hemos echado en clase y por ser siempre los primeros en apuntarnos a prácticas. A* **Javi**, *un compañero de prácticas inmejorable. A* **Natalia**, *mi chófer particular durante la carrera. A* **Verónica**, *porque cuando hablo con ella siempre me levanta el ánimo. A* **Ana** *y* **Loreto**, *por todo el apoyo moral que me han prestado durante el proyecto. A* **Aguilar**, **Alba**, **Héctor**, **Manu**, **Miki**, **José**, **Miguel**, **Cros**, **Diego** *y tantos otros, gracias por todos los buenos momentos que hemos pasado juntos.*

A mis amigos de toda la vida, **Diego**, **Guillermo**, **Víctor**, **Alberto**, **Javier**, **Jorge** *y* **Luis** *por aguantarme todos estos años, y en especial a* **Enrique** *por nuestros duelos entre físicos e ingenieros. A* **Álvaro** *y* **David** *porque aunque nos vemos poco siempre estais ahí cuando hace falta.*

A **Álex**, *por haber estado siempre a mi lado en los momentos buenos y no tan buenos, y con el a toda la banda que conocí gracias a él,* **Sara**, **Sonia**, **Juan**, **Boldoba** *y el resto, sin olvidar a* **Vallejo** *que si no le nombro me mata.*

A mi familia y a todo el que alguna vez me ha ayudado a llegar donde he llegado, que seguro que me dejo unos cuantos, gracias.

DESARROLLO DE UN AGENTE SNMPv3 PARA LA GESTIÓN TÉCNICA DE DISPOSITIVOS EN ESCENARIOS DE ATENCIÓN DOMICILIARIA

RESUMEN

En este proyecto se ha desarrollado un agente SNMPv3 (*Simple Network Management Protocol*) para la gestión técnica de dispositivos involucrados en un escenario de atención domiciliaria. Este agente va a permitir recoger la información técnica de los dispositivos médicos (báscula, termómetro...) asociados al *Compute Engine* (CE), el cual es el dispositivo concentrador de datos utilizado en la casa del paciente para reunir información de los dispositivos y enviarla posteriormente al centro sanitario correspondiente. Además, el agente permitirá la gestión técnica del CE, permitiendo así conocer información de recursos propios de dicho dispositivo y la detección de posibles problemas que afecten al correcto funcionamiento de éste. El diseño del agente se ha adaptado a la utilización de dispositivos médicos que implementen el estándar X.73 para las comunicaciones con el CE. Para desarrollar las herramientas que componen el sistema de gestión se ha utilizado software Java y bases de datos MySQL.

El agente no sólo ofrece la ventaja de una centralización de la información proveniente de los dispositivos médicos, sino que permite una escalabilidad total. Para llevar a cabo estas funciones, se han diseñado los bloques que componen el sistema completo: establecimiento y control de comunicaciones SNMP entre agente y gestores, diseño de la MIB de gestión de la información técnica, control de las comunicaciones con el manager X.73 y diseño del interfaz del agente y del interfaz de simulación del manager X.73. Además, el agente desarrollado recogerá información de los recursos propios del CE mediante encuestas SNMP transparentes al gestor. De esta forma, el agente propuesto permite la gestión de la información técnica de los dispositivos, dejando a elección del gestor la configuración de alarmas y eventos.

Por último, se han realizado pruebas mediante simulación del sistema de gestión propuesto para comprobar el alcance de la gestión del sistema y asegurar que todas las capacidades que el sistema se supone debe cumplir para realizar una gestión completa realmente se llevan a cabo de forma satisfactoria. Mediante la consecución de estas pruebas, se presenta un resumen de los resultados obtenidos que demuestra la eficacia del sistema de gestión de dispositivos médicos propuesto.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y Objetivos

1.1 Introducción

Las tecnologías de comunicaciones han avanzado en gran medida la última década. Gracias a la expansión de Internet, existen multitud de tareas de gestión de la vida cotidiana que pueden realizarse sin salir de casa, desde comprar un libro hasta mirar el parte meteorológico. Todas esas acciones que nos parecen tan sencillas requieren una compleja gestión de sistemas. Esa es la razón por la que la gestión de redes ha sido una de las áreas de mayor investigación en los últimos años.

Dentro de este entorno de gestión domiciliaria, uno de los ámbitos de mayor interés actualmente es el de la telemonitorización de pacientes debido a los numerosos beneficios que reporta. Desde el punto de vista del paciente, se reducen el número de visitas que debería realizar a los centros médicos, una ventaja destacable en casos de personas con movilidad reducida; aumenta la calidad de vida del paciente, permitiéndole tener controlado su estado de salud regularmente sin salir de casa. Desde el punto de vista del centro sanitario, se reducen las listas de espera en centros de atención primaria y se aumenta el tiempo que puede dedicársele a cada paciente, además de suponer un descenso en los gastos sanitarios que supone la atención al paciente.

Como se puede ver en la figura 1.1, de manera general un sistema de telemonitorización domiciliaria está compuesto por lo tanto, por dos entidades: el hogar del paciente y el centro sanitario. En el domicilio del paciente se encuentran los dispositivos médicos (báscula, medidor de presión arterial, glucómetro...), los cuales se conectan a un dispositivo concentrador de datos el cual se ha denominado como

CE (*Compute Engine*) en este proyecto. Este dispositivo es el encargado de transmitir la información posteriormente al centro sanitario. En el centro médico se ubica el MS (*Monitoring Server*), el cual es el dispositivo al que se conectan todos los CEs para enviar los datos recogidos.

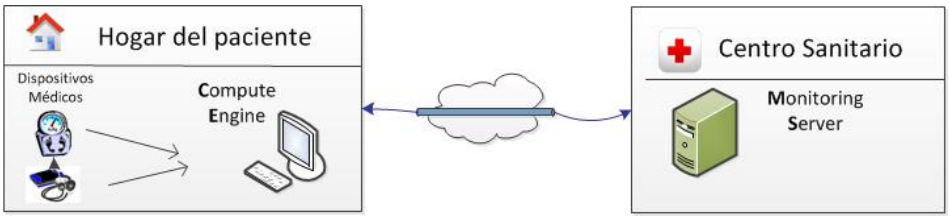

Figura 1.1. Aquitectura general de un sistema de telemonitorización.

En este escenario de telemonitorización se trata con dos tipos de datos (médicos y técnicos), por lo que nos encontramos con dos tipos de gestión a realizar en este entorno: gestión clínica y técnica. El objetivo de la gestión clínica es realizar la transferencia de datos clínicos (medidas realizadas por el paciente) de forma correcta al centro sanitario así como su evaluación para la detección de posibles cambios en la evolución del paciente. El objetivo de la gestión técnica es recoger y evaluar toda la información referente al estado físico de los dispositivos, como puede ser el estado de las baterías o el ancho de banda que están generando. Este proyecto está orientado a realizar la gestión técnica de los dispositivos, la cual es tan importante como la médica. Es necesario verificar el correcto funcionamiento de los dispositivos médicos para garantizar que el comportamiento de los mismos no afecta a la calidad de las medidas realizadas ni por lo tanto al seguimiento del paciente. Por lo tanto la problemática que se va a solucionar es la relacionada con el tratamiento, almacenamiento e interpretación de los datos técnicos de los dispositivos presentes en el domicilio del paciente. En caso de que surgiera algún problema en el funcionamiento de un dispositivo, se avisaría al gestor encargado de supervisar los dispositivos para que se corrigiera el problema con la mayor celeridad posible. De esta forma permite saber que los datos médicos que se obtengan mientras hay un problema técnico pueden no ser correctos y no lleven a un diagnóstico inadecuado.

Todos los dispositivos médicos, así como el CE y el MS se pueden ver (ya que de hecho son) como elementos de una red de comunicaciones. Es por ello que sería de gran interés poder desarrollar para su gestión una arquitectura basada en estándares de gran relevancia para las redes de comunicaciones basadas en la pila de protocolos TCP/IP.

Utilizar técnicas de gestión de redes nos va a permitir realizar una intensa recogida de datos de forma eficiente ademas de proporcionar seguridad en la tranmisión de la información. En este entorno de trabajo la seguridad es un factor primordial a la hora de elegir la técnica de gestión adecuada, ya que aunque la información con que trabajamos no exige privacidad, la manipulación de la misma puede afectar a la gestión médica. Por ello, para realizar esta gestión, en este proyecto fin de carrera se decidió utilizar el protocolo SNMP (1) (*Simple Network management Protocol*), el cual es el actual estándar de facto para la gestión de redes. En este proyecto fin de carrera se ha propuesto el diseño y desarrollo de un agente (aplicación capaz de responder a peticiones realizadas por un gestor de la red) SNMP versión 3 (2) para permitir la gestión técnica de los dispositivos en un escenario de telemonitorización domiciliaria. Este agente va a permitir integrar los datos técnicos de los dispositivos, asi como detectar anomalías en el funcionamiento de los dispositivos y controlar que todo funciona de forma correcta, posibilitando a su vez la transmisión de mensajes a fin de avisar al gestor en caso de datos erróneos o fuera del rango óptimo.

Como aplicación, se ha propuesto integrar en el agente la gestión técnica de dispositivos médicos que utilicen el estándar ISO/IEEEE 11073 (X.73) (3) para la transferencia de datos al CE. X.73 es el estándar europeo para la transmisión de dispositivos médicos, el cual se ha propuesto como solución para la integración e interoperabilidad en la transmisión de datos entre los MDs (*Medical Devices*) y el CE. De esta forma, el agente propuesto actuará también como proxy con un manager X.73 encargado de recoger los datos enviados por los dispositivos médicos. La información, tanto técnica como médica, se transmite de los MDs al CE. Como se puede ver en la figura 1.2, la información X.73 (enviada por los agentes X.73 instalados en los MDs) es procesada por el manager X.73 presente en el CE.

1.2 Estado del arte

Actualmente no existen muchas propuestas que traten con el problema de la gestión de dispositivos pertenecientes a entornos de telemonitorización de atención domiciliaria. Para poder realizar la gestión de estos dispositivos se requiere de un protocolo que permita la gestión del sistema en toda su extensión y proporcionar seguridad. El sistema de gestión elegido debe de poder facilitar la interoperabilidad con distintos módulos

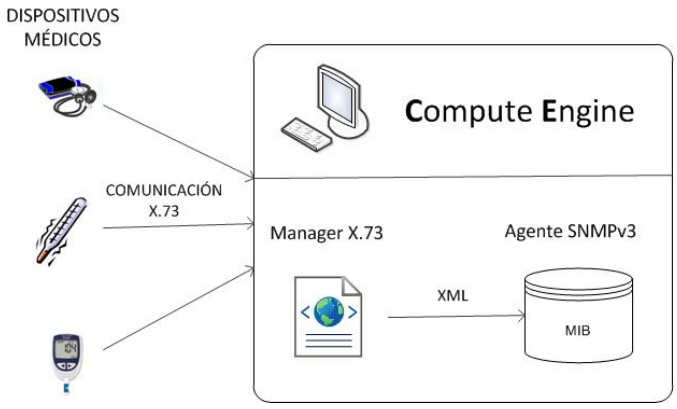

Figura 1.2. Funcionamiento del manager X.73.

con los que deba comunicarse y utilicen distinos lenguajes para ello. A pesar de que la arquitectura elegida finalmente es SNMP, durante la búsqueda y documentación inicial sobre arquitecturas y tecnologías de gestión de redes se encontraron alternativas interesantes.

CORBA (4) (*Common Object Request Broker Arquitecture*) es una arquitectura basada en un mecanismo intermediario llamado *broker*. La idea es que esta arquitectura nos permite comunicar aplicaciones definidas en diferentes lenguajes y ejecutados en distintas plataformas de manera trasparente sin tener que realizar ningún proceso intermedio. CORBA se encarga de la comunicación entre sistemas. El funcionamiento de CORBA es el siguiente: tenemos un cliente y un servidor conectados mediante CORBA; el cliente desea realizar una petición y se la envía al *broker*, éste lo traduce y lo envía al servidor, el cual cursa la petición y envía la respuesta al *broker*, y finalmente se reenvía al cliente. Esta arquitectura es capaz de soportar más de veinte lenguajes de programación diferentes, lo que evitaría problemas de incompatibilidad entre sistemas.

Para el sistema que se plantea CORBA podría ser muy interesante, ya que mediante su uso nos olvidarnos de traducir porque la arquitectura se encarga de todo. Además este protocolo tiene diferentes versiones diseñadas para equipos con pocos recursos como los sensores, para gastar menos energía. Por otra parte, SNMP nos permite una gestión más amplia y está muy extendido entre la mayoría de fabricantes. Además, este sistema no es tan heterogéneo como para apreciar las ventajas que una aplicación de este tipo nos ofrece.

MANNA (5) (*Management Architecture for Wireless Sensor Networks*) es una tecnología de gestión creada para gestionar redes de sensores wireless. Esta arquitectura

propone un sistema integrado de gestión de la seguridad desde el que podemos administrar cada nodo sensor a través de una página web. MANNA divide la gestión de los sensores en 3 dimensiones: áreas funcionales, niveles de gestión y funcionalidades de WSN (*Wireless Sensor Networks*).

Dentro de los módulos funcionales, MANNA dispone de una interfaz de usuario web desde la que se puede monitorizar la red y analizar la información del sistema; el módulo de gestión de la información de los sensores puede mostrar los datos en gráficas y permite al administrador cambiar el periodo de las mediciones, el módulo de gestión del nodo sensor permite gestionar el sensor o un grupo de sensores y pedir información a los mismos; el módulo de información del nodo sensor permite la autenticación y registro de usuarios autorizados; finalmente, el módulo de seguridad de la red proporciona información del estado de la seguridad de la red.

Realmente la idea de MANNA es parecida a la que queremos desarrollar nosotros, un sistema de gestión que ofrezca seguridad y permita controlar los parámetros de los sensores. Sin embargo SNMP tiene la ventaja de que es un sistema muy extendido y que muchos de los dispositivos existentes tipo routers o gateways lo tienen implementado, así que aunque MANNA no ofrece ventajas sustanciales sobre SNMP, éste último si las tiene frente a MANNA.

NETCONF (6) es un protocolo de gestión muy reciente basado en el protocolo XML. Es el nuevo protocolo del IETF (*Internet Engineering Task Force*) para la gestión de varios dispositivos simultáneamente, incluso de diferentes fabricantes. Este protocolo permite mantener 3 configuraciones diferentes de gestión del equipo: *running*, la configuración actual del dispositivo; *candidate*, configuración en standby que puede modificarse con la de running funcionando; y la de *startup*, que es la configuración inicial del dispositivo. Utiliza una estrucutura en capas para separar de forma adecuada la gestión de datos del protocolo de transporte que va por debajo. Un protocolo basado en XML tiene las ventajas de ser muy flexible definiendo estructuras de datos, existen muchas APIs (*Application Programming Interface*) gratuitas, es fácil de leer y de transmitir por canales seguros.

SNMP es una arquitectura que presenta muchas ventajas en el ámbito de la gestión de redes, por eso es el estándar de facto, sin embargo no carece de limitaciones. NETCONF ha sido diseñado expresamente para suplir las carencias de SNMP respecto a la gestión de dispositivos, presentando una alternativa de futuro a SNMP para sistemas donde los

gestores necesiten escribir grandes cantidades de información en las bases de datos y para la gestión de muchos equipos simultáneamente.

Tras evaluar las distintas propuestas, hemos comprobado que SNMP tiene ventajas tanto sobre CORBA como sobre MANNA, sin embargo NETCONF es más potente que SNMP. El factor determinante de la elección realizada ha sido la amplia distribución del protocolo a nivel mundial. NETCONF es un protocolo de reciente aparición, mientras que SNMP se implementa en la mayoría de routers, gateways y otros dispositivos de diferentes compañías, además de que como no queremos escribir grandes cantidades de información en la base de datos, sus limitaciones no nos afectan sustancialmente.

Existen otros trabajos que utilizan también SNMP como solución de gestión en entornos sanitarios similares, como se puede apreciar en (7), (8) y (9). Sin embargo, la aplicación que se propone en este proyecto a la utilización del estándar X.73 resulta novedosa, además de presentarse, a diferencia de los trabajos anteriores, una arquitectura que permite una gestión global en un escenario de telemonitorizacion domiciliaria.

1.3 Propuesta

En este proyecto se propone el desarrollo de un agente SNMPv3 para la gestión técnica de dispositivos médicos en entornos de atención domiciliaria. En concreto, se va integrar la gestión técnica de los MDs que utilizan el estándar X.73 para la transmisión de datos así como el CE dentro de la arquitectura SNMP. De manera que la arquitectura propuesta de gestión integra los dos estándares. Esta arquitectura nos permite almacenar la información obtenida de los dispositivos médicos de forma estructurada en unas bases de datos propias de esta arquitectura, conocidas como MIB (*Management Information Base*). El agente propuesto, además de funcionar como agente SNMP para establecer comunicaciones con gestores externos, funcionará como proxy X.73. De esta forma, el agente establecerá comunicaciones con un manager capaz de comunicarse con los dispositivos médicos mediante dicho protocolo, aplicándose así el diseño del agente a la utilización de este protocolo por parte de los dispositivos médicos.

Por lo tanto, la información con la que se evaluará el funcionamiento de los dispositivos se almacenara en una MIB privada, separada en tablas dependiendo del tipo de información que se trate. La MIB va a permitir conocer información técnica de los MDs, así como gestionar recursos propios del CE. La gestión del CE se va a llevar a

cabo mediante la realización de encuestas periódicas a un agente SNMP presente en el mismo, y finalmente va a permitir la definición de eventos y alarmas ante la detección de problemas técnicos tanto en los MDs como en el CE. Como se observa en la figura 1.3, en la que se representa la estructura general del sistema, cualquier gestor que se autentique de forma correcta en nuestro agente podrá acceder a la información que ésta almacena, pudiendo tanto examinar el comportamiento de los dispositivos a lo largo del tiempo como configurar el agente para que le avise cuando haya algún problema con alguno de los dispositivos conectados.

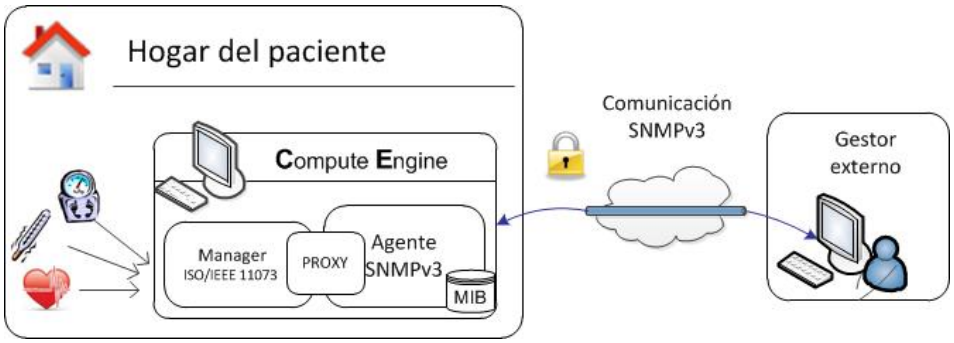

Figura 1.3. Esquema general de la arquitectura propuesta.

Como previamente hemos resaltado, queremos y necesitamos seguridad en nuestras comunicaciones, así que se ha decidido utilizar la versión 3 del protocolo. Esta última versión del protocolo nos aporta varios mecanismos de seguridad: USM (10) (*User-based Security Model*) nos proporciona protección frente a ataques de usuarios que traten de autenticarse en nuestro sistema sin autorización, el VACM (11) (*View-based Acces Control Model*) define la política de control de acceso para el agente y determina a qué partes del la MIB puede acceder cada gestor, y lo más importante es que proporciona cifrado y privacidad a las comunicaciones.

El uso de SNMP en el diseño del agente nos proporciona las siguientes ventajas:

- El agente nos permite gestionar un número indeterminado de dispositivos y mantener un número indeterminado de gestores.
- SNMP nos permite la definición de traps, mensajes asíncronos enviados por el agente al gestor para la notificación de cambios importantes en la monitorización de un recurso. Su definición en este entorno puede ser de gran utilidad para la definición y gestión de alarmas relacionadas con el funcionamiento de los dispositivos médicos.

- La versión 3 del protocolo nos proporciona confidencialidad e integridad en las comunicaciones mediante los métodos previamente comentados.

- Gracias a la integración del CE en la arquitectura SNMP, y utilizando el protocolo X.73 para la transmisión de información, conseguimos una plataforma de gestión integrada e interoperable que concentra en un sólo módulo de gestión toda la información técnica procedentes de las diferentes fuentes (CE y MDs) que componen el escenario de telemonitorización.

1.4 Objetivos

El objetivo principal de este proyecto es el diseño de un agente SNMP para la gestión de dispositivos médicos en entornos de atención domiciliaria. Para llevar a cabo la realización del proyecto se han completado los siguientes objetivos:

1. Diseño tecnológico del agente

    - Documentación sobre el funcionamiento del protocolo SNMP e implementación de bases de datos MIB, así como la revisión de las especificaciones de dispositivos del estándar X.73 y revisión bibliográfica sobre técnicas de gestión de dispositivos médicos y técnicas comunes de gestión de redes.
    - Diseño de las comunicaciones entre el agente y el gestor para una conexión segura.
    - Diseño y desarrollo de la MIB privada en la que se almacenen los datos obtenidos de los dispositivos, así como las alarmas y eventos que los gestores quieran utilizar.
    - Implementación de las comunicaciones con el manager X.73 mediante intercambio de ficheros XML.
    - Diseño del interfaz gráfico del agente.

2. Aplicación del agente en un escenario simulado: implementación y pruebas

    - Diseño de un interfaz gráfico para simular un entorno domiciliario en el que varios dispositivos médicos envían información médica al manager X.73

y éste establece las comunicaciones con el agente diseñado para el cual está encargado de su control técnico.

- Definición de los escenarios de prueba necesarios para comprobar el correcto funcionamiento del agente.
- Presentación y análisis de los resultados obtenidos.

1.5 Materiales y herramientas utilizadas

El desarrollo de la aplicación informática se llevó a cabo sobre el lenguaje de programación Java y el entorno de desarrollo SDK Eclipse. Para el almacenamiento estático de los datos se hizo uso de una base de datos MySQL. La conexión entre Java y la base de datos se realizó mediante el conector de bases de datos JDBC (*Java DataBase Conectivity*). Para la implementación de las comunicaciones SNMP entre el agente y el gestor se ha utilizado un framework de Java llamado SNMP4j (12). Además se utilizó el programa MIB Browser de MG-Soft (13) como gestor para comprobar y demostrar el correcto funcionamiento del agente.

Tanto el lenguaje MySQL como la herramienta de programación son multiplataforma, lo que nos proporciona las ventajas de portabilidad e independencia del sistema operativo que se utilice.

1.6 Organización de la memoria

En el capítulo 1 se ha desarrollado una breve introducción al PFC y los objetivos principales que se han perseguido.

En el capítulo 2 se describen los bloques que componen la arquitectura, detallando además las características más importantes de la arquitectura SNMP así como de la información técnica X.73.

En el capítulo 3 se describe la organización de la MIB así como se detallan los campos de las tablas y grupos que la componen.

El capítulo 4 recoge las pruebas realizadas y resultados obtenidos de la evaluación del funcionamiento del agente, además de la evaluación y el análisis final de la eficacia demostrada por el mismo ante las simulaciones a las que se le sometió.

Finalmente, en el capítulo 5 se exponen las conclusiones obtenidas sobre el resultado del proyecto, y las posibles líneas futuras de investigación y mejora que se podrían seguir.

Capítulo 2

Arquitectura del Agente

2.1 Descripción del agente

El agente se encarga de almacenar información que proviene de los dispositivos médicos facilitando de esta forma su gestión técnica a un gestor externo. Como el programa va a ser intermediario de todas las peticiones y acciones que se van a realizar en el sistema global, podemos situarlo en un único dispositivo, creando un sistema centralizado que nos ahorra recursos y facilita la instalación y mantenimiento del mismo. Por lo tanto, el software necesario se verá reducido a la instalación del agente en una máquina accesible por toda la red y al mantenimiento de gestores SNMP en aquellos sistemas operativos desde los que queramos tener acceso a nuestro agente.

La arquitectura general del agente SNMPv3 está compuesto por 3 bloques.

1. Bloque de comunicaciones SNMP entre agente y gestor.

   El agente y el gestor se comunican mediante el protocolo SNMPv3. El agente se encarga de atender las peticiones que le llegan del gestor, del mismo modo que es capaz de enviar mensajes asíncronos al gestor cuando alguna de las alarmas definidas por el mismo se activa.

2. Bloque de comunicaciones con el manager X.73.

   Las comunicaciones entre agente y manager se realizan mediante documentos XML. El agente se encarga de traducir la información que le llega del manager X.73 en valores que él pueda almacenar en la base de datos, además puede enviar XML al manager para pedirle actualizaciones de información que los dispositivos

le hayan comunicado.

3. Bloque de la MIB de gestión de dispositivos médicos.
   La MIB va a recoger toda la información de monitorización, tanto la proporcionada por los MDs como la proporcionada por el CE además de permitir la definición y configuración de alarmas. El agente se encarga tanto de crear las tablas, como de escribir, modificar y borrar datos en ellas, siguiendo una estructura interna que explicaremos más adelante.

En la figura 2.1 se puede apreciar el resultado final a través del esquema global del sistema. El funcionamiento general del sistema es el siguiente. Los dispositivos médicos se comunican con el manager X.73 instalado en el dispositivo concentrador. Éste transmite la información recibida al agente mediante el uso de un protocolo basado en el intercambio de documentos XML. Este protocolo funciona de manera que cuando el dispositivo médico realiza un cambio de estado o envía información al manager, el manager se lo comunica al agente mediante el XML generado. El agente se encarga de traducir y almacenar la información en la MIB, a la vez que se ocupa de forma estática de obtener datos del tráfico en la red y uso del procesador.

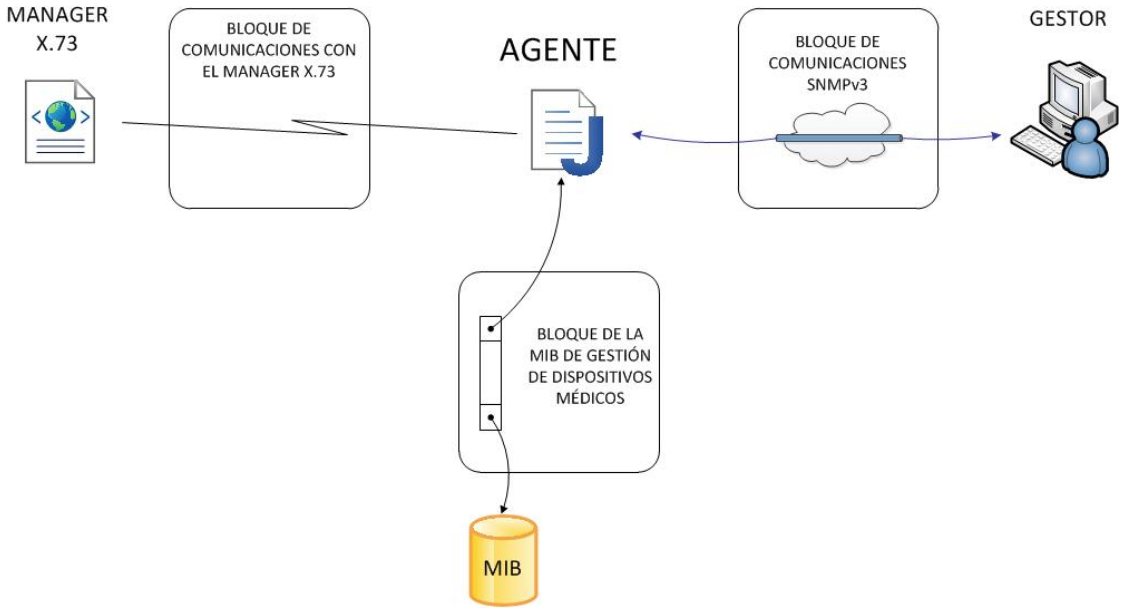

Figura 2.1. Esquema de bloques del proyecto.

La obtención de datos del CE se realiza mediante encuestas SNMP. El CE, además de nuestra MIB, tiene disponibles otras MIBs para realizar consultas, como es la

MIB2, donde se almacenan parámetros de tráfico y del sistema. Nuestro agente manda peticiones de consulta a la MIB2, y a través de unos datos concretos de esa MIB podemos calcular el ancho de banda o el uso de CPU del CE.

Cuando el gestor se conecta al agente puede consultar toda la información almacenada hasta la fecha, además de poder actualizar el estado de los dispositivos a través del agente, que se encarga de proporcionarle todos los servicios que precise. De esta forma el proceso que se lleva a cabo es totalmente transparente de cara al usuario, el cual se encarga solamente de conectar los dispositivos a la máquina que contiene el agente, y también transparente de cara al gestor, que exclusivamente tiene que lanzar peticiones y configurar alarmas y el agente se encarga de procurarle lo que necesita. En la figura 2.2 se muestra más detalladamente las comunicaciones entre los distintos bloques que componen el agente.

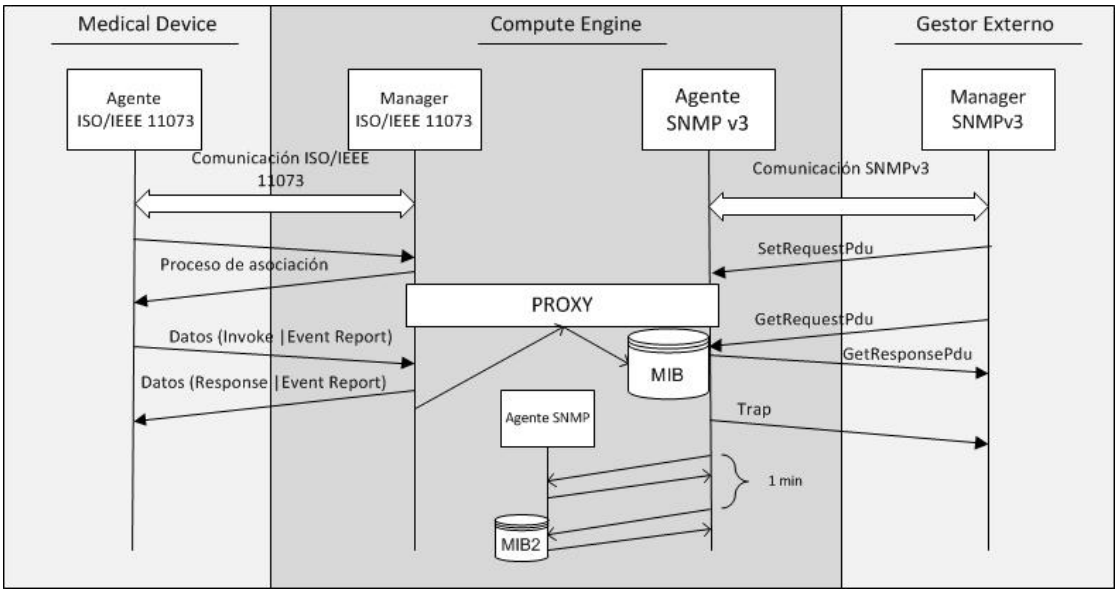

Figura 2.2. Bloques de comunicaciones del agente.

En definitiva, el agente propuesto nos ofrece una solución centralizada y transparente al usuario, con posibilidades de integración en una red que ya posea una gestión de redes con SNMP.

2.2 Descripción de la arquitectura SNMP

SNMP (1) (2) es la arquitectura de gestión de redes más utilizada en redes TCP/IP debido a su simplicidad y su escaso gasto de recursos. SNMP define un protocolo para el intercambio de información de gestión además de definir un formato para la representación de esa información y un marco para organizar sistemas distribuidos en gestores y agentes.

La primera versión del protocolo tenía tanto fallos de seguridad como funcionales, tales como la imposibilidad de pedir grandes cantidades de información en un mismo paquete. En la segunda versión se solucionan los problemas funcionales, pero la seguridad que implementaba no fue aceptada por fallos en su definición, así que se llegó a una tercera versión en la que se implementa un sistema de seguridad efectivo, manteniendo el funcionamiento de las versiones anteriores. SNMPv3 describe toda una arquitectura con estructuras de mensajes y características de seguridad, pero el *payload* (parte del mensaje transmitido por la red que contiene los datos y la información que identifica a la fuente y al destino) de SNMPv1 y SNMPv2 se mantiene intacto. Como resumen podríamos decir que SNMPv3 es SNMPv2 añadiendo seguridad y administración.

2.2.1 Conceptos básicos de SNMP

Las comunicaciones mediante SNMP involucran invariablemente a dos participantes, un agente y un gestor.

- Gestor: es el encargado de pedir información y modificarla según las necesidades de funcionamiento de la máquina en la que reside el agente. El gestor o gestores de la red poseen un interfaz gráfico para poder emitir comandos y examinar en forma de tablas los datos que les llegan de los agentes. Estos sistemas incluyen como mínimo aplicaciones para monitorización, control de configuración y realización de informes.

- Agente: programa situado en el equipo que se va a monitorizar, sus misiones son recolectar y guardar información local, así como responder ante peticiones del gestor y enviar información de forma asíncrona cuando sucede algún evento.

Todas las aplicaciones de la red comparten normalmente un protocolo de gestión común para facilitar las comunicaciones, y cada agente tiene una MIB donde almacena

información como por ejemplo de configuración local o parámetros de tráfico. La verdadera potencia del protocolo está en que las MIBs están estandarizadas y hay un gran número de ellas definidas, así que implementando las que nosotros queramos en cada agente recogeremos la información que necesitemos.

El protocolo SNMP define una serie de mensajes para el intercambio de información. En concreto, para la versión 3, que es la que se ha desarrollado en el proyecto, existen los siguientes tipos de mensajes de implementación obligatoria en un agente o gestor según corresponda:

- Mensajes enviados por el gestor al agente:
  - GetRequest: petición de una o varias variables incluidas en el mensaje de respuesta Response.
  - GetNextRequest: petición de la inmediatamente siguiente variable a la variable que mandamos y con valor no nulo, cuya respuesta se incluye también en un mensaje Response.
  - GetBulkRequest: tipo de petición presente solo a partir de la versión 2. Nos permite recibir una lista de variables consecutivas para cada variable que le mandamos nosotros. El número de variables consecutivas viene determinado por el valor *max-repetitions*. Además podemos pedir variables sin repetir, cuyo número viene determinado por el valor *non-repeaters*. De este modo podemos pedir más información en una cantidad menor de mensajes, con lo que aumentamos la efectividad.
  - SetRequest: mensaje que modifica una variable de la base de datos. Si ha habido algún error a la hora de modificar, el agente lo comunicará en el mensaje de respuesta mediante el código de error correspondiente.
- Mensajes enviados por el agente al gestor:
  - Response: mensaje de respuesta para los mensajes que nos envía el gestor. En él se incluyen los valores de las variables requeridas, y el código del error en caso de que ocurra alguno.
  - Trap: mensaje generado y transmitido de forma asíncrona como respuesta a un evento excepcional. En él se incluye el *sysUpTime*, que es el tiempo que

lleva encendido el dispositivo, además de las variables que correspondan al tipo de trap generado.

- Mensajes enviados de gestor a gestor:

  - InformationRequest: mensaje para enviar una alerta de un gestor a otro. Como nuestra aplicación no requiere comunicación entre gestores, no se incluirá en el proyecto.

2.2.2 Bases de datos en SNMP: MIB

Además de las comunicaciones gestor-agente y de los mensajes estandarizados, la arquitectura SNMP también cuenta con unas bases de datos también estandarizadas conocidas como MIB. Estas bases están organizadasn jerárquicamente en forma de árbol, y definidas a través del lenguaje SMI (14) (*Structure of management Information*). La filosofía de este lenguaje de programación de bases de datos es favorecer la simplicidad y extensibilidad de las MIB, permitiendo sólo estructuras simples de datos. Cada objeto dentro de la MIB está formado por una variable, que es el valor del objeto, y un identificador conocido como OID (*Object Identifier*). El OID está formado por una secuencia de números y puntos, representando cada uno un salto de nivel dentro del árbol que compone la MIB, como podemos observar en la figura 2.3.

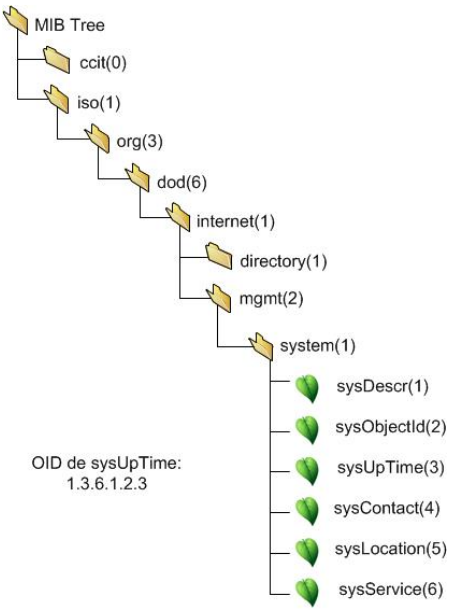

Figura 2.3. Ejemplo de creación del OID de un objeto.

Las MIBs se dividen en dos tipos, públicas y privadas. Las públicas están definidas mediante estándares y proporcionan información general del sistema. Las privadas están definidas por los fabricantes y ofrecen información mas detallada y concreta. La MIB más conocida es la MIB II, dado que tiene una amplia información tanto de la red, por ejemplo parámetros estadísticos de tráfico, como de dispositivos, por ejemplo es uso de CPU o de memoria, aunque no contiene información de más alto nivel, referente a aplicaciones o al sistema operativo.

2.2.3 Características específicas de SNMPv3

Como hemos introducido previamente, la versión 3 (2) se diferencia principalmente de las otras en que nos permite confidencialidad y privacidad en las comunicaciones a través de la autenticación del usuario y del cifrado de la comunicación.

Para la definición y aplicación de políticas de seguridad se utilizan dos bloques dentro de la propia entidad SNMP, llamados USM (*User-based Security Model*) y VACM (*View-based Access Control Model*). USM se encarga de la gestión de usuarios e información necesaria para el cifrado y seguridad de las comunicaciones. Para ello utiliza funciones resumen del cuerpo del mensaje (MD5 o SHA-1) para evitar la modificación del mensaje por parte de un tercero o ataques de suplantación de identidad. Para asegurar la confidencialidad SNMPv3 se apoya en algoritmos de cifrado (DES-CBC o AES de 128 bits). VACM gestiona los privilegios que cada usuario USM posee y las tablas y objetos a los que puede acceder. Para ello asigna a cada usuario una pareja de valores ≪securityModel, securityName≫ con los que determina el nivel de seguridad aplicado a cada usuario. En la figura 2.4 podemos observar cómo se lleva a cabo este proceso de autenticación y cifrado.

El agente y el gestor necesitan sicronizarse antes de comenzar la comunicación. Para realizar la sincronización, el gestor copia los valores de *snmpEngineTime* (indica el tiempo que ha transcurrido desde la última vez que se reinició por última vez), *snmpEngineBoots* (parámetro que indica cuántas veces se ha reiniciado ese agente) y *latestReceivedEngineTime* del agente. Para mantener la sincronización, el agente manda sus parámetros temporales junto con su *snmpEngineID* en cada mensaje, que es un identificador único de cada entidad SNMP, y si esos valores son correctos el gestor actualiza los valores.

Se establece una ventana en la cual los mensajes recibidos no se consideran con retardo y son válidos. Hay que tener cuidado de no poner una ventana demasiado pequeña o grande. En caso de que el *timeout* expire, el mensaje se considera inválido, se devuelve un error de notInTimeWindow y se espera la retransmisión del mensaje.

2.3 Funcionalidades del agente como proxy

El agente diseñado además de su funcionalidad básica como agente se comporta como un proxy X.73. El resultado es que además de la funcionalidad básica, sea capaz de comportarse como un proxy con el manager X.73, leyendo la informacion X.73 de los mensajes que éste le envía e insertando la información en la MIB. Esta comunicación entre dispositivos médicos y manager se lleva a cabo mediante el estándar X.73, mientras que el manager y el agente se comunican mediante documentos XML, protocolos que explicaremos a continuación.

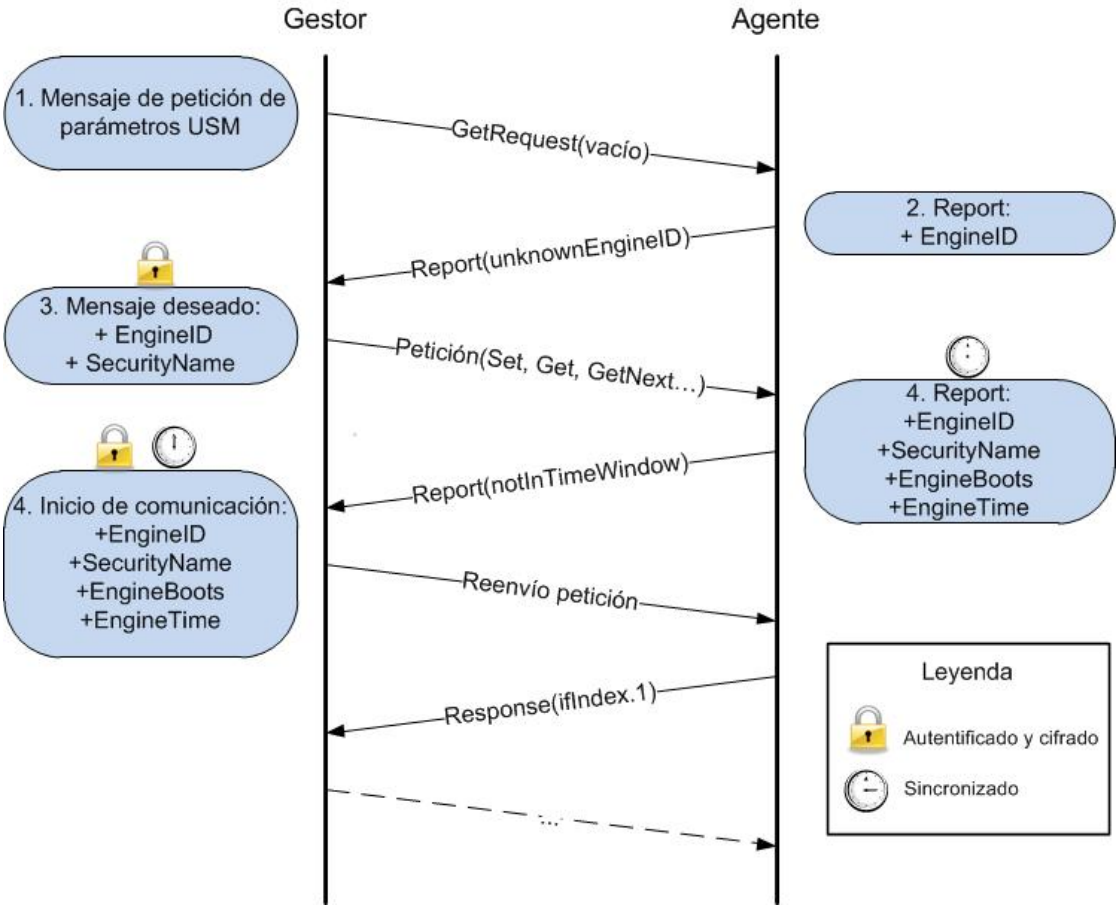

Figura 2.4. Sincronización y conexión entre agente y gestor SNMP.

2.3.1 Estándar ISO/IEEE 11073

El estándar ISO/IEEE 11073 (3) (15) (16), también conocido como estándar X.73, fue definido para permitir la comunicación entre dispositivos médicos y de asistencia sanitaria con sistemas informáticos externos. Se encarga de proveer de forma automática información detallada sobre el usuario y sus signos vitales, así como información del funcionamiento del dispositivo. Los principales objetivos de este protocolo son obtener información médica en tiempo real, facilidad a la hora de la instalación (*"Plug-and-play"*) y que pueda ser procesado por muchos tipos de aplicaciones.

El núcleo del estándar es el llamado DIM (*Domain Information Model*). El DIM describe una serie de objetos con atributos que caracterizan la información que puede ser mandada desde los MDs al CE (por ejemplo medidas de datos o información genérica del dispositivo). Dentro de esta estructura, la clase MDS representa el identificador y los atributos del agente X.73, los que contienen la mayor parte de la información técnica que los dispositivos médicos pueden proporcionar. En la tabla 2.1 podemos ver algunos ejemplos de información técnica general que envía X.73. Además, algunas especificaciones definen información técnica adicional, presente en el modo de configuración extendida de los dispositivos, relacionada con las singularidades o el modo específico de trabajo del MD. Por ejemplo, las especificaciones del pulsioxímetro definen información relacionada con el estado del dispositivo o el sensor tales como detección deficiente del sensor o irregularidades en la señal. El diseño de las MIB está realizado para cubrir la información proporcionada por los dispositivos médicos incluida en el estándar. Por lo tanto esta MIB contiene toda la información descrita en la tabla 2.1.

2.3.2 Protocolo de comunicaciones agente SNMP-Manager X.73

Para conocer los datos pertenecientes a los dispositivos, el agente tiene que establecer una comunicación con el manager del estándar X.73 que los recibe en el CE. Dicha comunicación se lleva cabo mediante el intercambio de documentos XML donde se incluye la información deseada en el formato que define el propio lenguaje XML.

Se trata de una comunicación bidireccional en la que el manager X.73 envía información de forma asíncrona o puede responder a peticiones realizadas por el agente. Esta última opción está pensada para una versión futura en la que el manager tenga

Clase: atributo-subatributo	Valor	Descripción
MDS: System-Type-Spec-List	MDC_DEV_SPEC_-PROFILE_SCALE_xxx	Tipo de dispositivo (glucómetro, báscula...)
MDS: System-Id	OctetString	Identificador único del MD.
MDS: System-Model::model	OctetString	Model del MD.
MDS: System-Model.manufacturer	OctetString	Fabricante del MD.
MDS: Dev-Configuration-Id	Standard config/Extended config	Tipo de configuración utilizada por el MD.
MDS: Power-Status	onMains(0), onBattery(1), chargingFull(8)	Muestra si el MD está usando batería o conectado a la red.
MDS: Battery-Level	Int (percentage)	Porcentaje de batería restante.
MDS:: Remaining-Battery-Time	MDC_DIM_MIN, MDC_DIM_HR, or MDC_DIM_DAY	Timepo restante de batería.

Tabla 2.1. Ejemplo de etiquetas de la clase MDS.

implementada esa posibilidad, de momento sólamente es capaz de devolver al agente los últimos datos recibidos de los dispositivos.

La comunicación se realiza mediante socket TCP/IP dentro de la misma máquina. Se utiliza una serie de etiquetas para indentificar la información que se está transmitiendo: *deviceInfo* para la información técnica, *measurementInfo* para la información clínica y *stateInfo* para la información del estado del dispositivo. Los comandos que usa el agente para pedir información son *getDeviceInfo* y *gestStateInfo*. Además, cuando la comunicación la inicie el agente, el XML llevará un *requestId* para identificar al mensaje, y relacionarlo con la respuesta enviada por el manager X.73 y saber a qué petición está respondiendo. Cuando el mensaje es enviado de forma asíncrona por el manager no hace falta mandar el *requestId*. En la figura 2.5 se aprecia la comunicación entre manager y agente. En la parte superior se encuentra la comunicación iniciada por el agente pidiendo información al manager, y en la parte inferior la comunicación asíncrona del manager cuando recibe datos de los dispositivos médicos.

El primer paso que realizan los dispositivos para poder transmitir información es

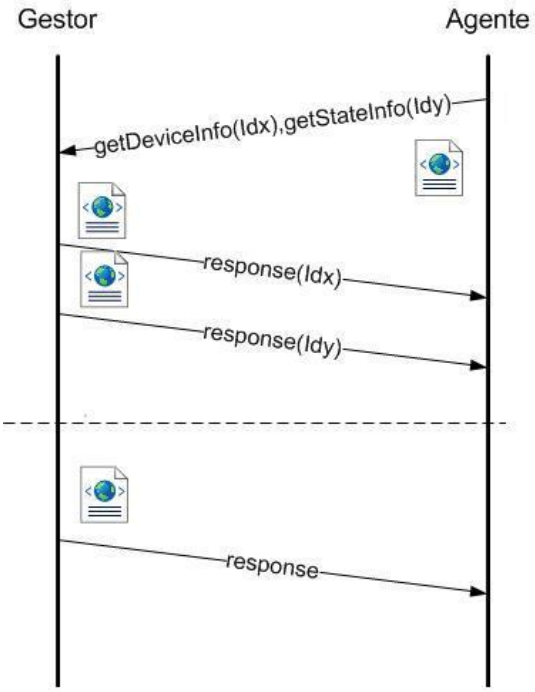

Figura 2.5. Comunicación entre manager X.73 y agente SNMP.

la asociación con el manager. Esto consiste comunicar que su estado es AVAILABLE. El manager se lo comunica al agente, el cual extrae del documento el indentificador del dispositivo (*SystemId*), así como la configuración que está utilizando. Cuando un dispositivo se retira del CE, enviará un estado de NOTAVAILABLE. En la figura 2.6 se presenta un ejemplo de documento XML enviado por un dispositivo médico.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <REQUEST>
  <PROTOCOL>X.73</PROTOCOL>
  - <DEVICE>
    <ID>11111</ID>
    - <COMMAND>
      <ID />
      - <DEVICEINFO>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_ID_MODEL</ID>
            <VALUE>Raytek</VALUE>
            <VALUE>Thermalert TX</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_SYS_ID</ID>
            <VALUE>11111</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_SYS_TYPE</ID>
            <VALUE>MDC_DEV_SPEC_PROFILE_TEMP</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_POWER_STAT</ID>
            <VALUE>0x00</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_VAL_BATT_CHARGE</ID>
            <VALUE>70</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_TIME_BATT_REMAIN</ID>
            <VALUE>6</VALUE>
            <VALUE>MDC_DIM_HR</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_DEV_CONFIG</ID>
            <VALUE>EXTENDED</VALUE>
          </ATTRIBUTE>
        </DEVICEINFO>
      - <MEASUREMENTINFO>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_ID_TYPE</ID>
            <VALUE>MDC_TEMP_ZZZ</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_NU_VAL_OBS_SIMP</ID>
            <VALUE>40</VALUE>
          </ATTRIBUTE>
        - <ATTRIBUTE>
            <ID>MDC_ATTR_UNIT_CODE</ID>
            <VALUE>MDC_DIM_DEGC</VALUE>
          </ATTRIBUTE>
        </MEASUREMENTINFO>
      </COMMAND>
    </DEVICE>
```

Figura 2.6. Ejemplo de documento XML.

Capítulo 3

Diseño de la MIB MedicalDevicesManager

3.1 Introducción

Una MIB (*Management Information Base*) es una base de datos estructurada que sirve para almacenar información intercambiada a través del protocolo SNMP. En este proyecto, para contener la información proporcionada por el manager X.73 se ha diseñado una MIB privada. Esta base de datos se divide en varias tablas, cada una responsable de un tipo de información diferente dentro del sistema. Los gestores pueden aceder a la información recogida en estas tablas e incluso configurar algunas de ellas para mejorar su capacidad de control sobre la información tabulada, mediante la definición de alarmas y eventos, un método muy eficaz de controlar los valores anómalos de forma pasiva por parte del gestor. También se les permite solicitar actualizaciones de los dispositivos, bien de su estado de conexión o de los datos almacenados, así como la configuración del tipo de aviso mediante traps que van a recibir del agente.

La definición formal de las MIB se realiza mediante un lenguaje de definición de datos, el estándar SMI (*Structure of Management Information*). La filosofía de este lenguaje prima la sencillez a la hora de definir datos, nos permite una gran flexibilidad y escalabilidad a la hora del diseño, de manera que esta base de datos se puede expandir sin demasiada dificultad para aumentar su capacidad de gestión sobre los dispositivos en un futuro. Como la definición mediante el estándar SMI es una obligación que impone la arquitectura, permite que cualquier gestor que se conecte al agente sea capaz de

interpretar la estructura de la base de datos y de acceder a sus valores con la misma facilidad que si de una MIB pública se tratara, necesitando únicamente el gestor tener la definición formal de la MIB dentro de su registro de MIBs. El estándar utilizado nos permite realizar una descripción de cada variable dentro de la MIB, de forma que cualquier gestor sea capaz de analizar la información contenida en las tablas sin necesidad de conocimientos previos de la organización y propósito de la MIB. La definición formal se incluye en el anexo E.

Además la definición de alarmas nos permite tanto la emisión de mensajes tipo Trap, como la creación de entradas en una tabla de *Logs*, para poder comprobar un histórico de valores anómalos, la cual será descrita en el siguiente apartado junto con las demás tablas.

3.2 Estructura de la MIB Medical Devices Manager

Siguiendo la estructura en árbol definida por SNMP, la MIB se incluye dentro del grupo de las MIB privadas, en un grupo llamado ZaragozaNetworkManagementResearchGroup, creado en la Universidad de Zaragoza para contener MIBs creadas por sus grupos de investigación, en el cual se recogen diseños anteriores de bases de datos para la gestión de dispositivos. Siguiendo con el orden del grupo, nuestra MIB tendrá el índice 4 dentro del mismo, con lo que el OID que identificará la base de datos MedicalDevicesManager será 1.(iso). 3(org). 6(dod). 1(internet). 4(private). 1(enterprises). 28308(zaragozaNetWorkManagementResearchGroup). 4(medicalDevicesManager). En la figura 3.1 se observa la estructura de la MIB dentro de la organizacion jerárquica de SNMP.

La estructura de la MIB MedicalDevicesManager se dispone en 5 grupos, los cuales contienen 10 tablas en total, separados según la información que contienen y la relación entre las tablas que contienen. En el primer grupo, ComputeEngineControlInfo, encontramos la información relacionada con los recursos propios del CE. El siguiente, MedicalDeviceInfo, contiene las tablas que almacenan información de los dispositivos médicos, tanto de control, como de datos y estados, y finalmente de los errores sucedidos.

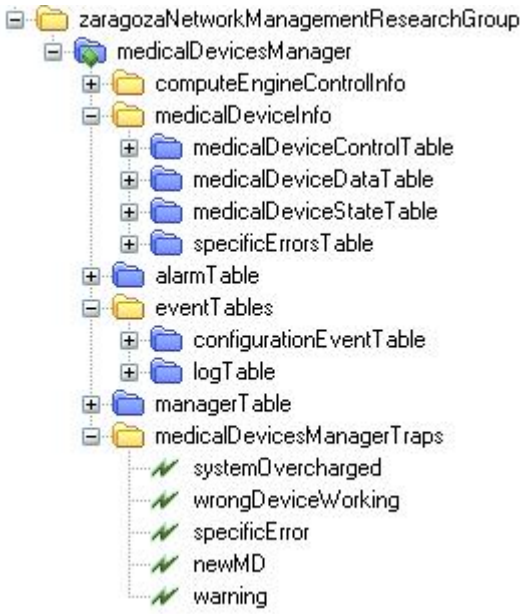

Figura 3.1. Estructura general de la MIB.

El grupo AlarmTable contiene las alarmas definidas por el gestor. EventTables es el grupo que contiene las tablas de eventos definidos por el gestor, y los logs resultantes de los eventos. El siguiente grupo contiene la tabla donde se almacena la información relativa a los gestores del sistema, y finalmente nos encontramos con las posibles traps que puede enviar el agente. Para el diseño de la MIB se ha seguido la filosofía de RMON (17). RMON se define como una extensión de las MIB que basa su funcionamiento en la organización de la MIB que implementa en tabla de control y tabla de datos, consiguiendo de esta forma soportar nuevas funciones además de las de SNMP: configuración e invocación de acciones.

En los subapartados siguientes se explica de forma más detallada la organización de cada tabla y su funcionalidad. La información parámetro a parámetro de cada una de ellas se encuentra en el anexo C.

3.2.1 Grupo ComputeEngineControlInfo

En este grupo no encontramos ninguna tabla, sino "hojas", como se observa en la figura 3.2, en las que se almacenan información estática que caracteriza al CE, tales como su identificador dentro del sistema, tipo de dispositivo, estado de trabajo en el que se encuentra o número de MDs asociados en ese momento; asimismo también encontramos información de los recursos propios del CE, como son el ancho de banda

utilizado y el uso de memoria y de procesador, recogidos con periodicidad de un minuto de las tablas MIB II mediante encuestas SNMP al propio dispositivo y de forma transparente al gestor, incluyendo la fecha en la que se recogieron por última vez estos valores. También se conserva información de contacto con el usuario que posee el CE. Finalmente encontramos dos variables escribibles, updateRequestStateMDs y updateRequestTechnicalDataMDs, que le permiten al gestor pedir una actualización del estado o los datos técnicos de todos los MDs que tiene asociados, respectivamente.

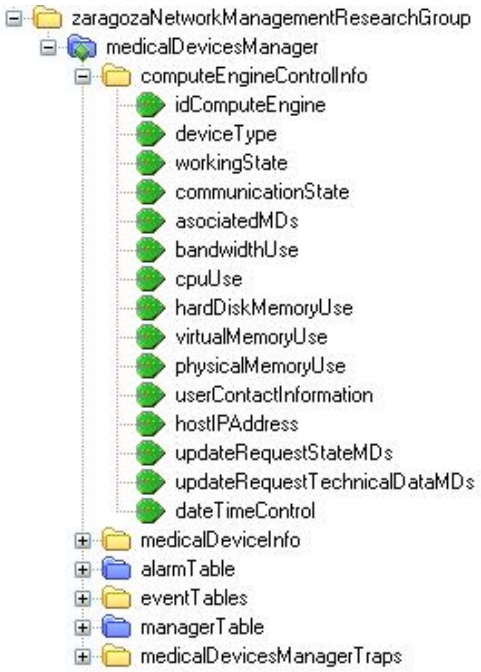

Figura 3.2. Objetos de ComputeEngineControlInfo.

La forma de acceder a cada uno será mediante un OID, que se contruirá a partir del el identificador de la rama madre más un índice consecutivo empezando por el 1 para cada hoja, tal que así: 1 .3 .6 .1 .4 .1 .28308 .4(MedicalDevicesManager) .1(computeEngineControlInfo) .X, siendo X un valor del 1 al 15, que es el número de hojas de esta rama.

3.2.2 Grupo MedicalDeviceInfo

El grupo medicalDeviceControlInfo contiene toda la información relativa a los datos y recursos pertenecientes a los dispositivos médicos asociados. Toda esa información se organiza en 4 tablas, divididas por información de control, datos técnicos de los

dispositivos, errores sucedidos durante la recogida de esos datos, y cambios en el estado de asociación de los MDs.

A cada MD dentro de nuestra base de datos vamos a proporcionarle un identificador que servirá para indexar varias de las tablas de la MIB. Ese identificador serán números consecutivos en orden de la primera conexión al CE. La forma de indexar las entradas de las tablas de este apartado será mediante el identificador del dispositivo responsable de la entrada, y para las entradas correspondientes a datos, estados y errores, también se utilizará el índice del dato, estado o error en cuestión, quedando un OID de esta forma: 1 .3 .6 .1 .4 .1 .28308 .4(MedicalDevicesManager) .2(MedicalDeviceInfo) .1(MedicalDeviceControlTable) .A(identificador de la tabla dentro del grupo) .1(Entry de la tabla) .B(indentificador del parámetro dentro de la tabla) .C(identificador del dispositivo) .D(identificador del dato, estado o error, cuando se requiera).

3.2.2.1 MedicalDeviceControlTable

Esta tabla contiene información estática de los dispositivos. Entre ellos se incluyen el fabricante, modelo e identificador del MD, además del tipo de dispositivo, protocolo, configuración utilizada y la fecha en que se conectó por primera vez al CE. Finalmente contiene dos parámetros de escritura similares a los descritos en el grupo anterior, updateRequestState y updateRequestTechData, que en este caso se utilizan para permitir al gestor pedir la actualización del estado o información técnica del dispositivo en cuestión.

Se creará una entrada nueva en esta tabla cada vez que un nuevo dispositivo se registre en el CE. Es posible que la primera vez que un dispositivo se conecte no nos proporcione información suficiente para llenar todos los parámetros de la nueva entrada. Si es así, todos los valores no facilitados permanecerán con un valor nulo hasta que el MD los actualice en próximas transmisiones. Para solicitar una actualizacion de valores, el gestor activará la actualización poniendo el valor 1 en los campos de updateRequestState o updateResquestTechnicalData. Posteriormente estos valores volverán al valor anterior para no estar realizando peticiones de actualización continuamente.

Figura 3.3. Objetos de MedicalDeviceControlTable.

3.2.2.2 MedicalDeviceDataTable

Esta tabla contiene información relativa al funcionamiento técnico de los MDs tales como el tipo de alimentación, el nivel de batería restante y la estimación de la duración de la batería, o si se ha producido algún error en esa medición. Además guardamos también la fecha en que se realizó esa entrada en la tabla.

Para cada uno de los dispositivos esta tabla permite registrar un total de 20 entradas relativas a sus datos técnicos, lo que permitirá realizar comparativas o permitirá observar su comportamiento a lo largo del tiempo. Cuando un dispositivo envía por primera vez un dato, se incializan sus 20 entradas a un valor nulo y se van rellenando conforme van llegando más mediciones. Una vez completas todas las entradas disponibles, se empiezan a sobreescribir las entradas más antiguas y se actualiza la posición de cada entrada para que la nueva entrada pase a ser la primera en la tabla. A la hora de evaluar las entradas desde el gestor, ignoraremos aquellas que todavía sigan con valor nulo.

3.2.2.3 MedicalDeviceStateTable

Cuando el manager detecta un cambio de estado en un dispositivo lo hace saber mediante el documento XML correspondiente. En esta tabla se almacenan esos cambios

de estado. Además del nuevo estado, en esta tabla podemos encontrar la fecha en que se recibió la información y los identificadores correspondientes al dispositivo y el número de captura de la entrada. Para esta tabla se han seguido las mismas reglas respecto al tamaño y la inicialización que en la tabla anterior.

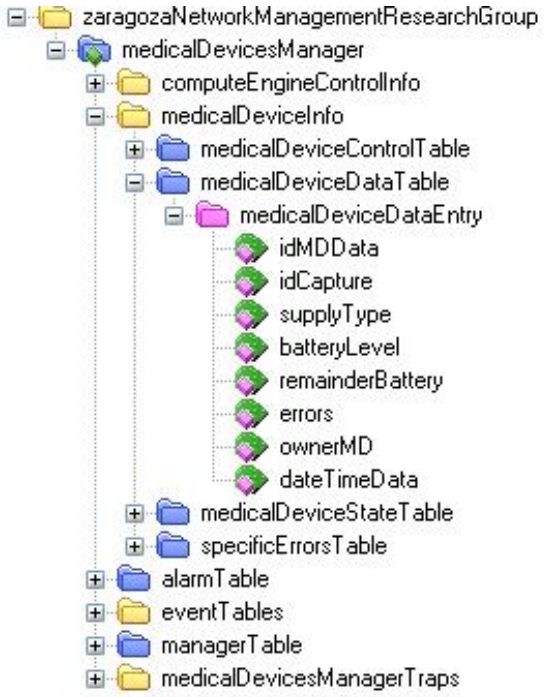

Figura 3.4. Objetos de MedicaDeviceDataTable.

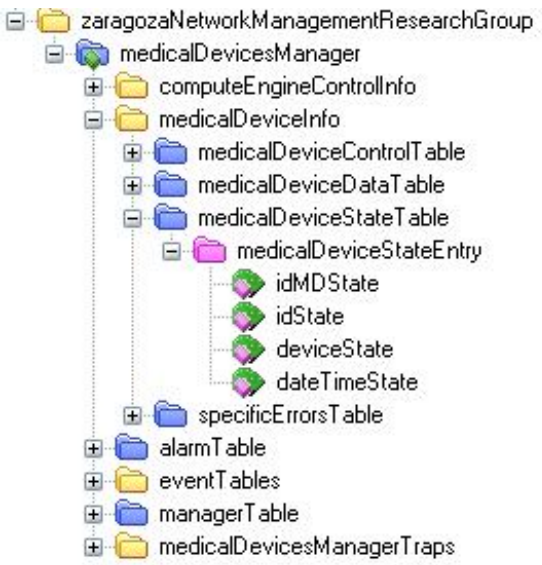

Figura 3.5. Objetos de MedicalDeviceStateTable.

3.2.2.4 SpecificErrorsTable

A veces los datos recibidos de los dispositivos muestran un problema de funcionamiento del mismo. La información relacionada con los errores existentes en la recepción de datos técnicos se incluyen en una entrada en esta tabla. Los errores que se van a monitorizar son en el funcionamiento del dispositivo en sí, en el funcionamiento del sensor en caso de que el dispositivo tuviera uno, en la conexión del dispositivo con el CE, las posibles interferencias que pudiera tener el sensor y que invalidaría la medida o un fallo en la señal, como por ejemplo una señal errática o demasiado pobre para medirse adecuadamente. Realmente la mayoría de los dispositivos no son capaces de transmitir estos problemas específicos, únicamente el pulsioxímetro dentro de los dispositivos que vamos a tratar.

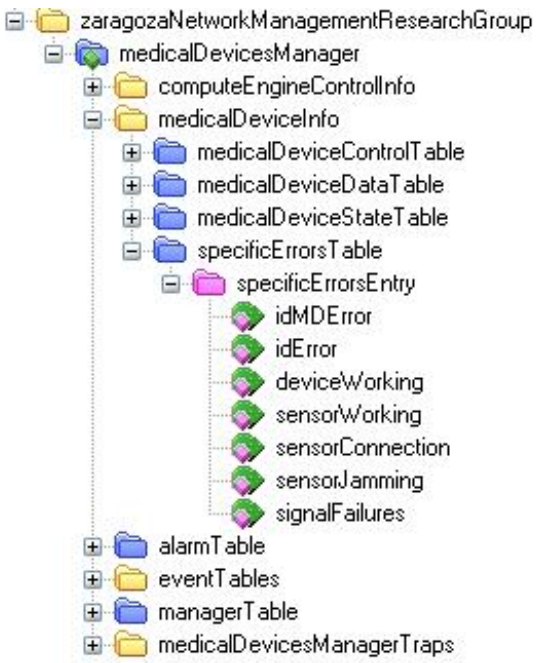

Figura 3.6. Objetos de SpecificErrorsTable.

Además de los datos técnicos recibimos también el dato médico del dispositivo para tratarlo en la búsqueda de errores. Mientras que los errores específicos son enviados mediante una entrada en el documento XML que se recibe, un error en el funcionamiento normal del dispositivo se descubre analizando la información médica. Del dato médico se obtiene tanto su valor como la unidad en que se ha realizado y el tipo de dato. Se evalúa si el valor entra dentro de los parámetros correctos para ese tipo de medida y esas unidades. En caso de que no sea el caso, se activa el parametro de error de la entrada

en la tabla de datos y se crea una nueva entrada en la tabla de errores especificando el tipo de error que se ha observado.

3.2.3 Grupo AlarmTable

Este grupo está formado en exclusiva por una tabla que almacena las alarmas definidas por los gestores del agente. Como es una tabla configurable por el gestor, incluímos un parámetro de estado de la entrada para saber si la entrada esta en creación o ya creada y validada. Se permite la definición de 3 tipos de alarmas: alarmas por valor, por un valor que sobrepase un umbral definido por encima o que lo sobrepase por debajo. Para cada una habrá que definir el OID de la variable que vamos a monitorizar, así como el valor de la misma o los umbrales que no debe sobrepasar y el evento que se va a producir al activar la alarma, dependiendo de la alarma que se defina. Además se debe especificar el identificador del gestor que ha definido dicha alarma, para saber cuál es la dirección IP del gestor en caso de que haya que enviarle una Trap.

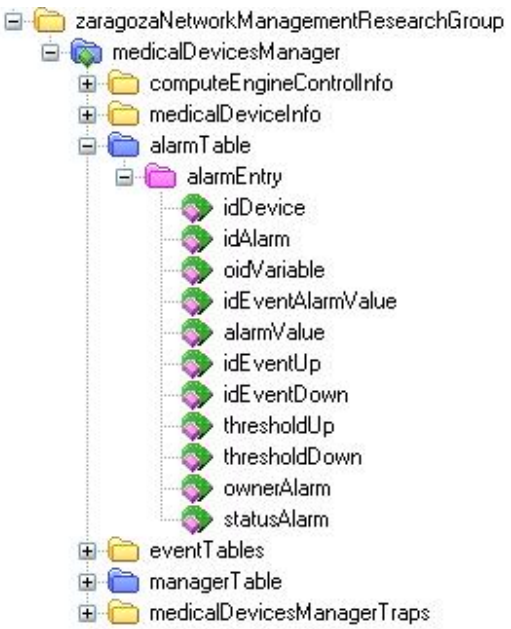

Figura 3.7. Objetos de AlarmTable.

En este grupo la indexación se realizará mediante el identificador del dispositivo y el índice de la alarma, quedando un OID final similar a los anteriores, 1 .3 .6 .1 .4 .1 .28308 .4(MedicalDevicesManager) .3(AlarmTable) .1(AlarmEntry) .A(identificador del parámetro dentro de la tabla) .B(identificador del dispositivo) .C(índice de la alarma).

Para crear una entrada en la tabla de alarmas se tiene que mandar primero una petición de creación de entrada, lo que si se ha realizado correctamente dará lugar a una nueva entrada en la tabla con todo sus valores nulos menos los índices, necesarios para realizar la indexación. Más tarde habrá que ir rellenando los valores del OID de la variable a monitorizar, del gestor y de al menos una de la tres alarmas disponibles antes de que se nos permita validar la entrada. Hasta que la entrada no se valida la alarma no se considera definida y no se activará cuando debería. Esta forma de crear una entrada se adecúa al uso de la filosofía RMON.

3.2.4 Grupo EventTables

En este grupo podemos encontrar la información relacionada con la gestión de los eventos activos en el sistema. Se subdivide en dos tablas, una para almacenar los datos necesarios para configurar los eventos que crean los gestores, y otra que contiene los logs de los eventos que así lo dispongan en su configuración.

A la hora de indexar este grupo, el identificador del evento será obligatorio, añadiendo el índice de log para la tabla de logs, resultando un OID como el siguiente: 1 .3 .6 .1 .4 .1 .28308 .4(MedicalDevicesManager) .4(EventTables) .A(identificador de la tabla).1(Entry de la tabla) .B(identificador del parámetro en la tabla) .C(identificador del evento) .D(índice de log, cuando se requiera).

3.2.4.1 ConfigEventTable

En esta tabla se almacenan los eventos que los gestores han definido para controlar las acciones que se llevan a cabo cuando se activa una alarma. Para ello se han de definir en cada evento el tipo de evento que es, el tipo de trap que se manda cuando se activa, en caso de mandar alguna, y el gestor al que se debe avisar en caso de activación.

La forma de crear entradas es del mismo estilo que para la entrada de alarmas. La diferencia radica en que en este caso necesitamos tener la entrada definida en su totalidad antes de validar.

3.2.4.2 LogTable

Esta tabla registra los logs resultantes de la activación de una alarma. No todas las alarmas generan un log, eso depende de si el evento asociado a ella lo permite. Como los

logs depende tanto de la alarma como de evento asociado, para cada log registramos la alarma y el evento asociado que lo crearon. Además se añade una descripción de lo que ha pasado para que se creara la entrada y la fecha en que se creó.

Un evento puede tener varias alarmas asociadas, mientras que una alarma está asociada a un único evento. Por ello para indexar la tabla utilizaremos el índice del evento, además del índice del log.

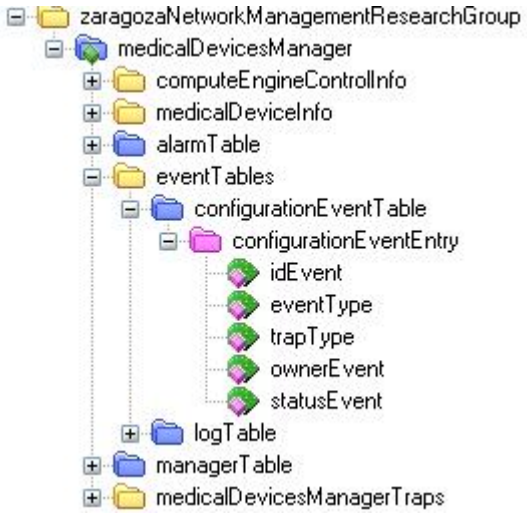

Figura 3.8. Objetos de ConfigEventTable.

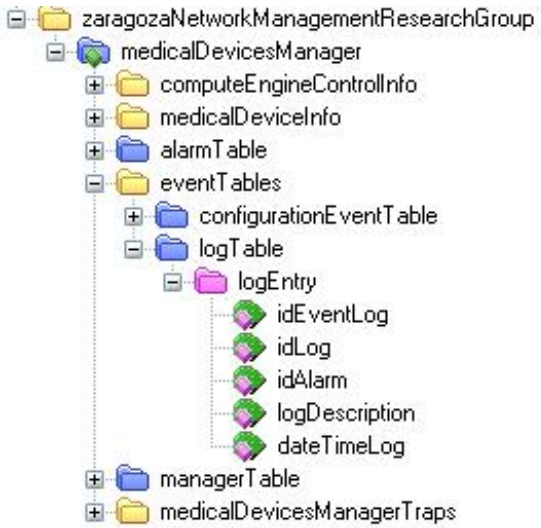

Figura 3.9. Objetos de LogTable.

3.2.5 Grupo ManagerTable

Grupo compuesto por una única tabla, permite registrar gestores a los que luego está permitido el envío de traps. Esta información se compone de la IP del gestor para recibir los traps, un teléfono de contacto, dirección de correo electrónico y lugar de trabajo. También encontramos un parámetro que nos permite en un futuro diferenciar los permisos que tiene cada gestor dentro de la MIB. Además existe un campo que nos permite indexar los gestores de forma consecutiva.

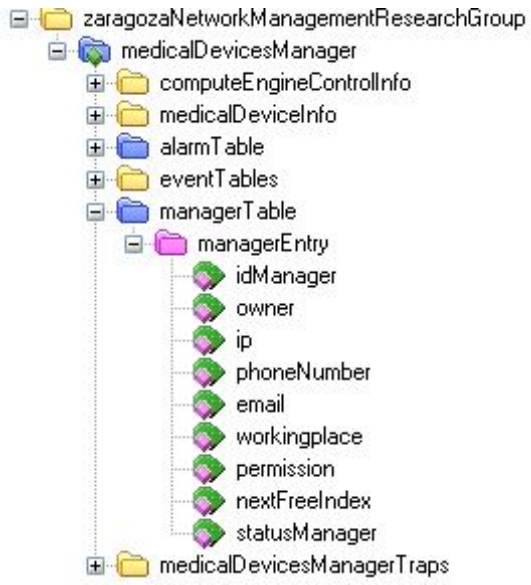

Figura 3.10. Objetos de ManagerTable.

Para indexar utilizaremos precisamente el índice del gestor, lo que resultrá en un OID como este: 1 .3 .6 .1 .4 .1 .28308 .4(MedicalDevicesManager) .5(ManagerTable) .1(ManagerEntry) .A(identificador del parámetro en la tabla) .B(identificador del gestor).

3.2.6 Definición de los Traps

Para informar al gestor de los problemas técnicos de los dispositivos se han diseñado 5 tipos de Traps para abarcar todas las posibles situaciones: *SystemOvercharged*, *WrongDeviceWorking*, *SpecificError*, *NewMD* y *Warning*.

Un Trap es un mensaje enviado por un agente de forma asíncrona al puerto 162 (puerto por defecto para la recepción de Traps) de un gestor de la red, para informar de ciertos cambios importantes sobre el estado de un proceso. La definición y envío de

los Traps permite a los gestores conocer los problemas técnicos de los dispositivos en tiempo real, reduciendo el tiempo de reacción necesario para solucionar el problema. Con cada mensaje no sólo se envía el tipo de Trap, sino también una serie de valores que nos ayudan a evaluar el problema sin tener que buscar en las tablas de la MIB qué valor ha hecho saltar las alarmas.

Cuando el uso de alguno de los recursos propios del CE supera un cierto umbral, se lanza una Trap de tipo *SytemOvercharged*. De esta forma evitamos que se sobrecargue el CE y tengamos problemas en la recepción de la información de los dispositivos técnicos. Las variables que se adjuntan son el ancho de banda, uso de CPU y uso de memoria del CE, junto con la fecha en que ocurrió la sobrecarga.

Cuando alguna de las medidas realizadas por los dispositivos se salen del rango aceptable en caso de buen funcionamiento del dispositivo, se activa un Trap tipo *WrongDeviceWorking*. Este mensaje nos permite conocer el mal funcionamiento de alguno de los dispositivos. Los valores que se mandan son el identificador del dispositivo, el identificador de la captura y la fecha en que ocurrió el fallo.

Para errores más concretos que son comunicados a través de los archivos XML se utiliza el Trap *SpecificError*. Con este mensaje mandamos todos los parámetros de la entrada en la tabla de errores que ha creado esta situación. De esta forma el gestor puede ver cuál de los posibles errores específicos ha ocurrido.

También nos interesa saber cuándo se ha conectado un dispositivo nuevo, por eso definimos la Trap *NewMD*. Cada vez que un nuevo dispositivo se registra en el CE, se manda un Trap de este tipo a cada gestor registrado en la tabla ManagerTable. Para conocer la información relativa al dispositivo, se manda la entrada creada en la tabla de control de MDs junto al tipo de trap en el mensaje.

Finalmente, se define un trap genérico que se puede utilizar para todos aquellas variables de la MIB que no son cubiertas con los Traps anteriormente descritos, como por ejemplo una alarma de cambio de estado a un valor concreto. Este tipo de Trap se denomina *Warning*, e incluye el OID y valor que ha hecho activarse a la alarma.

3.3 Definición formal de la MIB

La definición formal de la MIB ha sido realizada siguiendo el SMIv2 (18), estructura de gestión de la información, que presenta el formato para la información

de administración, convenciones, sintaxis y reglas básicas para la construcción de MIBs. La definición completa y estructura formal de la MIB se recoge en el anexo E. Un ejemplo de la definición sería el siguiente:

```
idComputeEngine OBJECT-TYPE
SYNTAX OCTET STRING
ACCESS read-only
STATUS mandatory
DESCRIPTION ‘‘This is the identifier for this compute engine inside
the network.’’
::= { computeEngineControlInfo 1}
```

Cada objeto de la MIB tiene que tener especificados los siguientes parámetros: Nombre (usando la convención SMI), tipo de dato del ASN.1 (integer32,octetstring, objectidentifier,etc.) o de los tipos de datos definidos en la MIB (en este caso: Date, DisplayString EntryStatus, Mode, OperationValue, OwnerString), el tipo de acceso permitido para ese valor (lectura, escritura, lectura / escritura), soporte de la implementación requerida (status: definidos todos en este caso como obligatorios), una breve descripción del objeto y su relación con otros objetos de la MIB, es decir, su posición dentro del árbol, indicando a qué tabla pertenece y en qué posición se encuentra.

Capítulo 4

Implementación y pruebas

4.1 Definición de un entorno de simulación

Para comprobar el correcto funcionamiento del sistema integrado, se han realizado una serie de pruebas incluyendo todos los aspectos que el agente es capaz de gestionar. Hemos realizado las pruebas con un software de simulación de diseño propio que nos permitirá reproducir situaciones reales de funcionamiento.

El primer interfaz es el que nos aparece al lanzar el agente en un equipo. Este interfaz nos permite configurar el puerto y la IP en el que escuchará el agente, así como los parámetros de seguridad que regirán las comunicaciones. Estos parámetros están compuestos por el nombre de usuario, las contraseñas de autenticación y privacidad y el nivel de seguridad de la comunicación (privacidad y autenticación, sólo autenticación o ninguna de las dos). Una vez lanzado el interfaz, podemos conectarnos al mismo mediante un gestor SNMP comercial. En nuestro caso utilizamos MIB Browser. Además de escuchar en un puerto a la espera de la conexión del gestor, el agente también abre un socket que le permitirá recibir los documentos XML proporcionados por el manager X.73. El interfaz se puede observar en la figura 4.1, en el anexo A se puede comprobar con más detalle la forma en que el agente se conecta al gestor y recibe estos documentos.

El segundo interfaz nos permitirá simular la presencia de dispositivos médicos en el sistema. Vamos a simular 4 tipos diferentes de dispositivos: termómetro, báscula, pulsioxímetro y medidor de presión arterial. El interfaz simula el comportamiento de estos dispositivos, permitiendo enviar modificaciones en el estado o mensajes con información técnica. Tras seleccionar el tipo de información que se va a transmitir,

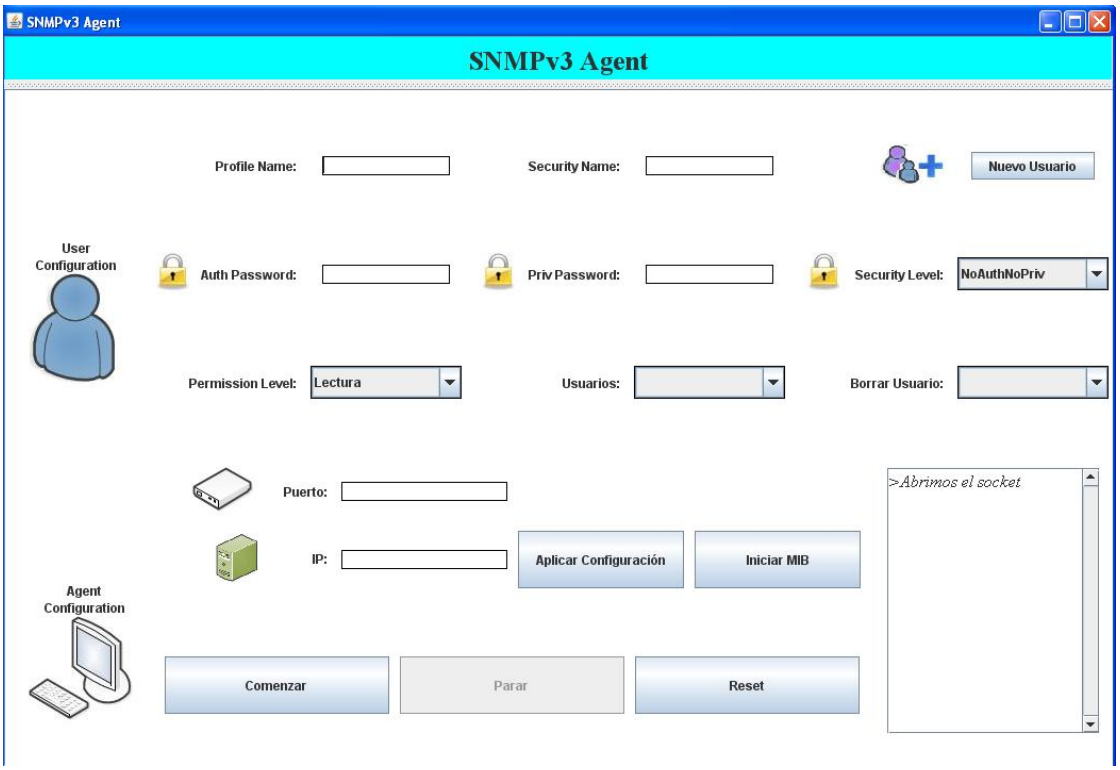

Figura 4.1. Interfaz gráfico del agente desarrollado.

se crea un documento XML que representará dicha información mediante las etiquetas correspondientes del protocolo X.73 para cada parámetro. Esta información se transmite utilizando un socket como si del verdadero manager se tratara. Este interfaz se explica más detalladamente en el anexo B.

4.2 Entorno de pruebas

Para evaluar los resultados obtenidos en las pruebas realizadas al agente, vamos a mostrar el comportamiento del agente mediante un ejemplo que ilustraremos con capturas de pantalla en el que se seguirá el proceso lógico de conexión y envío de datos de un dispositivo, además de la creación de alarmas y eventos y su posterior funcionamiento. Para comprobar que los valores en las tablas son correctos nos conectaremos al agente con el *MIB Browser* y visualizaremos las tablas desde su interfaz gráfico.

4.2.1 Recogida de información del CE

Cuando activamos el agente, aparte de abrir el socket y quedarse escuchando a la espera de conexiones por parte del gestor, también empieza la recogida de información de recursos propios del CE. El agente envía peticiones SNMP a la MIB II presente en el CE en la que se recoge información del tráfico y nivel de utilización de los sistemas del CE. Estos datos se almacenan en la tabla de control del CE. El resto de valores de esa tabla son estáticos en su mayoría y se escriben al iniciar el agente por primera vez. En la figura 4.2 vemos los objetos pertenecientes a la tabla de control tras consultar los recursos propios del sistema.

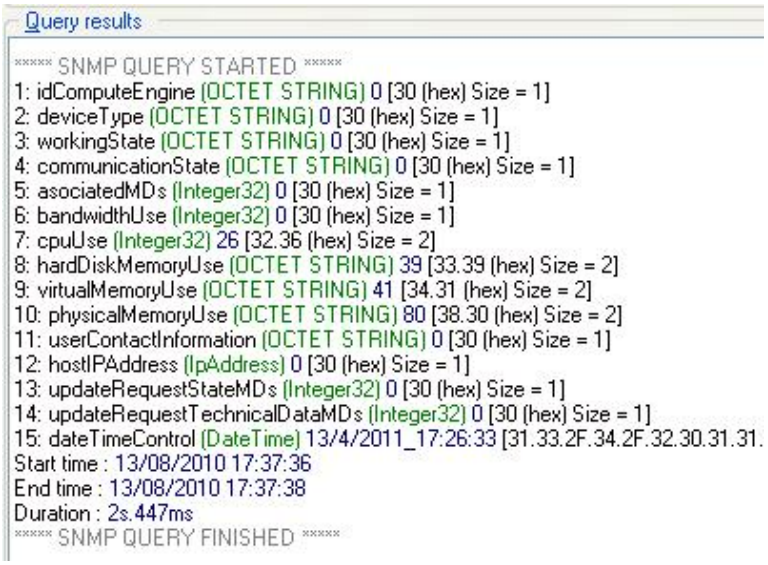

Figura 4.2. Visualización del contenido de la tabla de control del CE.

4.2.2 Recepción y almacenamiento de datos de los dispositivos médicos

Uno de los principales objetivos del agente es el mantenimiento de una MIB con la información que le llega de los dispositivos médicos. Para comenzar las pruebas, partimos del supuesto de que el agente acaba de ser instalado en el CE y no se ha conectado todavía ningún dispositivo, por lo tanto sus tablas están vacías. Así podemos comprobar que desde el principio el agente funciona correctamente.

En este caso vamos a simular la conexión de un termómetro al CE. Al conectarse el botón rojo del interfaz gráfico pasa a color verde si se ha realizado la conexión, como

vemos en la figura 4.3. Lo primero que hacen los dispositivos al conectarse por primera vez es mandar su estado de funcionamiento. Cuando un dispositivo se conecta por primera vez debe enviar un estado de AVAILABLE. A partir de ese momento el dispositivo puede cambiar de estado, pero no podrá enviar datos hasta que se encuentre en estado ASSOCIATED. La primera vez que llegue a este estado el dispositivo enviará información técnica, y a partir de entonces cada vez que el dispositivo se asocie podrá enviar información, cambiando su estado a OPERATING. En la figura 4.4 se muestra un ejemplo de XML.

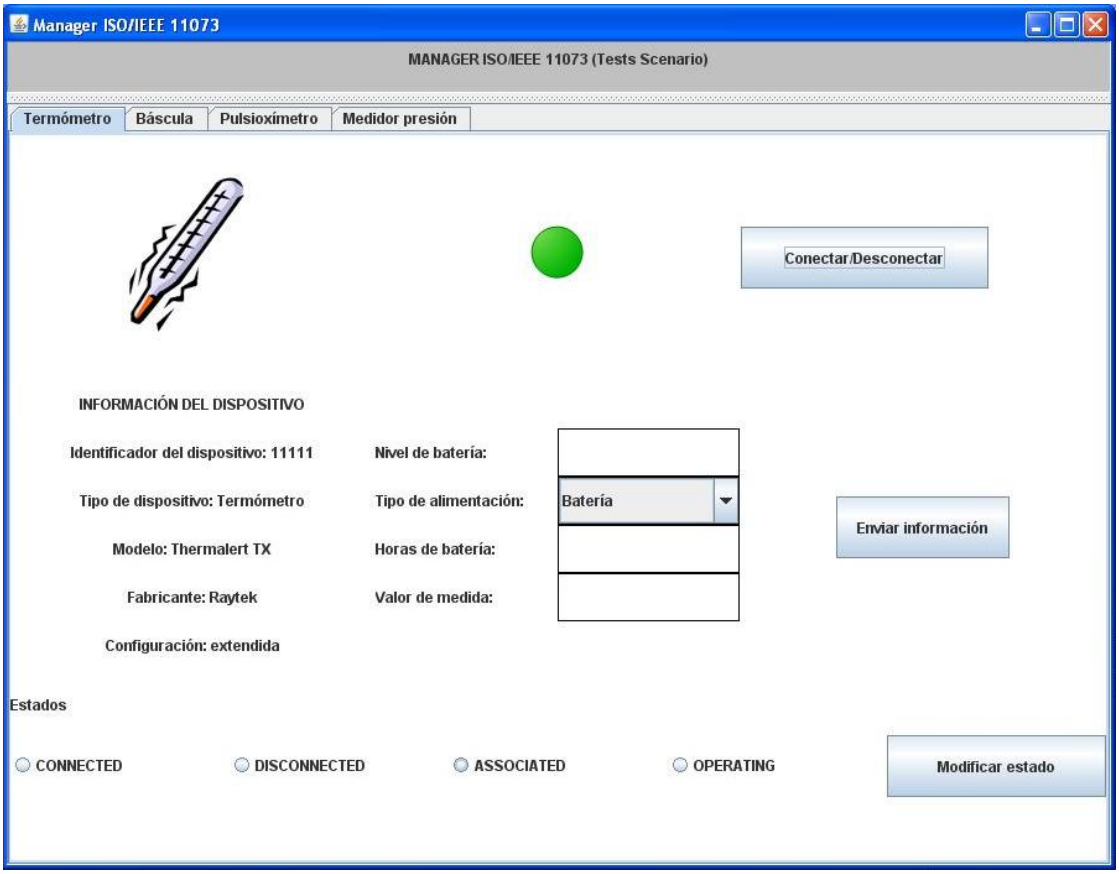

Figura 4.3. Interfaz con un termómetro conectado al sistema.

Para interpretar la información presente en el documento, el agente lee las etiquetas que deben llevar los parámetros para poder ser interpretadas por el protocolo X.73. En la figura 4.4 vemos algunos ejemplos de etiquetas. Unas etiquetas se utilizan para información como la etiqueta STATEINFO para imformar de un estado. También hay etiquetas de atributos como MDC_ATTR_DEV_CONFIG, que sirve para indicar que el valor siguiente es la configuración del dispositivo.

Dentro del XML, la información se divide en apartados según el tipo de

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <REQUEST>
    <PROTOCOL>X.73</PROTOCOL>
  - <DEVICE>
      <ID>11111</ID>
    - <COMMAND>
        <ID />
        <STATEINFO>CONNECTED</STATEINFO>
      </COMMAND>
    </DEVICE>
  </REQUEST>
```

Figura 4.4. Ejemplo de XML que contiene información de estado.

información que se trate: técnica, médica o de estado. Dentro de cada apartado se transmiten diferentes atributos, cada uno con un identificador (como el MDC_ATTR_DEV_CONFIG) y el valor o valores correspondientes. Los atributos médicos también incluyen la escala con la que están tomados.

Tras interpretar la información contenida en el documento, el agente almacena dicha información en la tabla que corresponda. Lo primero que se comprueba es si el dispositivo que ha enviado el mensaje está registrado previamente en la MIB. De no ser así, lo registramos en la tabla de control de MDs y le asignamos un identificador dentro de la MIB. Además, actualizamos el número de MDs conectados al CE en la tabla de control del CE. Si la información era técnica, se crea una entrada en la tabla de datos. Los datos médicos incluídos en un XML con la información técnica servirán para comprobar si las medidas realizadas por el dispositivo entran dentro de los márgenes aceptables de trabajo. Es decir, si un termómetro está registrando valores de 50 °C, hay algo que no funciona bien. Si el dispositivo transmite un error específico o el dato médico se sale de los umbrales de aceptación, se crea a su vez una entrada en la tabla de errores. Finalmente, la información de estado se escribirá en la tabla de estados. Como explicamos en el capítulo anterior, las tablas de datos y estados han sido limitadas en extensión para evitar sobrecargar al agente con tablas demasiado grandes.

Siguiendo con el ejemplo que se ha iniciado al conectar un nuevo dispositivo, seguimos realizando cambios en su estado, para posteriormente poder enviar información. En la figura 4.5 podemos observar los estados por los que ha pasado el dispositivo antes de poder enviar información.

El siguiente paso es el envío de información por parte del dispositivo, cuyo registro

en las tablas de la MIB podemos apreciar en la figura 4.6. A continuación conectaremos otro dispositivo más (figura 4.7) y enviaremos mensajes con datos que contienen errores, tanto generales de funcionamiento como específicos a traves del pulsioxímetro. En las figuras 4.8 y 4.9 podemos observar el funcionamiento de la MIB en esos casos.

4.2.3 Comprobación de alarmas y eventos

El gestor crea una alarmas o evento mediante el envío de peticiones SNMP. Para ello nuestro agente tiene que tratar esas peticiones, comprobar qué tipo de petición es, y tratarla como corresponda.

Se van a definir una serie de alarmas para probar los diferentes tipos de Traps que se pueden enviar. Primero se define una alarma que mande un trap de tipo *Warning* cuando el primer dispositivo que se conectó envíe un estado de DISCONNECTED. En la figura 4.11 se puede apreciar la configuración de la alarma.

Hay que tener en cuenta que para poder definir una alarma antes se deben tener creados los eventos a los que se va a asociar, como se puede comprobar en la figura

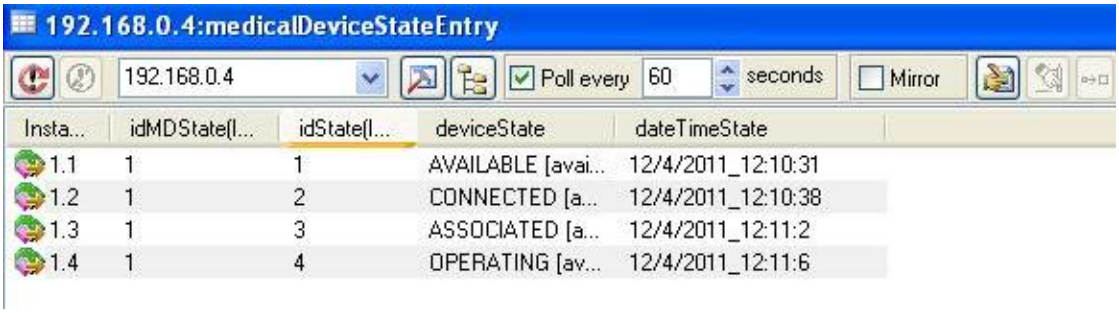

Figura 4.5. Vista desde MIB Browser del contenido de la tabla de estados.

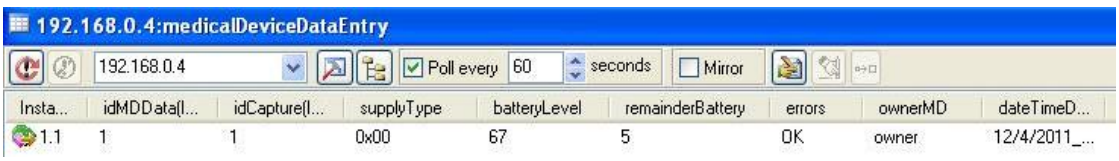

Figura 4.6. Vista desde MIB Browser del contenido de la tabla de datos.

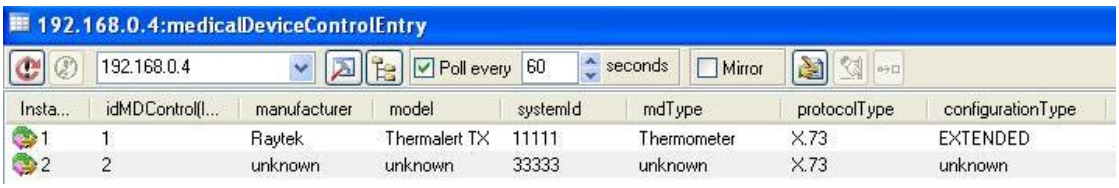

Figura 4.7. Conexión de varios dispositivos.

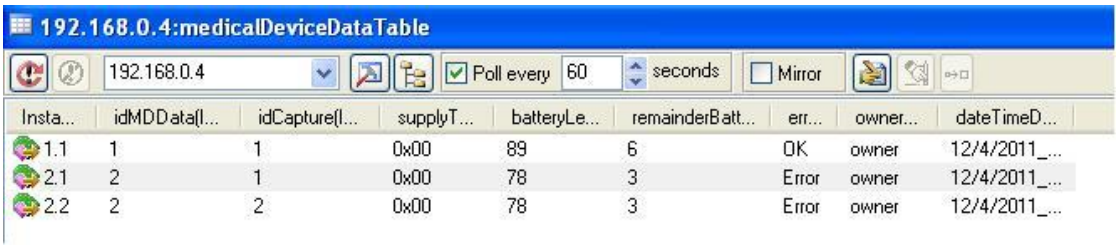

Figura 4.8. Ejemplo de datos con errores.

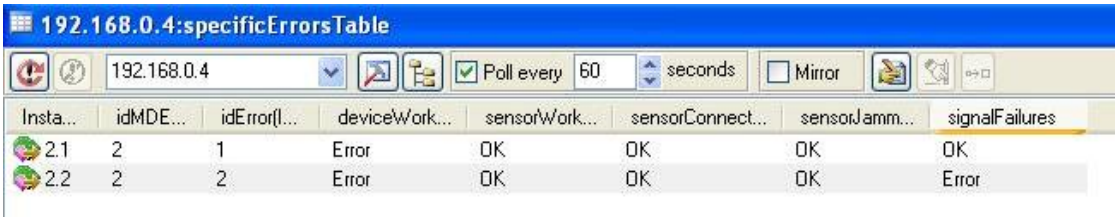

Figura 4.9. Vista de la tabla de errores.

4.10. Tampoco se puede validar la alarma hasta que no se hayan definido un número

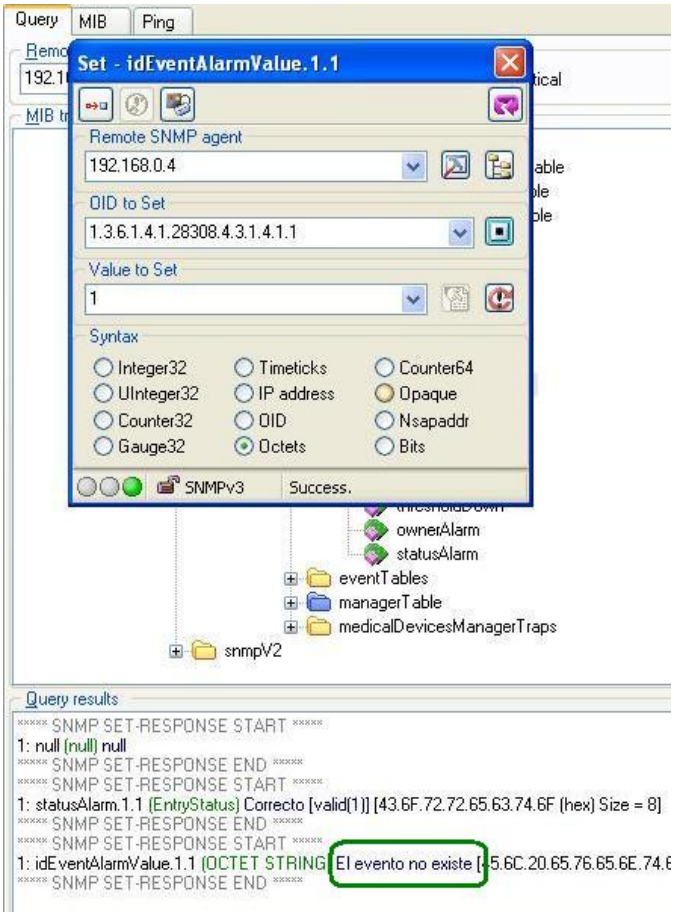

Figura 4.10. Error al crear alarma al no estar creado el evento asociado.

determinado de objetos dentro de la tabla, y hasta que la alarma no se haya validado no será evaluada a la hora de determinar si se activan o no las alarmas.

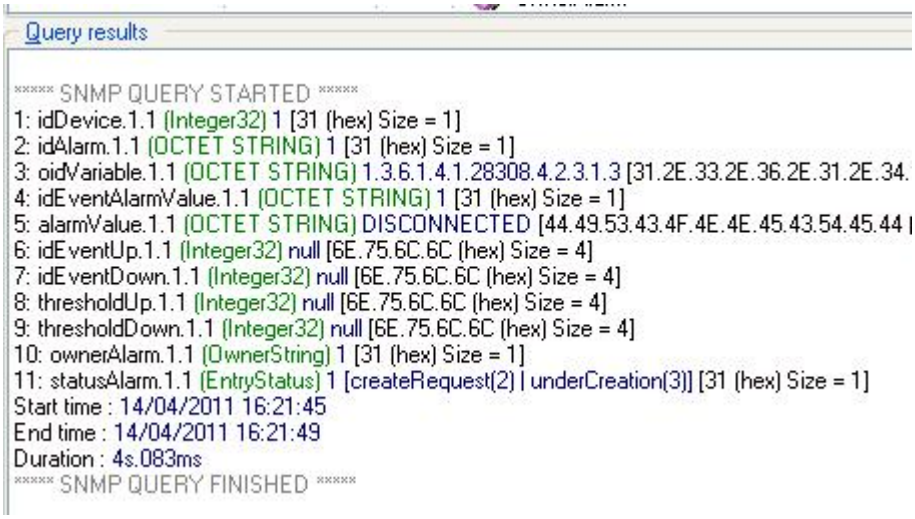

Figura 4.11. Alarma configurada para el estado DISCONNECTED.

Para evaluar si se activa una alarma, cada vez que se introduce un nuevo dato en una tabla, el agente comprueba si su OID coincide con alguno perteneciente a una alarma definida y validada. En el caso de que así sea, comprueba que el valor de la variable cumple las restricciones impuestas por la alarma, esto es, el valor no excede los umbrales o no es un valor concreto que activa la alarma. En caso de activarse la alarma, el evento asociado a la misma indica lo que se debe hacer a continuación. Según sea el caso se transmite un trap al gestor que definió la alarma, se crea una entrada en la tabla de logs o ambas acciones.

En la figura 4.12 vemos cómo llega una Trap al gestor indicando, además del tipo de Trap, el valor que ha hecho activarse la alarma.

4.2.4 Actualización de los MDs

Para simular el comportamiento del manager X.73 ante una petición de actualización de un dispositivo, se dispone de una serie de XML con información que responde a todos los tipos de peticiones que pueden llegar de parte del agente. Tras enviar la petición, se simulará que el manager la ha recibido, la ha procesado y responde con un XML. La respuesta es procesada a su vez por el proxy y posteriormente se trata como si de un mensaje asíncrono enviado por el manager se tratara, actualizando las tablas que sean necesarias.

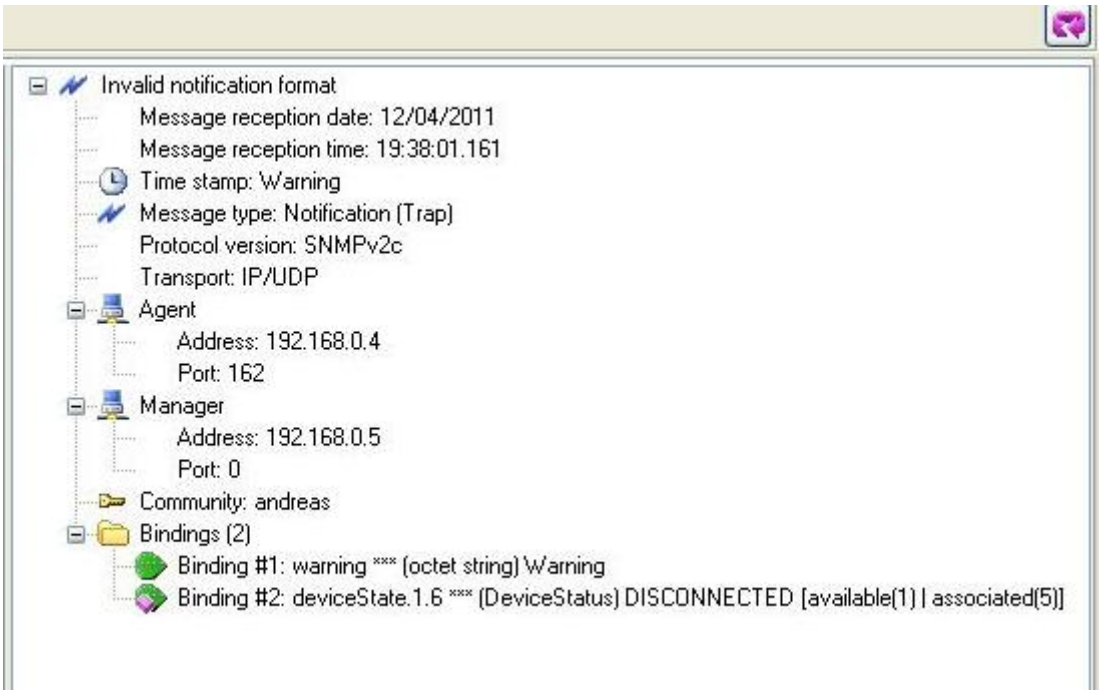

Figura 4.12. Trap enviada por la alarma anterior.

Para ilustrarlo en la figura 4.14 se puede comprobar como tras la petición de actualización del dispositivo, se ha producido una actualización en su estado.

Figura 4.13. Estados del dispositivo antes de actualizar.

Figura 4.14. Estados del dispositivo después de actualizar.

4.2.5 Borrado de tablas y objetos

Cuando un dispositivo se desvincula del CE, es decir, pasa al estado NOTAVAILABLE, ya no es necesario guardar su información en la MIB, así que se procede al borrado de toda la información vinculada al dispositivo.

Se va a borrar el primer dispositivo que se conectó, para poder evaluar todos los pasos que se realizan a la hora de eliminar información. En las figuras 4.15 y 4.16 se aprecia la situación de las tablas de control y estados antes de borrar el dispositivo, y en las figuras 4.17 y 4.18 el estado de las mismas tras proceder a la eliminación del MD.

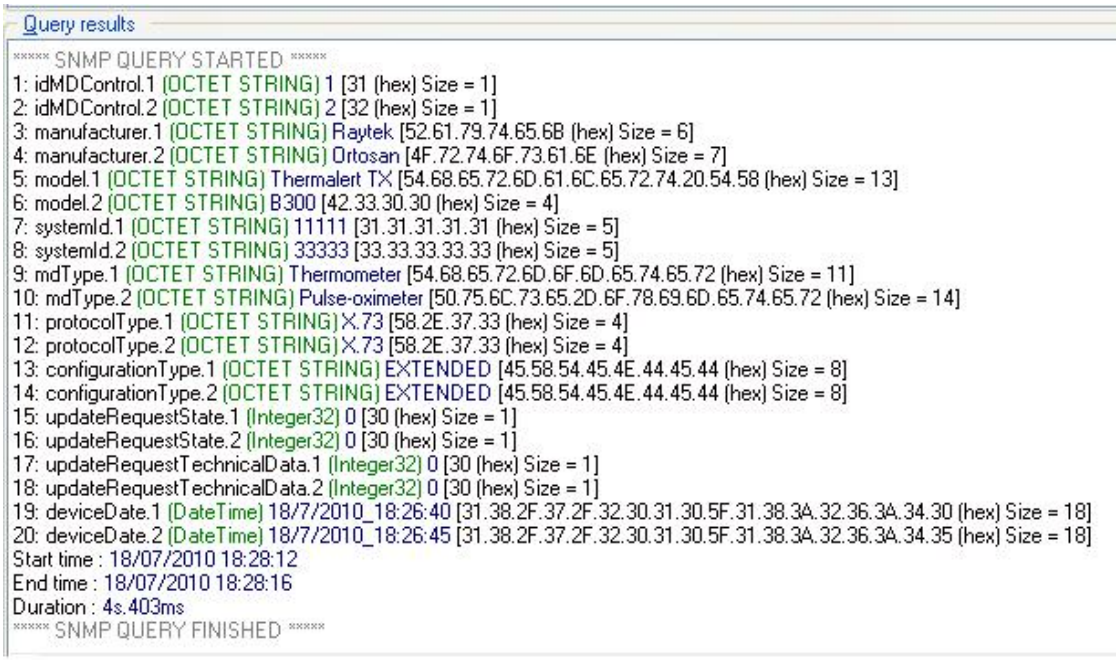

Figura 4.15. Tabla de control antes de eliminar dispositivo.

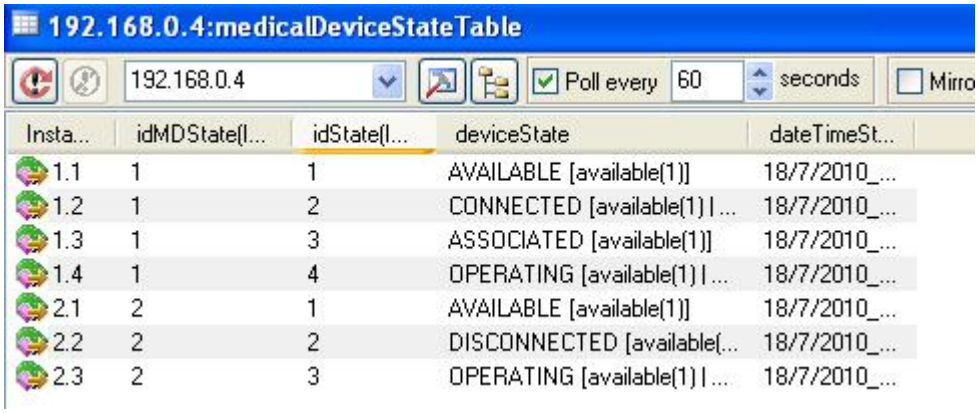

Figura 4.16. Tabla de estados antes de eliminar dispositivo.

Figura 4.17. Tabla de control después de eliminar dispositivo.

Figura 4.18. Tabla de estados después de eliminar dispositivo.

Como se aprecia en las figuras posteriores a la eliminación, al eliminar el primero de los dispositivos, se actualizan los identificadores de los dispositivos que se registraran posteriormente, lo que conlleva una actualización de los índices utilizados para indexar tablas. Además de borrar la información de control del dispositivo, se deberán eliminar las entradas correspondientes a sus datos, estados, errores y alarmas.

En cuanto a las alarmas y eventos, se puede borrar una alarma o evento de forma individual cambiando su valor de status a 4. La eliminación de un evento conlleva además la de los logs generados por el mismo.

Capítulo 5

Conclusiones y líneas futuras

5.1 Conclusiones

En este proyecto se ha desarrollado un agente SNMPv3 para la gestión de dispositivos médicos en entornos de atención domiciliaria. La utilización de esta arquitectura proporciona una estrutura de datos en la que se incluirá la información obtenida de los dispositivos médicos organizada de forma lógica y ordenada en tablas (MIB), así como del protocolo y la base de las comunicaciones con los gestores de la red y de las comunicaciones con el manager X.73. Para desarrollar todas las herramientas que componen el sistema de gestión se ha utilizado el lenguaje de programación Java.

El agente constituye un sistema totalmente centralizado que puede instalarse como un módulo más en cualquier sistema que ya posea una gestión de redes vía SNMP. Además, no sólo ofrece la ventaja de una centralización de la información proveniente de los dispositivos médicos, sino que permite una escalabilidad total. Para el desarrollo completo del agente se han diseñado e implementado los bloques que integran el sistema completo: establecimiento y control de comunicaciones SNMP entre agente y gestores, diseño de la MIB de gestión de la información técnica, control de las comunicaciones con el manager X.73 y diseño de interfaz del agente y de interfaz de simulación de manager X.73. De esta forma, el agente propuesto permite la gestión de la información técnica de los dispositivos, dejando a elección del gestor la configuración de alarmas y eventos para controlar de una manera efectiva los problemas presentes en los dispositivos. También se deja al usuario la elección de los parámetros de seguridad del agente, así como el puerto y la interfaz de salida al sistema, permitiendo mantener un registro de usuarios para

configurar rápidamente los parámetros característicos del agente.

Para el almacenamiento de la información se ha diseñado e implementado una MIB privada. Esta MIB nos permite disponer de una estructura organizada en la que almacenar la información (datos técnicos, estados, datos de control, errores, alarmas, eventos y logs) necesaria para la gestión técnica de los dispositivos. La implementación física de la MIB se ha realizado mediante una base de datos MySQL. Ademas, para la organización interna de la misma se ha diseñado una estructura basada en la utilización de tablas que puede ser utilizada en la definición de una MIB genérica. Adicionalmente se han definido unos traps que permitirán al agente avisar en caso de recibir alguna medida anómala o la conexión de un nuevo dispositivo.

Para facilitar la configuración del agente por parte del usuario, se ha diseñado un interfaz desde el que se pueden establecer los parámetros de seguridad para la conexión de la versión 3 de SNMP. Además el interfaz permite tanto guardar como cargar datos de configuración establecidos previamente, lo que agiliza el proceso. Una vez establecida la seguridad, permite seleccionar puerto e IP por los que el agente va a estar escuchando, y lanzar y parar el agente mediante dos sencillos botones. Finalmente, y como funcionalidad añadida, permite iniciar la MIB para asegurarnos de que hay tabla creadas, o resetear la MIB.

Por último, se han realizado pruebas mediante simulación del sistema de gestión propueto para comprobar el alcance de la gestión del sistema y asegurar que todas las capacidades que el sistema se supone debe cumplir para realizar una gestión completa realmente se llevan a cabo de forma satisfactoria. Mediante la consecución de estas pruebas, se presenta un resumen de los resultados obtenidos que demuestra la eficacia del sistema de gestión de dispositivos médicos propuesto.

Podemos concluir que el agente desarrollado constituye un sistema integrado y unificado de gestión de información técnica en entornos de telemonitorización domiciliaria. Proporciona una gestión total de la información técnica que se transmite de los MDs al CE, y a su vez, al haber tenido en cuenta en el diseño el uso del estándar X.73 por parte de los MDs, contribuye a implementar una plataforma estándar donde la interoperabilidad y la capacidad de gestión global están garantizadas.

5.2 Líneas futuras

La telemonitorización de pacientes es un tema de gran interés en la actualidad por las posibilidades que ofrece, y por lo tanto las líneas de investigación que ofrece son numerosas.

Las posibles vías de desarrollo del proyecto de gestión de dispositivos médicos en entornos de gestión domiciliaria son las siguientes:

- Aplicar el sistema presentado a la problemática de gestión de información médica de los dispositivos médicos.
- Crear un gestor propio para facilitar y agilizar la gestión de la MIB creada en contrapartida de soluciones comerciales como el *MIB Browser* de MG-Soft.
- Aumentar la capacidad de gestión añadiendo parámetros nuevos de los dispositivos médicos y mejorando los traps creando traps nuevos más específicos.
- Aumentar la variedad de dispositivos médicos que el agente puede gestionar.

Bibliografía

[1] W.Stallings. "SNMP and SNMPv2: The Infraestructure for Network Management". *IEEE Communications Magazine*, 1998.

[2] W.Stallings. "SNMPv3: A Security Enhancement For SNMP". *IEEE Communications Surveys*, 1998.

[3] ISO/ IEEE11073. Health informatics Medical Devices Communication [P11073-20601. Application profile-Optimized protocol]. http://standards.ieee.org/. Primera edición: 2006.

[4] M.Rodríguez, R.Sanz, S.Galán, C.García, R.Chinchilla, A.Yela. "CORBA: Una plataforma software para el futuro".

[5] Shinkyung Lee, Namje Park, Dooho Choi. "Security Mangement System for Sensor Network". *International Conference of Convergence and Hybrid Information Technology 2008*, 2008.

[6] James Yu, Imad al Ajarmeh. "An Empirical Study of the NETCONF Protocol". *2010 Sixth International Conference on Networking and Services*, 2010.

[7] Jungho Seo, Jungil Heo, Suyoung Lim, Jinsoo Ahn, Wooshik Kim. "A Study on the Implementation of a Portable u-Healthcare System using SNMP and AODV". *Proceeding of the 29th Annual International Conference of the IEEE EMBS Cité Internationale, Lyon, France*, 2007.

[8] Yen Yang Lim, Marco Messina, Frank Kargl, Leena Ganguli, Martin Fisher, Tommy Tsang. "SNMP-Proxy For Wireless Sensor Network". *Fifth Intenational Conference on Information Technology: New Generations*, 2007.

[9] "WilocT: localización y trazabilidad en entornos sanitarios". *Reportaje de RFID Magazine*, 2009.

[10] U. Blumenthal and B.Wijnen. "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)". *Technical Report RFC 2574, Internet Engineering Task Force (IETF)*, 1999.

[11] B. Wijnen, R. Presuhn, and K. McCloghrie. "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)". *Technical Report RFC 2575, Internet Engineering Task Force (IETF)*, 1999.

[12] Framework SNMP4j. El API de SNMP orientada a objetos para agentes y gestores Java. Disponible en: http://www.snmp4j.org/.

[13] MIB Browser. Gestor comercial de *MG-Soft*. Disponible en: http://www.mg-soft.si/.

[14] K. McCloghrie, D. Perkins, and J. Schoenwaelder. "Structure of Management Information Version 2 (SMIv2)" *Technical Report RFC 2578, Internet Engineering Task Force (IETF)*, 1999.

[15] Borrador de comunicación de datos entre manager X.73 y agente Versión 1.6. *Communications Technologies Group (GTC), Universidad de Zaragoza*, 2011.

[16] N.Lasierra, A.Muñoz, A.Alesanco, J.Escayola, J.García. "SNMPv3-ISO/IEEE 11073 integration for technical management in home-based telemonitoring scenarios", 2011.

[17] S. Waldbusser. "Remote Network Monitoring Management Information Base" *Technical Report RFC 2819, Internet Engineering Task Force (IETF)*, 2000.

[18] K. McCloghrie, D. Perkins, and J. Schoenwaelder. "Structure of Management Information Version 2 (SMIv2)". *Technical Report RFC 2578, Internet Engineering Task Force (IETF)*, 1999.