

Modelo de Gestión de Redes incorporando la difusión Masiva de Información

Diego Marcos Jorquera¹, Francisco Maciá Pérez¹, Francisco José Mora Gimeno¹,
Juan Antonio Gil Martínez-Abarca¹, Iren Lorenzo Fonseca²

¹Universidad de Alicante. Departamento de Tecnología Informática y Computación
{dmarcos, pmacia, fjmora, gil}@dtic.ua.es

²Instituto Superior Politécnico José Antonio Echevarría.
Centro de Estudio de Ingeniería y Sistema
{ilorenzo}@ceis.cujae.edu.cu

Resumen. En el presente artículo se propone un modelo de gestión de redes que incorpora en su definición la gestión y difusión masiva de la información sobre redes no propietarias como Internet. Este modelo viene a solucionar uno de los problemas más comunes para cualquier aplicación, tanto de gestión como de usuario, proporcionando un mecanismo sencillo, intuitivo y normalizado para transferir grandes volúmenes de información, con independencia de su localización o los protocolos de acceso. Las principales áreas de aplicaciones del modelo se encuentran en: los sistemas de gestión como las copias de seguridad en red, sistemas para la continuidad en el negocio y alta disponibilidad o los sistemas de mantenimiento de redes de computadoras; aplicaciones de usuario como las aplicaciones *eBusiness*, sistemas de compartición de archivos o sistemas multimedia; y, en general, cualquier tipo de aplicación que conlleve la transferencia de importantes volúmenes de información a través de redes de área amplia como Internet.

1 Introducción

El concepto de gestión de redes aglutina al conjunto de aplicaciones, herramientas, protocolos y sistemas informáticos que ayudan a los administradores a realizar tareas de gestión, monitorización o implantación de dispositivos, servicios, aplicaciones o, en general, cualquier componente informático de una red [1].

Si bien inicialmente los primeros sistemas y estándares para la gestión de redes se responsabilizaban básicamente de la gestión de los elementos de networking (enrutadores, concentradores, servidores,...) cada vez son más los elementos que se gestionan, incorporando tanto las estaciones de trabajo y los dispositivos móviles que se conectan a la red, como los servicios y aplicaciones que se ejecutan en éstos [2, 3].

En la actualidad, la gestión y distribución de la información dentro de las redes es uno de los problemas a los que se enfrentan los administradores de sistemas. Realizar copias de seguridad, replicas de archivos y bases de datos o gestionar los sistemas NAS son algunos de las múltiples tareas asociadas a la información que en la actualidad deben ser contempladas en la gestión.

Otra característica relevante de los nuevos servicios electrónicos es que en muchas ocasiones la información es enviada de forma simultánea a más de un destinatario, como es el caso de la distribución de software o la replicación de contenidos multimedia o de bases de datos. En estos casos, los sistemas tradicionales, basados en las comunicaciones punto a punto, producen saturaciones en la red de comunicaciones debido a la necesaria retransmisión de la información por cada destinatario.

El incremento de los volúmenes de información que manejan actualmente las aplicaciones imposibilita que en campos de la informática tan importantes como la multimedia, los sistemas de almacenamiento distribuido o la gestión de redes se pueda seguir avanzando en tiempos y con costes aceptables. Los problemas para transferir grandes volúmenes de información entre los distintos nodos que conforman la red es una de las principales causas que provoca esta situación.

El incremento de los volúmenes de información que manejan actualmente las aplicaciones imposibilita que en campos de la informática tan importantes como la multimedia, los sistemas de almacenamiento distribuido o la gestión de redes se pueda seguir avanzando en tiempos y con costes aceptables. Los problemas para transferir grandes volúmenes de información entre los distintos nodos que conforman la red es una de las principales causas que provoca esta situación.

Si bien es cierto que se están aplicando soluciones parciales a este problema, el hecho de que la difusión de información no esté contemplada desde la propia concepción de los sistemas imposibilita que se pueda afrontar el problema de una forma global.

Dentro de todas estas aplicaciones, la propia gestión de redes también se mueve en la actualidad con estos volúmenes de información y es uno de los primeros problemas a resolver a la hora de abordar los nuevos retos que la informática plantea. Con los actuales sistemas de gestión de redes no se pueden asegurar los niveles de confiabilidad y continuidad en el negocio que requieren estos sistemas. En concreto, tareas de mantenimiento que requieran la transferencia de grandes volúmenes de información no se pueden realizar de forma integrada en el sistema de gestión y que éstas se realicen de forma eficiente y en tiempos acotados. Es por ello que el ámbito donde se centra el presente trabajo es el de la gestión de redes, donde se pretende aportar soluciones que permitan resolver la transmisión de grandes volúmenes de información.

Como hipótesis de partida de esta investigación se propone la incorporación de mecanismos para la gestión de la difusión masiva de información en los modelos de gestión de redes, lo cual posibilitará la propuesta, de manera sistemática, de arquitecturas y sistemas de gestión viables con los requerimientos actuales.

A partir de las técnicas actuales usadas en *grid computing* para la autogestión o gestión colaborativa y las usadas en los procesos de *streaming* de archivos y *multicast* para la difusión de información, que resuelven parcialmente el problema, se puede componer un modelo de difusión de información masiva a un conjunto hipotéticamente alto de destinatarios y que ésta se realice de forma eficiente y en tiempos acotados, a la vez que aporte escalabilidad al sistema y un bajo impacto sobre la red que administra.

2 Antecedentes

La gestión de red trata sobre la planificación, la organización, la supervisión y el control de los elementos que componen una red de comunicaciones. Los objetivos principales de la gestión de red consisten en garantizar la disponibilidad y mejorar el rendimiento de los elementos del sistema, así como incrementar su efectividad.

Debido al éxito que han tenido las redes de computadores en los últimos tiempos, en la actualidad, son consideradas una parte importante y estratégica de las empresas, industrias u otros tipos de organizaciones. Como consecuencia de las cada vez mayores dimensiones que están adoptando, resulta aun más importante su control y gestión con el fin de obtener la mejor calidad de servicio posible.

Uno de los principales problemas a los que se ha enfrentado la gestión de redes es la alta heterogeneidad de los elementos que componen las redes de computadores: dispositivos de *networking*, nodos de red, servicios, aplicaciones, etc., cada uno, además, de fabricantes distintos. Esto ha promovido la aparición de estándares abiertos con el fin de compatibilizar protocolos e información. De esta forma, durante la década de los noventa, se han ido desarrollando diversas iniciativas con el objetivo de ofrecer recomendaciones y estándares abiertos para tratar de dar solución a estas nuevas problemáticas, como es el caso de los protocolos de gestión SNMP o el CMIP [4].

Sin embargo hasta la aceptación de los actuales estándares surgieron multitud de propuestas para sistemas de gestión de redes promovidas por diversos fabricantes. Estas soluciones propietarias aportaban arquitecturas de gestión que eran adecuadas para los productos de dichos fabricantes pero con un bajo grado de interoperabilidad con otros sistemas, lo cual, unido a la creciente complejidad de las actuales redes de comunicaciones, los convirtió en sistemas en desuso [3].

Como alternativa a estos sistemas propietarios los estándares para la gestión de redes, fundamentalmente propuestas derivadas del estándar OSI propuesto por ISO que aportan una solución mucho más interoperable en las actuales complejas y heterogéneas redes de computadores.

A mediados de los 80 la *Organización Internacional para la Estandarización* (ISO) en colaboración con *Comisión Electrotécnica Internacional* (IEC) propusieron un modelo de red descriptivo denominado *Open System Interconnection* (OSI), como un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones [5]. Dentro de esta propuesta se establecía un marco para la creación de sistemas de gestión de redes.

Distributed Management Task Force (DMTF), una organización conformada por las más importantes compañías de la industria (AMD, Cisco, Dell, Fujitsu, HP, Hitachi, IBM, Intel, Microsoft, Novell, Oracle, Sun Microsystems,...), propuso *Common Information Model* (CIM), un esquema conceptual que permite modelar los elementos que componen un sistema TI. Este modelo representa los elementos como un conjunto de objetos y sus relaciones y usa para su definición UML [6]. CIM, además, permite integrarse con la información de otros modelos, como es el caso de SNMP. El objetivo de CIM es ser la base de información para otras tecnologías que se sustentan en él [7], como es el caso de los estándares WBEM, SMASH o SMI-S.

Web Services for Management o *WS-Management* es un estándar abierto que utiliza un protocolo basado en SOAP para la gestión de sistemas TIC [8]. En la actualidad el

estándar está gestionado por el DTMF. Además de realizar tareas de gestión como la lectura o escritura de objetos de gestión como base para la administración de un elemento, *WS-Management* permite la ejecución de métodos específicos para la gestión, permitiendo realizar tareas más sofisticadas que otras propuestas. Además, el hecho de estar basado en *Web Services* permite su integración con otros sistemas y su composición con otros servicios [9].

Web-Based Enterprise Management (WBEM) es un conjunto de tecnologías estándares para la gestión en Internet desarrollado para unificar la gestión de entornos distribuidos, facilitando el intercambio de información a través de plataformas heterogéneas. WBEM está fundamentado en CIM y utiliza como protocolos una implementación de CIM en XML (CIM-XML) y WS-Management. En la actualidad están surgiendo muchas propuestas para el uso de WBEM en la gestión de sistemas [10] [11], incluido la gestión de dispositivos embebidos [12].

Common Management Information Protocol (CMIP) es un protocolo de gestión de red fundamentado en el modelo OSI. Este protocolo ofrece un mecanismo de transporte que sustenta un conjunto de servicios implementados con el esquema *petición-respuesta*. El protocolo proporciona una comunicación entre las aplicaciones de gestión y los agentes situados en los dispositivos a gestionar. El MIB que utiliza CMIP está basado en objetos de gestión complejos que se relacionan para describir los elementos que son gestionados. CMIP aporta un conjunto amplio de operaciones de gestión y así como mecanismos de seguridad, autorización y control de acceso.

Propuesto por el IETF, *Simple Network Management Protocol* (SNMP) es el protocolo de gestión de redes más extendido en la actualidad y se ha convertido en el estándar más utilizado en Internet. Las primeras propuestas para la gestión de red en Internet aparecieron en 1987 con una serie de protocolos como el SGMP (*Simple Gateway Monitoring Protocol*), el HEMS (*High-Level Entity Management System*) y el CMOT (versión de CMIP sobre TCP). Estos protocolos de gestión se ocupaban básicamente de la gestión de los dispositivos de *internetworking* como *routers*. SNMP se fundamenta en la distribución de un conjunto de agentes en los dispositivos a gestionar y unos elementos gestores o NMS (*Network Management Systems*) que solicitan a los agentes la realización de las tareas de gestión. El protocolo utilizado entre estas dos entidades es un protocolo relativamente sencillo (de ahí el nombre de simple) y con pocas operaciones, básicamente una operación de lectura de objetos (GET), otra de escritura (SET) y otra de notificación (TRAP). El protocolo utiliza una serie de mensajes que, si bien estos fueron pensados para trabajar sobre UDP, por su concepción pueden integrarse sobre otros protocolos, lo cual permitió la expansión de SNMP.

2 Modelo de Gestión de Redes

Como punto de partida para la propuesta de un modelo de gestión se ha utilizado el marco de trabajo propuesto en el modelo de referencia básica de OSI [13] [5], marco utilizado por la mayoría de los sistemas de gestión de redes actuales. En esta propuesta se consideran diversos aspectos relacionados con la gestión de redes, incluyendo la gestión de elementos tan dispares como dispositivos de red,

aplicaciones, servicios, protocolos o los propios usuarios y administradores del sistema. Todos estos elementos conforman el **Entorno de Red** (*OSI Enviroment – OSIE*). Si bien este marco de trabajo ha sido utilizado en muchas propuestas de sistemas de gestión la principal novedad del presente trabajo es el hecho de contemplar no solo los elementos tradicionales de gestión de redes, sino también la información que se almacena y transfiere en el sistema, permitiendo realizar una gestión de la misma de forma integrada con el resto del sistema de gestión. Su inclusión en la propia definición del modelo permitirá que actividades de gestión que necesiten la difusión de información masiva puedan realizarse de forma integrada y eficiente al poder contemplar el resto de elementos que componen el sistema.

A lo largo de esta sección, y siempre que se haga referencia a términos descritos en el estándar OSI, se utilizará la nomenclatura utilizada por el ISO para clarificar los conceptos que se describen.

El objetivo principal de este trabajo es proponer un modelo de gestión de redes (*Network Management Model – NMM*) con capacidad de gestión de la información que denominaremos **NMM_{INF}**.

Dentro del modelo de gestión, como se indica en el modelo OSI, se pueden distinguir cuatro sub-modelos bien diferenciados que ayudan a desarrollar de forma separada y ordenada un sistema de gestión: un **Modelo de Organización** (*Organizational model – OM*) donde se describen los componentes que conforman la red (los elementos a administrar, el conjunto de gestores y agentes de gestión y la relación entre éstos; un **Modelo de Información** (*Information Model – IM*) que define la información asociada a la gestión de redes (*Management Information Base – MIB*) así como su estructura (*Structure of Management Information – SMI*); un **Modelo de Comunicación** (*Communication Model – CM*) que define la forma de comunicarse entre los gestores y los agentes de gestión; y un **modelo funcional** (**FM**) que establece las tareas a realizar por parte del sistema de gestión.

Esta separación hace que la descripción del modelo esté mucho más estructurada y permita su formalización de forma más clara y sistemática.

De manera formal podríamos definir nuestro modelo de gestión como la siguiente tupla:

$$NMM_{INF} = \langle OM, IM, CM, FM \rangle \quad [1]$$

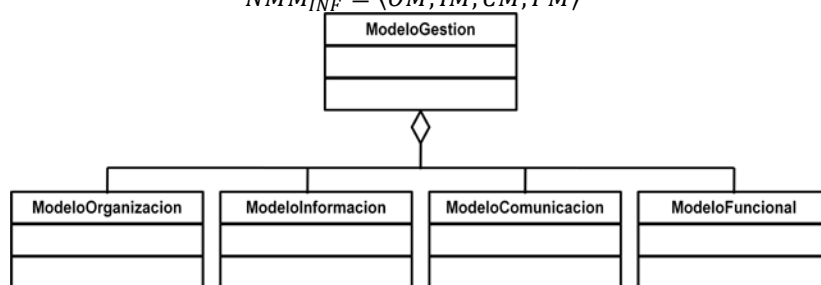


Fig. 1. Diagrama de agregación del modelo de gestión.

De manera formal, mediante la *Extensión de Eriksson y Penker* para UML podríamos definir el proceso de creación del *Modelo de Gestión* propuesto como se describe en la fig. 2.

En ciertos momentos, donde se requiera tomar decisiones de diseño, el modelo será instanciado en el protocolo SNMP ya que, como se vio en el capítulo 2, es el sistema de gestión actual más extendido y que además tiene las características de extensibilidad y flexibilidad necesarias para adaptarlo a nuestras necesidades.

A continuación se describen en profundidad cada uno de los cuatro sub-modelos presentados.

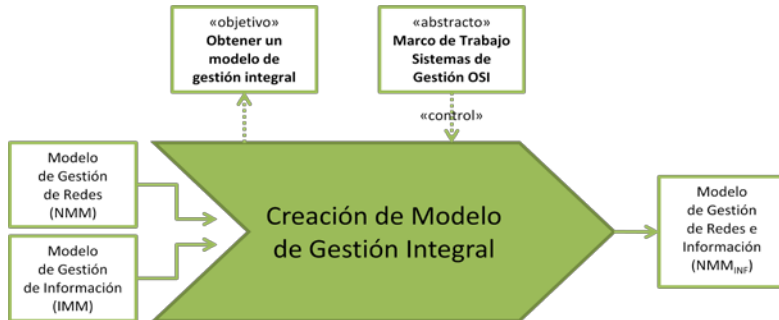


Fig. 2. Proceso de creación del Modelo de Gestión.

2.1 Modelo de Organización

En un entorno distribuido como las redes de computadores los procesos de gestión también se distribuyen entre los elementos del entorno.

Siguiendo el estándar OSI de gestión de redes se pueden diferenciar los siguientes elementos: un **Entorno de Gestión** (*OSI Enviroment – OSIE*), compuesto por el conjunto de dispositivos que conforman el sistema y la red de comunicaciones que los interconecta, un conjunto de **Entidades** (*System Management Application-Entity – SMAE*), responsables de la gestión del sistema, y un conjunto de protocolos de red (*Network Protocols – NP*)

$$OM = \langle OSIE, SMAE, NP \rangle \quad [2]$$

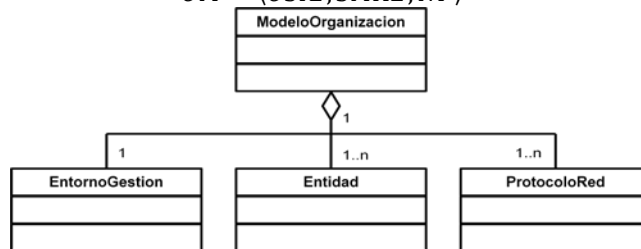


Fig. 3. Diagrama de agregación del Modelo de Organización.

El entorno de gestión refleja la parte más física y tangible del modelo de organización. Principalmente está compuesto por el conjunto heterogéneo de **Dispositivos a Gestionar** (*Management Devices – MD*) que aglutina los distintos elementos que conforman una red de computadores: servidores, computadoras personales, dispositivos móviles, routers o cualquier otro dispositivo con capacidad de

comunicación. En nuestro sistema estos dispositivos son el objetivo final de las tareas de gestión.

$$MD = \{md_1, md_2, \dots, md_n\} \quad [3]$$

En el sistema también se distinguen otros dispositivos que no están sujetos a ningún tipo de mantenimiento, o que al menos no están contemplados como objeto de gestión en nuestro sistema. Sin embargo estos dispositivos también forman parte del entorno de gestión, bien porque dan soporte a algún elemento del mismo bien porque facilitan alguna de las tareas del sistema de gestión como es el caso de *routers*, *firewalls* o cualquier otro dispositivo de *networking*. A estos dispositivos los denominaremos **Dispositivo de Soporte** (*Support Device – SD*).

$$SD = \{sd_1, sd_2, \dots, sd_n\} \quad [4]$$

Al conjunto total de dispositivos que conforman el entorno de gestión lo denominaremos **Dispositivos de Red** (*Network Devices – ND*).

$$ND = MD \cup SD \quad [5]$$

Evidentemente también existen otros elementos que interactúan con los dispositivos de red: usuarios, aplicaciones externas, administradores, etc. Estos elementos son considerados externos al sistema de gestión y no forman parte en si del entorno de gestión.

Para interconectar todos los elementos del sistema existirá una red de comunicaciones que denominaremos **Red de Computadores** (*Network – N*). La red la formalizaremos como un pseudografo dirigido (un grafo dirigido que acepta bucles en un nodo) y etiquetado compuesto por un conjunto de nodos que conformarán los *Dispositivos de Red* (ND), un conjunto de **Aristas** (E) que unirán estos nodos, el conjunto de **Etiquetas** posibles (C) y una **Función de Etiquetado** (p).

$$N = (ND, E, C, p) \quad [6]$$

$$E \subseteq ND \times ND \quad [7]$$

Las aristas del grafo no tienen por qué corresponderse con interconexiones físicas entre los nodos. Una arista representa que es posible comunicar los dos extremos de la misma en una dirección determinada independientemente de la infraestructura de red que exista por debajo.

La ponderación del grafo establece las características que cumplen cada una de las aristas, es decir, propiedades como ancho de banda, velocidad de transferencia, latencia, etc. Cada una de esas características se definirá como:

$$C_i = \{c_1^i, c_2^i, \dots, c_n^i\} \quad [8]$$

Por lo tanto las propiedades de cada arista del grafo vendrán definidas por el producto cartesiano de las características.

$$C = C_1 \times C_2 \times \dots \times C_n \quad [9]$$

De esta forma la ponderación es una función que asocia cada arista con un conjunto de características (fig. 4).

$$p: E \rightarrow C \quad [10]$$

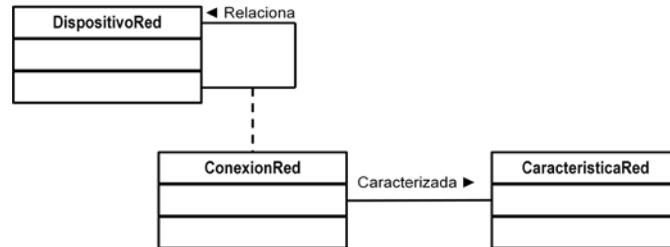


Fig. 4. Diagrama de relación de la red de computadores.

Para llevar a cabo la administración del sistema existirá un conjunto de *Entidades* que trabajarán en conjunción para realizar las tareas de administración. Las *Entidades* conforman un sistema distribuido que es la base fundamental del modelo de gestión. A continuación se describen los diferentes tipos de entidades que se distinguen en nuestro modelo.

Cada uno de los dispositivos a gestionar tendrá asociado un agente específico para su gestión que llamaremos **Entidad de Gestión** (*Management Entity – ME*). Las *Entidades de Gestión* son las responsables finales de llevar a cabo las tareas de mantenimiento dentro de cada uno de los dispositivos que gestionan, así como de recabar la información necesaria del dispositivo y, en caso de ser necesario, lanzar alertas cuando se identifica un problema.

$$ME = \{me_1, me_2, \dots, me_n\} \quad [11]$$

$$\forall me \in ME \exists md \in MD \equiv Gestiona(me, md) \quad [12]$$

Si bien se podría considerar la posibilidad de que en un mismo dispositivo existieran varias *Entidades de Gestión*, cada una de ellas llevando a cabo tareas de gestión específicas, para simplificar el sistema vamos a considerar que conceptualmente ambos se comportan como una única entidad que aglutina la suma de todas las tareas.

El comportamiento de las *Entidades de Gestión* es principalmente reactivo, es decir, reciben peticiones de otras entidades, llevan a cabo la tarea de gestión encomendada y devuelven el resultado de la misma. Complementariamente también pueden estar monitorizando alguna variable del dispositivo a gestionar y, cuando se cumple alguna circunstancia determinada, emite una alerta a alguna otra entidad.

Por otro lado en el sistema también cuenta con gestores o **Entidades de control** (*Control Entity – CE*) que se encarga de solicitar a las diferentes *Entidades de Gestión* del sistema que lleven a cabo las tareas de mantenimiento. Es en este elemento donde se suelen centrar las labores de coordinación y planificación del mantenimiento y suele ser el punto de acceso al sistema de gestión para los administradores.

$$CE = \{ce_1, ce_2, \dots, ce_n\} \quad [13]$$

Para su funcionamiento la *Entidad de Control* residirá en un *Dispositivo de Soporte* que le provea de un adecuado entorno de ejecución.

$$\forall ce \in CE \exists sd \in SD \equiv DaSoporte(sd, ce) \quad [14]$$

Una *Entidad de Control* solicitará realizar tareas de gestión a las *Entidades de Gestión* que se encuentran distribuidas por el sistema. Estas operaciones son iniciadas siempre por la *Entidad de Control* (parte activa) y son servidas por las *Entidades de*

Gestión (parte pasiva). Las *Entidades de Control* también son las responsables de recibir y procesar las alertas emitidas por las *Entidades de Gestión*. En este caso el rol es distinto, siendo la *Entidad de Gestión* la parte activa de la operación. Si bien éste es el esquema general de operación del sistema de gestión también puede existir algún tipo de delegación tanto a nivel de *Entidad de Gestión* como de *Entidad de Control* (ver fig. 5). Esto permitiría la concepción de sistemas de gestión más sofisticados donde podría darse algún tipo de relación jerárquica entre las diferentes entidades que lo componen.

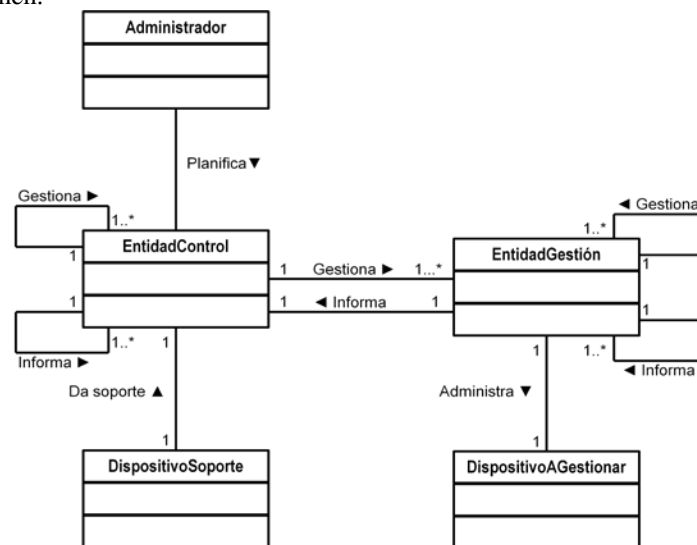


Fig. 5. Relación entre entidades de control y de gestión.

Esta distribución del sistema de gestión, donde tenemos entidades de gestión situadas en los dispositivos a gestionar y entidades de control que supervisan y controlan a las primeras, es una arquitectura muy extendida en los sistemas de gestión actuales. Por ejemplo SNMP utiliza este modelo llamando *Agentes de Gestión* a las *Entidades de Gestión* y *Sistemas Administradores de Red* (NMS) a las *Entidades de Control* [14].

Dado que el modelo de gestión que se propone tiene que contemplar la gestión no solo de los dispositivos, aplicaciones y servicios que conforman un sistema, sino también la información que estos utilizan, incorporaremos el concepto de **Información** (*Information – I*) dentro del modelo de organización. Para ello definiremos la información como el conjunto de los diferentes paquetes de datos que pueden existir en el sistema. Estos paquetes pueden ser archivos, recursos de información, paquetes software, paquetes de configuración y, en general, cualquier elemento de información que puede ser gestionado, transferido y almacenado de forma independiente.

$$I = \{i_1, i_2, \dots, i_n\} \tag{15}$$

Tradicionalmente las entidades del sistema de gestión no contemplaban la gestión y transferencia de la información en su concepción, de forma que si en el desarrollo de una tarea de gestión se necesitaba un paquete de información determinado, se

utilizaba un sistema de transporte externo al sistema de gestión para obtener dicho paquete. Además el sistema de gestión debía conocer a priori dónde residía la información y con qué protocolo podía ser accedido.

En nuestro modelo daremos a las *Entidades de Gestión y Control* la posibilidad de poder transferir de forma integrada en el sistema de gestión esta información. Es en este punto donde radica la principal aportación del modelo ya que al hacer esta transferencia de forma integrada conseguiremos las siguientes ventajas.

- Al realizar la transferencia de la información a nivel de gestión no será necesario realizar esta tarea a nivel de aplicación. Por ejemplo si necesitamos replicar la información que utiliza un servidor web no será necesario utilizar ninguna aplicación de réplica adicional, el propio sistema de gestión podrá realizar esta tarea.
- Al tener las entidades de gestión distribuidas en nuestro entorno de gestión permite potencialmente el acceso a la información que se encuentre en cualquier parte del sistema. Gracias a esto una entidad podrá obtener un paquete de información siempre que se encuentre accesible por otra entidad de gestión.
- El carácter distribuido del sistema de gestión hace que la transferencia de la información pueda realizarse de forma colaborativa. Con ello todas las entidades que soliciten la información podrán colaborar para optimizar el proceso de transferencia, es más, incluso elementos que no requieran dicha información podrán colaborar en la propia transferencia.

Por lo tanto, en nuestro sistema de gestión tanto las entidades de gestión como las de control tienen la capacidad de solicitar la transferencia de estos paquetes de información. Dado que es posible que existan nodos en el sistema que contengan información pero que no tengan asociado una entidad de gestión, definiremos las **Entidades de Información** (*Information Entity – IE*) que serán las responsables de gestionar y suministrar la información en estos casos. Se trataría de entidades asociadas a: servidores web, servidores ftp, servidores de ficheros, servidores de contenido multimedia, etc.

Si bien las *Entidades de Información* dentro de nuestro modelo tienen como única labor la de gestionar la información, está también podrá ser gestionada por las *Entidades de Gestión* y las *Entidades de Control*. Por lo tanto un paquete de información podrá estar accesible y podrá ser suministrado por cualquiera de estas entidades o, incluso, por más de una si existen réplicas de este paquete.

Las *Entidades de Información* no tienen por qué residir en el mismo dispositivo donde se almacenan los paquetes de información. Podrán estar sustentadas por *Dispositivos de Soporte* y gestionar la información que reside físicamente en otros dispositivos.

En este sistema distribuido donde pueden existir multitud de *Entidades* y de *Paquetes de Información* se hace necesario incluir algún tipo de sistema de descubrimiento que permita conocer a cada uno de las entidades cuales son las otras entidades del sistema, así como que paquetes de información existen, donde se ubican y mediante qué protocolo puede descargarse. De esta forma se podría desacoplar todas las partes del sistema dando una visión unificada de las *Entidades* y los *Paquetes de Información* a todos los elementos.

Para llevar a cabo el descubrimiento en nuestro modelo utilizaremos unas **Entidades de Registro** (*Registry Entity – RE*) que nos permitan llevar a cabo las labores de registro y de búsqueda de entidades e información.

$$RE = \{re_1, re_2, \dots, re_n\} \quad [16]$$

Las *Entidades de Registro* actúan básicamente como repositorios donde el resto de entidades son responsables de registrarse con la información necesaria para su localización. Así mismo, las entidades que tengan información disponible, incluidas las *Entidades de Información*, también registrarán dicha información con su localización y la información necesaria para su acceso. Si bien el registro puede ser visto como un lugar centralizado donde almacenar información de localización de los elementos del sistema, para ganar en escalabilidad y tolerancia este registro puede ser replicado o distribuido entre diferentes entidades de registro (ver fig. 6).

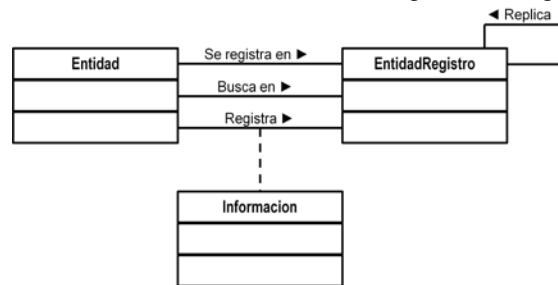


Fig. 6. Entidad de Registro.

Por lo tanto el conjunto total de entidades que conforman el sistema serán:

$$SMAE = ME \cup CE \cup IE \cup RE \quad [17]$$

En la fig. 7 se puede ver el diagrama de clases para las entidades del sistema.

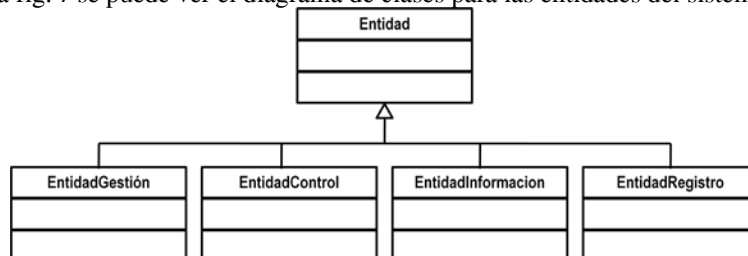


Fig. 7. Diagrama de clases de las entidades del sistema.

La comunicación entre las distintas entidades del sistema se realizará mediante el uso de diversos **Protocolos de Red** (*Network Protocols – NP*). Una vez vistos los diferentes tipos de entidades del sistema y la relación entre ellos en el modelo se distinguirán tres tipos distintos de protocolos: **Protocolos de Gestión de Redes** (*Network Management Protocols – NMP*), **Protocolos de Transporte** o transmisión de datos (*Transport Protocols – TP*) y **Protocolos de Descubrimiento** (*Discovery Protocols – DP*). Los protocolos serán descritos posteriormente en el *Modelo de Comunicación*.

$$NMP = \{nmp_1, nmp_2, \dots, nmp_n\} \quad [18]$$

$$TP = \{tp_1, tp_2, \dots, tp_n\} \quad [19]$$

$$DP = \{dp_1, d, \dots, dp_n\} \quad [20]$$

$$NP = NMP \cup TP \cup DP \quad [21]$$

En la fig. 8 se muestra un diagrama resumen con el esquema general del *Modelo de Organización*. Se ha destacado aquellos elementos fundamentales que caracterizan al modelo por su capacidad de gestión de la información.

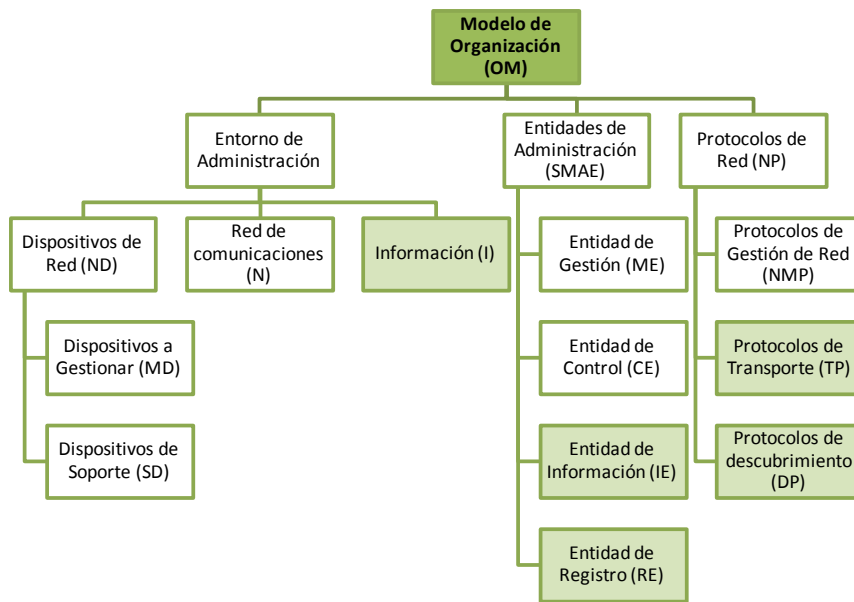


Fig. 8. Esquema general del Modelo de Organización.

2.2 Modelo de Información

Dado que el propio sistema de gestión de redes es en sí un sistema distribuido la información que gestiona también se distribuye entre los distintos elementos que conforman el sistema.

Para ello cada una de las entidades que conforman el sistema tendrá asociada una **Base de Información de Gestión** (*Management Information Base – MIB*). En los MIB se representa la información referente a la gestión que cada uno de las entidades necesita.

Dentro de cada *Dispositivo a Gestionar* se pueden distinguir diferentes **Objetos de Gestión** (*Management Object – MO*). Cada uno de estos objetos representa un elemento de la realidad. Además de los propios dispositivos de red que el agente está gestionando el MIB estaría compuesto por todo tipo de objetos como aplicaciones, servicios, usuarios, equipos, datos, etc.

De esta manera se podría realizar una gestión integral de la red contemplando cualquier elemento. Es importante destacar que en nuestra propuesta los datos en sí

(archivos, flujos de información, configuraciones, etc.) también van a ser contemplados como objetos a gestionar. Esto, a diferencia de los modelos de gestión tradicionales, nos permitirá incluir la propia gestión de la información (almacenamiento, transporte, tratamiento o difusión) como parte de la gestión de la red.

Los *Objetos de Gestión* son elementos fundamentales en el *Sistema de Gestión*. Nos permiten modelar los *Dispositivos a Gestionar* de forma que las entidades del sistema ven a éstos como una colección de *Objetos de Gestión*. Una vez modelados los dispositivos, aplicaciones, servicios y la información, las tareas de gestión se resumen en realizar acciones sobre los *Objetos de Gestión*. De esta manera los *Objetos de Gestión* nos abstraen de los detalles reales que existen en los sistemas que gestionamos, ofreciendo una visión unificada de los objetos reales que hay por detrás. Por ejemplo un objeto puede modelar una NIC de un dispositivo, de forma que el *Objeto de Gestión* que lo modela refleje información referente a estadísticas de paquetes enviados y recibidos por esta NIC, sin importar que modelo de NIC sea. La agrupación de todos los *Objetos de Gestión* conforma el MIB de la entidad. El MIB para cada *Dispositivo a Gestionar* encapsula y actúa como interfaz para realizar las tareas de gestión sobre él, como se puede ver en la Fig. 9.

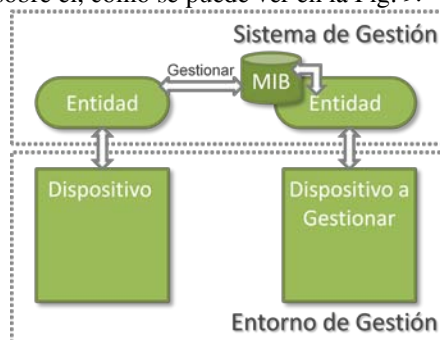


Fig. 9. MIB de las entidades del sistema de gestión.

El MIB no es una base de datos al uso, que almacena y gestiona los datos que contiene, es una representación formal de los datos de gestión que virtualmente asocia un objeto del MIB a un objeto de la realidad. De esta forma un objeto del MIB podría ser el estado de un dispositivo, por ejemplo si está apagado o encendido, y cambiando dicho estado podríamos apagar el dispositivo.

Para conseguir la compatibilidad con SNMP que indicamos anteriormente partiremos de la definición realizada en este estándar del MIB. Este MIB es lo suficientemente flexible para incorporar los elementos nuevos que deseamos integrar en nuestro modelo.

La forma en que se formaliza el MIB viene definida por la **Estructura de la Información de Gestión** (*Structure of Management Information – SMI*).

En SNMP el MIB sigue una estructura de directorio organizado en forma de un árbol jerárquico donde los nodos de este árbol son los *Objetos de Gestión*. Cada uno de estos objetos tendrá asociado un nombre único que lo identificará dentro de cada subárbol, es decir no podrán existir dos objetos con el mismo nombre que cuelguen del mismo objeto padre. En SNMP el nombrado de cada objeto puede ser textual o

numérico, siendo el primer hijo de cada padre el objeto 0. De esta manera los objetos se podrán identificar indistintamente por su nombre o por su número.

El árbol que conforma el MIB de SNMP es extensible, es decir, cada entidad tiene la capacidad de expandir alguna de las ramas del árbol con nueva información.

En la fig. 10 se muestra la estructura raíz básica del MIB estándar utilizado por SNMP [15]. Normalmente se utiliza el nodo *mgmt* para extender el MIB base con información sobre gestión. El nombre completo de este nodo sería *iso.org.dod.internet.mgmt* usando una notación donde se nombran todos los nodos del camino, separándolos por puntos. Alternativamente el nodo *mgmt* también se puede nombrar como 1.3.6.1.2, usando la notación numérica. A este nombre único de cada objeto se le conoce como **Identificador de Objeto** (*Object Identification – OID*).

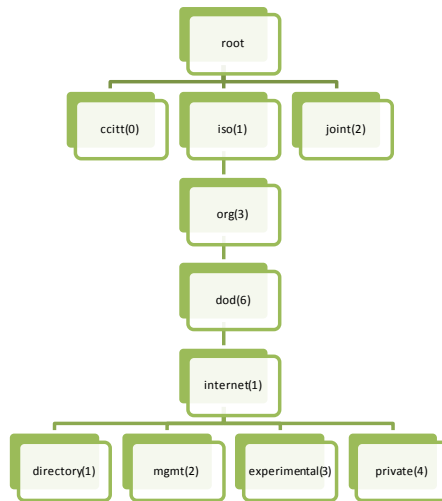


Fig. 10. Esquema general del MIB de SNMP.

Otras zonas del árbol normalmente utilizada en SNMP son la rama *experimental* (1.3.6.1.3) donde se sitúan propuestas no estándares que están aún en proceso de validación y la rama *private* (1.3.6.1.4) de donde cuelgan los MIBs propuestos por diversas empresas privadas para dar soluciones de gestión específicas para sus productos.

Dentro de la rama *mgmt*, donde se recogen los estándares, destaca el MIB más utilizado en la actualidad por la mayoría de los sistemas que incorporan SNMP. Se trata del subárbol definido como el estándar MIB-II [16]. Sus principales ramas se pueden ver en la fig. 11.



Fig. 11. Esquema general del MIB-II de SNMP.

El estándar MIB-II está soportado por la mayoría de los dispositivos que incluyen SNMP y recoge datos referentes a interfaces, protocolos, sistema, etc., datos que normalmente son comunes a todos los *Dispositivos a Gestionar*.

En el MIB cada objeto es de un tipo determinado. El tipo indica la sintaxis que se aplica a dicho objeto. Los tipos de datos que incluirá nuestro modelo son los soportados por SNMP. Estos tipos son un subconjunto de los definidos en el estándar ASN.1 [17].

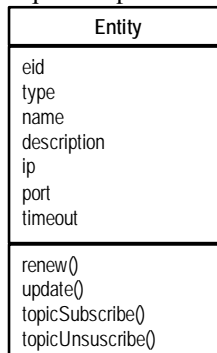
Los objetos definidos en el MIB pueden ser de *lectura*, si solamente pueden ser consultados o de *lectura/escritura* si estos pueden ser además modificados. Además, cada atributo puede ser *opcional* u *obligatorio* en función de si siempre es requerido o no.

Según lo que hemos visto en el *Modelo de Organización* se han analizado un conjunto de nuevas características y funcionalidades que se han incorporado a nuestro *Modelo de Gestión* para que el sistema pueda incorporar la gestión y transferencia de la información. Para poder reflejar esto dentro del modelo de información de SNMP a continuación se van a definir un conjunto de extensiones en forma de MIBs donde se refleje la información asociada al registro y a los paquetes de información.

Para gestionar el descubrimiento se propone la creación de un MIB específico que denominaremos **MIB-Registry**. Este MIB que será expuesto por las *Entidades de Descubrimiento*, tiene como objetivo representar los datos necesarios para realizar la publicación y descubrimiento de entidades e información.

Formalmente y de forma independiente del sistema de gestión seleccionado para el modelo, definimos el MIB de descubrimiento como un conjunto de objetos que nos van a permitir gestionar el registro de entidades e información.

En la fig. 12 observamos la definición en UML de la clase *Entity*. En ella se recogen los campos y operaciones que componen cada una de estas entidades.

**Fig. 12.** Definición de la clase *Entity*.

En la tabla 1 se puede ver los diferentes campos que componen una entidad en el sistema de registro.

Cada entidad estará identificada de forma única por un **Identificador de Entidad** (*Entity ID* – **EID**). Además por cada entidad se podrá establecer un nombre y una descripción de la misma. Para su localización también se registrará la información de

la dirección IP y el puerto en el que estará trabajando la entidad en cuestión. Por último, cada entidad tendrá asociada un *timeout* de registro, cumplido el cual la entidad será eliminada del mismo. Con esto evitamos que entidades que desaparecen de forma anómala del sistema, por ejemplo debido a un apagado del equipo donde residen, no permanezcan indefinidamente registrados en el sistema. La entidad por lo tanto es responsable, no sólo de registrarse en una entidad de registro, sino también de renovar periódicamente dicho registro. En cualquier caso, antes del cumplimiento del *timeout*, si la entidad de registro está configurado para ello, podrá enviar una alerta a la entidad en cuestión para que esté renueve su *timeout*.

Tabla 1. Definición de los campo de una entidad en el registro.

Nombre	Tipo	Descripción
eid	INTEGER	Identificador único de entidad.
type	INTEGER	Tipo de entidad: <ul style="list-style-type: none"> • 1 – Entidad de gestión • 2 – Entidad de control • 4 – Entidad de información • 8 – Entidad de registro
name	OCTET STRING	Nombre de la entidad.
description	OCTET STRING	Descripción de la entidad.
ip	IpAddress	Dirección IP de la entidad.
port	INTEGER	Puerto de la entidad donde reside el sistema de gestión.
timeout	INTEGER	Segundos pendientes para la permanencia de la entidad en el registro. El valor 0 indica que no caduca nunca.

En la fig. 13 se puede ver la definición de la clase *EntitySet*, que representa el conjunto de entidades registradas. Se trata básicamente de una lista de entidades sobre las que podemos operar para publicar o consultar.

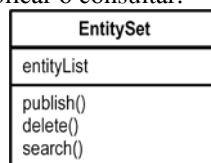


Fig. 13. Definición de la case *EntitySet*.

De forma similar en el registro se gestionan los objetos que representan a los paquetes de información del sistema. La fig. 14 ilustra la clase *Information*.

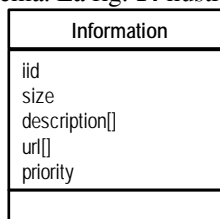


Fig. 14. Definición de la clase *Information*.

De cada información del sistema tendremos registrados los datos identificados en la tabla 2.

Tabla 2. Definición de los campos de un paquete de información en el registro.

Nombre	Tipo	Descripción
iid	INTEGER	Identificador único de paquete de información
size	INTEGER	Tamaño del paquete. Si no se conoce 0.
description	OCTET STRING	Descripción de la información. Pueden existir varias.
url	OCTET STRING	URL para el acceso a la información. Pueden existir varias.
priority	INTEGER	Prioridad o importancia del paquete.

Cada paquete de información tendrá un **identificador de información** (*Information ID* – **IID**). Dado que este identificador tiene que ser único para cada paquete, éste podrá ser calculado a partir de una función resumen de los datos contenidos en el paquete. De esta forma dos entidades que tengan el mismo paquete de datos lo identificarán con el mismo IID independientemente de si la información es un archivo, un registro almacenado en una base de datos o un flujo de información emitido por un servidor de *broadcasting*. De esta forma dos paquetes de información serán iguales si y sólo si el contenido de los mismos es exactamente igual. Para garantizar esto muchas propuestas utilizan funciones de resumen que, a partir del contenido de los datos, obtiene un identificador asociado de forma que la función resumen asegura que la probabilidad de que dos *Paquetes de Información* distintos tengan el mismo resumen sea prácticamente nula.

La descripción de la información (campo *description*) son textos que ayudan a identificar el contenido de la información. Diversas entidades pueden aportar descripciones distintas y complementarias de una misma información, por lo que descripción será un vector de múltiples descripciones. Las descripciones podrían ser, por ejemplo, los distintos nombres de archivo que puede tener una misma información en cada equipo donde reside.

De forma similar tendremos un vector de URLs que nos indicará los distintos protocolos y direcciones de acceso que podemos tener para una misma información. La URL especifica la información necesaria (protocolo, direcciones, puertos, rutas, nombres, etc.) para que una entidad pueda localizar y transferir una información determinada.

El conjunto de Paquetes de Información registrados estará reflejado en la clase *InformationSet* (ver fig. 15).

InformationSet
informationList
publish() delete() search()

Fig. 15. Definición de la clase *InformationSet*.

Cada entidad es responsable de mantener el conjunto de paquetes de administración que gestiona. Por lo tanto, cada paquete de información estará asociado a todas las entidades que lo gestionan y será dado de baja del sistema cuando todas estas entidades se hayan dado de baja en el sistema.

Una de las principales características de las que se quiere dotar al sistema es que en el propio modelo de gestión se recoja la distribución de la información en el sistema, de forma que la información se pueda replicar, mover o eliminar en función de las necesidades cambiantes del sistema.

En este sentido, cuando una entidad registra un paquete de información éste puede ser interesante a otras entidades. Para conseguir esto se proponen dos enfoques diferentes. En uno de estos enfoques, basado en *polling* o consulta, las entidades estarían continuamente consultando el registro de información para determinar cuándo hay una nueva entrada de interés. Este enfoque es poco escalable y podría saturar el sistema de registro cuando tenemos un alto número de entidades consultando o un gran número de paquetes de información en el sistema. Otro enfoque, mucho más adecuado para nuestra propuesta, es un modelo basado en eventos en el cual cuando una entidad registra un nuevo paquete de información, el propio registro es responsable de avisar mediante avisos a las entidades interesadas en recibir notificaciones.

En ambos enfoques una de las cuestiones que hay que determinar, principalmente si se quiere ganar en escalabilidad, es acotar tanto las consultas como los avisos generados definiendo de alguna forma cual es el interés de las entidades en la información.

En nuestro caso vamos a modelar estos intereses mediante un conjunto de temas o categorías (*topics*). Cada paquete de información podrá estar asociado a un conjunto ilimitado de categorías, de forma que cada categoría representa un área de interés para las entidades. Por ejemplo un paquete de actualización para el sistema operativo *Microsoft Windows XP* podría estar suscrito a las categorías: *software*, *actualización* y *windowsxp*. En este caso los administradores serían los responsables de asociar la información a las categorías adecuadas.

Esta categorización de la información haría que las entidades pudieran localizar de forma más eficiente la información que buscan. De esta manera en sus consultas las entidades podrán filtrar la información en función de las categorías que requieran. Por ejemplo los equipos con sistema operativo *Windows* podrían estar interesados en toda información etiquetada en la categoría *windows*, y un repositorio de *software* en toda información etiquetada con *software*.

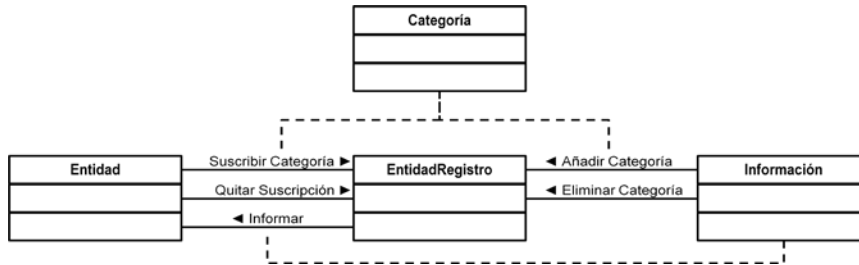


Fig. 16. Relación entre las distintas clases del registro.

Para el caso de un modelo basado en eventos las entidades, al registrarse, tendrían que suscribirse a las categorías en las que tengan algún interés. De esta forma, cuando se registra un nuevo paquete de información en el registro, el propio registro enviaría un aviso a todas las entidades suscritas a alguna categoría a la que perteneciera dicho paquete de información (fig. 16).

En este esquema las categorías actúan como un eje central que permite relacionar entidades e información en función de sus intereses (ver fig. 17).

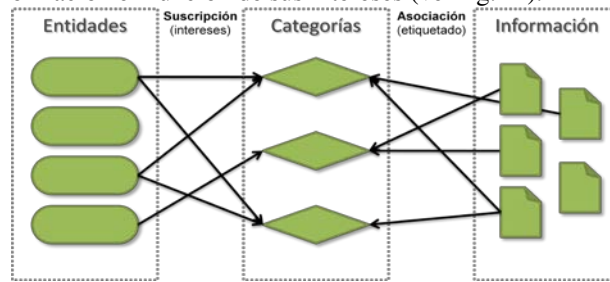


Fig. 17. Esquema de la relación entre entidades, información y categorías.

En algunos casos existirán entidades que estarán interesadas en todos los paquetes de información, por ejemplo repositorios generales de información o una entidad de registro secundaria. Para estos casos existirá una categoría virtual que denominaremos *global* y que estará asociada de forma automática a todos los *Paquetes de Información*. De esta forma, cuando una entidad se suscriba a la categoría *global* recibirá un aviso con la publicación de todos los paquetes de información. El uso de la suscripción a esta categoría deberá ser utilizado con medida ya que podría comprometer la escalabilidad del sistema de registro.

En el modelo de distribución de la información, el otro aspecto que permite guiar la gestión de los *Paquetes de Información* es la *prioridad*. Al registrar una entidad un *Paquete de Información* indica cuál es la prioridad de dicho paquete. Mediante este valor otra entidad puede decidir si transfiere o no dicho paquete. Por ejemplo un repositorio de software, al detectar la presencia de un nuevo *Paquete de Información* de interés, decidirá que lo transfiere sólo en el caso de que la prioridad supere un determinado umbral. En otros casos la prioridad también podrá ser utilizada para que, en caso de no disponer de más espacio de almacenamiento, un repositorio descarte los paquetes cuyo prioridad sea inferior, y así poder obtener y almacenar paquetes de prioridad más alta.

Dado que vamos a instanciar nuestro modelo utilizando el MIB de SNMP hemos realizado un proceso de adaptación de los objetos de registro para conformar el *MIB-Registry*. El proceso puede verse ilustrado en la fig. 18.

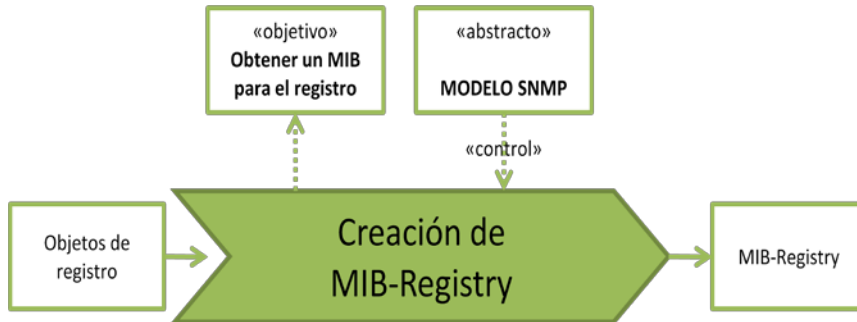


Fig. 18. Proceso de creación del MIB-Registry.

Básicamente tendrá tres ramas una donde residirán objetos que permitirán la gestión de la propia entidad de descubrimiento (**management**), otra donde se gestionaran las entidades (**entity**) y una última donde se gestionaran los paquetes de información (**information**). Dentro de las dos últimas ramas cada uno de ellas se desglosará en dos ramas, una para la publicación (**publish**) y otra para la consulta (**inquiry**). Este esquema se desglosa en la fig. 19. El subárbol de descubrimiento estará referenciado en el MIB-II de SNMP dentro de la rama *experimental*.

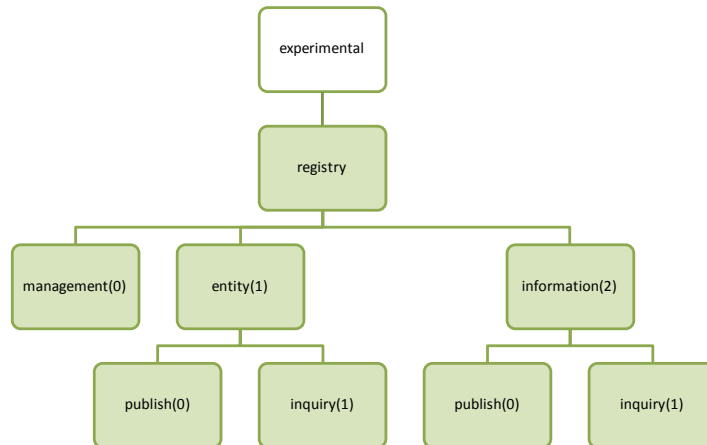


Fig. 19. Esquema general del MIB-Discovery.

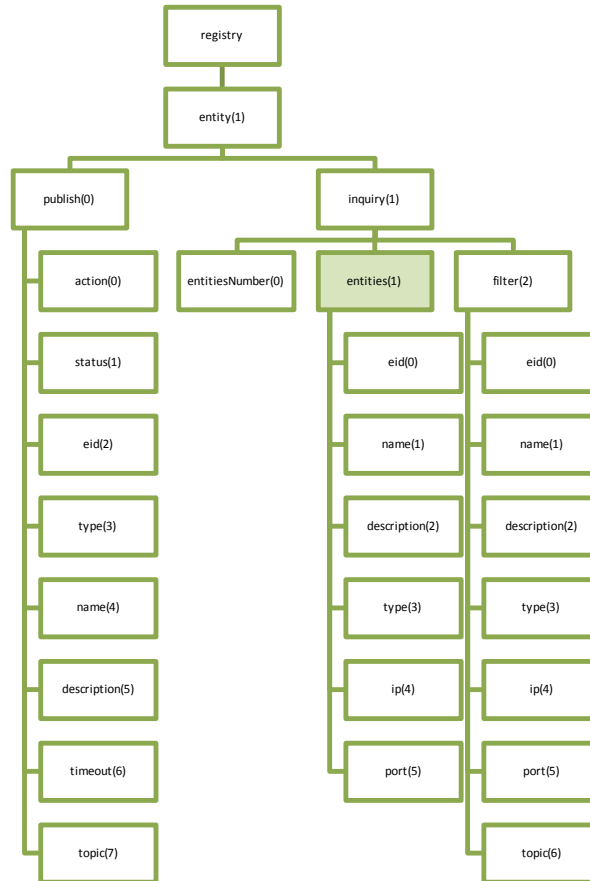


Fig. 20. Rama *entity* del subárbol *registry*.

En la rama *entity* (fig. 20) encontraremos la información necesaria para gestionar la publicación y consulta de las diferentes entidades que se encuentran en el sistema.

Dentro de la rama *entity*, la rama *publish* aporta los objetos necesarios para realizar la publicación de una entidad. Para ello la entidad que desea publicarse deberá establecer los datos de registro en los diversos objetos que se encuentran en la rama *publish*. Una vez establecido dichos valores se establecerá el valor del objeto *action* a 1, lo cual materializará la publicación de la entidad. Al ser la propia entidad la que se auto-registra, no será necesario indicar la dirección IP y el puerto que se utilizará para la publicación ya que estos datos podrán ser obtenidos del mismo proceso de comunicación.

El campo *action* nos permite además realizar otras operaciones sobre el registro: eliminar entradas, actualizarlas o renovar *timeout*. La tabla 3 resume estas acciones. En la tabla se indica qué campos son obligatorios y cuales son optativos o ignorados.

Tabla 3. Operaciones de gestión de entidades.

Acción	Valor	Descripción	Campos					
			eid	type	name	description	timeout	topic
None	0	Sin acción.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
New	1	Nueva entrada.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete	2	Elimina una entrada. La entrada es identificada por el EID.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Update	3	Actualiza una entrada. La entrada es identificada por el EID.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Renew	4	Renueva el <i>timeout</i> de una entrada. La entrada es identificada por el EID.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TopicAdd	5	Subscribe la entidad a una categoría de información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TopicDel	6	Quita la subscripción de la entidad a una categoría de información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

■ Campo obligatorio □ Campo opcional — Campo ignorado

Una vez se establezca el campo *action* a un valor, la acción correspondiente será realizada, y posteriormente el valor *status* podrá ser consultado para comprobar el resultado de la acción. Un valor 0 indicará que la operación ha sido realizada con éxito. Cualquier otro valor indicará un error en la misma. En la tabla 4 se puede ver los diferentes valores que puede tomar *status*, en función de la acción que se realiza.

Tabla 4. Códigos de resultado de las operaciones sobre entidades.

Valor	Descripción
0	Acción realizada correctamente
1	Código de acción incorrecto
2	Error general
3	No existe la entidad
4	Error en un campo

Dado que para hacer una acción será necesario realizar varias operaciones, y para evitar que acciones paralelas interfieran entre si, por ejemplo dos altas simultáneas, la *Entidad de Registro* gestionará cada una de las acciones en un contexto distinto, de forma que se puedan realizar acciones en paralelo sin interferirse.

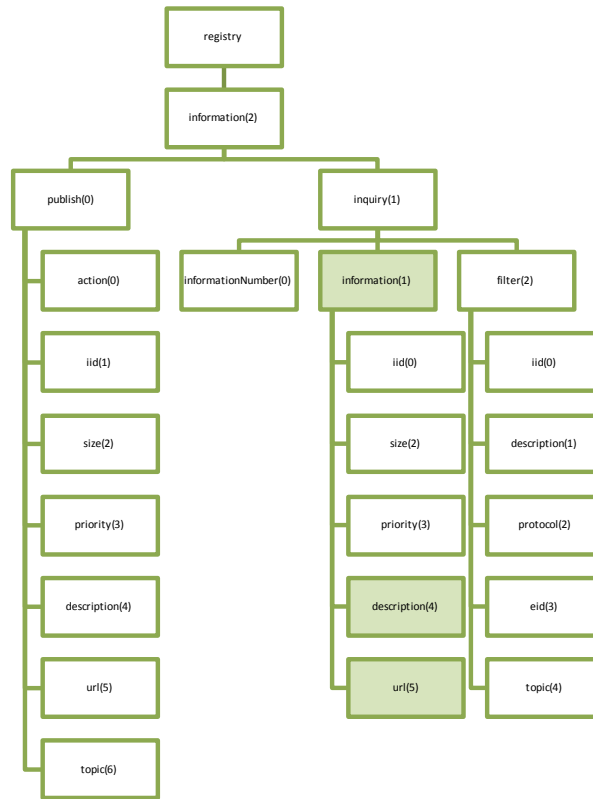


Fig. 21. Rama *Information* del subárbol *registry*.

Posteriormente, una vez registradas las entidades, cualquier entidad podrá consultar qué entidades están presentes en el sistema. Para ello en la rama *inquiry* contiene un objeto *entities* que es una vector de entidades con los datos de sólo lectura descritos en la tabla 1. El número de entidades que hay en el vector está disponible en el objeto *entitiesNumber*.

Para facilitar las labores de búsqueda en la rama *entity* también existe un objeto *filter* que me permite filtrar los datos de entidades a consultar. Cuando se establece algún valor en un objeto de *filter* el vector de entidades así como el *entitiesNumber* se actualiza en función del resultado del filtro. Si establezco más de un criterio de filtrado se combinan para obtener las entidades que cumplen todos los criterios.

De forma similar a las entidades podemos registrar y localizar los *paquetes de información* accesibles en nuestro sistema. Para ello en la rama *discovery* también tenemos una rama *information* con una funcionalidad similar a la de *entity* y que se muestra en la fig. 21.

Tabla 5. Operaciones de gestión de información.

Acción	>	Descripción	Campos
--------	---	-------------	--------

			iid	size	priority	description	url	topic
None	0	Sin acción.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
New	1	Nueva entrada.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delete	2	Elimina una entrada. La entrada es identificada por el IID.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Update	3	Actualiza una entrada. La entrada es identificada por el EID.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DescrAdd	4	Añade una descripción a la información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DescrDel	5	Elimina una descripción a la información. Si <i>description</i> está vacía borra todas las descripciones.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UrlAdd	4	Añade una URL a la información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UrlDel	5	Elimina una descripción a la información. Si <i>url</i> está vacía borra todas las URLs.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TopicAdd	5	Añade una categoría a la información.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TopicDel	6	Elimina una categoría a la información. Si <i>topic</i> está vacía borra todas las categorías.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> Campo obligatorio <input type="checkbox"/> Campo opcional <input type="checkbox"/> Campo ignorado								

En este caso las acciones que se pueden realizar en la rama *publish* son las que se muestran en la tabla 5. Su funcionamiento es similar al de las acciones descritas para el caso de las entidades, actuando el campo *action* como identificador de la acción que se quiere realizar.

Hemos visto como nuestro modelo refleja en el MIB-Registry la información necesaria para publicar y descubrir *Paquetes de Información* así como *Entidades*. Dado que el registro es una pieza opcional de nuestro sistema, también es necesario proponer algún mecanismo que permita a una entidad exponer a otra entidad que *Paquetes de Información* gestiona, sin necesidad de usar un registro. Para ello toda entidad que gestione información extenderá su MIB con un subárbol específico para tal efecto que denominaremos **MIB-Information**.

La extensión del MIB para la información reflejará los objetos necesarios para que se pueda consultar los *Paquetes de Información* que una entidad gestiona. El proceso de creación de este MIB, al igual que el del *MIB-Registry*, partirá de los objetos, en este caso los relacionados con la información, y siguiendo el estándar SNMP se confeccionará el *MIB-Information* (ver fig. 22).

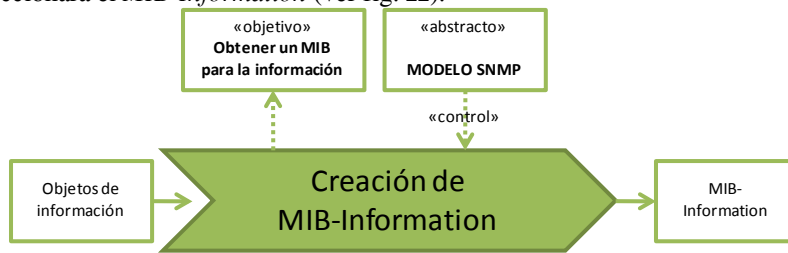


Fig. 22. Proceso de creación del *MIB-Information*.

En este caso el *MIB-Information* sólo tendrá una interfaz de consulta, ya que no será necesario ofertar una interfaz de publicación al tratarse de registros internos. El resultado del proceso puede verse en la fig. 23. Su funcionamiento es similar al de la rama *inquiry* de la rama *information* del *MIB-Registry*.

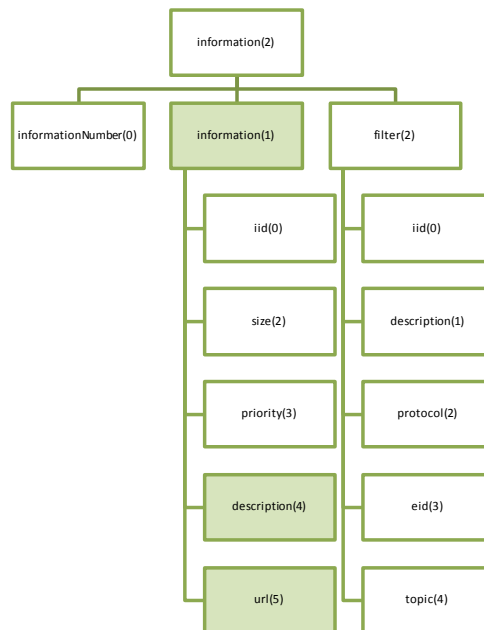


Fig. 23. Rama *information*.

2.3 Modelo de Comunicación

En el modelo de comunicación se definen los procesos y protocolos de comunicación utilizados por las entidades del sistema.

Como se indicó en el *Modelo de Organización* existen tres tipos de protocolos: el de gestión, el de descubrimiento y los de transporte. En nuestro caso y dado que hemos reflejado el descubrimiento dentro del propio MIB de SNMP, tanto el protocolo de gestión como el protocolo de descubrimiento coincidirán en SNMP. En cuanto a los protocolos de transporte estos serán tratados en profundidad en los capítulos 4 y 5.

Conceptualmente la comunicación entre las distintas entidades se realiza mediante el paso de mensajes, actividad que es realizada a través de la red de comunicaciones. Formalizaremos el paso de mensajes mediante dos funciones: una para enviar mensajes a la red, *sndMsg*, y otra para recibir mensajes desde la red. Cuando una entidad, conocida como emisor o *srcEntity*, envía un mensaje establece cual es la entidad destinataria del mismo, conocida como destinatario o *dstEntity*.

En el modelo, el paso de mensajes es no confiable, es decir, el hecho de que una entidad envíe un mensaje a otra entidad no implica que esta reciba el mensaje. Esto refleja la conducta normal de las redes de comunicaciones donde la transferencia de información no siempre puede ser garantizada. En caso de pérdida tanto la entidad origen como la destino no son conscientes de esta pérdida. Por ello las entidades destinatarias, ante la recepción de un mensaje, deberán contestar con otro mensaje de confirmación o asentimiento (ACK) o bien con un mensaje de respuesta, cuando se quiere garantizar la confiabilidad. El control de errores, por tanto, deberá ser gestionado por la entidad origen. Si un mensaje se pierde en la red o se pierde el mensaje de confirmación o respuesta, el emisor del mensaje será responsable de, pasado un tiempo sin recibir confirmación alguna, reenviar el mensaje asumiendo que éste ha sido perdido. En este esquema clásico de la comunicación entre procesos es posible que un destinatario reciba por duplicado un mismo mensaje, ya que los retardos en la comunicación pueden producir un retraso en la contestación que puede ser interpretado por el emisor como un error de comunicación.

El protocolo SNMP está fundamentado en el envío de mensajes, normalmente encapsulados en un paquete UDP. Cada mensaje o paquete SNMP está compuesto por dos partes básicas, una cabecera y un cuerpo o DPU.

Cada cabecera es dependiente de la versión del protocolo que se está utilizando. En este capítulo vamos a describir la versión 3 del protocolo que es la más reciente hasta la actualidad.

El protocolo SNMP en su versión 3 define una cabecera común para todos mensajes que se envían. Esta cabecera precede a la unidad de Datos de Protocolo (*Protocol Data Units* – DPU) donde se codifican los datos del mensaje SNMP. El cuerpo del mensaje en SNMP básicamente contiene una serie de campos.

El protocolo de gestión utiliza un conjunto de mensajes que permiten la realización de las tareas de gestión y que se codifican en la PDU de los paquetes.

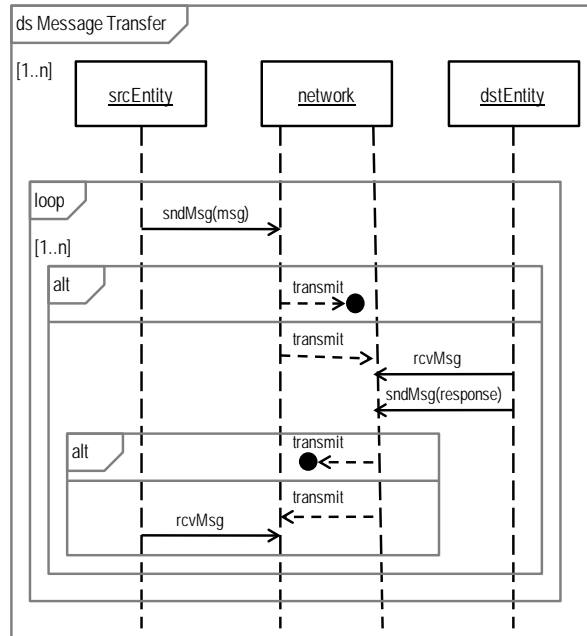


Fig. 24. Paso de mensajes con confiabilidad.

Los mensajes utilizado por SNMP están resumidos en la tabla 6 (Presuhn *et al.*, 2002).

Mediante los mensajes el sistema puede llevar a cabo las distintas operaciones que se definen para el protocolo de gestión

Mediante los mensajes las entidades del sistema de gestión puede realizar operaciones entre ellas.

Tabla 6. Mensajes de SNMP.

Mensaje	Descripción
GetRequest	Solicita leer un valor del MIB de la entidad destino.
GetNextRequest	Solicita el siguiente valor del MIB. Permite realizar una consulta de recorrido en el árbol.
GetBulkRequest	Solicita un conjunto de valores del MIB.
SetRequest	Solicita modificar valor del MIB de la entidad destino
Response	Respuesta de una petición.
Trap	Envía un aviso a una entidad que no requiere respuesta.
InformRequest	Envía un aviso a una entidad que requiere respuesta.

Las operaciones pueden ser de dos tipos: **comandos** que son operaciones que tienen respuesta o **notificaciones**, que no tienen respuesta. El tipo de cada operación está en función de la naturaleza de la misma.

En la tabla 7 se describen las operaciones que se pueden realizar indicando cuál es el mensajes de solicitud y cuál el de respuesta.

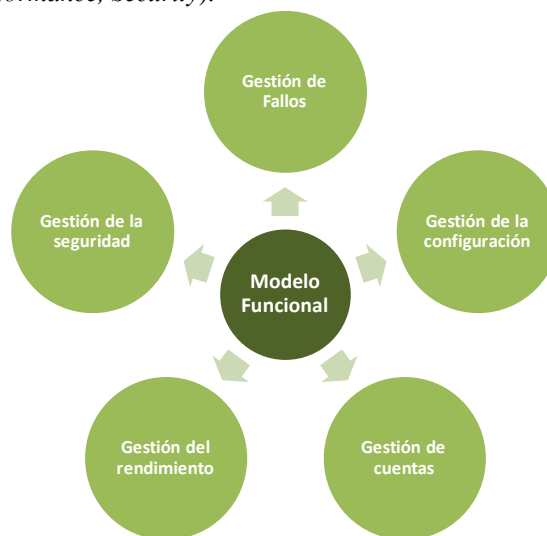
Tabla 7. Operaciones de SNMP.

Operación	Solicitud	Respuesta
GET	GetRequest	Response
SET	SetRequest	Response
GETNEXT	GetNextRequest	Response
GETBULK	GetBulkRequest	Response
TRAP	Trap	
INFORM	Inform	Response

Básicamente el protocolo permite leer (GET) o modificar (SET) un objeto de gestión o bien enviar notificaciones síncronas (INFORM) o asíncronas (TRAP). Las operaciones GETNEXT y GETBULK permiten optimizar la lectura secuencial de varios objetos de gestión.

2.4 Modelo funcional

El modelo funcional describe las cinco áreas en las que tradicionalmente se ha dividido la gestión de red: gestión de fallos, gestión de la configuración, gestión de prestaciones, gestión de contabilidad y gestión de seguridad (fig. 25). A este modelo se le conoce normalmente por el acrónimo de FCAPS (*Fault, Configuration, Accounting, Performance, Security*).

**Fig. 25.** Áreas del modelo funcional

El objetivo de la *gestión de fallos* es la detección, aislamiento, corrección y registro de operaciones anormales que ocurren en el sistema.

Determinar el máximo de información posible sobre los fallos es un elemento fundamental para su buena gestión. Por ello el sistema de gestión de fallos está muy

asociado con la monitorización de los elementos del sistema, para detectar un cambio en el estado de alguno de ellos.

Otro aspecto importante a contemplar es el análisis de tendencias para poder predecir errores e intentar garantizar que la red siempre está disponible.

Cuando una *entidad de gestión* detecta un fallo en un componente está enviará una notificación a una *entidad de control* para que, en la medida de lo posible, realice las acciones pertinentes que solventen dicho fallo.

En función de la notificación recibida la *entidad de control* identificará el fallo producido. Es posible que un mismo fallo produzca múltiples notificaciones, incluso notificaciones provenientes de entidades de gestión distintas. Es por ello que la *entidad de control* deberá realizar un proceso de correlación que permita asociar notificaciones relacionadas con un mismo fallo.

Una vez correlacionadas las notificaciones la *entidad de control* podrá tener una hipótesis del fallo producido. A partir de ese momento podrá realizar pruebas de localización para obtener más información sobre el fallo.

Finalmente, una vez localizado el fallo, la entidad de control podrá realizar las acciones correctivas pertinentes y posteriormente validar la corrección.



Fig. 26. Fases de la gestión de fallos.

La gestión de configuración es un conjunto de facilidades que permiten controlar, identificar, recoger y proporcionar datos de configuración a elementos gestionados. Entre las tareas relacionadas se puede destacar la definición de información de configuración en los recursos, la modificación de las propiedades de los recursos, la definición y modificación de relaciones entre los recursos, la inicialización y terminación de servicios de red o bien la distribución de software.

Todos los cambios de hardware y de software son coordinados a través de este proceso incluido la instalación de nuevas aplicaciones o equipamiento, la modificación de sistemas existentes y la eliminación de sistemas y programas obsoletos.

Según las redes incrementan su tamaño, una tarea importante es la configuración automatizada. Mediante la automatización se intenta minimizar la relación del sistema de gestión con los administradores, de forma que las tareas de administración se realicen con altos grados de autogestión.

La *gestión de cuentas* o tarificación son un conjunto de procedimientos que permiten medir y gestionar el uso de determinados elementos e identificar costes por el uso de éstos

El objetivo es reunir estadísticas sobre usuarios y otros elementos del sistema y su relación que el consumo de recursos que realizan: utilización de disco, consumo de memoria, tiempo de CPU, conexiones, etc.

En esta área se engloban también las operaciones de gestión de usuarios (usuarios, contraseñas y permisos) operaciones sobre equipos y servicios como realizar copias de seguridad y la sincronización y también labores de inventario.

La gestión del Rendimiento hace referencia al conjunto de procedimientos dedicadas a evaluar el comportamiento de elementos gestionados y la efectividad de determinadas actividades. Entre los indicadores de prestaciones se pueden definir los que están orientados al servicio, como la disponibilidad, el tiempo de respuesta, y la fiabilidad. En cambio, otros indicadores están orientados a la eficiencia o al grado de utilización.

La gestión del rendimiento permite a los administradores planificar la red para el futuro, así como a determinar la eficiencia de la red actual, por ejemplo, en relación con las inversiones realizadas para establecerla.

Algunos parámetros de red que se miden en la gestión del rendimiento son el porcentaje de utilización, las tasas de error y los tiempos de respuesta. Recolectando y analizando estos datos se pueden detectar problemas o tendencias de capacidad o fiabilidad y a qué servicios está afectando.

Se pueden establecer umbrales de rendimiento que serán utilizados lanzar una alarma. La alarma sería notificada y gestionada por el proceso de gestión de fallos

La *gestión de seguridad* está relacionada con todos los elementos asociados a la seguridad en los recursos de la red: generación, distribución y almacenamiento de claves de cifrado, información de usuarios y contraseñas, control de acceso y autorización. La gestión de seguridad no hace referencia a la propia seguridad del sistema de gestión, si no a la seguridad de los sistemas que gestiona.

En este área se proporcionan facilidades para incorporar mecanismos de seguridad contra los ataques a las comunicaciones, como protección contra interrupción del servicio, captura no autorizada de información, modificación de información o suplantación de entidad.

4 Conclusiones

En esta investigación se ha creado un modelo general de gestión de redes compatible con los estándares basado en la incorporación de mecanismos para la gestión de la difusión masiva de información a partir de técnicas de grid computing, multicast y streaming de información, lo cual posibilita la propuesta, de manera sistemática, de arquitecturas y sistemas de gestión viables con los requerimientos actuales.

Los resultados obtenidos tras la aplicación del modelo en un sistema de gestión de red real demuestran que mediante la integración de la gestión y transferencia de la información dentro de dicho sistema de gestión se consigue: que se puedan construir aplicaciones de gestión sin la necesidad de solucionar los problemas de transferencia de información, ya que ésta ya está contemplada en el propio sistema de gestión; que las aplicaciones de gestión accedan a la información presente en la red de forma transparente, sin importar la ubicación o los protocolos de acceso utilizados para ello; que la información se distribuya de forma automática en función de la importancia de dicha información y de las necesidades de los elementos integrados en el sistema de gestión, aportando mayores niveles de rendimiento y tolerancia a fallos.

Como línea futura se va a estudiar la incorporación de semántica a la gestión de redes para conseguir altos niveles en la automatización de la gestión.

Referencias

1. Voruganti, R.R. (1994). *A global network management framework for the '90s*. IEEE Communications Magazine. Vol. 32 issue 8 pp. 74-83.
2. Alon Y. Halevy, Zachary G. Ives, Jayant Madhavan, Peter Mork, Dan Suciu, and Igor Tatarinov (2004). The Piazza Peer Data Management System. IEEE Transactions on Knowledge and Data Engineering. ISSN: 1041-4347, pp: 787- 798. EEUU.
3. Antoni Barba Martí (2006). Gestión de Red. Editorial Ataraxiainc.
4. Nakamura, N. ; Kashimura, N. ; Motomura, K. (1995) *CMIP to SNMP translation technique based on rule description*. Fourth International Conference on Computer Communications and Networks, 1995. Proceedings. pp. 266–271.
5. ISO/IEC (1994). Information technology – Open Systems Interconnection – Basic Reference Model: The basic model. ISO/IEC 7498-1:1994(E).
6. Britton, J.P., deVos, A.N. (2005). CIM-based standards and CIM evolution. IEEE Transactions on Power Systems. Vol. 10, Issue 2, pp: 758-764.
7. Debusmann, M., Schmidt, M., Schmid, M., Kroeger, R. (2003). Unified service level monitoring using CIM. Enterprise Distributed Object Computing Conference, 2003. Proceedings. Seventh IEEE International. pp: 75-86.
8. DMTF (2008). Web Services for Management (WS-Management) Specification. Version: 1.0.0. Document Number: DSP0226.
9. ZhiHui Lu, JiaJun Wang, Yu Wu, Jie Wu, YiPing Zhong (2009). MWS-MCS: A Novel Multi-agent-assisted and WS-Management-based Composite Service Management Scheme. IEEE International Conference on Web Services, 2009. ICWS 2009. pp. 1041-1042.
10. Jong-Geun Park, Chang-Won Ahn, Hee-Nam Ch, o Il-Soo Byun, F. Desmons, Seong-Woon Kim (2006). A method for representing and transporting CIM operations using binary XML in the WBEM architecture. The 8th International Conference Advanced Communication Technology, 2006. ICACT 2006. pp. 2049-2053.
11. Sundaram, S., Jong-Cheol Seo, Abdurakhmanov, A., Young-Tak Kim (2006). Design and Implementation of WBEM-based Network Management System for Inter-AS Traffic Engineering. Fourth International Conference on Software Engineering Research, Management and Applications, 2006.
12. Hutter, M., Szekely, A., Wolkerstorfer, J. (2009). Embedded system management using WBEM. IFIP/IEEE International Symposium on Integrated Network Management, 2009. IM '09. pp. 390-397.
13. ISO/IEC (1989). Information processing Systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framawork. ISO/IEC 7498-4 (E).
14. Douglas Mauro y Kevin Schmidt (2005). *Essential SNMP, 2nd Edition*. O'Reilly.
15. J. Case, K. McCloghrie, M. Rose, S. Waldbusser (1996). *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*. Request for Comments 1907. Network Working Group.
16. K. McCloghrie, M. Rose (1991). *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. Request for Comments 1213. Network Working Group.
17. M. Rose, K. McCloghrie (1990). *Structure and Identification of Management Information for TCP/IP-based Internets*. Request for Comments 1155. Network Working Group.