INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELECTRICA

UNIDADAD CULHUACAN

TESINA

Seminario de Titulación:
“Las tecnologías aplicadas en redes de computadoras”
DES/ ESIME-CUL 5092005/07/2009

IMPLEMENTACIÓN DE UN SISTEMA DE GESTIÓN DE SOPORTE TÉCNICO REMOTO CENTRALIZADO SOBRE UNA RED LAN

Que como prueba escrita de su examen Profesional para obtener el Título de: Ingeniero en Comunicaciones y Electrónica.

Presenta:

ALANIS CUEVAS JOSEPH
ANAYA PINEDA EDGAR ELIHU
MARTINEZ MONTELLANO ALDO AUGUSTO

México D.F **Noviembre 2009.**

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD CULHUACAN
TESINA

POR LA OPCIÓN DE	SEMINARIO DE TITULACIÓN DES/ESIME-CUL/5092005/07/09
QUE PARA OBTENER EL TITULO DE	INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
PRESENTAN:	ALANIS CUEVAS JOSEPH ANAYA PINEDA EDGAR ELIHU MARTÍNEZ MONTELLANO ALDO AUGUSTO

IMPLEMENTACIÓN DE UN SISTEMA DE GESTION DE SOPORTE TÉCNICO REMOTO CENTRALIZADO SOBRE UNA RED LAN

EL PRESENTE TRABAJO SURGE DE LA NECESIDAD DE LAS EMPRESAS Y PRINCIPALMENTE DE LOS INGENIEROS EN SISTEMAS DE TENER UN SISTEMA DE MONITOREO Y GESTIÓN DE LA RED Y LOS DISPOSITIVOS QUE LA COMPONEN, ESTE SISTEMA SE DESARROLLO EN LA PLATAFORMA DE VISUAL STUDIO, PENSADA POR SER UN LENGUAJE BASICO SIN PROBLEMAS DE COMPATIBILIDAD CON EL MODULO WINSOCK, DICHO SISTEMA SE BASA EN LA INTEGRACIÓN Y COMUNICACIÓN DE LOS PROTOCOLOS DE RED ICMP, SNMP Y EL MODULO DE TCP/IP WINSOCK, UTILIZANDO LAS HERRAMIENTAS DE WINDOWS Y LAS LIBRERÍAS DE MSDN PROPUESTAS POR MICROSOFT. PERMITIENDO EL MONITOREO DEL HOST, EL ESCANEO DE PUERTOS, EL ECHO ICMP HACIA OTROS HOST, ALMACENANDO NUESTRA INFORMACIÓN PARA SU REVISIÓN VÍA WEB Y ENVIANDO POR CORREO AL ADMINISTRADOR DEL SISTEMA, TAMBIÉN INCLUYE OTRAS HERRAMIENTAS ARA SOPORTE DE PRIMER NIVEL, TODO ESTO EN UNA INTERFAZ AMIGABLE Y PORTABLE ACCESIBLE DESDE UN SERVIDOR REMOTO.

CAPITULADO

México D.F. 28 de Noviembre de 2009

M. en C. Diana Salomé Vázquez Estrada
Coordinador Académico del Seminario

Ing. Patricia Cortés Pineda.
Asesor.

Ing. Ignacio Monroy Ostria
Jefe del Departamento de Ingeniería
en Comunicaciones y Electrónica

AGRADECIMIENTOS

Esta tesina, producto de horas de investigación y desarrollo la dedico principalmente a mi Madre, gracias por tu inmenso cariño, por tu paciencia y tus sacrificios, porque gracias a ti tengo una educación y puedo decir que soy una persona de bien, gracias por desvelarte para mantenerme despierto y trabajando, gracias por todo tu apoyo, gracias por toda la paciencia que me tienes cuando me llamas y no voy por estar frente a la computadora.

Agradezco a la niña de mis ojos Mercedes, que ha jugado un papel crucial en mi vida, gracias por ese inmenso amor que me demuestras día con día. Esta tesina no hubiera podido ser terminada sin tu ayuda, gracias por ayudar con la investigación, gracias por entenderme cuando no te daba el tiempo que merecías, sin embargo aquí te presento el producto de ese tiempo que no pase contigo.

Agradezco de corazón a la M. en C. Diana Salomé Vázquez Estrada, por todo su apoyo, su exigencia y su cariño. Como profesora le agradezco mucho, pues usted ha sido una extraordinaria fuente de conocimientos y como persona es Inmedible el aprecio que le tengo, pues es gracias a usted, a su motivación y confianza que se ha desarrollado esta tesina y su correspondiente sistema.

Gracias a estas 3 Mujeres me será posible obtener el título de Ingeniero, prometo no desperdiciar su valioso esfuerzo.

Anaya Pineda Edgar Elihu.

INDICE

CAPITULO III

CAPITULO IV

CAPITULO V

INTRODUCCIÓN

Este proyecto surge de la necesidad de las empresas y principalmente de los administradores de sistemas de contar con herramientas integrales que faciliten realizar tareas de administración desde cualquier punto de la red, independizándolo de esta manera de la plataforma necesaria para ejecutar aplicaciones de gestión.

Este set de herramientas pretende aprovechar los módulos y variables que proporcionan los protocolos de red y el socket de Windows para implementar la comunicación e integración entre estos mismos, permitiendo de esta forma, una gestión a primer nivel por medio de una interfaz sencilla de la red acoplada y la administración de dispositivos. Debido a que esta tecnología es relativamente nueva, pocos investigadores se adentraron en el tema hasta ahora. Estas herramientas están desarrolladas en Visual Basic, uno de los lenguajes con mayor alcance en cuanto a desarrollo y librerías se refiere, que además de tener un completo control sobre las variables del entorno de Microsoft Windows, permite una completa compatibilidad con todas las versiones de este sistema operativo.

Uno de los beneficios más notables de las herramientas de gestión remota es que los desarrolladores de aplicaciones no tienen por qué conocer los detalles de los protocolos de gestión para manejar dispositivos remotos. Adicionalmente esto permite abstraer los diferentes protocolos y unificarlos con una única visión.

Abordaremos el área de la gestión de redes, y compararemos las distintas Herramientas de gestión tradicionales basadas en SNMP. Por otro lado se planea desarrollar un conjunto de herramientas que sean rápidamente implementables y permitan al Ingeniero de red comunicarse con sus usuarios y realizar algunas operaciones de administración en agentes del tipo pc/routers, ver estadísticas, estado y evolución de estos dispositivos.

OBJETIVOS GENERALES

- Desarrollar un sistema de gestión de redes, y comparar sus ventajas con las Herramientas de gestión tradicionales basadas en SNMP desde el punto de vista de la seguridad, eficiencia, costo, interfaz amigable, escalabilidad, expansión y servicios.
- Contar con herramientas que faciliten las tareas de administración desde cualquier punto de la red, independizándolo de la plataforma necesaria para ejecutar aplicaciones de gestión.

- Construir una infraestructura de gestión, que inicialmente soporte el protocolo de administración de redes SNMP, y que haga posible en trabajos posteriores la migración a otros protocolos de gestión.

OBJETIVOS ESPECÍFICOS

- Adquirir los conocimientos necesarios sobre protocolos de gestión (SNMP y CMIP).
- Explorar los mecanismos de gestión de red basados en WinSock
- Permitir a los ingenieros en redes realizar las tareas de administración de sus dispositivos desde cualquier punto a través de una VPN.
- Implementar una interfaz de ambiente grafico que permita visualizar de forma amigable el estado y evolución de los dispositivos de red.
- Implementar la captura de SNMP Traps a través de la base de datos y generar E-mails a los administradores registrados.
- Construir una aplicación de gestión elemental que utilice los principios enunciados en los puntos anteriores.

JUSTIFICACION

La tecnología crece cada vez más rápido y con esto las empresas aprovechan sus beneficios, por tanto tratan de estar a la vanguardia de las herramientas existentes. Actualmente las actividades de gestión usan técnicas y herramientas muchas veces aisladas e incompatibles entre sí. Se requiere por tanto de una pronta solución que además de simple, se convierta en una plataforma unificada para gestionar no solo redes, sino también sistemas y aplicaciones.

El desarrollo de una herramienta de gestión de redes y sistemas centralizado en un servidor por acceso remoto es un acercamiento promisorio que puede proveer una solución de gestión realmente integrada.

CAPITULO I

PROTOCOLO SNMP

1 PROTOCOLO SNMP

En esta primera parte se trata la situación actual de los dos protocolos de gestión de red más importantes: el SNMP (Simple Network Management Protocol, protocolo simple de gestión de red) y el CMIP (Common Management Information Protocol, protocolo común de gestión de información). Se presentan las ventajas y desventajas de cada uno.

A finales de los años 70 las redes de computadoras experimentaron un espectacular crecimiento y empezaron a conectarse entre sí. A estas nuevas redes se les llamó inter-redes o internets. Pronto se hicieron muy difíciles de gestionar, y se hizo necesario el desarrollo de un protocolo de gestión.

El primer protocolo que se usó fue el SNMP [RFC 1157]. Se diseñó como un recurso provisional para "ir tirando" hasta que se desarrollara otro protocolo más elaborado.

En los años 80 aparecieron dos nuevos protocolos: por un lado, la segunda versión del SNMP, que incorporaba muchas de las funciones del original (que sigue en uso) e incluía nuevas características que mejoraban sus deficiencias. Por otro, el CMIP, que estaba mejor organizado y contenía muchas más funciones que las dos versiones del SNMP.

Pronto el público general tendrá que elegir entre CMIP [RFC 1189] y SNMPv2 [RFC 1441], y esta decisión será de gran importancia: no en vano una empresa gasta alrededor del 15% del presupuesto asignado a sistemas de información en gestión de red.

Los criterios de elección deben basarse en las necesidades del usuario, esto es, un buen sistema de seguridad de red, una interfaz amigable, implementación relativamente barata y una reducción del tiempo empleado en gestión. Estos serán algunos de los criterios que se seguirán en la comparación de ambos protocolos.

1.1 INTRODUCCIÓN A SNMP

Para el desarrollo de la gestión de redes en inter-redes basadas en TCP/IP, el IAB (Internet Activities Board) decidió seguir la estrategia de usar a corto plazo el Simple Network Management Protocol (SNMP) para gestionar los nodos, proponiendo para largo plazo la estructura de gestión de redes OSI. Se escribieron entonces dos documentos para definir la gestión de la información: RFC 1065 que definía la Estructura de la Información de gestión (Structure of Management Information, SMI), y RFC 1066, que definía la Base de Información de gestión (Management Information

Base, MIB). Ambos documentos fueron diseñados para ser compatibles con la estructura SNMP y la de gestión de redes OSI.

Posteriormente se observó que los requerimientos de SNMP y los de gestión de redes OSI diferían más de lo esperado en un principio, por lo que los requerimientos de compatibilidad entre el SMI y el MIB fueron suspendidos.

La IAB ha designado al SNMP, a la SMI, y a la Internet MIB inicial como "Protocolos Estándar", con status de "Recomendado". Por medio de esta acción, la IAB recomienda que todas las implementaciones de IP y TCP sean gestionables por red, y los adopten y apliquen.

Así pues, la actual estructura para gestión de redes basadas en TCP/IP consiste en:

- Estructura e Identificación de la Información de gestión para redes basadas en TCP/IP, que describe cómo se definen los objetos gestionados contenidos en el MIB tal y como se especifica en la RFC 1155.
- Protocolo de gestión de Redes Simples, que define el protocolo usado para gestionar estos objetos, según se expone en la RFC 1157.

1.2 ARQUITECTUTA DEL PROTOCOLO SNMP

Implícita en el modelo de arquitectura del SNMP existe una colección de estaciones de gestión de red y de elementos de red. Las estaciones ejecutan aplicaciones de administración que monitorizan y controlan los elementos de red. Los elementos de red son dispositivos como hosts, gateways, servidores de terminal, y parecidos, que poseen agentes de gestión para realizar las funciones de administración de red solicitadas por las estaciones de gestión de red. El SNMP es usado para transmitir información de administración entre las estaciones de gestión y los agentes en los elementos de red.

1.2.1 PROPÓSITOS DE LA ARQUITECTURA

El SNMP explícitamente minimiza el número y complejidad de las funciones de gestión realizadas por el propio agente de gestión. Esta meta es atractiva al menos en cuatro aspectos:

El coste de desarrollo del software del agente de gestión necesario para soportar el protocolo se reduce acordemente.

El grado de complejidad de funciones de gestión soportado remotamente se incrementa, posibilitando un uso completo de los recursos de Internet en la tarea de gestión imponiendo las mínimas restricciones posibles en la forma y sofisticación de herramientas de gestión.

Los conjuntos simplificados de funciones de gestión son fácilmente entendibles y usados por los creadores de herramientas de gestión de red.

Un segundo objetivo del protocolo es que el paradigma funcional para monitorizar y controlar sea lo suficientemente flexible como para posibilitar aspectos de gestión y operación de la red adicionales y posiblemente no anticipados.

Un tercer propósito es que la arquitectura sea en lo posible independiente de los mecanismos de hosts o gateways particulares.

1.2.2 ELEMENTOS DE LA ARQUITECTURA

La arquitectura SNMP formula una solución al problema de gestión de redes en términos de los siguientes puntos:

- Alcance de la información de gestión comunicada por el protocolo.
- Representación de la información de gestión comunicada por el protocolo.
- Operaciones soportadas por el protocolo en la información de gestión.
- Forma y significado de los intercambios entre entidades de gestión.
- Forma y significado de las referencias a la información de gestión.

El alcance de la información de gestión transmitida por operaciones del SNMP es exactamente el representado por casos de todos los tipos de objetos no agregados, definidos en el estándar MIB [RFC 1156] de Internet, o definidos en cualquier otro sitio de acuerdo a las convenciones expuestas en el estándar SMI [RFC 1155] de Internet.

El SNMP modela las funciones del agente de gestión como lecturas (get) o escrituras (set) de variables. Esta estrategia posee al menos dos consecuencias positivas:

Limita el número esencial de funciones de gestión realizadas por el agente de gestión a dos.

Evita introducir el soporte de comandos de gestión imperativos en la definición del protocolo.

La estrategia plantea que la monitorización del estado de la red se puede basar a cualquier nivel de detalle en el sondeo (poll) de la información apropiada en la parte de los centros de monitorización. Un número limitado de mensajes no solicitados (traps) guían el objetivo y la secuencia del sondeo.

Las funciones de los pocos comandos imperativos actualmente soportados pueden ser fácilmente implementadas en este modelo de modo asíncrono. El intercambio de mensajes SNMP sólo requiere un servicio de datagramas poco fiable, y todo mensaje se representa por un único datagrama de transporte.

El SMI [RFC 1155] requiere que la definición de un protocolo de gestión contemple:

- Resolución de referencias MIB ambiguas: debido a que el alcance de cualquier operación SNMP está conceptualmente confinado a los objetos relevantes a un único elemento de red, y ya que todas las referencias SMI a objetos MIB son por medio de nombres de variables únicos, no hay posibilidad de que una referencia SNMP a cualquier tipo de objeto definido en el MIB se pueda resolver entre múltiples casos de ese tipo.

- Resolución de referencias entre versiones MIB: el objeto referenciado por cualquier operación SNMP es exactamente el especificado como parte de la operación de petición, o en el caso de una operación get-next su sucesor en el conjunto de MIB. En particular, una referencia a un objeto como parte de una versión del MIB estándar de Internet, no se aplica a ningún objeto que no sea parte de dicha versión, excepto en el caso de que la operación sea get-next, y que el nombre del objeto especificado sea el último lexicográficamente entre los nombres de todos los objetos presentados como parte de dicha versión.

- Identificación de los casos de objetos: cada caso de un tipo de objeto definido en el MIB se identifica en las operaciones SNMP por un nombre único llamado su "nombre de variable". En general, el nombre de una variable SNMP es un identificador de objeto de la forma x.y, donde x es el nombre del tipo de objeto no agregado definido en el MIB, e y es un fragmento de un identificador de objeto que de forma única para dicho tipo de objeto, identifica el caso deseado. Esta estrategia de denominación admite la completa explotación de la semántica de la PDU GetNextRequest, dado que asigna nombres para variables relacionadas de forma que sean contiguas en el orden lexicográfico de todas las variables conocidas en el MIB.

1.3 ESPECIFICACIONES DEL PROTOCOLO SNMP

El protocolo de administración de red es un protocolo de aplicación por el que las variables del MIB de un agente pueden ser inspeccionadas o alteradas.

Las entidades de protocolo se comunican entre sí mediante mensajes, cada uno formado únicamente por un datagrama UDP. Cada mensaje está formado por un identificador de versión, un nombre de comunidad SNMP y una PDU (Protocol Data Unit - Unidad de datos de protocolo). Estos datagramas no necesitan ser mayores que 484 bytes, pero es recomendable que las implementaciones de este protocolo soporten longitudes mayores.

Todas las implementaciones del SNMP soportan 5 tipos de UNA PDU:

- GetRequest-PDU
- GetNextRequest-PDU
- GetResponse-PDU
- SetRequest-PDU
- Trap-PDU

1.3.1 ELEMENTOS Y ESTRUCTURA DE UNA PDU

Se describirán a continuación las acciones que realiza una entidad de protocolo en una implementación SNMP. Definiremos dirección de transporte como una dirección IP seguida de un número de puerto UDP (Si se está usando el servicio de transporte UDP).

Cuando una entidad de protocolo envía un mensaje, realiza las siguientes acciones:

Construye la PDU apropiada como un objeto definido con el lenguaje ASN.1.

Pasa esta PDU, junto con un nombre de comunidad y las direcciones de transporte de fuente y destino, a un servicio de autentificación. Este servicio generará en respuesta otro objeto en ASN.1.

La entidad construye ahora un mensaje en ASN.1 usando el objeto que le ha devuelto el servicio de autentificación y el nombre de comunidad.

Este nuevo objeto se envía a la entidad destino usando un servicio de transporte.

Cuando una entidad de protocolo recibe un mensaje, realiza las siguientes acciones:

Hace un pequeño análisis para ver si el datagrama recibido se corresponde con un mensaje en ASN.1. Si no lo reconoce, el datagrama es descartado y la entidad no realiza más acciones.

Observa el número de versión. Si no concuerda descarta el datagrama y no realiza más acciones.

Pasa los datos de usuario, el nombre de comunidad y las direcciones de transporte de fuente y destino al servicio de autentificación. Si es correcto, este devuelve un objeto ASN.1. Si no lo es, envía una indicación de fallo. Entonces la entidad de protocolo puede generar una trampa (trap), descarta el datagrama y no realiza más acciones.

La entidad intenta reconocer la PDU. Si no la reconoce, descarta el datagrama. Si la reconoce, según el nombre de comunidad adopta un perfil y procesa la PDU. Si la PDU exige respuesta, la entidad iniciará la respuesta ahora.

Los datos que incluye una PDU genérica son los siguientes:

- RequestID: Entero que indica el orden de emisión de los datagramas. Este parámetro sirve también para identificar datagramas duplicados en los servicios de datagramas poco fiables.
- ErrorStatus: Entero que indica si ha existido un error. Puede tomar los siguientes valores, que se explicarán posteriormente:
- noError (0)
- tooBig (1)
- noSuchName (2)
- badValue (3)
- readOnly (4)
- genErr (5)
- ErrorIndex: Entero que en caso de error indica qué variable de una lista ha generado ese error.
- VarBindList: Lista de nombres de variables con su valor asociado. Algunas PDU quedan definidas sólo con los nombres, pero aún así deben llevar

valores asociados. Se recomienda para estos casos la definición de un valor NULL.

1.3.2 MENSAJES SNMP

Para realizar las operaciones básicas de administración anteriormente nombradas, el protocolo SNMP utiliza un servicio no orientado a la conexión (UDP) para enviar un pequeño grupo de mensajes (PDUs) entre los administradores y agentes. La utilización de un mecanismo de este tipo asegura que las tareas de administración de red no afectarán al rendimiento global de la misma, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a la conexión, por ejemplo TCP.

Los puertos comúnmente utilizados para SNMP son

- 161 SNMP
- 162 SNMP-trap

Los paquetes utilizados para enviar consultas y respuestas SNMP poseen el siguiente formato:

Versión → Comunidad → SNMP PDU

Donde:

Versión: Número de versión de protocolo que se está utilizando (por ejemplo 1 para SNMPv1);

Comunidad: Nombre o palabra clave que se usa para la autenticación. Generalmente existe una comunidad de lectura llamada "public" y una comunidad de escritura llamada "private";

SNMP PDU: Contenido de la unidad de datos del protocolo, el que depende de la operación que se ejecute.

Los mensajes GetRequest, GetNextRequest, SetRequest y GetResponse utilizan la siguiente estructura en el campo SNMP PDU:

Tipo → Identificador → Estado de error → Índice de error → Enlazado de variables

Donde:

Identificador: Es un número utilizado por el NMS y el agente para enviar solicitudes y respuesta diferentes en forma simultánea;

Estado e índice de error: Sólo se usan en los mensajes GetResponse´(en las consultas siempre se utiliza cero). El campo "índice de error" sólo se usa cuando "estado de error" es distinto de 0 y posee el objetivo de proporcionar información adicional sobre la causa del problema. El campo "estado de error" puede tener los siguientes valores:

0: No hay error;

1: Demasiado grande;

2: No existe esa variable;

3: Valor incorrecto;

4: El valor es de solo lectura;

5: Error genérico.

Enlazado de variables: Es una serie de nombres de variables con sus valores correspondientes (codificados en ASN.1).

1.3.2.1 GETREQUEST-PDU

Son PDU's que solicitan a la entidad destino los valores de ciertas variables. En el caso de GetRequest-PDU estas variables son las que se encuentran en la lista VarBindList; en el de GetNextRequest-PDU son aquellas cuyos nombres son sucesores lexicográficos de los nombres de las variables de la lista. Como se puede observar, GetNextRequest-PDU es útil para confeccionar tablas de información sobre un MIB.

Siempre tienen cero los campos ErrorStatus y ErrorIndex. Son generadas por una entidad de protocolo sólo cuando lo requiere su entidad de aplicación SNMP.

Estas PDU's siempre esperan como respuesta una GetResponse-PDU.

1.3.2.2 SETREQUEST-PDU

Estas ordenan a la entidad destino poner a cada objeto reflejado en la lista VarBindList el valor que tiene asignado en dicha lista. Es idéntica a GetRequest-

PDU, salvo por el identificador de PDU. Es generada por una entidad de protocolo sólo cuando lo requiere su entidad de aplicación SNMP. Espera siempre como respuesta una GetResponse-PDU.

1.3.2.3 GETRESPONSE-PDU

Es una PDU generada por la entidad de protocolo sólo como respuesta a GetRequest-PDU, GetNextRequest-PDU o SetRequest-PDU. Contiene o bien la información requerida por la entidad destino o bien una indicación de error.

Cuando una entidad de protocolo recibe una GetRequest-PDU, una SetRequest-PDU o una GetNextRequest-PDU, sigue las siguientes reglas:

1. Si algún nombre de la lista (o el sucesor lexicográfico de un nombre en el caso de GetNextRequest-PDU) no coincide con el nombre de algún objeto en la vista del MIB al que se pueda realizar el tipo de operación requerido ("set" o "get"), la entidad envía al remitente del mensaje una GetResponse-PDU idéntica a la recibida, pero con el campo ErrorStatus puesto a 2 (noSuchName), y con el campo ErrorIndex indicando el nombre de objeto en la lista recibida que ha originado el error.

2. De la misma manera actúa si algún objeto de la lista recibida es un tipo agregado (como se define en el SMI), si la PDU recibida era una GetRequest-PDU.

3. Si se ha recibido una SetRequest-PDU y el valor de alguna variable de la lista no es del tipo correcto o está fuera de rango, la entidad envía al remitente una GetResponse-PDU idéntica a la recibida, salvo en que el campo ErrorStatus tendrá el valor 3 (badValue) y el campo ErrorIndex señalará el objeto de la lista que ha generado el error.

4. Si el tamaño de la PDU recibida excede una determinada limitación, la entidad enviará al remitente una GetResponse-PDU idéntica a la recibida, pero con el campo ErrorStatus puesto a 1 (tooBig).

5. Si el valor de algún objeto de la lista no puede ser obtenido (o alterado, según sea el caso) por una razón no contemplada en las reglas anteriores, la entidad envía al remitente una GetResponse-PDU idéntica a la recibida, pero con el campo ErrorStatus puesto a 5 (genErr), y el campo ErrorIndex indicando el objeto de la lista que ha originado el error.

Si no se llega a aplicar alguna de estas reglas, la entidad enviará al remitente una GetResponse-PDU de las siguientes características:

- Si es una respuesta a una GetResponse-PDU, tendrá la lista varBindList recibida, pero asignando a cada nombre de objeto el valor correspondiente.
- Si es una respuesta a una GetNextResponse-PDU, tendrá una lista varBindList con todos los sucesores lexicográficos de los objetos de la lista recibida, que estén en la vista del MIB relevante y que sean susceptibles de ser objeto de la operación "get". Junto con cada nombre, aparecerá su correspondiente valor.
- Si es una respuesta a una SetResponse-PDU, será idéntica a esta, pero antes la entidad asignará a cada variable mencionada en la lista varBindList su correspondiente valor. Esta asignación se considera simultánea para todas las variables de la lista.

En cualquiera de estos casos el valor del campo ErrorStatus es 0 (noError), igual que el de ErrorIndex. El valor del campo requestID es el mismo que el de la PDU recibida.

1.3.2.4 TRAP-PDU

Es una PDU que indica una excepción o trampa. Es generada por una entidad de protocolo sólo a petición de una entidad de aplicación SNMP. Cuando una entidad de protocolo recibe una Trap-PDU [RFC 1215], presenta sus contenidos a su entidad de aplicación SNMP.

Los datos que incluye una Trap-PDU son los siguientes:

- enterprise: tipo de objeto que ha generado la trampa.
- agent-addr: dirección del objeto que ha generado la trampa.
- generic-trap: entero que indica el tipo de trampa. Puede tomar los siguientes valores:
- coldStart (0)
- warmStart (1)
- linkDown (2)

- linkUp (3)
- authenticationFailure (4)
- egpNeighborLoss (5)
- enterpriseSpecific (6)
- specific-trap: entero con un código específico.
- time-stamp: tiempo desde la última inicialización de la entidad de red y la generación de la trampa.
- variable-bindings: lista tipo varBindList con información de posible interés.

Dependiendo del valor que tenga el campo generic-trap, se iniciarán unas u otras acciones:

- Trampa de arranque frío (coldStart): La entidad de protocolo remitente se está reinicializando de forma que la configuración del agente o la implementación de la entidad de protocolo puede ser alterada.
- Trampa de arranque caliente (warmStart): La entidad de protocolo remitente se está reinicializando de forma que ni la configuración del agente ni la implementación de la entidad de protocolo se altera.
- Trampa de conexión perdida (linkDown): La entidad de protocolo remitente reconoce un fallo en uno de los enlaces de comunicación representados en la configuración del agente. Esta Trap-PDU contiene como primer elemento de la lista variable-bindings el nombre y valor de la interfaz afectada.
- Trampa de conexión establecida (linkUp): La entidad de protocolo remitente reconoce que uno de los enlaces de comunicación de la configuración del agente se ha establecido. El primer elemento de la lista variable-bindings es el nombre y el valor de la interfaz afectada.
- Trampa de fallo de autentificación (authenticationFailure): La entidad de protocolo remitente es la destinataria de un mensaje de protocolo que no ha sido autentificado.
- Trampa de pérdida de vecino EGP (egpNeighborLoss): Un vecino EGP con el que la entidad de protocolo remitente estaba emparejado ha sido seleccionado y ya no tiene dicha relación. El primer elemento de la lista variable-bindings es el nombre y el valor de la dirección del vecino afectado.

- Trampa específica (enterpriseSpecific): La entidad remitente reconoce que ha ocurrido algún evento específico. El campo specific-trap identifica qué trampa en particular se ha generado.

1.4 COEXISTENCIA ENTRE SNMPV1 Y SNMPV2

Se comentará a continuación qué hay que realizar para garantizar la compatibilidad y coexistencia de las dos versiones del protocolo SNMP [RFC 1908].

1.4.1 INFORMACIÓN DE GESTIÓN

La forma que tiene SNMPv2 para manejar los objetos gestionados no es mas que una extensión de SNMPv1. Así, ambas versiones utilizan el lenguaje ASN.1 para la notación. De hecho, lo que hace principalmente la versión 2 es normalizar la forma de definir los módulos MIB tal y como han dictado los años de experiencia trabajando con la primera versión.

Para que un módulo MIB o una declaración en SNMP se haga compatible con SNMPv2 necesita una serie de cambios. Normalmente estos cambios no exigen la invalidación de los objetos que contiene, ya que no son cambios muy graves. Son cambios referentes al vocabulario o la sintaxis (por ejemplo el guión se convierte en carácter prohibido en los nombres de variables), la definición de nuevos tipos (como integer32), o a la conversión de ciertas partes del MIB de opcionales a obligatorias (por ejemplo, ahora todos los objetos deben tener una cláusula DESCRIPTION). Hay cambios que son obligatorios y hay cambios que son sólo recomendados.

1.4.2 OPERACIONES DE PROTOCOLO

Se considerarán dos áreas: el comportamiento del intermediario entre una entidad SNMPv2 y una agente SNMPv1, y el comportamiento de entidades de protocolo bilingües actuando como administradoras.

1.4.2.1 AGENTE INTERMEDIARIO

Para conseguir la coexistencia a nivel de protocolo, se puede utilizar un mecanismo intermediario. Una entidad SNMPv2 actuando como agente puede ser implementada y configurada para realizar esta labor.

1.4.2.2 PASO DE SNMPV2 A SNMPV1

Para convertir peticiones de una entidad SNMPv2 administradora en peticiones a una entidad SNMPv1 agente:

1. Si es una GetRequest-PDU, una GetNextRequest-PDU o SetRequest-PDU, el agente intermediario la pasa sin alterar.
2. Si es una GetBulkRequest-PDU, el intermediario pone los campos non-repeaters y max-repetitions a cero, y la convierte en una GetNextRequest-PDU.

1.4.2.3 PASO DE SNMPV1 A SNMPV2

Para convertir respuestas enviadas de una entidad SNMPv1 agente hacia una entidad SNMPv2 administradora:

1. Si es una GetResponse-PDU, pasa por el intermediario sin alteraciones. No obstante hay que observar que aunque una entidad SNMPv2 nunca generará una PDU de respuesta con un campo error-status con un valor de "noSuchName", "badValue" o "readOnly", el agente intermediario no debe cambiar este campo. Así la entidad administradora podrá interpretar la respuesta correctamente. Si se recibe una GetResponse-PDU con el campo error-status con el valor "tooBig", el intermediario eliminará los contenidos del campo variable-bindings antes de propagar la respuesta. También aquí hay que señalar que aunque una entidad SNMPv2 nunca enviará una PDU de respuesta con un "tooBig" ante una GetBulkRequest-PDU, el agente itermediario debe propagar dicha respuesta.
2. Si se recibe una Trap-PDU, se convertirá en una Trap-PDU de SNMPv2. Esto se consigue colocando en el campo variable-bindings dos nuevos elementos: sysUpTime.0, que toma el valor del campo timestamp de la Trap-PDU, y snmpTrapOID.0, que se calcula así: Si el valor del campo generic-trap es "enterpriseSpecific", entonces el valor usado es la concatenación del campo

enterprise de la PDU con dos subidentificadores: '0', y el valor del campo specific-trap. Si no es así, se utiliza el valor definido para las Trap-PDU en la versión 2. En este caso se pone un elemento más en el campo variable-bindings: snmpTrapEnterprise.0, que toma el valor del campo enterprise de la PDU. Los destinos de esta Trap-PDU versión 2 se determinan según la implementación del agente intermediario.

1.4.2.4 ADMINISTRADOR BILINGÜE

Para conseguir la coexistencia a nivel de protocolo, una entidad de protocolo actuando como administradora podría soportar las dos versiones de SNMP. Cuando una aplicación de administración necesita contactar con una entidad de protocolo agente, la entidad administradora consulta una base de datos local para seleccionar el protocolo de gestión adecuado. Para dar transparencia a las aplicaciones, la entidad administradora debe mapear las operaciones como si fuera un agente intermediario.

1.5 VENTAJAS Y DESVENTAJAS DE SNMP

1.5.1 VENTAJAS DE SNMP

La ventaja fundamental de usar SNMP es que su diseño es simple por lo que su implementación es sencilla en grandes redes y la información de gestión que se necesita intercambiar ocupa pocos recursos de la red. Además, permite al usuario elegir las variables que desea monitorizar sin mas que definir:

• El título de la variable.

• El tipo de datos de la variable.

• Si la variable es de sólo lectura o también de escritura.

• El valor de la variable.

Otra ventaja de SNMP es que en la actualidad es el sistema más extendido. Ha conseguido su popularidad debido a que fue el único protocolo que existió en un principio y por ello casi todos los fabricantes de dispositivos como puentes y enrutadores diseñan sus productos para soportar SNMP.

La posibilidad de expansión es otra ventaja del protocolo SNMP: debido a su sencillez es fácil de actualizar.

1.5.2 DESVENTAJAS DE SNMP

El protocolo SNMP no es ni mucho menos perfecto. Tiene fallos que se han ido corrigiendo.

La primera deficiencia de SNMP es que tiene grandes fallos de seguridad que pueden permitir a intrusos acceder a información que lleva la red. Todavía peor, estos intrusos pueden llegar a bloquear o deshabilitar terminales.

La solución a este problema es sencilla y se ha incorporado en la nueva versión SNMPv2. Básicamente se han añadido mecanismos para resolver:

- Privacidad de los datos, que los intrusos no puedan tomar información que va por la red.
- Autentificación, para prevenir que los intrusos manden información falsa por la red.
- Control de acceso, que restringe el acceso a ciertas variables a determinados usuarios que puedan hacer caer la red.

El mayor problema de SNMP es que se considera tan simple que la información está poco organizada, lo que no lo hace muy acertado para gestionar las grandes redes de la actualidad. Esto se debe en gran parte a que SNMP se creó como un protocolo provisional y no ha sido sustituido por otro de entidad.

De nuevo este problema se ha solucionado con la nueva versión SNMPv2 que permite una separación de variables con más detalle, incluyendo estructuras de datos para hacer más fácil su manejo. Además SNMPv2 incluye dos nuevas PDU's orientadas a la manipulación de objetos en tablas.

CAPITULO II

PROTOCOLO CMIP

2 PROTOCOLO CMIP

Es un modelo de sistema de interconexión abierta que define como crear sistemas comunes de administración de redes

Tanto CMIP como SNMP definen estándares de administración de redes pero CMIP es más complejo y proporciona diversas características que deben observar los administradores de red

2.1 INTRODUCCIÓN

Tras la aparición de SNMP como protocolo de gestión de red, a finales de los 80, gobiernos y grandes corporaciones plantearon el Protocolo Común de gestión de Información CMIP (Common Management Information Protocol) que se pensó podría llegar a ser una realidad debido al alto presupuesto con que contaba. En cambio, problemas de implementación han retrasado su expansión de modo que solo está disponible actualmente de forma limitada y para desarrolladores.

CMIP [RFC 1189] fue diseñado teniendo en cuenta a SNMP, solucionando los errores y fallos que tenía SNMP y volviéndose un gestor de red mayor y más detallado. Su diseño es similar a SNMP por lo que se usan PDUs (Protocol Data Unit) como 'variables' para monitorizar la red.

En CMIP las variables son estructuras de datos complejas con muchos atributos, que incluyen:

- variables de atributos: representan las características de las variables.
- variables de comportamiento: qué acciones puede realizar.
- notificaciones: la variable genera una indicación de evento cuando ocurre un determinado hecho.

2.2 MODELOS DEL PROTOCOLO

Como CMIP es un protocolo de gestión de red implementado sobre OSI conviene introducir el marco de trabajo OSI en lo que respecta a gestión, ya que será la base para CMIP.

La gestión OSI posibilita monitorizar y controlar los recursos de la red que se conocen como "objetos gestionados". Para especificar la estandarización de la gestión de red se determina:

- Modelo o grupo de modelos de la inteligencia de gestión, hay 3 principales:
- *Modelo de organización:* describe la forma en que las funciones de gestión se pueden distribuir administrativamente. Aparecen los dominios como particiones administrativas de la red.
- *Modelo funcional:* describe las funciones de gestión (de fallos, de configuración, de contabilidad, de seguridad...) y sus relaciones.
- *Modelo de información:* provee las líneas a seguir para describir los objetos gestionados y sus informaciones de gestión asociadas. Reside en el MIB (Management Information Base).
- Estructura para registrar, identificar y definir los objetos gestionados.
- Especificación detallada de los objetos gestionados.
- Serie de servicios y protocolos para operaciones de gestión remotas.

2.3 FUNDAMENTOS DE CMIP

CMIP es un protocolo de gestión de red que se implementa sobre el modelo de Interconexión de Redes Abiertas OSI (Open Systems Interconnection) que ha sido normalizado por la ISO (International Organization for Standardization's) en sus grupos de trabajo OIW (OSI Implementors Workshop) y ONMF (OSI Network Management Forum). Además existe una variante del mismo llamado CMOT [RFC 1095] que se implementa sobre un modelo de red TCP/IP.

En pocas palabras, CMIP es una arquitectura de gestión de red que provee un modo de que la información de control y de mantenimiento pueda ser intercambiada entre un gestor (manager) y un elemento remoto de red. En efecto, los procesos de aplicación llamados gestores (managers) residen en las estaciones de gestión, mientras que los procesos de aplicación llamados agentes (agents) residen en los elementos de red.

CMIP define una relación igual a igual entre el gestor y el agente incluyendo lo que se refiere al establecimiento y cierre de conexión, y a la dirección de la información de gestión. Las operaciones CMIS (Common Management Information Services) se

pueden originar tanto en gestores como en agentes, permitiendo relaciones simétricas o asimétricas entre los procesos de gestión. Sin embargo, la mayor parte de los dispositivos contienen las aplicaciones que sólo le permiten hacer de agente.

En esta presentación nos vamos a centrar en CMIP ya que si éste apenas ha tenido éxito menos aún lo ha tenido el CMOT.

Como se puede ver, un sistema CMIP debe implementar una serie de protocolos de los cuales el CMISE (Common Management Information Service Element) es el que trabaja mano a mano con CMIP: todas las operaciones de gestión de red que crea CMISE, el CMIP las mapea en una operación en el CMIP remoto.

2.4 PROTOCOLOS EN CMIP

Para comunicarse entre sí dos entidades de aplicación pares del gestor y del agente se utilizan APDU's (Application Protocol Data Units). Como hemos visto, CMIP está compuesto de los protocolos OSI que siguen:

2.4.1 ACSE

ACSE (Association Control Service Element) se utiliza para establecer y liberar asociaciones entre entidades de aplicación. El establecimiento lo puede realizar el agente o el gestor; durante el proceso se intercambian los títulos de la entidad de aplicación para identificarse, y los nombres del contexto de aplicación para establecer un contexto de aplicación.

Servicios que ACSE proporciona a CMISE:

- A-ASSOCIATE, servicio confirmado utilizado para inicializar la asociación entre entidades de aplicación.
- A-RELEASE, servicio confirmado usado para liberar una asociación entre entidades de aplicación sin pérdida de información.
- A-ABORT, servicio no confirmado que causa la liberación anormal de una asociación con una posible pérdida de información.
- A-P-ABORT, servicio iniciado por el proveedor que indica la liberación anormal de la asociación del servicio de presentación con posible pérdida de información.

2.4.2 ROSE

ROSE (Remote Operation Service Element): es el equivalente OSI a una llamada de un procedimiento remoto. ROSE permite la invocación de una operación en un sistema remoto. CMIP usa los servicios orientados a conexión proporcionados por ROSE para todas las peticiones, respuestas y respuestas de error.

Servicios que ROSE proporciona a CMISE:

- RO-INVOKE, servicio no confirmado que es usado por un usuario de ROSE para invocar que una operación sea realizada por un ROSE invocado remoto.
- RO-RESULT, servicio no confirmado que un ROSE invocado usa para contestar a una previa indicación RO-INVOKE en el caso de que se haya realizado con éxito.
- RO-ERROR, servicio no confirmado que es usado por un usuario de ROSE invocado para contestar a una previa indicación RO-INVOKE en el caso de que haya fracasado.
- RO-REJECT, servicio no confirmado utilizado por un usuario de ROSE para rechazar una petición (indicación RO-INVOKE) del otro.

2.4.3 CMISE

CMISE (Common Management Information Service Element): Proporciona los servicios básicos de gestión confirmados y no confirmados para reportar eventos y manipular datos de gestión. CMISE hace uso de los servicios proporcionados por ROSE y ACSE. Servicios de CMISE: se denominan unidades funcionales y se resumen en la tabla siguiente las denominadas 'stand alone' (luego hay otras tres más). El número que sigue a cada unidad funcional está definido por el CMIP. También se especifica en cada caso si se trata de servicio confirmado (C) o no confirmado (NC) y no aplicable NA.

Se distinguen dos elementos en la comunicación:

- invoker es el invocador, el que llama a la ejecución de una operación remota
- performer es el que realiza la operación solicitada por un sistema remoto.

Existen otras tres unidades funcionales que sólo son válidas si se seleccionan junto con alguna de las vistas del tipo “stand alone”. Todas ellas son PDU's.

Unidad Funcional	Primitivas de Servicio	Modo
Conf. Event report invoker(0)	M-EVENT-REPORT Req/Conf	C
Conf. Event report performer(1)	M-EVENT-REPORT Ind/Rsp	C
event report invoker(2)	M-EVENT-REPORT Req	NC
event report performer(3)	M-EVENT-REPORT Ind	NC
Confirmed get invoker(4)	M-GET Req/Conf	NA
Confirmed get performer(5)	M-GET Ind/Rsp	NA
Confirmed set invoker(6)	M-SET Req/Conf	C
Confirmed set performer(7)	M-SET Ind/Rsp	C
Set invoker(8)	M-SET Req	NC
set performer(9)	M-SET Ind	NC
Confirmed action invoker(10)	M-ACTION Req/Conf	NC
Confirmed action performer(11)	M-ACTION Ind/Rsp	NC
Action invoker(12)	M-ACTION Req	NC
Action performer(13)	M-ACTION Ind	NC
Confirmed create invoker(14)	M-CREATE Req/Conf	NA
Confirmed create performer(15)	M-CREATE Ind/Rsp	NA
Confirmed delete invoker(16)	M-DELETE Req/Conf	NA
Confirmed delete performer(17)	M-DELETE Ind/Rsp	NA
Multiple reply(18)	Linked Identification	NA
Multiple object selection(19)	Scope, Filter, Sync.	NA
Extended service(20)	Extended Presentation	N/A

Tabla 2.1. Unidades Funcionales

Aparecen dos elementos en la interacción de la gestión de red, el gestor y el agente (gestionado), que negocian cuatro tipos de asociación que se establecen entre dos entidades de aplicación:

- Event: el gestionado puede enviar un M-EVENT-REPORTS.

- Event/Monitor: como el Event y además el gestor puede usar M-EVENT-REPORTS, peticiones M-GET y recibir respuestas M-GET.
- Monitor/Control: el gestor implementa M-GET, M-SET, M-CREATE, M-DELETE y M-ACTION, sin permitir el reporte de eventos.
- Full Manager/Agent: soporta todas las funciones.

Un sistema debe soportar al menos una de estas asociaciones, y puede hacer de gestor o gestionado pero no dentro de la misma asociación. Esta es una manera de reducir el tamaño del código: para determinado grupo de unidades funcionales se le da una asociación. Del mismo modo en la negociación, no se negocia cada unidad funcional sino una asociación, lo cual es más eficiente.

El proceso de negociación usa los servicios A-ASSOCIATE y A-RELEASE para determinar en una asociación quién va a ser el gestor o el gestionado, y el tipo de asociación. Estas son las reglas de negociación:

- Un sistema agente (gestionado) solo puede requerir una asociación Event y puede crearla sólo si tiene un evento que reportar y no tiene un gestor asociado.
- Un sistema gestor puede requerir cualquier tipo de asociación.
- Una asociación se crea mediante un requerimiento A-ASSOCIATE con el AE-TITLE del solicitante y la aplicación en uso. Entonces el receptor puede devolver un A-ASSOCIATE con su AE-TITLE para aceptarlo o un A-ASSOCIATE para rechazarlo.
- Un sistema gestionado puede pedir dentro de una asociación bajar a otra asociación (de Full a Monitor/Control o Event/Monitor o Event). El gestor puede rechazar la petición..

2.5 ESTRUCTURA DE LA ESTION DE INFORMACION

SMI (Structure of Management Information) define la estructura lógica de la información de gestión y cómo se identifica y describe. Este SMI está diseñado para usarse tanto en SNMP como en CMIP, pero cada uno lo implementa de manera específica. SMI define las siguientes funciones:

- Alcance: se utiliza para identificar los objetos gestionados que van a ser filtrados.

- Filtrado: se usa para seleccionar un subconjunto de objetos gestionados que satisfacen ciertas condiciones.
- Sincronización: una vez filtrados, se puede operar bajo el método 'best effort' por el cual, si falla una operación en un objeto sigue con el resto, y el método 'atomic' en el que la operación se realiza en todos o en ninguno.

El único requerido en CMIP es el de 'best effort' aunque el otro también puede ser soportado.

MIB (Management Information Base) viene especificado por SMI y define los objetos gestionados en la actualidad. Los objetos gestionados vienen definidos totalmente especificando los atributos o propiedades que tiene el objeto. Se entiende por atributos a los elementos de información que solo se pueden manipular como un todo y se les da un identificador.

Los objetos se jerarquizan según están contenidos unos dentro de otros (jerarquía de contención) y según sus propiedades (jerarquía de herencia).

2.6 VENTAJAS Y DESVENTAJAS DE CMIP

2.6.1 VENTAJAS DE CMIP

El principal beneficio que aporta el protocolo CMIP es que no sólo se puede enviar información de gestión de o hacia un terminal, sino que es posible desarrollar tareas que serían imposibles bajo SNMP. Por ejemplo, si un terminal no puede encontrar un servidor de ficheros en un tiempo predeterminado, CMIP notifica el evento al personal adecuado. En SNMP el usuario tendría que guardar el número de intentos de acceso al servidor mientras que en CMIP de esto se encarga el propio protocolo.

CMIP soluciona varios de los fallos de SNMP. Por ejemplo, tiene incluidos dispositivos de gestión de la seguridad que soportan autorizaciones, control de acceso, contraseñas. Como resultado de la seguridad que de por sí proporciona CMIP no necesita de posteriores actualizaciones.

Otra ventaja de CMIP es que haya sido creado no sólo por gobiernos sino también por grandes empresas, en los que puede tener en el futuro un mercado fiel.

2.6.2 DESVENTAJAS DE CMIP

Si todo lo dicho hace a CMIP tan bueno, uno puede preguntarse: ¿por qué no se usa? La respuesta es que CMIP significa también desventajas: CMIP requiere 10 veces más recursos de red que SNMP. En otras palabras, muy pocas redes de la actualidad son capaces de soportar una implementación completa de CMIP sin grandes modificaciones en la red (muchísima más memoria y nuevos protocolos de agente). Por eso mucha gente piensa que CMIP está destinado al fracaso.

La única solución es disminuir el tamaño de CMIP cambiando sus especificaciones. Así han aparecido varios protocolos que funcionan con la base de CMIP con menos recursos, pero todavía no ha llegado el momento de prescindir del tan extendido SNMP.

Otro problema de CMIP es su dificultad de programación: existe tal cantidad de variables que sólo programadores muy habilidosos son capaces de aprovechar todo su potencial.

2.7 IMPLEMENTACIÓN

SNMP es un conjunto de especificaciones de comunicación de red muy simple que cubre los mínimos necesarios de gestión de red exigiendo muy poco esfuerzo a la red sobre el que SNMP está implementado.

CMIP es un sistema de gestión de red muy bien diseñado que mejora muchas de las deficiencias del SNMP. El costo de estas mejoras es haberse convertido en un sistema tan grande y complejo que sólo las redes mejor equipadas pueden soportarlo.

Por tanto, es recomendable implementar SNMP antes que CMIP por los enormes recursos de sistema que CMIP requiere.

CAPITULO III

WINSOCK

3 WINSOCK

La API de Windows Sockets, que fue reducido a Winsock, es una especificación técnica que define cómo el software de red de Windows debe tener acceso a los servicios de red, especialmente TCP / IP. Define una interfaz estándar entre una aplicación cliente TCP/IP de Windows (como un cliente FTP o un navegador web) y el protocolo TCP/IP subyacente.

3.1 INTRODUCCION A LOS SOCKETS

Continuamente se oyen hablar de los "sockets" en cuanto a la capa de aplicación del modelo OSI se refiere, pues los sockets son una forma de comunicarse con otros programas usando descriptores de fichero estándar de Unix. Designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.

En los orígenes de Internet, las primeras computadoras en implementar sus protocolos fueron aquellas de la universidad de Berkeley. Dicha implementación tuvo lugar en una variante del sistema operativo Unix conocida como BSD Unix. Pronto se hizo evidente que los programadores necesitarían un medio sencillo y eficaz para escribir programas capaces de intercomunicarse entre sí. Esta necesidad dio origen a la primera especificación e implementación de sockets, también en Unix. Hoy día, los sockets están implementados como bibliotecas de programación para multitud de sistemas operativos, simplificando la tarea de los programadores.

Para que dos programas puedan comunicarse entre sí es necesario que se cumplan ciertos requisitos:

- Que un programa sea capaz de localizar al otro.
- Que ambos programas sean capaces de intercambiarse cualquier secuencia de octetos, es decir, datos relevantes a su finalidad.

Para ello son necesarios los tres recursos que originan el concepto de socket:

- Un protocolo de comunicaciones, que permite el intercambio de octetos.
- Una dirección del Protocolo de Red (Dirección IP, si se utiliza el Protocolo TCP/IP), que identifica una computadora.

- Un número de puerto, que identifica a un programa dentro de una computadora.

Los sockets permiten implementar una arquitectura cliente-servidor. La comunicación ha de ser iniciada por uno de los programas que se denomina programa cliente. El segundo programa espera a que otro inicie la comunicación, por este motivo se denomina programa servidor.

Un socket es un fichero existente en la máquina cliente y en la máquina servidora, que sirve en última instancia para que el programa servidor y el cliente lean y escriban la información. Esta información será la transmitida por las diferentes capas de red.

3.2 PROPIEDADES DE LOS SOCKETS

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es TCP, aunque también es posible utilizar UDP o IPX. Gracias al protocolo TCP, los sockets tienen las siguientes propiedades:

Orientado a conexión.

Se garantiza la transmisión de todos los octetos sin errores ni omisiones.

- Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.

Estas propiedades son muy importantes para garantizar la corrección de los programas que tratan la información.

El protocolo UDP es un protocolo no orientado a la conexión. Sólo se garantiza que si un mensaje llega, llegue bien. En ningún caso se garantiza que llegue o que lleguen todos los mensajes en el mismo orden que se mandaron. Esto lo hace adecuado para el envío de mensajes frecuentes pero no demasiado importantes, como por ejemplo, mensajes para los refrescos (actualizaciones) de un gráfico.

Consideremos la terminología siguiente:

Un socket es un tipo especial de manejador de fichero que utiliza un proceso para pedir servicios de red al sistema operativo.

Una dirección de socket es la tripleta: {protocolo, dirección-local, proceso-local}

En la familia TCP/IP, por ejemplo: {tcp, 193.44.234.3, 12345}

Una conversación es el enlace de comunicación entre dos procesos.

Una asociación es la quíntupla que especifica completamente los dos procesos que comprende una conexión: {protocolo, dirección-local, proceso-local, dirección-externa, proceso-externo}

En la familia TCP/IP, por ejemplo: {tcp, 193.44.234.3, 1500, 193.44.234.5, 21} podría ser una aso ciación válida.

Una media-asociación es: {protocolo, dirección-local, proceso-local} o {protocolo, dirección-externa, proceso-externo} que especifica cada mitad de una conexión.

La media-asociación se denomina también socket o dirección de transporte. Esto es, un socket es un punto terminal para comunicación que puede nombrarse y direccionarse en una red.

3.3 TIPOS DE SOCKET

Los sockets se clasifican en sockets de flujo o sockets de datagramas dependiendo de si el servicio utiliza TCP (orientado a conexión y fiable) o UDP.

3.3.1 SOCKETS DE FLUJO

Los sockets de flujo definen flujos de comunicación en dos direcciones, fiables y con conexión. Si envías dos objetos a través del socket en el orden "1, 2" llegarán al otro extremo en el orden "1, 2", y llegarán sin errores. Utilizan el protocolo TCP/IP

- Telnet. Si realizamos telnet a un sitio de la web sobre el puerto 80, y escribes " GET / ", recibirás como respuesta el código HTML.
- Navegadores que usan el protocolo HTTP , usan sockets de flujo para obtener las páginas.

Usan el Protocolo de Control de Transmisión, TCP que asegura que la información llega secuencialmente y sin errores. Y el Protocolo de Internet IP que se encarga básicamente del encaminamiento a través de Internet y en general no es responsable de la integridad de los datos.

3.3.2 SOCKETS DE DATAGRAMAS

Los sockets de datagramas también usan IP para el encaminamiento, pero no usan TCP, usan el UDP (Protocolo de Datagramas de Usuario)

Simplemente se monta un paquete, se introduce una cabecera IP con la información de destino y lo envías. Generalmente se usan para transferencias de información por paquetes. Uso de acuses (ACK).

- Tftp (Protocolo de transferencia de archivos trivial).
- Bootp (Protocolo de inicio, usado para obtener una ip).

3.4 INVOCACION DE UN SOCKET

Llamadas socket básicas

A continuación se lista algunas llamadas de la interfaz socket básica. En la próxima sección se verá un escenario ejemplo del uso de estas llamadas de la interfaz socket.

3.4.1 INICIALIZAR UN SOCKET

FORMATO: int sockfd = socket(int familia, int tipo, int protocolo)

Donde:

- Familia simboliza familia de direccionamiento. Puede tomar valores tales como AF_UNIX, AF_INET, AF_NS y AF_IUCV. Su propósito es especificar el método de direccionamiento utilizado por el socket.
- Tipo simboliza el tipo de interface de socket a usar. Puede tomar valores tales como SOCK_STREAM, SOCK_DGRAM, SOCK_RAW, y SOCK_SEQPACKET.
- Protocolo puede ser UDP, TCP, IP o ICMP.
- Sockfd es un entero (similar a un descriptor de fichero) que devuelve llamada socket.

Ligar (registro) un socket a una dirección de puerto

FORMATO: int bind(int sockfd, struct sockaddr *localaddr, int addrlen)

Donde:

Sockfd es el mismo entero devuelto por la llamada socket.

Localaddr es la dirección local que devuelve la llamada bind.

Nótese que después de la llamada bind, ya se tienen valores para los tres primeros parámetros dentro de la asociación 5-tupla:

{protocolo, dirección-local, proceso-local, dirección-externa, proceso-externo}

Indicar premura para recibir conexiones

FORMATO: int listen(int sockfd, int tamaño-cola)

Donde:

Sockfd es el mismo entero que devuelve la llamada socket.

Tamaño-cola indica el número de conexiones pedida que puede introducir en la cola el sistema mientras el proceso local no haya emitido la llamada accept.

3.4.2 ACEPTACION DE LA CONEXIÓN

FORMATO: int accept(int sockfd, struct sockaddr *dirección-externa, int addrlen)

Donde:

Sockfd es el mismo entero que devuelve la llemada socket.

Dirección-externa es la dirección del proceso externo (cliente) que devuelve la llamada accept.

Nótese que esta llamada “accept” la emite un proceso servidor en vez de un proceso cliente. Si existe una petición de conexión esperando en la cola por este socket de conexión, accept toma la primera petición de la cola y crea otro socket con las mismas propiedades que “sockfd”; en caso contrario, accept bloqueará al proceso llamador hasta que llegue una petición de conexión.

3.4.3 PETICIÓN DE CONEXIÓN AL SERVIDOR

FORMATO: int connect(int sockfd, struct sockaddr *dirección-externa, int addrlen)

Donde:

Sockfd es el mismo entero devuelto por la llamada socket.

Dirección-externa es la dirección del proceso externo (servidor) que devuelve la llamada connect.

Nótese que esta llamada la emite un proceso cliente mejor que un proceso servidor.

Enviar y/o recibir datos

Las llamadas read(), readv(sockfd, char *buffer, int addrlen), recv(), readfrom(), send(sockfd, msg, len, flags), write() pueden usarse para recibir y enviar datos en una asociación de socket establecida (o conexión).

Nótese que estas llamadas son similares a las llamadas del sistema de E/S de ficheros estándar read y write.

3.4.4 CIERRE DEL SOCKET

FORMATO: int close(int sockfd)

Donde sockfd es el mismo entero que devuelve la llamada socket.

3.5 TECNOLOGÍAS

La especificación API de WinSock define dos interfaces: la API utilizada por los desarrolladores de aplicación, y el SPI, que proporciona un medio para que los desarrolladores de software de red puedan agregar nuevos módulos de protocolo para el sistema. Cada interfaz representa un contrato. La API garantiza que una aplicación funcionará correctamente con una implementación del protocolo de cualquier proveedor de software de red. El contrato de SPI garantiza que un módulo de protocolo puede ser agregado a Windows y por lo tanto podrá ser utilizado por una aplicación API. Aunque estos contratos fueron importantes cuando Windows Sockets fue lanzado por primera vez, de múltiples como entornos de red, exige el apoyo multi-protocolo. Ahora solo son de interés académico. Incluido en la versión de Windows Sockets API 2.0 son funciones para utilizar IPX / SPX, pero ninguna

aplicación comercial se sabe que utilice este transporte, ya que el protocolo era obsoleto, pero ahora con WSA 2,0, Microsoft ha enviado una pila TCP/IP de alta calidad con todas las versiones recientes de Windows, y no hay alternativas importantes independientes. Tampoco ha habido un gran interés en la aplicación de otros protocolos de TCP / IP.

Windows Sockets se basa en los Sockets de BSD, pero proporciona funcionalidad adicional para que la API pueda cumplir con el modelo estándar de programación de Windows. La API de Windows Sockets abarca casi todas las características de los sockets BSD de la API, pero hay algunos obstáculos inevitables que en su mayoría surgieron de las diferencias fundamentales entre Windows y Unix (aunque para ser justo Windows Sockets difieren menos de los sockets BSD que los últimos hechos a partir de STREAMS). Todas las llamadas a funciones en la API comienzan con el nombre de WSA, por ejemplo, WSASend () para el envío de datos en un socket conectado. Los WinSocks se amplían sobre la funcionalidad de los sockets BSD, ofreciendo "no-bloqueo" o Sockets asíncronos (acceso mediante la adición de WSAAsync antes de la función deseada, por ejemplo, WSAAsyncGetHostByName ())

Sin embargo, era un objetivo de diseño de WinSocks que deberían ser relativamente fáciles de utilizar para los desarrolladores de aplicaciones basadas en sockets y puertos de Unix a Windows. No se consideró suficiente para crear una API que sólo era útil para los nuevos programas escritos de Windows. Por esta razón, Windows Sockets incluye una serie de elementos que fueron diseñados para facilitar el cambio al operador. Por ejemplo, aplicaciones de Unix fueron capaces de utilizar la misma variable errno para registrar tanto los errores de red como los errores detectados en estándar de funciones de biblioteca de C. Dado que esto no era posible en Windows, WinSocks introdujo una función específica, WSAGetLastError (), para recuperar información de error. Estos mecanismos son útiles, pero la asignación de puertos de aplicación sigue siendo extremadamente compleja. Muchas aplicaciones tradicionales TCP/IP han sido implementadas por usar el sistema de características específicas de Unix, tales como pseudo-terminales y el sistema de retencion de llamadas, y esa funcionalidad de reproducción en Windows era problemática. Dentro de un tiempo relativamente corto, asignar puertos dio paso al desarrollo de las aplicaciones dedicadas de Windows.

3.6 WINSOCK

Winsock: Windows Socket API. Son las especificaciones técnicas que definen como el software de redes de Windows debería acceder a los servicios de red,

especialmente TCP/IP, es decir define la interfaz estándar entre una aplicación cliente y el protocolo que lo soporta:

- TCP/IP
- NWLink (Novell)
- IPX/SPX (Novell)
- NetBIOS (IBM)
- AppleTalk. (Macintosh)

3.6.1 VENTAJAS DE WINSOCK

- Proporciona una red familiar de la API para programadores que utilizan Windows o UNIX.
- Ofrece compatibilidad binaria entre Windows basado en TCP/IP y los proveedores de servicios públicos.
- Soporta tanto los protocolos orientados a conexión y sin conexión.
- Proporciona una interfaz que las aplicaciones puedan utilizar para acceder a muchos espacios de nombres diferentes, como nombres de dominios (DNS)
- Protocolo multipunto independiente y multidifusión.

3.6.2 ARQUITECTURA DE WINSOCK

Su arquitectura está estructurada en directivas almacenadas en ficheros .dll basados en programación UNIX, estos ficheros son instalados por defecto en Microsoft Windows a partir de la versión 2000, en el caso de los sistemas operativos anteriores es necesario descargar los service pack donde vienen integrados estos ficheros.

Windows Socket 1.1

Son las interfaces programadas que definen el funcionamiento básico de Winsock. Se mantuvo muy cerca de la interfaz de sockets de Berkeley existentes para simplificar la conservación de las aplicaciones existentes. Algunas extensiones

específicas se han añadido, principalmente para las operaciones asincrónicas con notificaciones basadas en mensajes

Windows Sockets 2.0

Es una extensión compatible de Winsock 1.1. Agregó soporte para el protocolo DNS, operaciones asincrónicas con notificaciones basadas en eventos y rutinas de conclusión, implantación del protocolo de capas, multidifusión, y QoS (calidad del servicio). También formalizó el apoyo para múltiples protocolos, incluidos IPX/SPX y DECnet. La nueva especificación permite a los sockets ser opcionalmente compartidos entre los procesos, solicitudes de conexión de entrada para ser aceptadas condicionalmente, y ciertas operaciones que se realizan en grupos de sockets en lugar de sockets individuales. Aunque la nueva especificación difería sustancialmente de Winsock 1, esta provee la fuente y compatibilidad a nivel binario con la API de Winsock 1.1. Una de las adiciones menos conocidas fue la interfaz de proveedor de servicios (SPI) de la API y Proveedores de servicio estructurado.

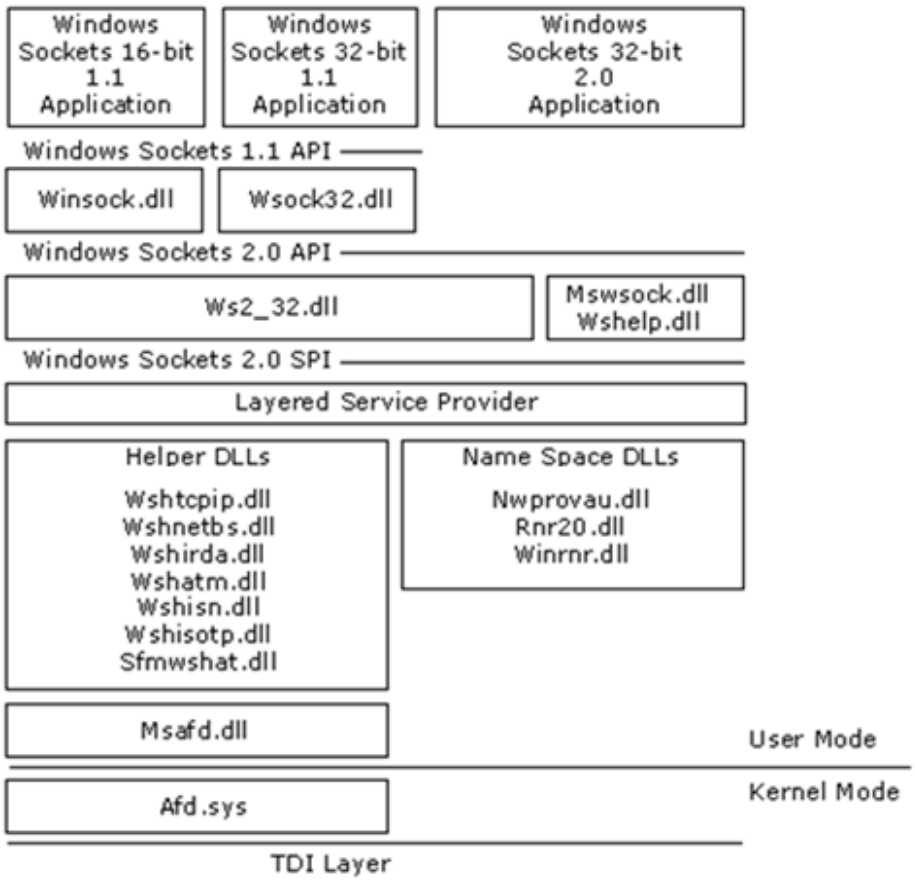

Figura 3.1 Arquitectura del Socket

3.6.3 FICHEROS DLL

Los ficheros Dll incluidos en los sistemas operativos de Microsoft Windows a partir de la versión 2000:

Fichero	Función
Winsock.dll	Winsock 1.1 de 16 bits
Wsock32.dll	Winsock 1.1 de 32 bits
Ws2_32.dll	Winsock 2.0 Principal
Mswsock.dll	Extensiones de Microsoft . Mswsock.dll es una interfaz API que suministra servicios que no son parte de Winsock.
Ws2help.dll	Utilidades de Plataforma-especifica. Ws2help.dll suminustra codigo especifico del sistema operativo que no es parte de Winsock
Wshtcpip.dll	Fichero de ayuda para TCP
Wshnetbs.dll	Fichero de ayuda para NetBT
Wshirda.dll	Fichero de ayuda para IrDA
Wshatm.dll	Fichero de ayuda para ATM
Wshisn.dll	Fichero de ayuda para Netware
Wshisotp.dll	Fichero de ayuda para transporte OSI
Sfmwshat.dll	Fichero de ayuda para Macintosh
Nwprovau.dll	Proveedor de resolucion de nombres para IPX
Rnr20.dll	Resolucion de nombres principal
Winrnr.dll	Resolucion de nombres LDAP
Msafd.dll	Interfaz Winsock para kernel
Afd.sys	Interfaz Kernel de Winsock para el transporte de protocolos TDI

CAPITULO IV

DESCRIPCION DE LA APLICACIÓN DESARROLLADA

4 DESCRIPCION DE LA APLICACIÓN DESARROLLADA

4.1 SISTEMA DE GESTION DE SOPORTE REMOTO

Para iniciar la descripción de la aplicación debemos aclarar que la aplicación no necesita instalarse pues se clasifica como una aplicación portable, ofreciendo muchas ventajas entre ellas es no requerir de Install-Shield para su ejecución, únicamente debe ser almacenada en algún dispositivo USB o en alguna carpeta creada por el administrador de redes, o finalmente puede ser accedida desde un servidor de aplicaciones. Se describirán en esta sección las secciones principales de las que consta el sistema, se tiene pensado continuar agregando mas utilidades según las necesidades del lugar donde se implemente.

4.1.1 APLICACIONES

Dado que el sistema mantiene una estructura de la forma cliente- servidor, tenemos 2 aplicaciones una que deben ser ejecutadas, la primera de nombre “Destino” debe ser ejecutada en el inicio de sesión en cada una de las maquinas monitoreadas para lograr esto debemos agregar la ruta de acceso de esta aplicación en un “valor de cadena” en el registro de Windows en la llave:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Para poder ingresar al registro, desde la ventana de ejecutar escribimos “regedit”, de esta manera esta aplicación se iniciara automáticamente al ingresar a nuestra sesion

La segunda aplicación, que es el sistema de gestión central debe ser accedida según la ruta que establezca el administrador de redes.

4.1.2 CONTRASEÑA

Al ejecutar la aplicación lo primero que nos mostrara será el sistema de contraseñas con 45 segundos como tiempo límite para ingresar nuestro usuario y contraseña otorgados por el programador con un máximo de 3 intentos (todos estos valores configurables en el momento de compilación con posibilidad de tener una base con estos datos).

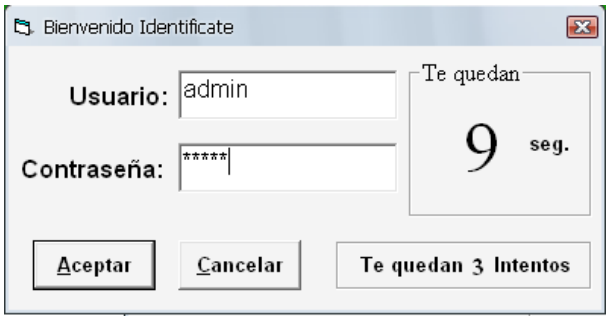

Figura 4.1 Sistema de contraseñas

4.1.3 VENTANA PRINCIPAL

Una vez comprobados el nombre de usuario y contraseña, se despliega el sistema principal, a partir del cual tenemos acceso a todo nuestro set de herramientas del sistema.

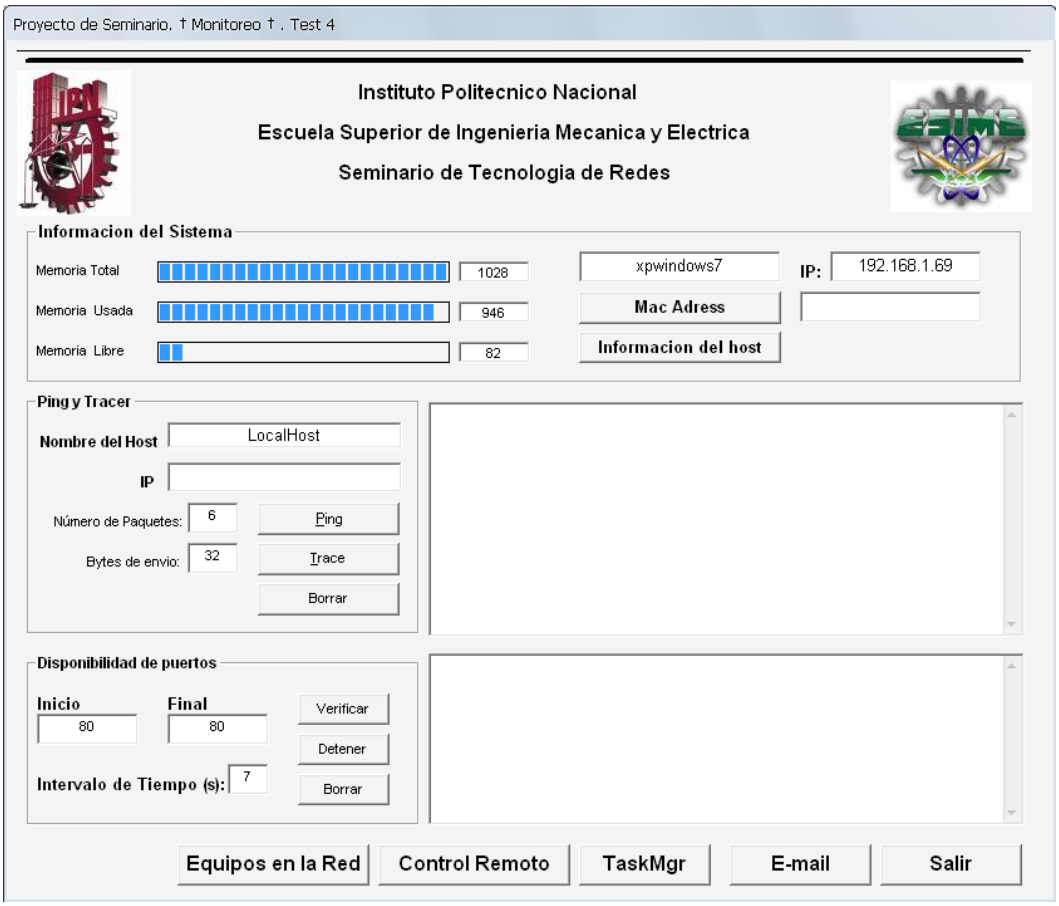

Figura 4.2 Ventana principal de la herramienta de gestión.

4.1.4 INFORMACIÓN DEL SISTEMA

Obteniendo de esta sección el rendimiento de nuestro equipo evaluado en MB de memoria RAM, nos despliega nuestro nombre de sistema, la dirección IP y la dirección física de nuestra tarjeta de red.

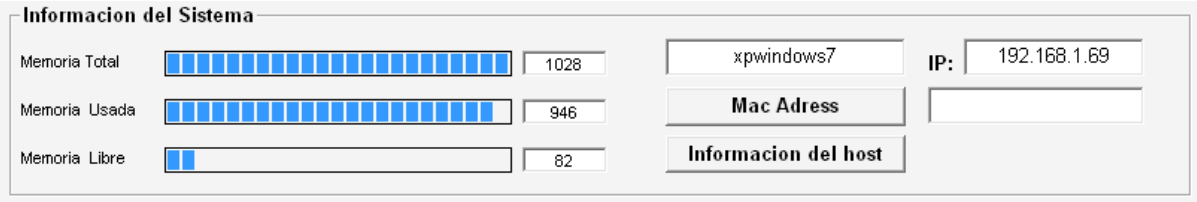

Figura 4.3 Información del sistema

En esta sección encontramos uno de las utilidades de mayor importancia para nosotros pues al presionar el botón de “Información del Host” accedemos a una ventana en la que podemos obtener todas las características del host monitoreado, incluyendo privilegios, contraseñas y datos de la sesión y del usuario, el cual además podemos guardar en un documento HTML para su carga en el servidor web

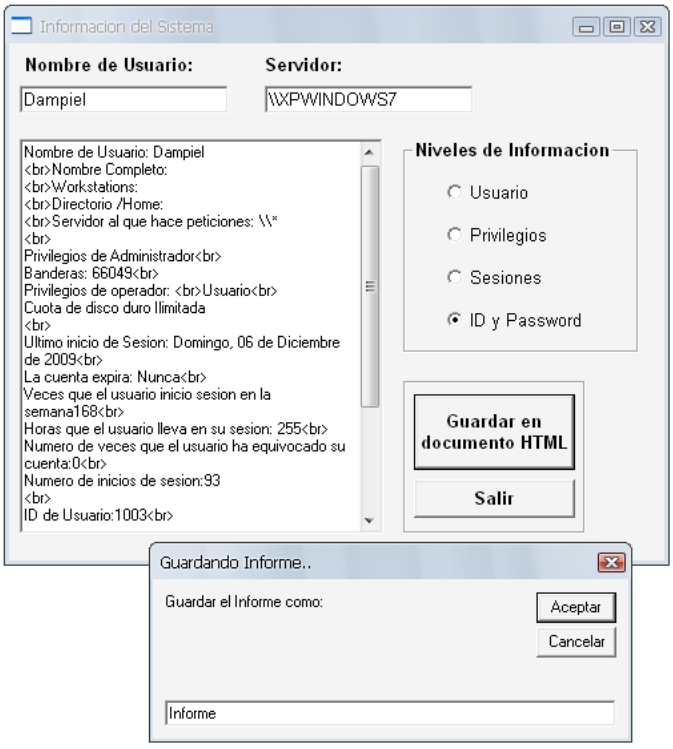

Figura 4.4 Información del Host y reporte.

4.1.5 PING Y TRAZADO DE RUTAS

Constantemente es necesario el monitoreo de hosts, guests y dispositivos de red, además de algunas webs, es por esto que fue diseñada esta sección en el que por medio del envió de paquetes configurados en número y en tamaño podemos saber si existe la comunicación con cierto nodo que nosotros introduciremos ya sea por IP o por nombre del host. Además para un administrador de redes es muy importante saber el costo de rutas hacia ciertos nodos es por esto que también se diseño el trazado de rutas hacia la IP que nosotros le indiquemos

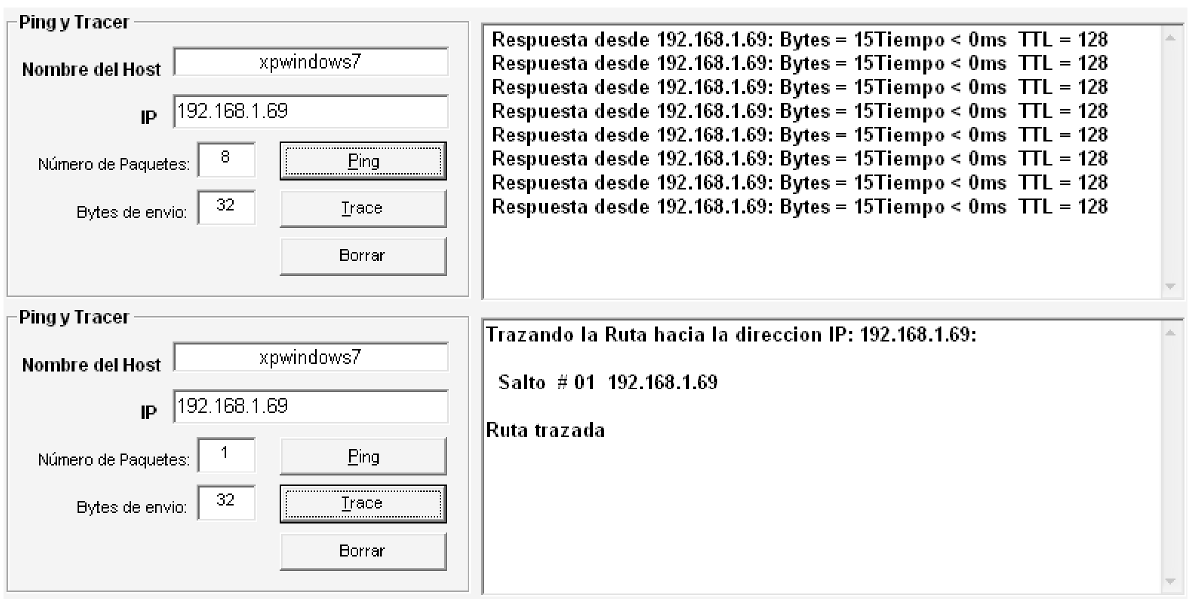

Figura 4.5 Ping y Trazado de rutas hacia un nodo de la red

4.1.6 DISPONIBILIDAD DE PUERTOS

Para un administrador de redes es importante conocer que puertos permanecen abiertos en un Host, ya sea por seguridad de saber que nadie se está introduciendo a dicho puerto o en caso de tener alguna aplicación empresarial especifica que maneje algún puerto y se encuentre provocando errores de conexión, lo primero que se debe verificar es precisamente la comunicación con ese puerto, pensando en eso se realizo esta sección permitiendo colocar un rango de puertos a monitorear en la IP que se ha establecido.

Figura 4.6 Disponibilidad de puertos

4.1.7 EQUIPOOS EN LA RED

Algo tan importante como es la detección de equipos dentro de la red, ya sea equipos de cómputo dispositivos de red como impresoras y dispositivos que permitan su identificación a través de un nombre, esta sección iniciada a partir del botón “Equipos en la red” nos muestra cualquier dispositivo incluido en la red de trabajo o dominio

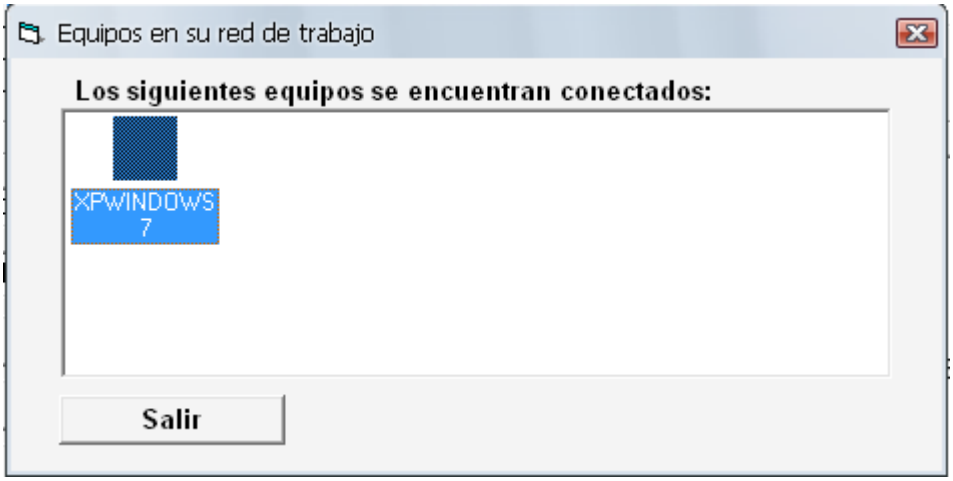

Figura 4.7 Dispositivos en la red de trabajo

4.1.8 APLICACIONES Y MENSAJES REMOTOS

Es esencial la ejecución de aplicaciones a otras maquinas, incluso se puede hacer una aplicación recursiva en la que se tenga almacenado el sistema dentro de las maquinas monitoreadas y para accesar a ellas lo hagamos desde esta sección de la aplicación que nos permite la ejecución de un programa desde una equipo remoto.

Otra aplicación que se le puede dar al hecho de que en la maquina monitoreada se esté ejecutando el programa “destino” que se encuentra a la escucha de nuestro sistema de gestión es el envió de mensajes.

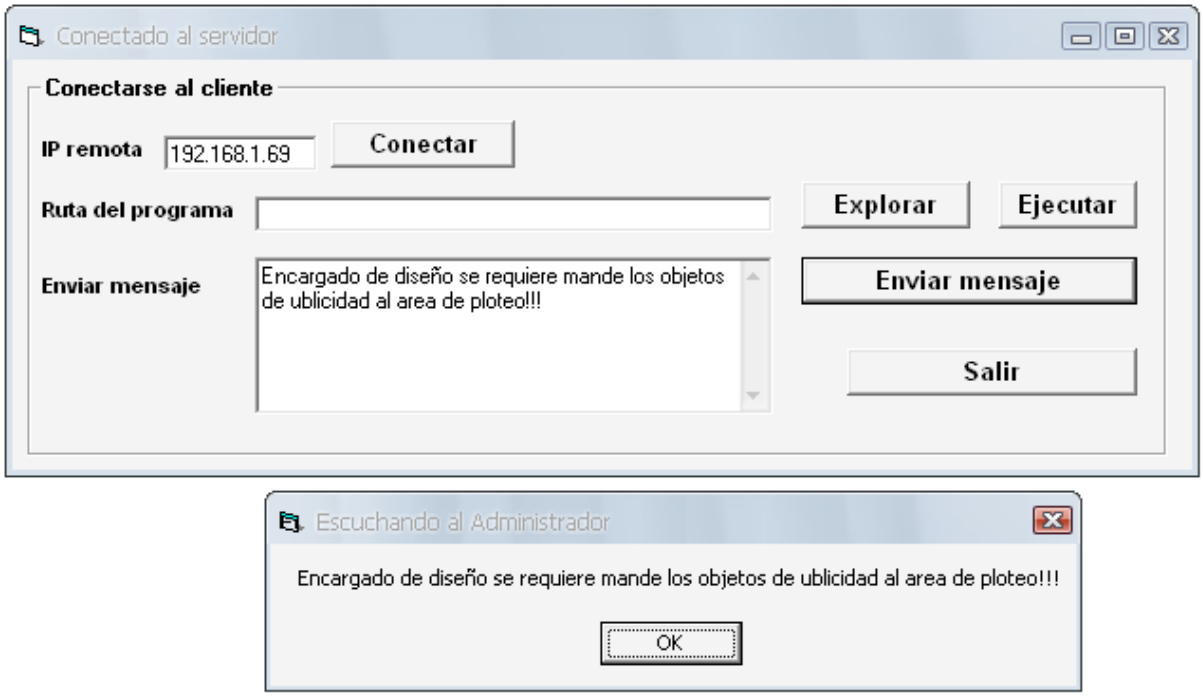

Figura 4.8 Envió de mensaje del administrador y recepción del mismo en la maquina monitoreada

4.1.9 ADMINISTRADOR DE TAREAS

Como administradores es importante el poder monitorear la acción de los hosts pensando en eso desarrollamos esta aplicación que nos despliega cada una de las tareas que se están ejecutando en la maquina monitoreada, no solo como precaución de observar lo que el usuario se encuentra realizando sino como monitoreo del sistema pues aquí podemos ver si existe algún proceso extraño o ajeno al sistema que pueda poner en riesgo la integridad de la información, pudiendo en este terminar u ocultar al usuario un proceso ya sea por seguridad o por modificación del aplicación, además de poder tener un informe de dicha tarea

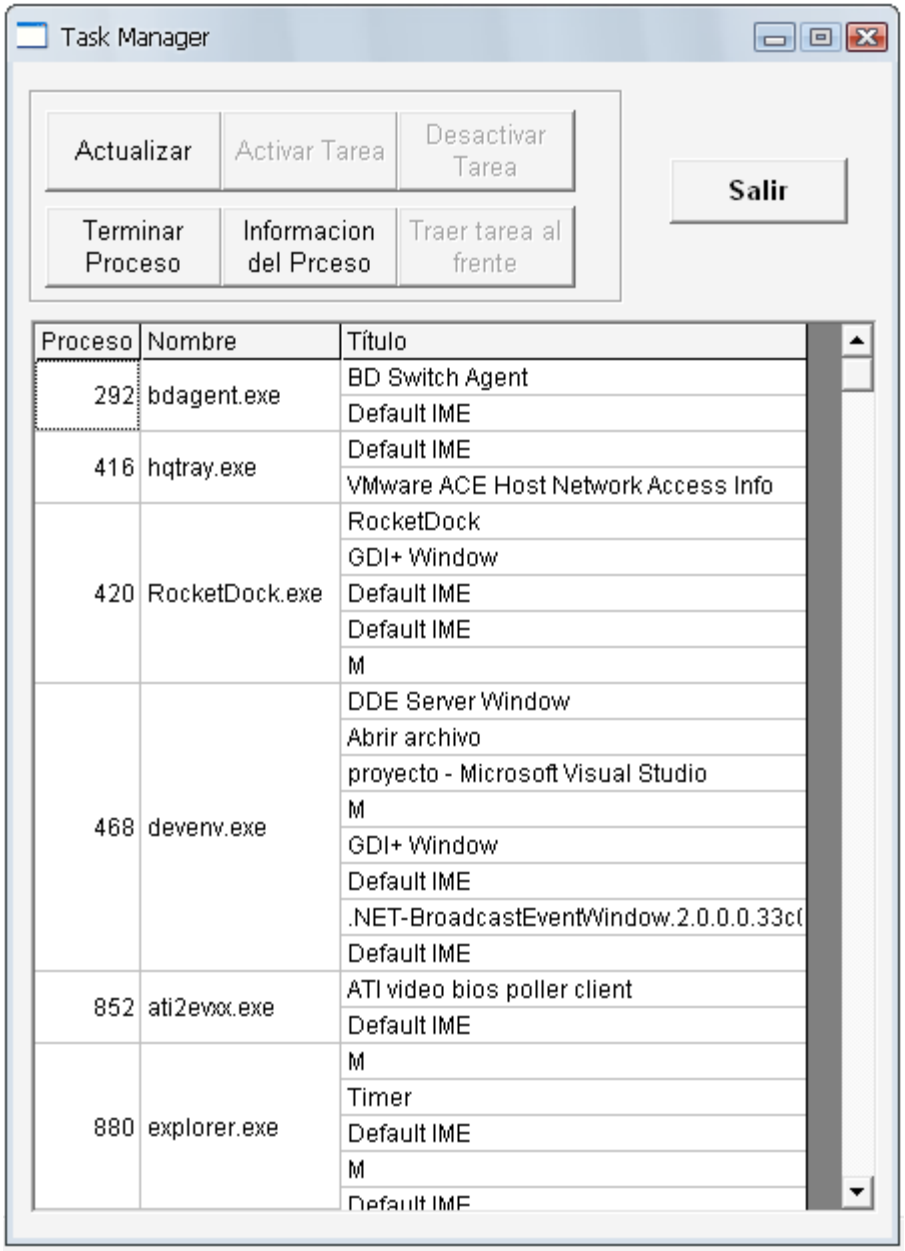

Figura 4.9 Administrador de tareas o Task Manager

4.1.10 ENVIO DE CORREO ELECTRONICO

Otra utilidad sumamente importante es el poder enviar los datos obtenidos del host o incluso cualquier otro dato adicional al correo y asea del encargado de soporte de alguna aplicación o incluso al nuestro para su posterior análisis, por lo que esta sección se realizo pensando en poder tener esta facilidad, permitiéndonos utilizar el

correo configurado en nuestro equipo para el envió de información a cualquier destinatario.

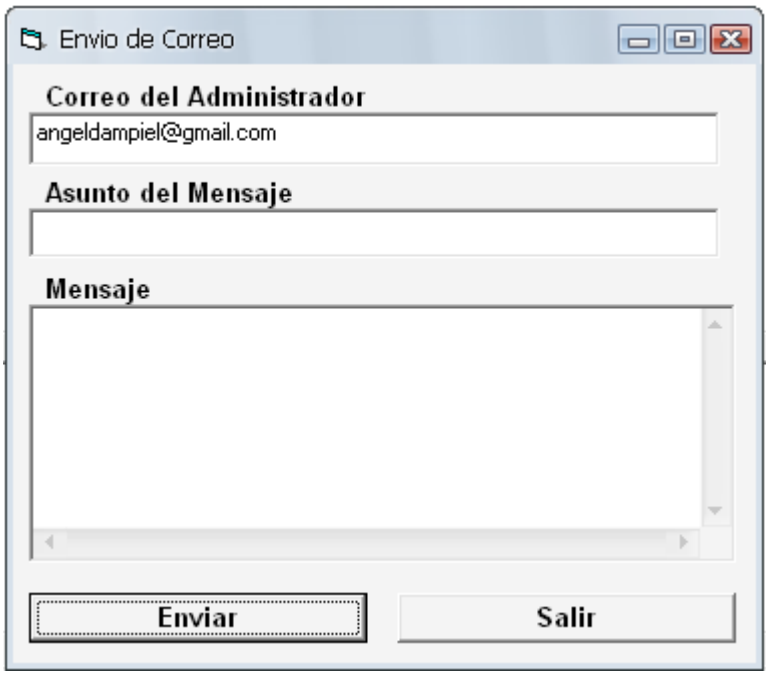

Figura 4.10 Envío de correo electrónico.

4.2 COMPARACION DE SISTEMAS DE GESTIÓN CONVENCIONALES

Para una mejor comprensión del valor de los sistemas de gestión, es necesario realizar la comparación entre nuestra herramienta y la forma convencional de administrar dispositivos y redes. Del análisis mencionado surgirán las ventajas y limitaciones de esta tecnología.

En la comparación se pone especial énfasis en las siguientes características:

- eficiencia
- escalabilidad
- niveles de abstracción de funcionalidad
- expansión
- costo
- interfaz amigable

- seguridad

4.2.1 EFICIENCIA

Existen problemas de eficiencia por parte del protocolo SNMP, problemas que pueden ser mejorados mejorarla en ciertos casos. Pero la idea no es reemplazar SNMPv1 por otro protocolo, sino combinarlo con otros protocolos para hacer más eficaz el uso de agentes SNMPv1.

SNMP fue desarrollado para proporcionar una implementación básica de un protocolo de administración en ambientes basados en TCP/IP. Muchas de las redes y dispositivos en Internet todavía son administrados con SNMPv1. Esta versión de SNMP tiene algunas limitaciones.

Una de las mayores deficiencias de SNMPv1 es la recuperación de grandes volúmenes de datos, como lo es, por ejemplo, una tabla de rutas (routing table) completa. La razón es que SNMPv1 proporciona el getnext-request para recuperar el próximo elemento en orden lexicográfico. Esta es la única manera de recuperar una tabla MIB entera desde un agente, y consecuentemente se requiere una solicitud separada para cada entrada en la tabla.

Si bien SNMPv2 resuelve este problema con el uso de getbulk-request, existe el inconveniente de la escasa difusión en las redes de estos agentes.

El hecho de hacer muchas solicitudes SNMP, no es un problema en si el tiempo de respuesta es bajo (ejemplo 0.02 seg.), pero si el tiempo de request-response es relativamente alto (por ejemplo 1 seg.), es posible tener un retardo significativo para recuperar grandes tablas MIB.

Otro retardo significativo suele presentarse desde que el agente procesa la solicitud entrante hasta que retorna la respuesta. Considerando este retardo para una gran cantidad de solicitudes, también puede transcurrir un tiempo apreciable hasta recuperar una tabla MIB completa.

4.2.2 ESCALABILIDAD

Esta sección discute los límites de escalabilidad que tiene la arquitectura de gestión SNMP. Primero se mencionan los problemas que pueden presentarse y posteriormente se explica cómo el uso herramientas puede evitar estos problemas de escalabilidad.

Los agentes SNMPv1 son generalmente incapaces de manejar muchas solicitudes simultáneamente, lo que puede ocurrir cuando el mismo agente es controlado por varias estaciones de administración. Si un agente es monitorizado con exceso, son necesarios recursos adicionales del sistema para poder manejar todas esas solicitudes simultáneas. Un agente SNMPv1 generalmente se ejecuta sobre un sistema con recursos limitados, ya que son muy costosos. Afortunadamente esta situación no se presenta en forma frecuente.

4.2.3 EXPANSIÓN

Conforme avance la tecnología y aumente la variedad de aplicaciones y dispositivos que se integren a las redes, será necesaria la adición de nuevas funciones a estas herramientas, lo cual conlleva a que el código fuente deba ser modificado.

Los programadores se encuentran con el inconveniente de tener que utilizar el lenguaje original con que fue desarrollado el software. Comprender el código escrito por otra persona puede llevar mucho tiempo. El software tiene que ser completamente o en parte recompilado, lo que implica detenerlo temporalmente hasta tener la nueva versión disponible.

Afortunadamente con el uso de visual basic, que mantiene una estructura organizada de los scripts utilizado en cada uno de los botones y eventos, permite una manipulación más sencilla del código que permita aumentar las herramientas, y el nivel de reconocimiento de dispositivos que incluyan no solo terminales de trabajo, además de telefonía IP, dispositivos de impresión, dispositivos de red, etc. Este fue uno de los principales motivos de haber realizado la programación en este lenguaje, pues es fácilmente adaptable, de tal manera que las nuevas herramientas puedan crearse en nuevos formularios que únicamente se agreguen a la aplicación, y se realice una nueva compilación.

En general una herramienta expansible de gestión debería posibilitar lo siguiente:

- Prototipación: deberían desarrollarse versiones de nuevas funciones utilizando herramientas que permitan una rápida implementación.
- Creación de nuevas funciones en tiempo de ejecución: las herramientas existentes deberían quedar disponibles durante la creación de nuevas funciones.
- Compilación sólo de nuevas funciones

- Distribución de la funcionalidad sobre múltiples plataformas.

4.2.4 REQUERIMIENTOS

Muchas de las herramientas tradicionales de gestión requieren demasiado hardware para trabajar apropiadamente. Como el hardware es costoso, en general se implementa una sola estación central para una red local entera. Los usuarios de esta estación pueden tener acceso desde la consola local.

Para nuestra aplicación podemos decir que los requisitos necesarios son un servidor HTTP para la publicación de informes y detección de errores y algunos scripts que deben ser escritos para servir de interfaz entre las herramientas de administración, el servidor y los dispositivos monitoreados, estas estaciones no requieren tantos recursos como normalmente lo piden otras aplicaciones, el rendimientos de los dispositivos no se altera, pidiendo finalmente como recursos mínimos:

- Sistema Operativo Microsoft Windows 2000 o posteriores (en caso de trabajar con Windows 98 únicamente actualizar el service pack)
- Memoria RAM de 32 MB
- Almacenamiento 1MB

Finalmente, para poder acceder a estas herramientas de gestión es necesario una red virtual que permita la conexión a la estación central que contenga la aplicación en caso de no mantenerla de forma portable.

Es de gran importancia mantener una interfaz de usuario estándar para el conjunto de herramientas de gestión basada en Web. Si cada vendedor instala en sus dispositivos sus propias herramientas de administración, se plantea el peligro de que cada uno tenga su propia interfaz de usuario. Consecuentemente el administrador del sistema tendría que aprender a usar diferentes herramientas de diferentes proveedores.

4.2.5 INTERFAZ AMIGABLE

Muchas herramientas convencionales de administración están equipadas con potentes interfaces gráficas de usuario (GUI) que además de requerir demasiada memoria para su ejecución despliegan demasiadas interfaces independientes encadenadas, es decir para poder acceder a los privilegios de usuario es necesario

entrar a la interfaz de usuario que se encuentra en la ventana de administración. Un ejemplo de GUI es Tk. Tk14 es una GUI para programar aplicaciones en lenguaje Tcl. Una herramienta que utiliza Tk es Tkined.

Nuestra aplicación de gestión está estructurada en pocas interfaces o ventanas, desplegables desde una sola ventana principal y en orden de eventos, de esta manera, se puede tener rápido acceso a cualquiera de las herramientas, la GUI es muy ligera, pues uno de los objetivos es no interferir con el rendimiento del sistema, la segunda interfaz que se utiliza son los navegadores web, que a partir de la configuración personalizada que nosotros deseemos implementar podemos aprovechar los distintos beneficios de las interfaces web tales como:

- hiperenlaces: una característica muy usada en interfaces gráficas como Windows para presentar páginas de ayuda.
- Botones clickeables.
- Formularios: permiten transferir una consulta simple y efectiva al servidor.
- Frames: permiten ver múltiples documentos en frames separados sobre una pantalla.
- Tablas.
- Bar charts.
- Javascripts tags: incrementa la interactividad de las páginas HTML.
- Java applets incrustados en páginas HTML.

Un navegador presenta páginas HTML, java script y java, ofreciendo así un rico conjunto de características GUI comparables con muchas GUI existentes para herramientas de administración. Pero éste no es el principal argumento para justificar el uso de esta tecnología, sino el hecho de que el mismo navegador que permite acceder a Internet permite controlar redes locales y remotas, y con las mismas características de GUI con las que uno está familiarizado.

4.2.6 PORTABILIDAD

Una de las mayores ventajas de nuestro sistema de herramientas de gestión es su portabilidad, pues además de poder acceder de manera remota al servidor y tener

almacenada la aplicación ahí, esta herramienta permite ser almacenada en una memoria USB o en alguna carpeta creada por el administrador de la red, sin interferir con la capacidad del dispositivo de almacenamiento pues el tamaño de la herramienta es absolutamente pequeño ocupando apenas las unidades de Kb, se propone 1 MB con el propósito de incluir scripts personalizados de la red.

4.2.7 SEGURIDAD

En cuanto a la aplicación se refiere, el primer tope con el que se encuentran es con al acceso a la aplicación, ya que si esta está almacenada en un servidor o estación de trabajo dedicada, permitirá el acceso solo un usuario al administrador de redes, de esta forma, desde el sistema operativo uno otorga permisos a los usuarios de quien puede tener accesos a las aplicaciones almacenadas, en caso de permanecer en una memoria USB, la seguridad disminuye pues esa memoria tiene mayores posibilidades de caer en manos de un usuario sin privilegios, en esta parte entra la seguridad del propio sistema pues cuenta con una contraseña otorgada por el programador, se puede dar de alta a más de un usuario para su acceso, no conforme con esto esta herramienta solo permite “n” intentos de acceso a los cuales si el usuario falla la aplicación queda bloqueada, ultimando este sistema de contraseña, se da tan solo “n” segundos para que el usuario entre al sistema de esta forma se evita que se use alguna aplicación de acceso Hacker, pues solo se otorga el tiempo suficiente para que el administrador ingrese sus datos.

En cuanto al acceso web se refiere una de las mayores desventajas de la administración SNMPv1 convencional es que proporciona un mecanismo trivial de autenticación, por lo que SNMP es básicamente mejor para monitorizar que para controlar. SNMPv2 resuelve este problema proporcionando mayores facilidades de seguridad, pero no es muy común hoy en día en las redes. La pobre seguridad de SNMPv1 obliga a los administradores a bloquear todo el tráfico SNMP proveniente de la red externa.

De esta manera los usuarios externos no pueden tener acceso a ninguna información concerniente al rendimiento o disponibilidad de la red. Debería haber un camino para poder exportar información segura sin requerir el intercambio de PDU SNMP fuera del sistema.

Una solución simple a este problema es exportar información a través de HTTP. El único camino para los usuarios externos es ver cualquier información a través de un servidor HTTP extendido. El servidor recibe solicitudes y retorna los datos que obtiene indirectamente de la red. Esto implica que algunos procesos regulares de

monitorización de dispositivos de red deben almacenar la información en alguna base de datos. El servidor HTTP puede ver sólo la información de estado desde esa base de datos, haciendo imposible a un cliente acceder directamente a un agente vía el servidor HTTP.

4.3 TECNOLOGIA UTILIZADA

Si bien la mayoría de las herramientas y elementos principales de gestión ya fueron introducidos o descritos en los capítulos anteriores, en esta sección se pretende mencionar aquellas herramientas que se utilizaron para la construcción de la aplicación:

4.3.1 WEB BROWSER HTML

Los documentos HTML pueden incluir tanto información estática como dinámica. Los documentos estáticos son almacenados en el servidor y generalmente no cambian, por el contrario, una página dinámica es creada instantáneamente como resultado de una consulta que el cliente envía al servidor.

HTML proporciona mucha flexibilidad a los clientes para interactuar con el servidor a través de los formularios. Esto se utiliza generalmente cuando un cliente necesita enviar una consulta de administración al Web (que puede ser una solicitud de un valor de un objeto gestionado o una operación para actualizar uno o más objetos).

Los visualizadores presentan documentos HTML incluyendo varios formatos de texto y gráficos, esta característica puede ser considerada como una potente interfaz distribuida de usuario para aplicaciones de gestión que residan sobre el servidor. De este modo, HTML permite la visualización de información de gestión a los usuarios en forma de gráficos y estadísticas.

4.3.2 SNMP

Los periféricos que tienen integradas las capacidades para SNMP ejecutan un paquete de software agente para administración, cargado como parte de un ciclo de arranque o incrustado en la memoria fija (firmware) del dispositivo. Estos dispositivos que tienen agentes SNMP se denominan “dispositivos administrados”.

Los dispositivos administrados por SNMP se comunican con el software servidor SNMP que está localizado en cualquier parte de la red. El dispositivo se comunica con el servidor de dos formas: por sondeo o por interrupción.

El administrador SNMP maneja el software general y las comunicaciones entre los dispositivos que utilizan el protocolo de comunicación SNMP. Todos los dispositivos administrados por SNMP contienen el software agente SNMP y una base de datos llamada Base de Información sobre la Administración (Management Information Base, MIB).

La MIB tiene 126 áreas de información sobre el estado del dispositivo, el desempeño del dispositivo, sus conexiones hacia los diferentes dispositivos y su configuración. El administrador SNMP consulta al MIB a través del software agente y puede especificar los cambios hechos a la configuración. La mayor parte de los administradores SNMP consultan a los agentes en un intervalo regular, 15 minutos por ejemplo, a menos que el usuario indique otra cosa.

El software agente SNMP por lo general es bastante pequeño (comúnmente de 64KB) dado que el protocolo SNMP es sencillo. SNMP está diseñado para ser un protocolo de sondeo (polling), lo que quiere decir que el administrador produce mensajes para el agente. Los mensajes SNMP se colocan dentro de un datagrama UDP y se enrutan vía IP (aunque podrían utilizarse otros protocolos).

El puerto UDP 161 se utiliza para todos los mensajes, excepto para los traps, que llegan al puerto UDP 162. Los agentes reciben sus mensajes del administrador a través del puerto UDP 161 del agente.

Dado que UDP no tiene conexiones, no existe confiabilidad inherente en el envío de los mensajes. SNMP utiliza el sondeo, lo que ocupa una considerable cantidad de ancho de banda. Los intercambios entre SNMP y su sucesor, CMIP, en el futuro tomarán decisiones más difíciles concernientes al protocolo de administración.

4.3.3 SOCKETS

Los sockets no son más que puntos o mecanismos que permiten que un proceso se comunique (emita o reciba información) con otro proceso, incluso estando en distintas máquinas. Esta característica de interconectividad entre máquinas hace que el concepto de socket nos sea de gran utilidad.

La forma de hacer referencia a un socket por los procesos implicados es mediante un descriptor del mismo tipo que el utilizado para referenciar ficheros. Debido a esta

característica, se podrán realizar redirecciones de los archivos de E/S estándar (descriptores 0, 1 y 2) a los sockets y así combinar entre ellos aplicaciones de la red.

La comunicación entre procesos a través de sockets se basa en la filosofía Cliente-Servidor: un proceso en esta comunicación actuará de proceso servidor creando un socket cuyo nombre conocerá el proceso cliente, el cual podrá "hablar" con el proceso servidor a través de la conexión con dicho socket.

4.3.4 VISUAL BASIC

Visual Basic es lenguaje nacido del BASIC (Beginner´s All-purpose Symbolic Instruction Code) diseñado para facilitar el desarrollo de aplicaciones en un entorno grafico (GUI-GRAPHICAL USER INTERFACE), que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar.

Entre las características que hacen a Visual Basic un buen entorno de programación para la generación de aplicaciones están.

Diseñador de entorno de datos: Es posible generar, de manera automática, conectividad entre controles y datos mediante la acción de arrastrar y colocar sobre formularios o informes.

Los Objetos Activex son una nueva tecnología de acceso a datos mediante la acción de arrastrar y colocar sobre formularios o informes.

Asistente para formularios: Sirve para generar de manera automática formularios que administran registros de tablas o consultas pertenecientes a una base de datos, hoja de cálculo u objeto (ADO-ACTIVE DATA OBJECT)

Asistente para barras de herramientas es factible incluir barras de herramientas es factible incluir barra de herramientas personalizada, donde el usuario selecciona los botones que desea visualizar durante la ejecución.

En las aplicaciones HTML: Se combinan instrucciones de Visual Basic con código HTML para controlar los eventos que se realizan con frecuencia en una página web.

La Ventana de Vista de datos proporciona acceso a la estructura de una base de datos. Desde esta también acceso al Diseñador de Consultas y diseñador de Base de datos para administrar y registros.

CAPITULO V

CASOS DE USO

5 CASOS DE USO

Como sabemos, la administración de redes es el proceso de controlar complejas redes de datos para maximizar su eficiencia y productividad. Para una mejor definición del ámbito de la Administración de Redes, la ISO16 la divide en cinco áreas funcionales:

- Administración de Fallas (Fault Management): es el proceso de localizar problemas o fallas en la red.
- Administración de Configuración (Configuration Management): es el proceso de descubrir y configurar los dispositivos críticos en la red, es decir, los que controlan su comportamiento.
- Administración de Seguridad (Security Management): es el proceso de controlar el acceso a la información en la red.
- Administración de Rendimiento (Performance Management): involucra la medida del rendimiento de la red, hardware, software y medios de comunicación.

Utilizando técnicas de Administración de fallas, el ingeniero puede localizar y resolver los problemas mucho más rápido que sin ellas.

Una herramienta de Administración de configuración puede ayudar al ingeniero a determinar, por ejemplo, que versión de software está instalada en cada dispositivo de la red.

La Administración de seguridad brinda una manera de monitorizar los puntos críticos de la red y proporcionar registros de auditoría para potenciar la seguridad.

Usando información del rendimiento de la red, el ingeniero puede asegurarse de que la red tiene la capacidad suficiente que los usuarios necesitan.

Con herramientas de administración de cuentas, el ingeniero puede conceder y remover permisos de acceso, o aprender sobre qué usuarios utilizan cuales recursos.

El propósito de este apartado es describir algunos casos de uso (desde la perspectiva de las áreas funcionales definidas por la ISO) que ayuden al ingeniero de red a interpretar la información proporcionada por las Herramientas de gestión basada en Web.

La aplicación desarrollada se centró en brindar información correspondiente a las primeras tres áreas funcionales, Administración de Fallas, Administración de

Configuración y Administración de Rendimiento, si bien es posible obtener información también aplicable a la Administración de Seguridad y Administración de Cuentas.

5.1 ADMINISTRACIÓN DE FALLAS

Las Herramientas de gestión permiten testear la conectividad al nivel de la capa IP mediante mensajes ICMP (pings). Este tipo de herramienta es muy utilizada, particularmente si los hosts o dispositivos no disponen de capacidades sofisticadas para generar eventos.

Una vez que se produjo la falla (falta de conectividad) la herramienta alerta al administrador generando automáticamente un e-mail con la dirección IP y detalles de la fecha y hora.

Si los dispositivos en la red son lo suficientemente sofisticados como para reportar eventos, las herramientas disponen de un proceso que permite capturar traps e informar automáticamente al administrador definido en la base de datos mediante un e-mail.

5.1.1 INFORMACIÓN DEL SISTEMA

Las Herramientas de gestión de redes permiten monitorizar con cierto nivel de abstracción, los objetos más relevantes para la administración de fallas, como sysObjectID (que ayuda a clasificar las entidades por vendedor o a identificar al fabricante), sysServices (que nos dice a qué nivel del modelo OSI opera el dispositivo) y sysUptime (que nos informa desde cuando el sistema se encuentra en funcionamiento).

Con la ayuda de una Base de Datos es posible definir una entrada en la tabla host para monitorizar el valor del sysUptime cada cierto intervalo, y determinar de esa manera en qué momento la entidad fue reinicializada.

Si al realizar una consulta sobre la tabla state se observa que el valor crece monótonamente, se entiende que el dispositivo está funcionando correctamente, si un valor es menor que el anterior, la entidad fue reinicializada desde el último muestreo. Si el valor disminuyó, se puede asumir que el agente se reinicializó o que el sysUptime alcanzó el valor máximo. Una forma de determinar cuál de estos

eventos ocurrió, es observar el tiempo transcurrido desde el último muestreo (por defecto 5 o 10 minutos) y el último valor conocido.

Todos estos objetos también pueden ser monitorizados desde el Mib Web Browser, permitiendo además el acceso a varias entidades simultáneamente.

5.1.2 INFORMACIÓN DE LOS PROTOCOLOS

Si bien todos los objetos del grupo IP son útiles para la Administración de Fallas, la aplicación Network Monitor permite monitorizar los que considero más relevantes.

Es posible consultar la tabla de rutas y descubrir por ejemplo, cómo fue aprendida la información de ruteo (objetos ipRouteTable, ipRouteProto). Otro grupo IP que puede ayudar también a resolver problemas es ipNetTo MediaTable. Estos objetos mapean direcciones de red IP a direcciones de otros protocolos. Un caso común es ARP (Address Resolution Protocol) tabla que mapea direcciones IP a direcciones MAC.

Además es posible acceder a todos los objetos de la tabla de rutas y tabla de traducción de direcciones desde el Mib Web Browser

Si una entidad está recibiendo y enviando errores SNMP no necesariamente implica problemas en la red, esto podría significar que la entidad no está manejando apropiadamente los paquetes SNMP.

El número y tipo de error también podría indicar que la entidad está recibiendo paquetes SNMP con errores desde dispositivos en la red. La solución de estos errores frecuentemente reside en la configuración de los agentes y estaciones administradoras.

5.2 ADMINISTRACIÓN DE CONFIGURACIÓN

Aunque es una aplicación realmente simple permite una enorme gama de opciones de configuración anterior al momento de la compilación es decir en el código, pues podemos indicar además de una gran variedad de constantes según el lugar donde se esté aplicando podemos establecer otras constantes de seguridad, entre ellas los usuarios que tendrán acceso a la aplicación y las restricciones, la ventaja de tener un código abierto es que se puede recompilar con la información que se crea necesaria

5.2.1 INTERFAZ

Dentro de la interfaz, existen una serie de variables que el administrador puede manipular, obviamente el sistema a monitorear ya sea por dirección IP o por el nombre del host, sin restricciones.

Sin embargo se otorga la posibilidad de cambiar el nivel de profundidad de información que se desea obtener del host esto se hace en la sección de “información del host”, obteniendo únicamente la que se cree sea necesaria para el diagnostico del equipo, los nombres de los informes tienen un nombre en blanco por lo que es posible catalogarlo como se crea necesario, entrando en esta sección la posibilidad de poder manipular el código para establecer que se guarden con una serie consecutiva de nombres que le facilite al administrador la identificación de estos ahorrando tiempo.

Los correos se pueden preestablecer en serie según la cantidad de personas que administren la red, sin embargo es posible el envió a una matriz de correos, por defecto se establece uno solo, de igual forma se establece si se requiere la adjunción de archivos, nuevamente por defecto se incluye la información del Host

5.3 ADMINISTRACIÓN DE RENDIMIENTO

Las Herramientas de Gestión proporcionan una manera simple de conocer información básica de un dispositivo, sin tener que conocer detalles de la información de administración. Aunque la información establecida se mide en aspecto a la memoria RAM, es posible, observar con el mismo método la capacidad de los discos de almacenamiento, controlando de esta forma las cuotas, en esta sección aun se tiene planeado implementar más opciones según los requerimientos de los lugares en los que se implemente el sistema.

CONCLUSIONES

Finalmente en base de lo aprendido durante el seminario de “Las tecnologías aplicadas en redes de computadoras” y en las experiencias y necesidades laborales de los elaboradores de esta tesina, desarrollamos este proyecto con el fin de permitirle a los administradores de sistemas poder monitorear y administrar de manera remota todos los nodos de la red que mantienen, permitiéndole además poder dar un soporte de primer nivel a dichos nodos.

Se desarrolló un set versátil de herramientas que permita la movilidad y acceso desde cualquier punto ya sea dentro o fuera de la red empresarial a la información necesaria para el diagnostico de fallas de la red y sus dispositivos, enfocándose principalmente en los segundos. Se tomaron en cuenta muchas experiencias de encargado de sistemas, administrador de red, o encargado del área de ingeniería, para obtener las necesidades de estos, siendo entre ellas, la gestión remota, el envió de información vía mail de aspectos de nuestros sistemas e incluso la publicación de estos en una página web.

Después de haber obtenido información sobre los protocolos que gestionan la información de la red, sobre las ventajas de cada uno de estos y de haber analizado la estructura de las aplicaciones actuales, surgió la idea de combinarlos, sus propiedades, módulos e instrucciones para poder obtener una herramienta de fácil implementación y uso, para ello se recurrió a una de los parámetros quizás un poco olvidados dentro de los sistemas operativos de Microsoft Windows, que son los sockets y que nos permiten la comunicación entre protocolos y aplicaciones.

Todas estas tecnologías utilizadas abren una gama inmensa de posibilidades de gestión y monitoreo no explotadas aun al 100%, pero que sin embargo se tiene la idea de continuar con el proyecto para poder ofrecer una herramienta definitiva que ayude a la administración de redes y sistemas. Nos queda una experiencia grata aunque aun no satisfecha en la elaboración del código de estas herramientas que al final cumplieron con los objetivos de la tesina.

Esperamos que este proyecto de pie al desarrollo de muchas otras aplicaciones y herramientas con las mismas bases y quizás con los mismos objetivos, que al fin y al cabo es a nosotros a quienes nos es útil, facilitando nuestras tareas, ahorrando tiempo y manteniendo una estructura organizada como administradores de redes.

BIBLIOGRAFIA

SNMP, SNMPv2 and CMIP: The Practical Guide to Network Management Standards.

William Stallings. Addison Wesley.

Ed. Addison-Wesley

Redes para Proceso Distribuido.

Jesús García Tomás

Ed. Ra-ma

Redes con Microsoft TCP/IP.

Drew Heywood.

Ed. Prentice Hall

Cisco Press: Academia De Networking De Cisco Systems: Guía Del Segundo Año (Ccna 3 Y 4).

Ed. Cisco Systems

Essential SNMP.

Mauro, Douglas R., Schmidt, Kevin

Ed. O'reilly & Associates, Inc

http://technet.microsoft.com/en-us/default.aspx

http://msdn.microsoft.com/es-mx/default.aspx

GLOSARIO

API
Interfaz de Programación de Aplicaciones.

CMIP
Protocolo de Administración de Información Común.

CMISE
Elemento de Servicio de Información de Gestión Común.

CMOT
CMIP sobre TCP/IP.

DNS
Sistema de Nombre de Dominios.

HTML
Lenguaje de Marcas de Hiper-Texto.

HTTP
Protocolo de Transferencia de Hiper-Texto.

KERNEL
Es el núcleo del sistema operativo.

MIB
Base de Información Gestionada.

OIW
Taller de Implementaciones OSI.

ONMF
Foro de Administración de Redes OSI.

OSI
Modelo de referencia de interconexión de sistemas abiertos.

PDU
Protocolo unidades de datos.

PROXY
El término proxy hace referencia a un programa o dispositivo que realiza una acción en representación de otro.

QoS
Calidad de servicio.

RFC#
Mecanismo de petición de comentario utilizado para referirse a algún protocolo o servicio de red.

ROSE
Servicio de Operaciones Remotas Básicas.

Service pack (Microsoft Windows)
Paquete de actualizaciones que la empresa Microsoft libera para complementar a sus sistemas operativos.

SMI
Estructura de Administración de Información.

SNMP
Protocolo Simple de Administración de Red.

SOCKET
Designa un concepto abstracto por el cual dos programas pueden intercambiarse información.

SPI
Interface Periférica Serial.

TCP/IP
Protocolo de Control de Transmisión / Protocolo de Internet.

UNIX
Sistema operativo portable, multitarea y multiusuario.

WINSOCK
Socket de Windows.