UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

ANALISIS CAUSA RAIZ DE EVENTOS DE RED DE UN OPERADOR DE TELECOMUNICACIONES MEDIANTE TIVOLI NETCOOL PRECISION IP

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRONICO

PRESENTADO POR:

Omar Aguilar Huacan

PROMOCIÓN
2003 - I

LIMA – PERÚ
2008

ANALISIS CAUSA RAIZ DE EVENTOS DE RED

DE UN OPERADOR DE TELECOMUNICACIONES

MEDIANTE TIVOLI NETCOOL PRECISION IP

Dedico este trabajo a:
Mis padres, cuyo ejemplo e iniciativa se reflejan
en cada paso importante de mi vida.
A mi hermano Boris, quien fue mi mejor amigo
y compañero durante mi vida universitaria.

SUMARIO

En el presente trabajo se describen el análisis, la planificación y la implementación de un sistema de análisis causa raíz de alarmas de red de un operador de banda ancha. El sistema de análisis causa raíz ayuda a la detección de alarmas principales críticas cuando se tienen una gran cantidad de alarmas presentes en un sistema de gestión de alarmas de red, para ello utiliza información de la topología de red así como un inventario detallado de todos los equipos presentes dentro de la red.

Inicialmente se realiza un estudio de la red a monitorizar y de cada uno de los modelos de equipos que conforman la red. Tal estudio sirve para el modelamiento correcto de cómo están conformados los equipos y de como se interconectan entre si, este modelamiento se utiliza en el servidor que realiza el análisis causa raíz.

El análisis causa raíz se realiza sobre alarmas de red que son gestionadas por un sistema centralizado de gestión de alarmas de red, por esta razón en el capítulo II se hace un estudio del funcionamiento del sistema centralizado de gestión de alarmas de red con la finalidad de comprender como se recolecta la información de alarmas.

En el capítulo III se hace un estudio de la arquitectura del servidor Precision IP para comprender: como se modelan la red y los equipos que la conforman, como se hace la monitorización de red a partir del modelo de red, como se hace el intercambio de información entre el sistema centralizado de gestión de alarmas y el servidor Precision IP, y finalmente se estudia el componente que realiza el análisis causa raíz.

Seguidamente en el capítulo IV se definen los tipos de análisis causa raíz, estos tipos de análisis consideran el impacto de una alarma dentro de la red tomando en consideración información de la topología red, modelos de equipos e información de las alarmas.

Finalmente en el capítulo V se describe la infraestructura de los servidores que albergan al sistema de gestión de alarmas y la manera como se interconectan a la red de banda ancha a monitorizar.

INDICE

INTRODUCCIÓN

En la gestión de red de un operador de telecomunicaciones se presentan muchas alarmas procedentes de los distintos equipos que conforman la red del operador. Muchas de las alarmas presentes llevan información importante que puede tener un gran impacto tanto en el tiempo de respuesta como en el tiempo de resolución frente a la aparición de problemas de red críticos, con tal motivo se debe tener un mecanismo con el cual se pueda localizar alarmas críticas principales que puedan ser origen de la aparición de otras alarmas críticas o alarmas secundarias.

La presencia de un gran número de alarmas críticas, tales como la caída de equipos de red dentro de una red monitorizada, puede conllevar a un mal análisis de los problemas principales de una red por parte de los gestores de red. Generalmente la aparición de una gran cantidad de alarmas se debe a algún problema principal crítico, la identificación de este problema permite que los gestores de red actúen eficazmente sobre él.

Un sistema de gestión y recolección de alarmas de equipos de red y un inventario detallado con información de topología de red son la principal fuente de datos que utilizará el sistema de análisis causa raíz de alarmas. El sistema de análisis causa raíz de alarmas será quien determine cual de las alarmas es el problema raíz que desencadena la aparición de otras alarmas críticas y secundarias, una vez detectado el problema raíz, el sistema se encargará de marcarlo, así como también de marcar a los problemas secundarios. Mapas topológicos y toda la información referente a las alarmas se visualizan en un entorno gráfico integrado con el sistema de gestión de alarmas.

En este trabajo se describe el diseño e implementación de un sistema de análisis causa raíz de alarmas RCA (Root Cause Analisys), éste sistema forma parte de un módulo denominado de gestión de fallas, éste módulo a su vez forma parte de un sistema OSS (Operationa Support System) que es otro sistema de mayor cobertura que realiza las

funciones de administración, inventario, ingeniería, planeamiento y reparación de las redes de servicio de telecomunicaciones.

CAPITULO I
DESCRIPCION DE LA RED Y ESTUDIO DE INTERFACES

Un sistema de gestión de alarmas supervisa las alarmas de todos los equipos de red del operador de banda ancha. La red del operador está compuesta por otras redes de diferentes tecnologías que tienen una función determinada dentro de toda la red del operador de banda ancha.

Con la finalidad de entender la monitorización y el tipo de interfaces que presenta los diferentes equipos de red, se hace una descripción de la red y de los equipos que conforman la misma.

1.1. Tecnologías Involucradas

La red del operador de banda ancha esta compuesta por varias tecnologías, en resumen se puede representar la red con la figura 1.1

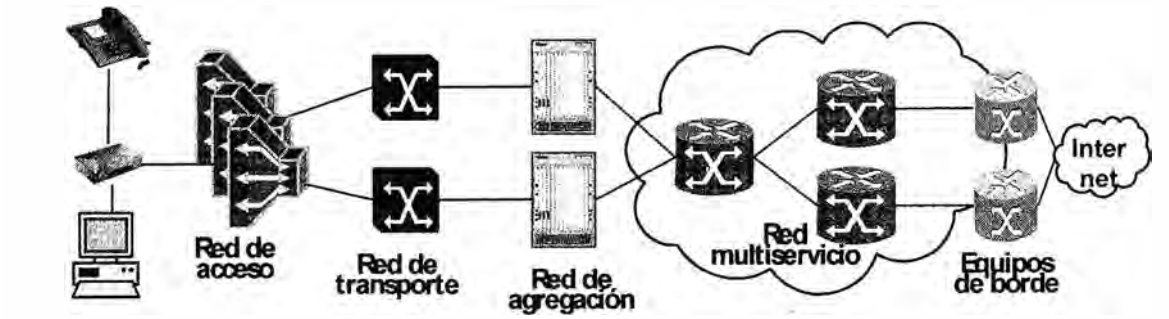

Figura 1.1 Esquema de red de operador de Banda Ancha

La figura 1.1 muestra una red de acceso, de transporte, de agregación, multiservicio y de equipos de borde.

1.1.1 Red de Acceso

La red de acceso está formada por los equipos que permiten concentrar enlaces ADSL en un único enlace ATM, permiten el paso del tráfico de los enlaces ADSL hacia la red de transporte. El equipo principal en esta red es el DSLAM (Multiplexor de Acceso DSL), los equipos utilizados en esta red son DSLAM Alcatel y DSLAM Huawei.

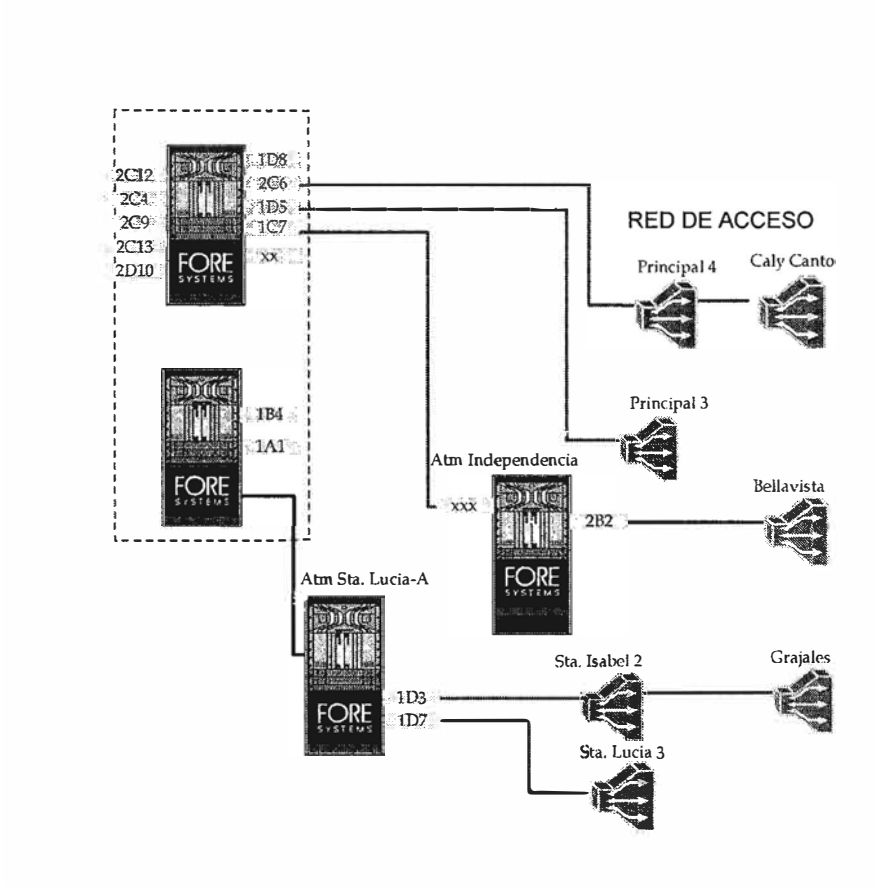

Figura 1.2 Red de acceso compuesta por DSLAM Alcatel y Huawei

1.1.2 Red de Transporte

Es la red que se encarga de llevar el tráfico de los DSLAM a la red de Agregación. La red de transporte está conformada por switches ATM CISCO y Marconi principalmente, los switches llevan el tráfico ATM de los DSLAM a los routers modelo ERX de Juniper. La velocidad de los enlaces son STM-1 en su mayoría.

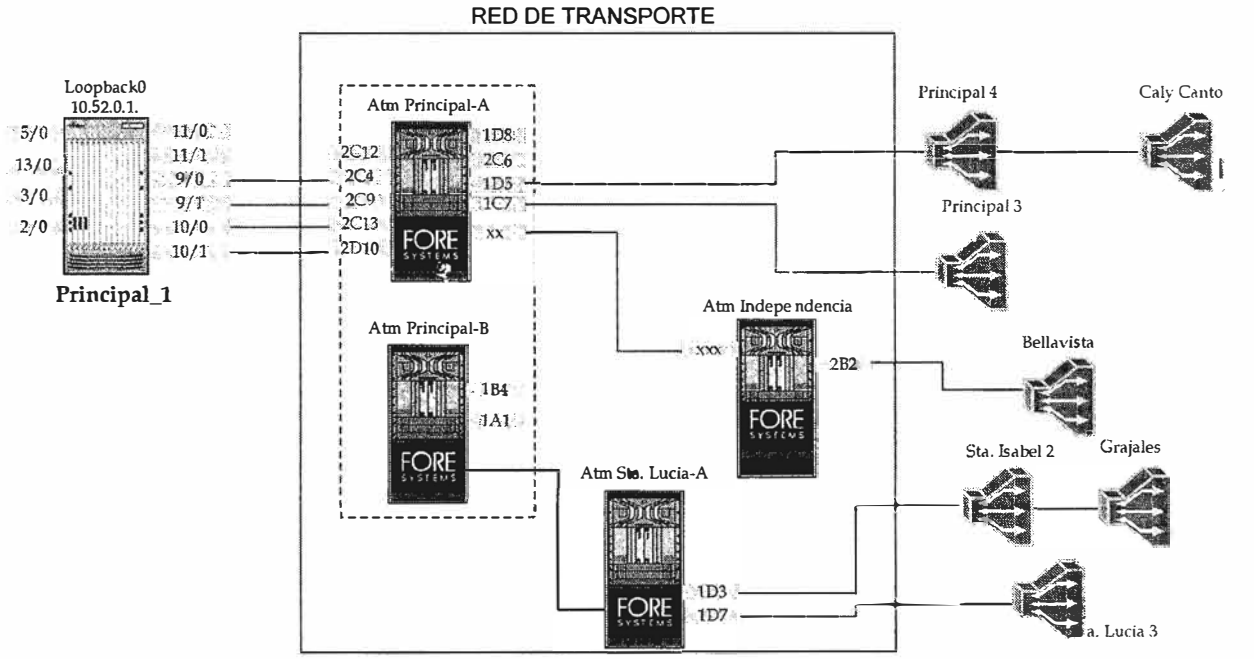

Figura 1.3 Red de transporte compuesta por los switches Marconi

1.1.3 Red de Agregación

Esta red está encargada de proporcionar el acceso del trafico ADSL proveniente de la red de transporte a la red multiservicio IP/MPLS. La red de agregación esta compuesta por equipos ERX de Juniper principalmente. Los equipos ERX reciben tráfico ATM y Gigabit Ethernet para luego direccionarlo a los equipos GSR a través de enlaces ATM, Gigabit Ethernet o POS según sea el caso.

Los equipos de esta red se encargan de hacer la validación y autorización de los usuarios para luego asignarles un ancho de banda y una calidad de servicio correspondientes.

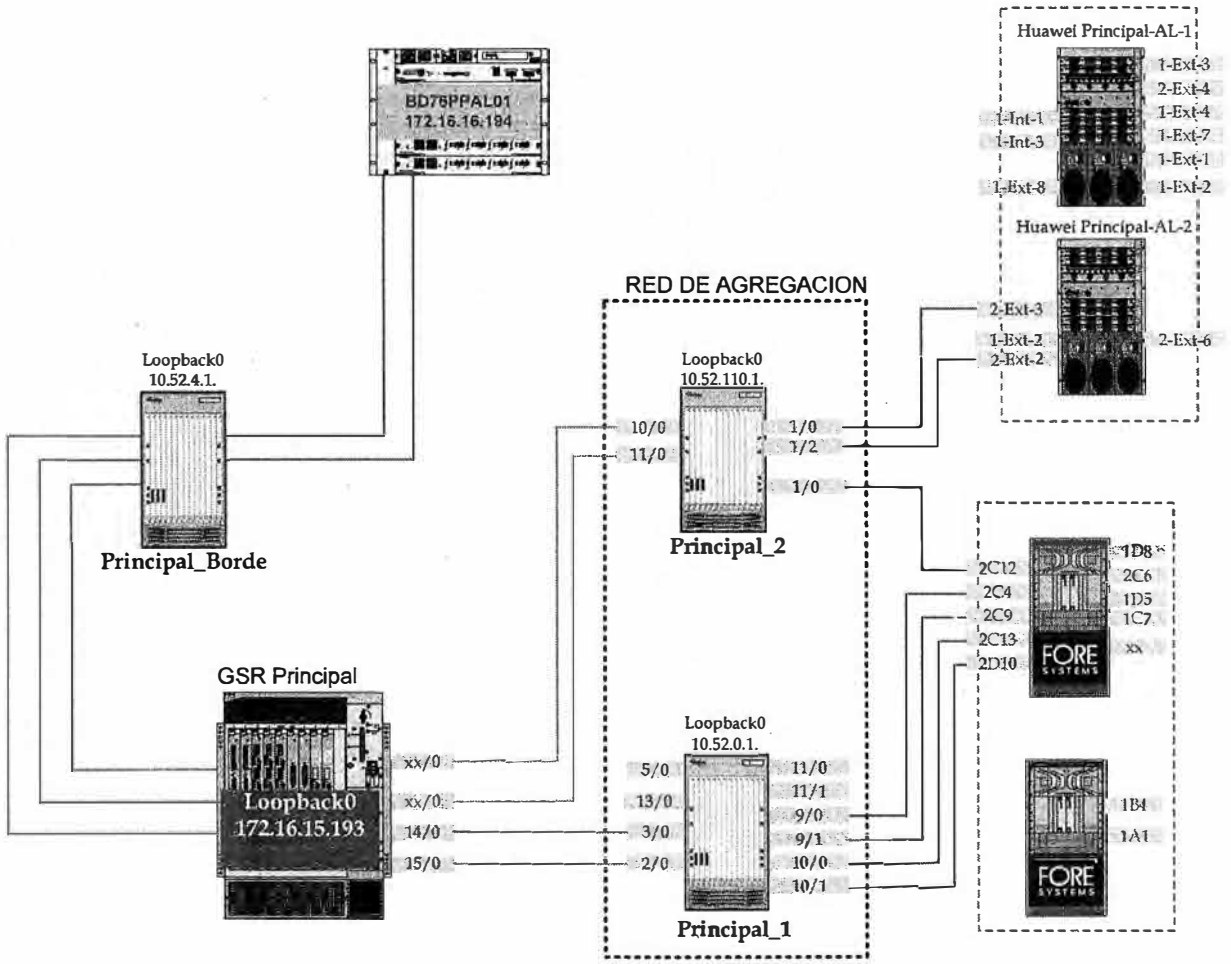

Figura 1.4 Red de agregación compuesta por routers Juniper ERX

1.1.4 Red Multiservicio

Es una red que permite el acceso de diferentes tipos de tráfico tales como ATM, IP y Ethernet, y está soportada por una red de transporte MPLS. Esta red está conformada principalmente por equipos GSR (Router Switch Gigabit) de Cisco. Los equipos GSR pueden recibir distinto tipo de tráfico, provenientes principalmente de los routers de modelo ERX de Juniper. Parte de la topología del operador se muestra en la figura 1.5

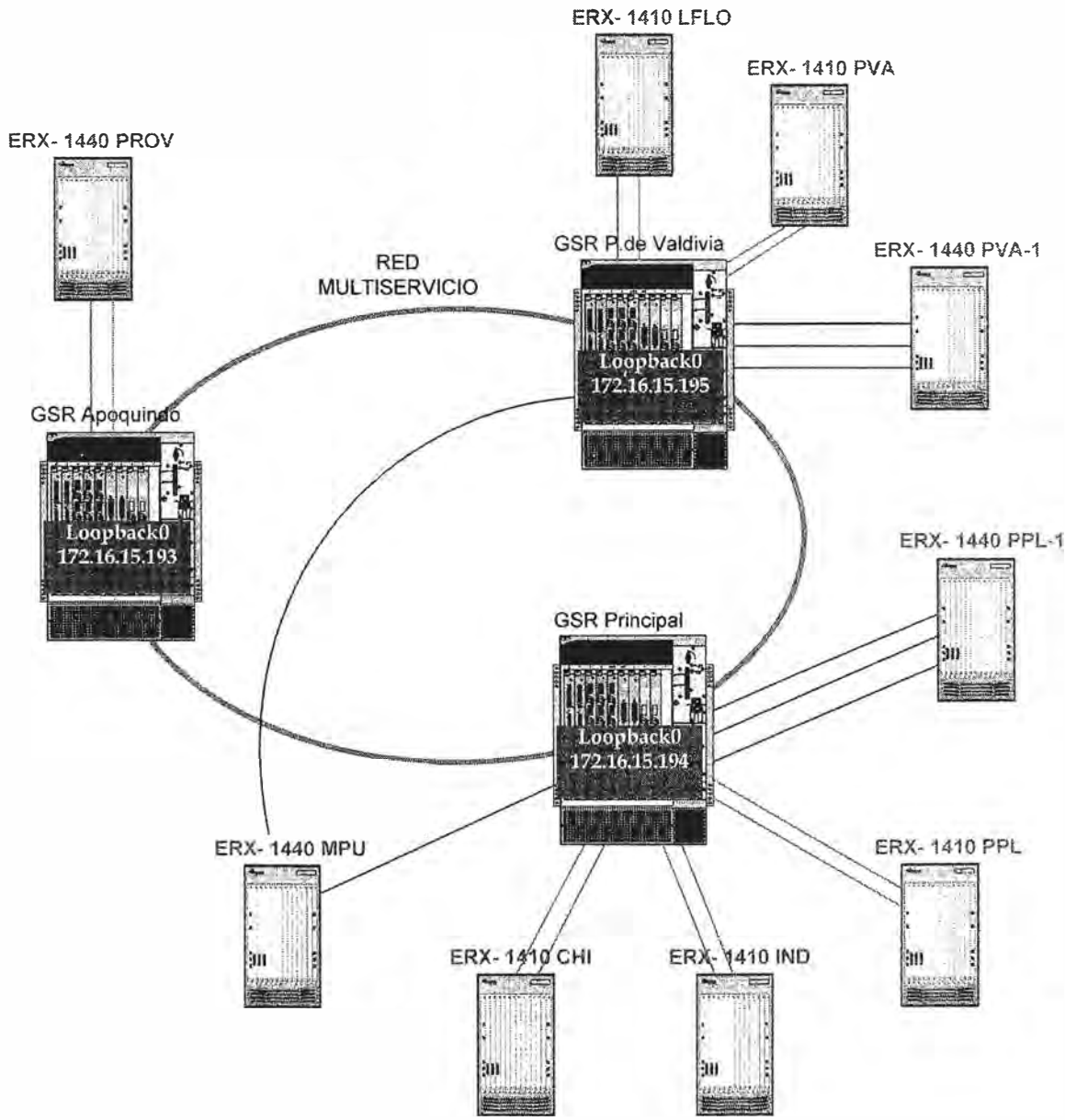

Figura 1.5 Red Multiservicio compuesto por Router Cisco Familia 12000

1.2 Especificación de interfaces

Como ya se mencionó anteriormente, existen distintos tipos de interfaces para los diferentes equipos. La mayoría de los equipos tienen diferentes notaciones para identificar sus interfaces. Si se sigue una convención para la notación de contención de interfaces, es decir como identificar si una tarjeta contiene otras tarjetas y como una tarjeta contiene varios puertos, cada tarjeta y puerto tendría un identificador. Así por ejemplo el puerto 12 de la tarjeta 2 del equipo A sería identificado por A[2 [12]]. Cada tarjeta y puerto de un equipo tendría un índice que lo identificaría dentro del equipo.

Los equipos pueden tener varios niveles para ser identificados, así los niveles que se pueden identificar dentro de un equipo se muestran el la tabla 1.1

Número de niveles	Rack	Chasis	Tarjeta	Puerto
4 niveles	X	X	X	X
3 niveles		X	X	X
2 niveles			X	X

Tabla 1.1 Número de niveles para la identificación de interfaces

En lo que resta de este capítulo se hará una descripción de todas las interfaces de los diferentes modelos de equipos que conforman la red del operador, también se detallará como se identifican los puertos de todas las interfaces de los equipos.

1.2.1 Multiplexor de Acceso de Línea Digital de Abonado Alcatel

Existen 3 modelos de equipos DSLAM Alcatel: SD, UD y HD. Todos los DSLAM Alcatel tienen la misma configuración para los racks y chasis, dicha configuración se muestra en la figura 1.6. La diferencia que existe entre estos modelos de DSLAM es el número de tarjetas y puertos que puede contener cada chasis.

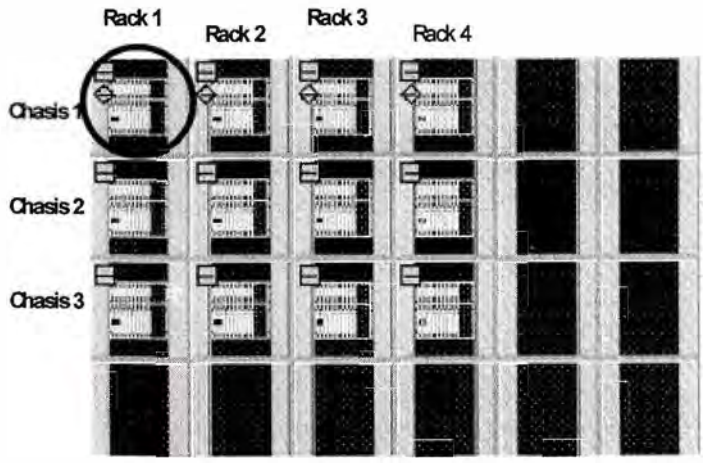

Figura 1.6 Configuración de racks y chasis para DSLAM Alcatel

a) DSLAM Alcatel SD

Estos equipos tienen 12 tarjetas, cada tarjeta posee 4 puertos, la configuración de un chasis de un DSLAM Alcatel SD se muestra en la figura 1.7

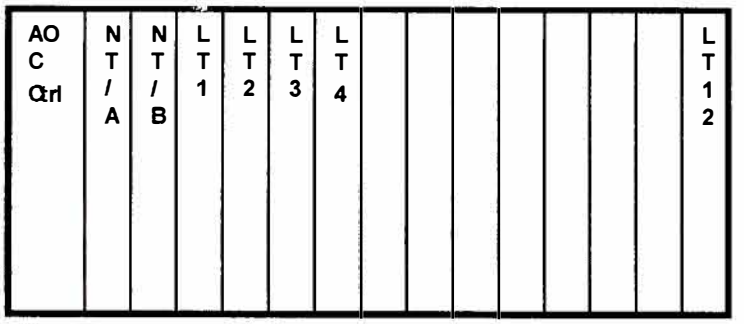

Figura 1.7 Chasis DSLAM Alcatel SD

b) DSLAM Alcatel UD

Estos equipos tienen 5 tarjetas en un chasis y 24 puertos por cada tarjeta. La configuración física de un chasis se muestra en la figura 1.8

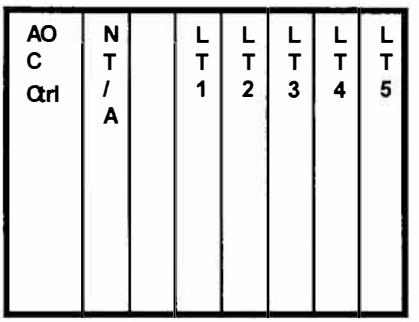

Figura 1.8 Chasis DSLAM Alcatel UD

c) DSLAM Alcatel HD

Estos equipos contienen 16 tarjetas por cada chasis y 12 puertos por cada tarjeta, tal como se muestra en la configuración física de la figura 1.9

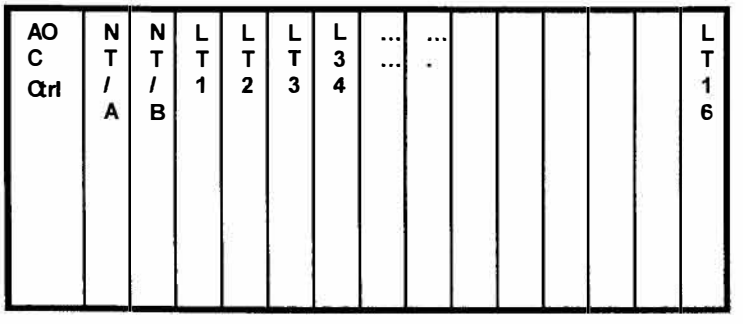

Figura 1.9 Chasis DSLAM Alcatel HD

1.2.2 Multiplexor de Acceso de Línea Digital de Abonado Huawei

Otro tipo de DSLAM son los DSLAM Huawei, las interfaces se detallan de acuerdo a la configuración que muestra la figura 1.10

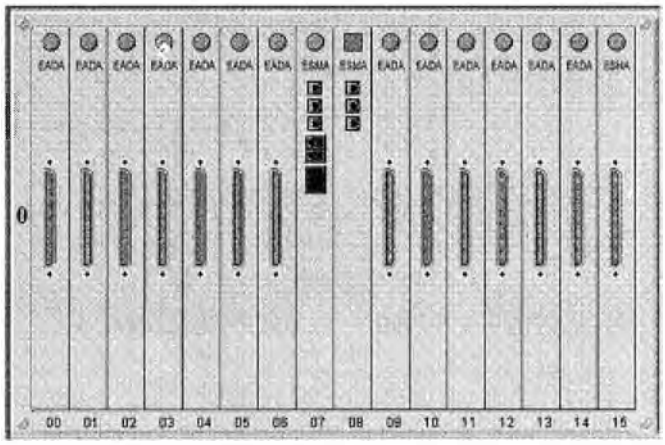

Figura 1.10 DSLAM Huawei

Estos equipos tienen un chasis, cada chasis contiene 16 tarjetas numeradas desde el cero, las tarjetas 7 y 8 son tarjetas de control, el resto de tarjetas contienen puertos de clientes, una tarjeta de cliente tiene 48 puertos, la numeración de los puertos comienza desde cero.

1.2.3 Switches Marconi

En estos equipos se disponen de tres modelos: ASX200BX, ASX1000 y ASX4000. Todos los modelos tienen una misma manera de identificar a un puerto. Un puerto se puede identificar mediante los identificadores de chasis, tarjeta y puerto. Chasis puede tomar valore enteros 1:n, las tarjetas tienen los valores A, B, C y D. Dependiendo del modelo de la tarjeta, cada tarjeta puede tener 1, 4 ó 16 puertos. Como ejemplo un puerto estaría especificado de la siguiente manera 'NOMBRE_EQUIPO[1 [B [0]]]'

a) ASX200BX

Este modelo de switch tiene una configuración de un chasis y tarjetas tal como se muestra en la figura 1.11

1A	1B
1C	1C
X1	Y1

Figura 1.11 Configuración de chasis y tarjetas ASX200BX

b) ASX1000

Este switch también tiene un chasis solamente, la configuración se muestra en la figura 1.12

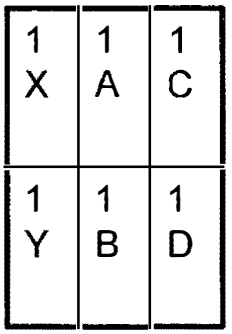

Figura 1.12 Configuración de chasis y tarjetas ASX1000

c) ASX4000

Este switch es de mayor capacidad por que tiene 2 chasis, su configuración física se muestra en la figura 1.13

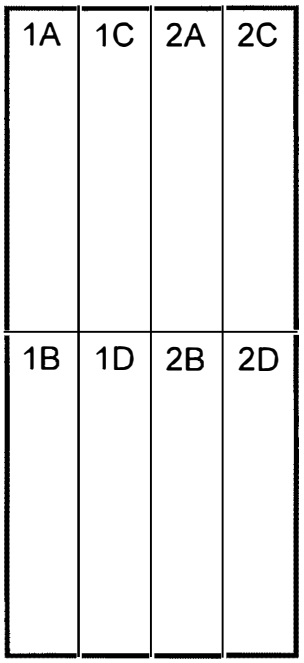

Figura 1.13 Configuración de chasis y tarjetas ASX4000

1.2.4 Routers Juniper

Se dispone de 2 modelos de estos equipos: ERX1410 y ERX1440. La manera de identificar los puertos se hace mediante tarjeta y puerto.

a) ERX1410

Estos equipos pueden contener tarjetas de diferentes interfaces tales como ATM, POS, Fast Ethernet y Gigabit Ethernet. Los puertos de las tarjetas ATM se enumeran de 0 a 4 al igual que los puertos de las tarjetas POS, los puertos de las tarjetas Fast Ethernet se enumeran de 0 a 7 y las tarjetas Gigabit Ethernet sólo tienen el puerto 0. De esta manera un puerto de los equipos ERX sería 'NOMBRE_EQUIPO[1 [0]]'

Un diagrama de enumeración de las tarjetas se muestra en la figura 1.14:

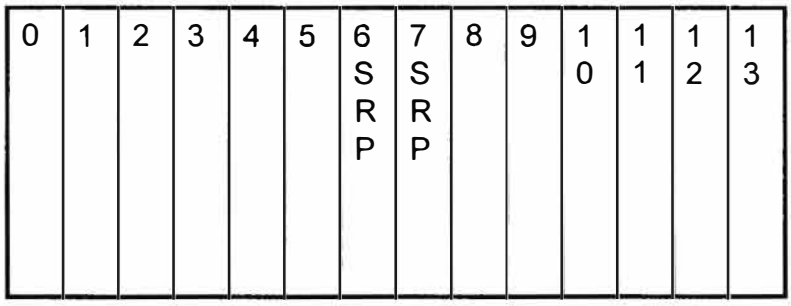

Figura 1.14 Configuración de tarjetas ERX1410

b) ERX1440

Estos equipos solo pueden tener tarjetas Gigabit Ethernet y tienen la misma configuración de tarjetas que la mostrada en la figura 1.14.

1.2.5 Routers Cisco Familia 7500

Esta familia de equipos Cisco tiene 3 niveles para identificar a un puerto: Tarjeta, Adaptador de Puerto y Puerto. El equipo contiene 7 u 8 tarjetas que pueden ser de diferentes interfaces: ATM, POS, GE y FE. Cada tarjeta esta conformado por dos adaptadores de puerto cuya numeración empieza en 0. Un puerto en particular tendría la siguiente notación 'NOMBRE_EQUIPO[0 [2 [0]]'. Un diagrama de las tarjetas y adaptadores de puerto se muestra en la figura 1.15

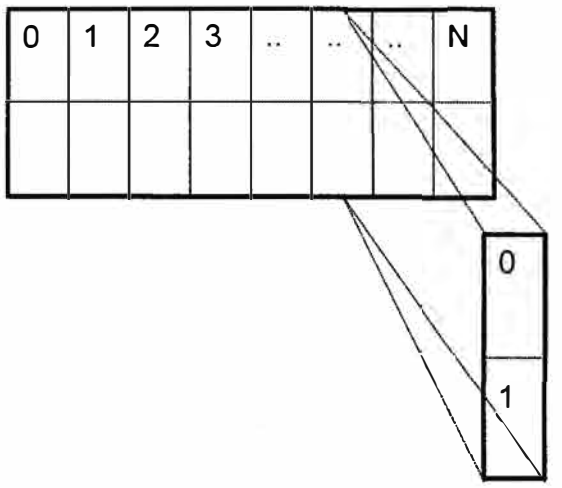

Figura 1.15 Configuración de tarjetas Cisco Familia 7500

1.2.6 Router Switch Cisco Familia 12000

Esta familia de dispositivos Cisco son los Router Switch Gigabit y tiene 3 modelos dentro de la red: Cisco 12416, Cisco 12016 y Cisco 12008.

a) Cisco 12416

Estos dispositivos soportan las interfaces: ATM, POS, FE y GE. Las tarjetas se enumeran desde 0 hasta 15 y los en numero de puertos por tarjeta depende del tipo de interfase, un puerto estaría denotado por 'NOMBRE_EQUIPO[TARJETA [PUERTO]]'. Un diagrama de las tarjetas se muestra en la figura 1.16

0	1	2	3	4	5	6	7 RP
8 RP	9	10	11	12	13	14	15

Figura 1.16 Configuración de tarjetas Cisco 12416

b) Cisco 12016/12008

A diferencia de los dispositivos Cisco 12416 éstos dispositivos soportan las interfaces: ATM, POS y FE. La numeración de tarjetas va desde 0 hasta 7, al igual que en el caso anterior, el número de puertos por tarjeta varía de acuerdo al tipo de interfaz. La especificación de un puerto estaría dada de igual manera que en el modelo Cisco 12416. Un diagrama de las tarjetas se muestra en la figura 1.17

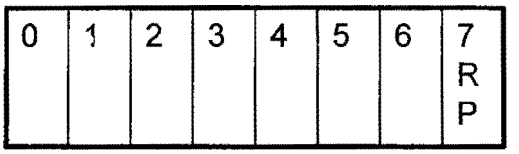

Figura 1.17 Configuración de tarjetas Cisco 12016/12008

CAPITULO II
RECOLECCION DE EVENTOS DE RED

2.1 Introducción

La red de un operador de telecomunicaciones esta conformada por muchos dispositivos que son de diferentes fabricantes y de distintos modelos, por lo general cada fabricante tiene un sistema de gestión de alarmas propietario que administra y recolecta todas las alarmas que son generados por los equipos de red del mismo fabricante. Al tener varios sistemas de gestión de alarmas, la respuesta a la aparición de una alarma de un equipo de red crítico y su consecuente resolución se ve afectada, para una gestión eficaz se hace necesario contar con un sistema centralizado de recolección de alarmas de equipos de red.

La finalidad de un repositorio centralizado de alarmas es la de recolectar traps, mensajes, etc. de equipos de red de diferentes fabricantes y distintos modelos en un solo sistema. Además, el sistema debe de ser flexible en cuanto a los métodos de recolección de alarmas de red.

El sistema de recolección de alarmas utilizado en este proceso tiene la función de recolectar alarmas de distintas formas, ya sea mediante recolección y revisión de traps, mensajes syslog, archivos de log, uso de protocolos propietarios de algunos fabricantes, etc.

2.2 Arquitectura del Repositorio de Alarmas

Netcool Omnibus es un conjunto de programas que facilita la administración y recolección de alarmas de equipos de red de diferentes fabricantes y de distintas tecnologías, está compuesto principalmente por una base de datos en memoria, denominado ObjectServer, donde se almacena la información de las alarmas de red; otro grupo importante de componentes son las sondas, que se especializan en la recolección de alarmas de los

distintos equipos de red. Todos estos componentes y sus relaciones se representan en la figura 2.1.

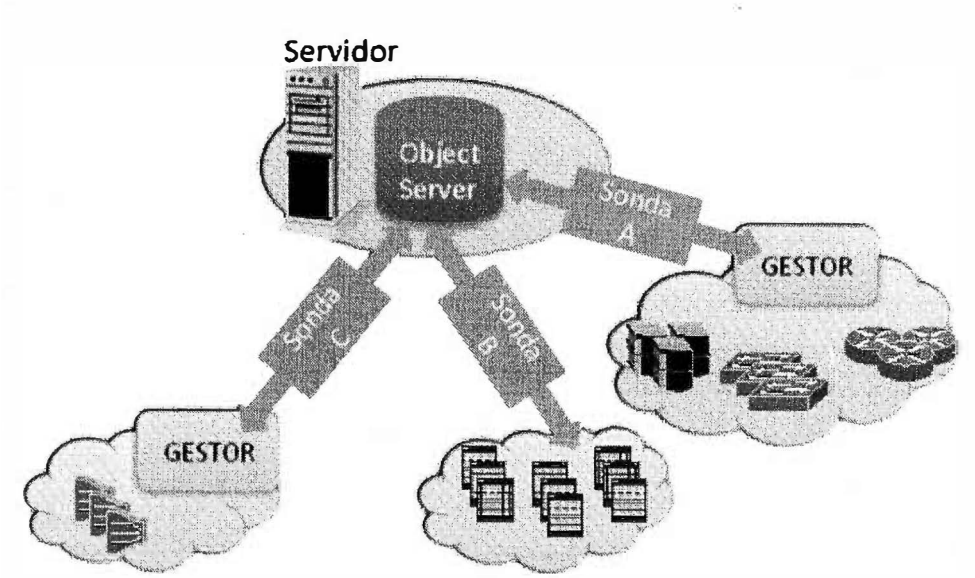

Figura 2.1 Arquitectura del repositorio de alarmas centralizado

2.2.1 Base de Datos de Alarmas

Es un servidor de base de datos en memoria y es el corazón de Netcool Omnibus donde todas las alarmas son almacenadas y gestionadas. A la base de datos de alarmas se le conoce como ObjectServer, y consolida la información de eventos tales como fallas, alarmas, mensajes de advertencia que son recolectados por las sondas desde diversos sistemas de gestión.

La base de datos tiene un modelo de datos que contiene varias columnas, muchas de las columnas son para uso interno del servidor mientras que otras columnas son importantes para el correcto funcionamiento de la base de datos, en la tabla 2.1 se listan los principales campos, donde se almacena la información de las alarmas de los equipos de red, y se presenta una breve descripción.

Tabla 2.1 Campos del ObjectServer

NOMBRE CAMPO	DESCRIPCIÓN
Identifier	Clave para sobre escribir varias ocurrencias de una misma alarma.
Serial	Identificador único secuencial
Node	Identifica el origen de la alerta
NodeAlias	Alias para el nodo, como una dirección IP
Manager	Nombre descriptivo del gestor que genera la alarma
Agent	Nombre descriptivo del sub-agente
AlertGroup	Grupo del manager o el agent.
AlertKey	Clave descriptiva con referencia a la alerta
Severity	Severidad de la alerta
Summary	Resumen de la alerta.
StateChange	Hora en que la alerta fue modificada por última vez según la hora del InfoServer local.
FirstOccurrence	Marca de tiempo de la hora origen de creación de la alarma.
LastOccurrence	Marca de tiempo de la hora de origen de la última ocurrencia de la alarma
Type	Tipo de alerta (Problema=1, Resolución=2 o desconocido=0)
Tally	Numero de veces que la alerta ha ocurrido
OwnerUID	Identificador del usuario al que pertenece una alerta
Acknowledged	Indica si una alerta ha sido reconocida
EventId	Identifica en nivel de severidad para el análisis causa raíz
LocalNodeAlias	Identificador del componente afectado para el análisis causa raíz
NmosObjInst	Identificador del dispositivo en modelo de red de Precision
NmosSerial	Serial de la alarma raíz si la alarma está inhibida
NmosCauseType	Tipo de alerta para el análisis causa raíz (raíz=1, síntoma=2, desconocido=0)

Los tres últimos campos NmosObjInst, NmosSerial y NmosCauseType, son de vital importancia para el análisis causa raíz, pues esto tres campos determinan si la alarma problema es de tipo raíz, síntoma o desconocido (ninguno de los dos anteriores).

2.2.2 Herramientas de Escritorio

Las herramientas de escritorio son un grupo de programas que permiten visualizar el estado de las alarmas en el ObjectServer, éstas pueden ser utilizadas desde una consola de Windows o Unix.

El estado de las alarmas puede ser visualizado así por los operadores de red. Este interfaz permite a su vez interactuar con las alarmas, pudiéndose modificar algunos valores para una mejor administración de los eventos de red.

2.2.3 Sondas de Recolección de Alarmas

Son programas que actúan como agentes de adquisición de datos, pueden recolectar información de las alarmas a través de traps SNMP o por otros mecanismos. Las sondas pueden obtener información de alarmas de diferentes fuentes, para luego enviar y escribir dichos eventos en el ObjectServer.

Las sondas permiten que los operadores puedan recolectar e interpretar información de diferentes sistemas de gestión de red, dispositivos de telefonía, redes de datos, LAN y WANs, y aplicaciones. Pueden recolectar información MIB de distintos fabricantes tales como Cisco, Cabletron, Bay Networks, etc. También hay sondas para ASCII, TL1 y estándares de Unix.

En las siguientes subsecciones describiremos brevemente las sondas más importantes:

a) **Sonda Multihilo de Traps**

   Se le conoce como la sonda mttrapd, es un proceso que actúa como un demonio del sistema que escucha la llegada de traps en los sockets udp y tcp.

b) **Sonda de Mensajes en Archivo**

   Es conocida como la sonda Glf, obtiene la información de eventos desde archivos log, puede separar los mensajes mediante caracteres especiales que pueden denotar separación de campos y separadores de línea.

c) **Sonda Mensajes de Sistema**

   Es la sonda Syslog y recopila información de las alarmas desde mensajes syslog recolectados por el demonio syslogd del sistema operativo Unix.

d) Sonda para el Gestor de red HP

Es la sonda nnm7 y ejecuta un subproceso dentro de un servidor que está ejecutando el sistema de administración de nodos de red de HP (conocido en el mercado como HP Open View). Esta sonda, al ejecutar un subproceso dentro del servidor, recolecta la información que proporciona el sistema de gestión HP y luego escribe esta información en el ObjectServer.

2.3 Recolección y procesamiento de Alarmas

Como ya se mencionó anteriormente, las sondas son los procesos que recolectan las alarmas, procesan la información de éstas y finalmente escriben la información relevante de estas alarmas en el ObjectServer. Cada una de las sondas tiene un archivo de reglas llamado archivo rule, donde se definen las reglas de cómo procesar la información proveniente de la recolección de alarmas, es decir, las reglas definidas en los archivos rules definen como se van a llenar cada uno de los campos del ObjectServer con la información proveniente de las alarmas.

En el sistema descrito (figura 2.2), se cuenta con diferentes sondas que recolectan información de diferentes fuentes, a continuación explicamos como, cada una de estas sondas, obtienen la información de alarmas de distintos fabricantes y sistemas de gestión de redes.

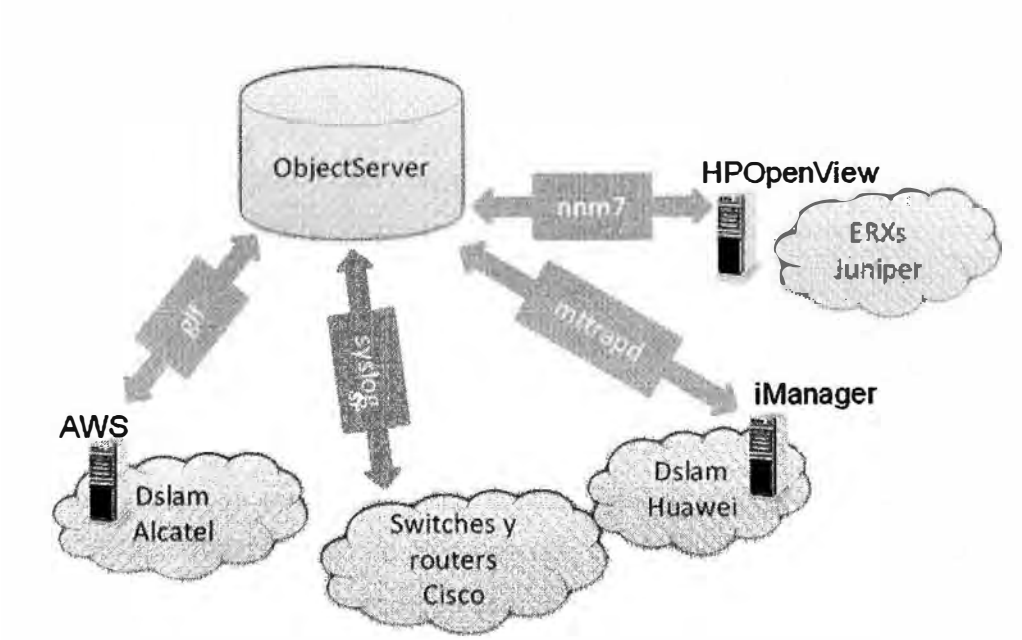

Figura 2.2 Recolección de eventos de red.

2.3.1 Recolección de Mensajes de Sistema

La sonda syslog obtiene información de los mensajes syslog generados por todos los dispositivos Cisco que están dentro de la topología de red considerada para la monitorización. Todos los equipos Cisco se configuran de tal manera que envíen los mensajes syslog al servidor donde esta el demonio syslogd quien es quien luego genera el archivo log que finalmente es leído por la sonda syslog.

La sonda syslog divide la información de los mensajes syslog en partes mas pequeñas, obteniendo varios campos en función de un carácter especial que denota la separación de campos dentro del mensaje syslog, por último son éstos campos los que son procesados y escritos en el ObjectServer finalmente.

2.3.2 Recolección de Alarmas de Equipos Alcatel

Para la recolección de Alarmas de red de los equipos Alcatel se ha utilizado una configuración especial que consta de dos partes.

Primeramente se cuenta con un script cuya finalidad es conectarse con el gestor propietario de equipos Alcatel (AWS) y procesar la información de las alarmas administradas por este gestor. La conexión y el mecanismo de petición de alarmas se consiguen mediante el uso del protocolo propietario TL1 de Alcatel.

Una vez conectado el script, obtiene la información de todos los eventos de red almacenados en el gestor. Con la información de los eventos, el script hace un procesamiento de los eventos dándole un formato especial de campos separados por un carácter especial para cada alarma y también separa las alarmas por otro carácter que representa un fin de línea, esto se hace con la finalidad de poder generar un archivo log uniforme que finalmente pueda ser leído y procesado por la sonda Glf.

Finalmente se utiliza la sonda Glf para procesar el archivo log generado por el script anterior. La sonda Glf procesa la información de todas las alarmas y seguidamente escribe la información relevante en cada uno de los campos del ObjectServer.

2.3.3 Recolección de Alarmas Desde un Gestor Propietario

Muchos de los equipos de red del operador también son administrados por otros entornos de gestión de alarmas de red, uno de ellos es el HP OpenView. Como el sistema a

implementarse es centralizado, se cuenta con otra sonda, la sonda nnm7, que se especializa en la recolección de eventos directamente desde el gestor HP OpenView.

HP OpenView es el gestor de eventos de red de varias plataformas tales como switches Marconi y routers ERX de Juniper. La sonda nnm7 es un proceso que se conecta al gestor de eventos de red HP OpenView 7.0 y obtiene toda la información de eventos de red de los equipos gestionados, luego se encarga de escribir la información de los eventos en las tablas del ObjectServer.

2.3.4 Recolección de Traps

Uno de los mecanismos de recolección de alarmas de red es mediante el uso del protocolo SNMP (Simple Network Management Protocol), muchos de los fabricantes implementan este protocolo para que sus dispositivos puedan ser gestionados. En el sistema centralizado también se cuenta con una sonda que hace la recolección de alarmas a través de SNMP.

La sonda mttrapd es la que procesa toda la información de traps snmp que es generada por los distintos dispositivos de la red. En el entorno generado, los equipos DSLAM Huawei envían traps a la sonda mttrapd a través de su gestor iManager, la sonda mttrapd procesa y escribe la información proveniente de los traps en los campos del ObjectServer.

2.4 Etiquetado de Eventos Para el Procesamiento en PRECISION IP

Todas las sondas tienen el mismo mecanismo para escribir la información de alarmas de red en el ObjectServer. La escritura de la información en los campos del ObjectServer se hace a través de un archivo de reglas (archivo rules), en el archivo de reglas se puede hacer un procesamiento básico de strings y enteros, además de que se pueden utilizar algunas condiciones lógicas básicas. Luego del procesamiento de la información de las alarmas, se asigna la información a cada uno de los campos del ObjectServer.

En la tabla 2.1 se ha descrito los principales campos que tiene la tabla de alarmas del ObjectServer, de esta tabla los campos EventId y LocalNodeAlias son dos campos importantes que se utilizan para proporcionar información de las alarmas para el análisis causa raíz del servidor Precision.

2.4.1 Campo Identificador de Eventos

Es el campo EventId del ObjetcServer. Para el análisis causa raíz de los eventos, el campo EventId se utilizará para identificar el tipo de alarma y también para asignarle una precedencia que proporcionará un nivel de prioridad sobre otras alarmas que pertenezcan a un mismo equipo.

Las alarmas de los equipos de red pueden tener información de fallas, de estados o de advertencias de los diferentes interfaces del equipo como tarjetas contenidas dentro del equipo, tarjetas contenidas dentro de otras tarjetas y puertos contenidos dentro de una tarjeta. Así por ejemplo puede haber alarmas que informen sobre la caída de un equipo o simplemente alarmas informativas sobre el estado del equipo; similarmente, las alarmas también pueden tener información sobre tarjetas y puertos dentro de un equipo.

En el diseño de análisis causa raíz se ha identificado 7 tipos de alarmas, a cada tipo le corresponde un valor en el campo EventId (Tabla 3.2).

Tabla 3.2 Valores del campo EventId

INTERFACE	EventId	DESCRIPCION
Equipo	NmosPingFail	Falta de respuesta frente a un ICMP request
	nodeupdown	Caída de equipo, informado desde el mismo equipo o desde un gestor
	nodesnmpfail	Disponibilidad de tráfico equipo menor a 100%
Tarjetas	shelfupdown	Caída de tarjeta segundo nivel o superior
	slotupdown	Caída de tarjeta primer nivel
Puertos	portupdown	Caída de puerto
	""	Alarmas informativas, pueden ser de cualquier parte del equipo (con menor prioridad)

Según la descripción de la tabla 3.2, cada valor del campo EventId está asociado a un tipo de interfaz dentro del equipo, los valores NmosPingFail, nodeupdown y nodesnmpfail están asociados al equipo en si, denotan que ha habido una caída del equipo, los valores shelfupdown y slotupdown están asociados a tarjetas y denotan la caída de una tarjeta, las tarjetas pueden contener tarjetas y además contienen puertos, el valor portupdown está asociado a la caída de puertos de comunicación. Cabe resaltar que de acuerdo al nivel de jerarquía dentro del equipo una alarma de mayor jerarquía dentro del equipo tendría mayor prioridad sobre otra alarma de menor jerarquía dentro del mismo equipo.

Sólo las alarmas que contengan algún valor diferente de vacío darán inicio al análisis causa raíz.

2.4.2 Campo Alias de Parte Afectada

Es el campo LocalNodeAlias del ObjectServer, este campo es una fuente de información para el análisis causa raíz, proporciona información de la parte del equipo de la cual proviene la alarma, es decir va a indicar si la alarma contiene información del equipo, de una tarjeta o de un puerto.

Con la finalidad de dar un formato único para cada interfaz y para cada equipo dentro de la red, se ha empleado el formato de la tabla 3.3:

Tabla 3.3 Formato del campo LocalNodeAlias

Parte	Campo LocalNodeAlias
Equipo	NOMBRE_EQUIPO
Tarjeta	NOMBRE_EQUIPO[ID_TARJETA]
Tarjeta	NOMBRE_EQUIPO[ID_TARJETA [ID_TARJETA]]
Puerto	NOMBRE_EQUIPO[ID_TARJETA [ID_TARJETA [ID_PUERTO]]]

Como se indicó en el capítulo de estudio de interfaces, los equipos tienen diferentes número de niveles para identificar a sus puertos y tarjetas, existe diferentes modelos de equipos de un mismo fabricante que tienen diferente número de niveles también. Por lo tanto se debe identificar correctamente el nivel en el que está cada tarjeta y puerto, esta identificación se hace en el archivo de reglas que procesa y escribe la información de alarmas.

Una identificación correcta conllevará a un análisis causa raíz, dentro de la topología de red y dentro del mismo equipo, que sea correcto; de esta manera el operador de red podrá interpretar mejor los resultados del análisis causa raíz.

CAPÍTULO III
ARQUITECTURA DEL SERVIDOR PRECISION IP

El servidor Precision es un repositorio de alarmas paralelo en donde se almacenan eventos y alarmas provenientes del repositorio central de alarmas, el ObjectServer. En Precision se almacena un conjunto de las alarmas del ObjectServer puesto que no todas las alarmas contienen información de caída de equipos, de tarjetas o de puertos, solamente se transfieren aquellas alarmas que contienen información que altera el estado de la topología de red.

Las alarmas que se encuentran en el repositorio de Precision tienen información relevante a la topología de red, es en este repositorio donde se correlan las alarmas para poder determinar el efecto que cada alarma tiene sobre el estado de la red, de esta manera se pueden identificar alarmas raíz, alarmas síntomas y alarmas desconocidas que no pertenecen a ninguno de los dos tipos anteriores. Luego de realizada la correlación de alarmas, todos los cambios efectuados sobre las alarmas dentro de Precision se actualizan en el ObjectServer.

El servidor Precision está constituido por componentes que realizan una función específica. Varios de los componentes necesitan de información actualizada de topología de red para poder realizar su función, por lo que estos componentes actualizan constantemente la información de topología de red desde el componente que administra y mantiene actualizada la topología de red. Otros componentes también necesitan interactuar con otros componentes, por lo que se realizará un constante intercambio de información entre los mismos.

En este capítulo se describe la función e interacción de los distintos servicios que conforman el servidor Precision.

3.1 Arquitectura servidor Precision IP

En la figura 3.1 se muestra un diagrama con los distintos servicios de Precision y la manera como se relacionan entre si.

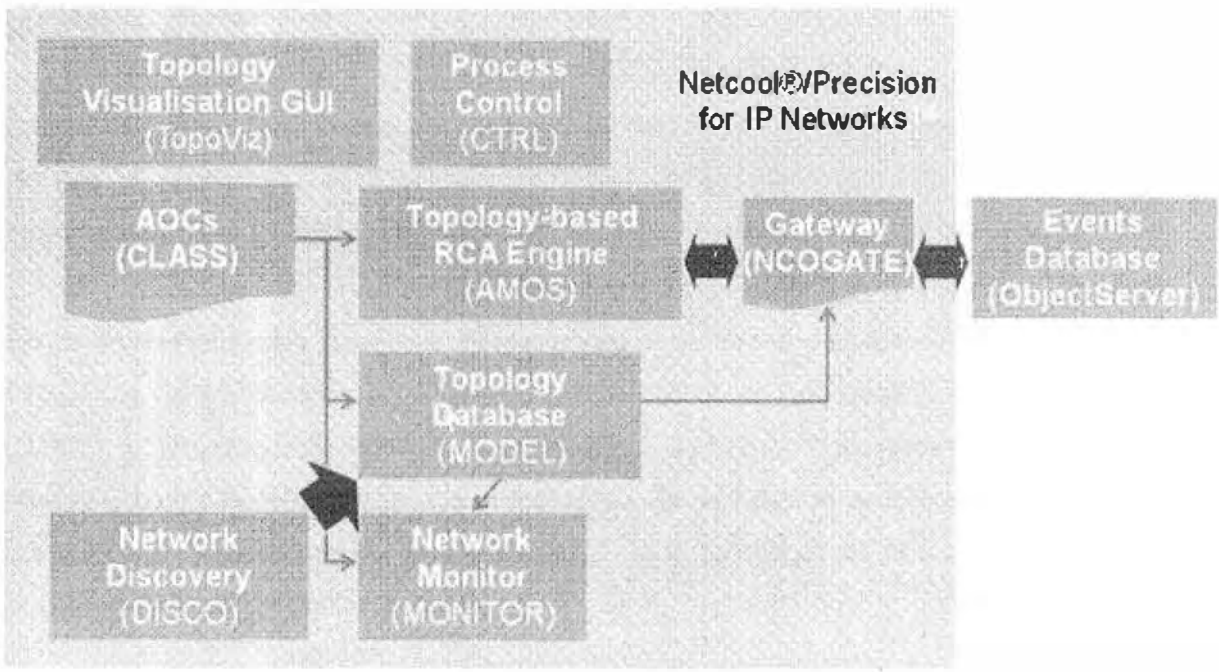

Figura 3.1 Diagrama de componentes de Precision

3.1.1 Sistema de Administración de Procesos

Este componente de Precision, conocido como CTRL, es el encargado de supervisar todos los demás procesos del servidor. Para iniciar el servidor Precision, primeramente hay que iniciar a CTRL, luego CTRL realiza una secuencia de inicio adecuada para cada uno de los procesos. Una vez iniciados todos los procesos, verifica que todos estén funcionando correctamente. Si uno de los procesos deja de funcionar se encarga de iniciarlo nuevamente.

3.1.2 Sistema de Descubrimiento de Red

El componente DISCO es el encargado de hacer el descubrimiento e inventario de la red a monitorizar, cuenta con varios mecanismos para poder obtener información de los dispositivos de red. Se puede conectar a un dispositivo de red mediante el protocolo SNMP, telnet u otros protocolos conocidos. DISCO obtiene información de características de los dispositivos tales como IPs, modelos, OIDs, etc., también obtiene información de las distintas interfaces de un equipo, así como los enlaces que existen entre distintos equipos dentro de la red. Con la información de las diferentes partes de un equipo DISCO hace una asignación a un modelo de contención dentro de Precision, así pues a cada

parte de un dispositivo se le asigna un nombre que se le denomina entidad, y dependiendo de que parte del equipo sea, éste podrá contener otras entidades o estará contenido dentro de otra entidad. Una entidad equipo contendrá entidades rack, una entidad rack contendrá entidades tarjetas, una entidad tarjeta contendrá entidades puertos.

Luego de haber descubierto la topología de red, Disco envía la información de topología de red al repositorio de topología de red MODEL para que éste administre y envíe información de la topología de red a los demás componentes que lo requieran.

En la implementación realizada, no se utilizan las funcionalidades de descubrimiento de DISCO, la carga de la topología de red la hace un script que hace la carga directamente a DISCO, finalmente DISCO envía la información de la topología de red a MODEL. El script utilizado carga la información de la topología de red desde archivos de texto plano que contiene información de los enlaces dentro de la topología de red, los archivos de texto son proporcionados por el operador.

3.1.3 Sistema de Administración de Clases

Es el componente CLASS y en él se almacena la parte fundamental de Precision, el modelamiento de la red. El modelamiento esta dividido en dos partes importantes: el modelamiento de objetos y el modelo de contención.

a) **Modelamiento de objetos**

El modelamiento de objetos consiste en agrupar varios dispositivos que tienen características en común para poder representarlos por un solo objeto y luego al momento de modelar la red, podemos tener varias instancias de un mismo objeto para los diferentes equipos de red que tienen características comunes, en el caso del operador la mayoría de los equipos de un mismo modelo de un mismo fabricante estarán representados por un mismo objeto.

En Precision la representación de las características y comportamiento de los dispositivos se obtiene mediante Clases de Objetos Activos (AOCs). La especificación, de los AOCs, se realiza mediante reglas en archivos de texto plano.

El modelo de objetos también contempla la propiedad de herencia, es decir que un objeto hijo puede heredar las características del objeto padre y además tener características adicionales.

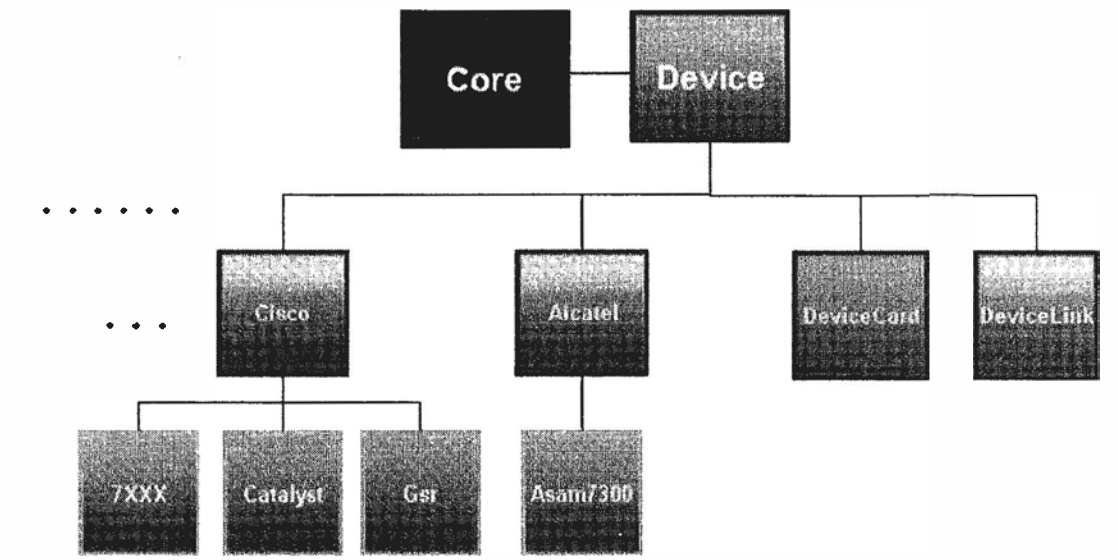

Figura 3.2 Jerarquía de clases.

Un bosquejo del modelamiento contemplado en la implementación actual se muestra en la figura 3.2. Como se puede ver, todas clases heredan de la clase principal Core, de esta clase se pueden desprender varias clases, en este caso se ha definido la clase Device que hereda de Core; de la clase Device se crearán clases para cada fabricante, cada clase de un fabricante determinado albergará los distintos modelos del mismo fabricante. Adicionalmente se ha creado la clase DeviceCard y la clase DeviceLink, la clase DeviceCard representará a las tarjetas de los equipos en general, la clase DeviceLink a su vez representará a los puertos contenidos dentro de una tarjeta, también tendrá información del puerto de otro equipo al que está conectado.

El modelo total de las clases se define en la tabla 3.1

Tabla 3.1 Modelo de clases en su totalidad.

Jerarquía de Clases	Clase padre	Class
Core.aoc		
Device.aoc	Core.aoc	
DeviceCard.aoc	Device.aoc	
DeviceLink.aoc	Device.aoc	
Alcatel.aoc	Device.aoc	
Asam7300.aoc	Alcatel.aoc	Asam7300
Huawei.aoc	Device.aoc	
MA5300.aoc	Huawei.aoc	MA5300
Cisco.aoc	Device.aoc	
Gsr.aoc	Cisco.aoc	GSR
7XXX.aoc	Cisco.aoc	7XXX
Catalyst.aoc	Cisco.aoc	Catalyst
3XXX.aoc	Cisco.aoc	3XXX
4XXX.aoc	Cisco.aoc	4XXX
Juniper.aoc	Device.aoc	
ERX.aoc	Juniper.aoc	ERX
T320.aoc	Juniper.aoc	T320
Marconi.aoc	Device.aoc	
ASX..aoc	Marconi.aoc	ASX
Lucent.aoc	Device.aoc	
Maxtnt.aoc	Lucent.aoc	Maxtnt

b) **Modelo de Contención**

El otro componente clave del modelamiento de la red es el modelo de contención. Un contenedor puede contener a otros objetos, como también puede contener a otros contenedores. Es decir, los contenedores son objetos que contienen elementos u otros contenedores. Los elementos y contenedores son representaciones de entidades físicas o lógicas como tarjetas, puertos físicos, puertos virtuales, etc.

Una representación del modelo utilizado se muestra en la figura 3.3

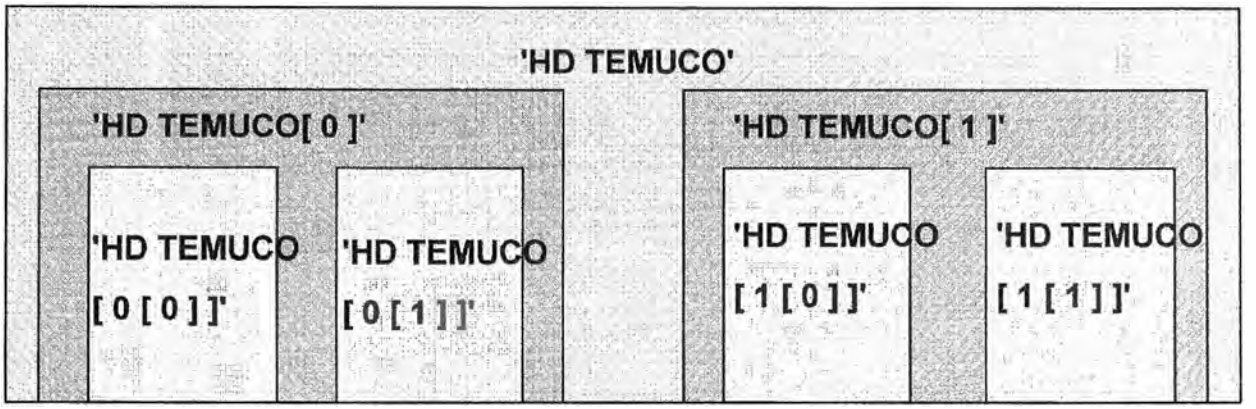

Figura 3.2 Modelo de contención.

En la figura se muestra la jerarquía física del modelo de contención, una entidad chasis (HD TEMUCO) contiene entidades tarjetas (HD TEMUCO[0] y HD TEMUCO[1]), a su vez cada entidad tarjeta contiene entidades puertos (HD TEMUCO[0 [0]]. HD TEMUCO[0 [1]], HD TEMUCO[0 [0]] y HD TEMUCO[0 [1]])

3.1.4 Repositorio de Topología de Red

Este es el componente MODEL, que almacena la topología de red, y es en éste donde se encuentra aplicado el modelo de contención.

MODEL obtiene la información de la topología de red desde el componente DISCO quien es el que realiza el descubrimiento de la red, es decir descubre los equipos de la red, una vez descubierto un equipo, DISCO obtiene información sobre las tarjetas y puertos que contiene el equipo. Con la información que proporciona DISCO, MODEL modela los distintos componentes de la red tales como equipos, tarjetas y puertos, los modela con objetos, cada objeto tiene un nombre que se le denomina entidad, a cada entidad le asocia una clase definida en la jerarquía de clases de CLASS, es decir un AOC, y también se le aplica el modelo de contención, finalmente cada entidad es almacenada en la base de datos de topología de MODEL, ésta es una base de datos orientada a objetos. La tabla de topología de MODEL se muestra en la tabla 3.2

Tabla 3.2 Tabla de topología.

Campo	Descripción
ObjectId	Clave de la entidad.
EntityName	Nombre de la entidad.
Address	Dirección IP o MAC de la entidad.
Description	Descripción
EntityType	Tipo de entidad: 1 Equipo, 2 Puerto, 5 Tarjeta.
ClassName	Nombre de clase asociada a entidad.
EntityOID	OID de entidad.
ExtraInfo	Información Adicional
RelatedTo	Entidad conectada a ésta entidad.
Contains	Nombres de entidades contenidas.
UpwardConnections	Nombre de entidad que contiene a entidad actual.

En la base de datos de topología tenemos entidades equipos, entidades tarjetas y entidades puertos cada uno de ellos con distintas propiedades. Cada equipo de la red está conformado por una entidad equipo (EntityType=1), entidades tarjetas (EntityType=5) que pueden tener varios niveles, y entidades puertos (EntityType=2).
Un ejemplo de cómo estaría representado un equipo se muestra en la siguiente tabla:

Tabla 3.3 Representación de un equipo en MODEL.

Parte	Nombre Entidad	Tipo Entidad
Equipo	CALDERA1	EntityType=1
Tarjeta nivel 1	CALDERA1[0]	EntityType=5
Tarjeta nivel 2	CALDERA1[0 [0]]	EntityType=5
Tarjeta nivel 3	CALDERA1[0 [0 [0]]]	EntityType=5
Puerto	CALDERA1[0 [0 [0 [0]]]]	EntityType=2

MODEL es el componente encargado de enviar o actualizar la información de topología de red a los distintos componentes del servidor Precision que lo requieran, con cada modificación que se realice a la topología de red, MODEL enviará un mensaje a los demás componentes para que el resto de componentes actualicen la información de la topología de red.

3.1.5 Sistema de Monitorización de Red

Otro componente del servidor Precision es MONITOR, que tiene la misión de monitorizar el estado de la red. La monitorización se realiza a través de varios mecanismos según esté especificado en la clase asociada a cada equipo. Para poder realizar la

monitorización a cada uno de los equipos de red, MONITOR obtiene información de la topología de red desde MODEL, lo que se hace con una copia de la base de datos de topología de red desde MODEL.

MONITOR tiene varios mecanismos de monitorización tales como ICMP requests, consultas SNMP, consultas telnet, etc., el tipo utilizado en la implementación actual es el de ICMP requests, MONITOR realiza pings con una frecuencia de 2 minutos a cada uno de los dispositivos de la base de datos de topología, cuando no obtiene respuesta de alguno de los dispositivos envía un mensaje a la sonda de monitor de pings, y ésta inserta el evento de falla de ping en el ObjectServer.

3.1.6 Sistema de Análisis Causa Raíz de Eventos

Es el repositorio de alarmas del servidor Precision y el motor de correlación de eventos para el análisis causa raíz, es conocido como AMOS. En AMOS se tiene una copia de la mayoría de los eventos del ObjectServer, se dice la mayoría, por que no todos los eventos del ObjectServer serán procesados por análisis causa raíz.

AMOS contiene una base de datos con los campos que se muestran en la tabla 3.4

Tabla 3.4 Tabla de datos de AMOS.

Campo	Descripción
EventId	Clave de alarma para AMOS
EntityName	Nombre de la entidad asociada a la alarma
ClassName	Clase asociada a la entidad asociada a la alarma
NcoSerial	Serial de alarma en el ObjectServer
Description	Descripción
EventName	Nombre de evento (campo EventId del ObjectServer)
RuleSet	Conjunto de reglas
RuleName	Regla a ejecutar
EventType	Tipo de evento, 0 evento, 2 alerta.
Severity	Severidad
CauseType	Tipo de falla, 0 indeterminado, 1 problema raiz, 2 problema síntoma.
AgentAddress	Estación que realiza monitorización de los equipos de red.
ExtraInfo	Información adicional.

3.1.7 Interfaz Bidireccional de Eventos

NCOGATE es el componente del servidor Precision que permite el flujo de eventos entre el ObjectServer y AMOS, NCOGATE revisa frecuentemente el cambio de los campos de las alarmas existentes, también revisa la aparición de nuevas alarmas en el ObjectServer, una vez detectado la aparición o cambios de alarmas, filtra las alarmas según un criterio para luego actualizar o insertar las alarmas en AMOS. Su función también es la de actualizar los cambios producidos en AMOS, luego de la correlación de eventos del análisis causa raíz, en el ObjectServer.

CAPITULO IV
ANALISIS CAUSA RAIZ

El análisis causa raíz (RCA por sus siglas en inglés) consiste en la correlación de alarmas, dentro de AMOS, para determinar alarmas raíz y sus alarmas síntomas correspondientes. Los criterios utilizados para determinar las alarmas raíces y síntomas se definen mediante reglas, las reglas de correlación se definen en archivos de reglas que son albergados dentro del componente CLASS.

Para poder hacer un análisis causa raíz efectivo en las alarmas que tengan información de eventos de red, se debe de contar con un mecanismo para poder determinar el estado actual de la red y que no dependa de otro sistema, también se necesita tener una copia actualizada de todas las alarmas, del ObjectServer, que tengan información de eventos de los equipos dentro de la red.

Precision tienen un mecanismo para determinar el estado de cada uno de los equipos de red y de esa manera puede obtener información actual del estado de toda la red. El componente MONITOR es quien realiza la monitorización mediante censado u otros mecanismos, finalmente toda la información recolectada es enviada al repositorio central de alarmas, el ObjectServer.

Con la finalidad de tener sincronizados al ObjectServer y AMOS, y mantener las alarmas actualizadas, se establece un flujo de eventos de manera bidireccional. En primera instancia existe un flujo de eventos desde el ObjectServer hacia AMOS, este flujo actualiza todos los cambios de alarmas del ObjectServer en AMOS, con los eventos actualizados en AMOS se procede la ejecución de políticas del análisis causa raíz que modifican las alarmas dentro de AMOS. Existe otro flujo de eventos desde AMOS hacia el ObjectServer, que actualiza los cambios realizados en las alarmas de AMOS en el ObjectServer.

4.1 Monitorización de la Red

Con la red descubierta y almacenada en MODEL, el servidor Precision puede obtener el estado actual de toda la red. Para obtener el estado de la red, el servidor Precision hace un censado u otro tipo de monitorización a cada equipo dentro de la base de datos de la topología de red, este censado lo hace con la finalidad de obtener el estado de cada dispositivo individualmente.

4.1.1 Proceso de Monitorización de Red

El proceso de monitorización de red se muestra en la figura 4.1

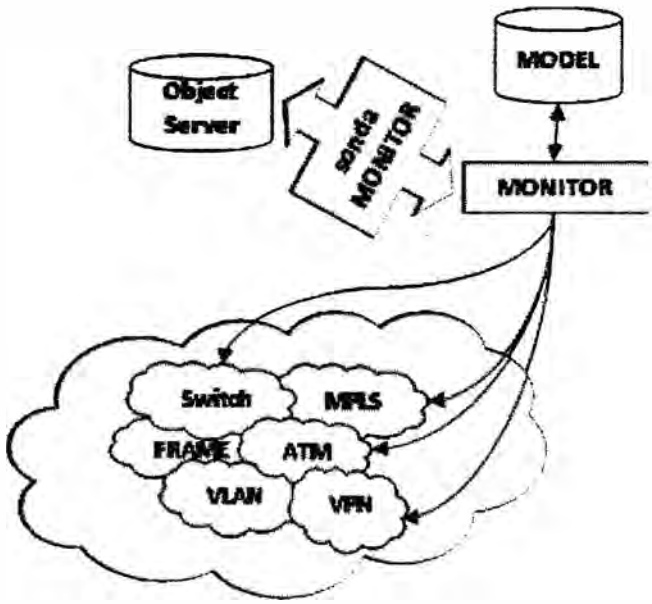

Figura 4.1 Monitorización de la red

Al término del proceso de descubrimiento de la red, Precision puede hacer una monitorización a los distintos dispositivos de la red para determinar el estado actual de la red. El proceso encargado de la monitorización es MONITOR, MONITOR tiene una copia local de la topología de red y realiza una monitorización específica para cada dispositivo. Dependiendo del tipo de dispositivo puede utilizar el protocolo ICMP, SNMP, etc. Una vez obtenido el estado de cada equipo, MONITOR envía información del estado del equipo mediante alarmas al ObjectServer. El envío de alarmas se hace de manera similar a la descrita en el capítulo 2, el servidor Precision dispone de una sonda, la sonda MONITOR. Esta sonda trabaja de manera similar al resto de sondas, hace un procesamiento previo sobre la alarma y luego hace una inserción en el ObjectServer.

4.2 Flujo Bidireccional de Eventos

Para poder realizar el análisis causa raíz dentro de AMOS se necesita tener una copia de los eventos del ObjectServer que sea actualizada, puesto que no todas las alarmas del ObjectServer representan eventos dentro de la topología de red, AMOS tendrá una copia menor de los eventos del ObjectServer y los actualizará constantemente. Todos los cambios realizados a consecuencia del análisis causa raíz serán actualizados en el ObjectServer.

El componente encargado de hacer el paso de eventos desde el ObjectServer hacia AMOS, y viceversa, es NCOGATE. NCOGATE realizará consultas periódicas al ObjectServer para poder identificar cualquier evento que sucedan en las alarmas, los eventos pueden comprender la aparición de una nueva alarma, la eliminación de alguna alarma, la actualización de algunos campos de las alarmas, y cambios de estados en las alarmas. Al detectar cualquiera de los eventos anteriores, NCOGATE intentará actualizar la información del evento en AMOS. Para poder identificar que eventos tienen la capacidad de insertarse en AMOS, NCOGATE va a tener que validar a cada evento detectado, cada evento será evaluado y procesado dentro de NCOGATE para finalmente ser insertado en AMOS.

El flujo de eventos se muestra en la figura 4.2

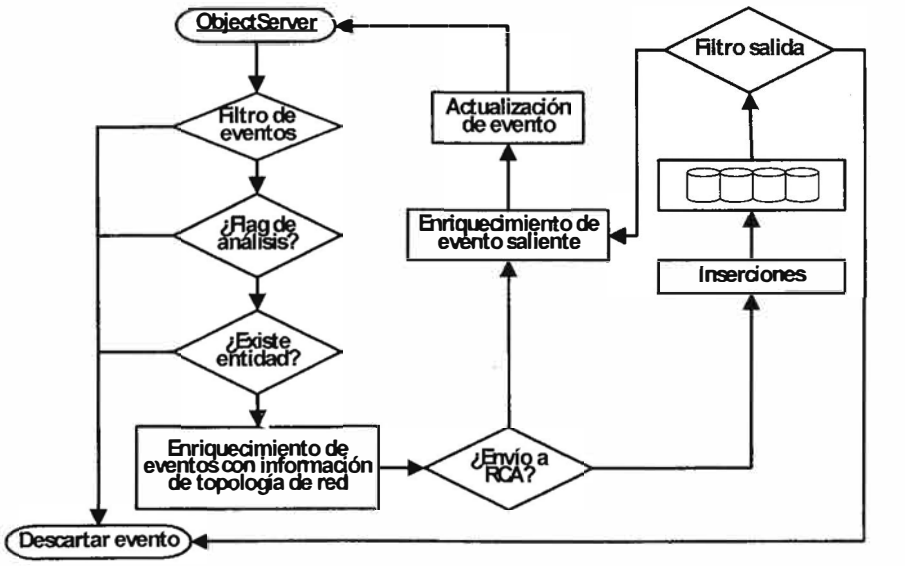

Figura 4.2 Diagrama de flujo de eventos

Cuando se detecte algún evento dentro del ObjectServer, NCOGATE realizará los siguientes pasos:

- Verificará si el evento pasa un filtro inicial, solo las alarmas que representen un problema y hayan sido totalmente enriquecidas dentro del ObjectServer podrán pasar este filtro, si el evento no pasa esta condición, se le descarta.
- Si el campo EventId corresponde a alguno de los valores de la tabla 3.2, la alarma corresponde a los eventos que pueden escribirse en AMOS y debe seguir el flujo, de lo contrario el evento se descarta.
- Si el evento pasa la condición anterior se trata de identificar a que entidad pertenece, NCOGATE realiza una búsqueda en la copia de la base de datos de topología de red que posee, si se logra localizar el equipo al que pertenece, el evento sigue el flujo, de lo contrario es descartado.

  La búsqueda en la base de datos de topología se realiza utilizando el campo LocalNodeAlias de la alarma, se hace una consulta para ver si existe alguna entidad cuyo nombre sea igual al campo LocalNodeAlias de la alarma, si la consulta retorna resultados, entonces si se logró identificar a la entidad a la que pertenece la alarma.
- Una vez identificada la entidad a la que pertenece la alarma, la alarma es asociada a la entidad para poder realizar el análisis causa raíz que utiliza la información de la topología de red.
- Hasta aquí todos los eventos van a pasar a formar parte del análisis causa raíz, para lo cual se hará una inserción en el motor del análisis causa raíz AMOS.
- Cuando la alarma es insertada en AMOS, inmediatamente se procede a correlar los eventos regidos por reglas que se definen en archivos de reglas. El resultado de la correlación de eventos conlleva a la identificación de las alarmas raíz y las alarmas síntomas.
- Una vez culminado el análisis causa raíz, todas las alarmas que hayan sufrido cambios son evaluadas por un filtro, donde no se permitirá pasar aquellos cambios que hayan eliminados alarmas dentro de AMOS con la finalidad de no reflejar el borrado de alarmas, por parte del Servidor Precision, en el ObjectServer.
- Antes de la culminación del ciclo, se hace un enriquecimiento final, previa a la realización de las actualizaciones en el ObjectServer.
- Finalmente se hace un mapeo en el cual se define que campos se van a actualizar en el ObjectServer, luego se realiza las actualizaciones en todo los registros que hayan sufrido cambios a consecuencia del análisis causa raíz.

4.3 Análisis Causa Raíz

Una vez explicado el flujo de eventos desde el ObjectServer hacia Precision, se puede pasar a explicar en detalle las partes del análisis causa raíz.

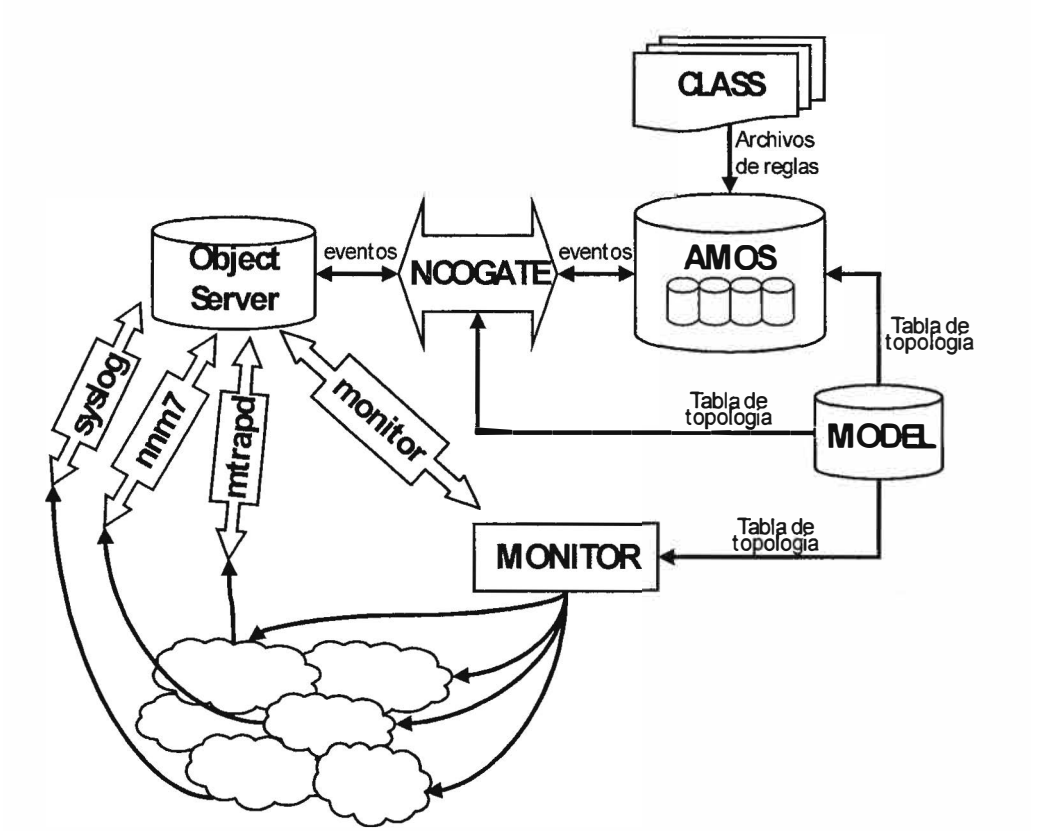

Figura 4.3 Flujo de eventos en el análisis causa raíz

4.3.1 Tipos de Análisis Causa Raíz

En el análisis causa raíz se han detectado 3 tipos de fallas principales en los equipos que pueden afectar el estado de la red, como alarmas de indisponibilidad, intranodo e internodo.

a) Alarmas de indisponibilidad

Dentro del estudio del tipo de alarmas que envían los equipos se han identificado alarmas del tipo de indisponibilidad de equipo, ante tal caso todas las alarmas que pertenecen al chassis del equipo deberían ser inhibidas por la alarma de indisponibilidad.

b) Intranodo

En este tipo de análisis causa raíz se inhibirían las alarmas de dispositivos que estuvieran contenidos dentro del dispositivo que presenta la falla.

c) **Internodo**

Este tipo de análisis causa raíz funciona para detectar si existen nodos aislados a consecuencia de la caída de un equipo. A la presencia de una alarma de caída de equipo, este tipo de análisis causa raíz detectará que equipos quedaron aislados e inhibirá las alarmas de los equipos aislados.

4.3.2 Características de las Alarmas

Para el análisis causa raíz tendremos 2 tipos de alarmas, alarmas dependientes y alarmas trigger.

Las alarmas dependientes son aquellas alarmas que no van a desencadenar el análisis causa raíz, sin embargo si van a ser afectadas por el análisis causa raíz, la característica de estas alarmas es que el campo EventId lo tendrán en blanco.

Las alarmas trigger son aquellas alarmas que si desencadenarán un análisis causa raíz y pueden afectar a otras alarmas, como también pueden ser afectadas por otras alarmas trigger, la característica de estas alarmas es que pueden tener los siguientes valores en el campo EventId: nodeupdown, NmosPingFail, nodesnmpfail, shelfupdown, slotupdown y portupdown

4.3.3 Estados de Alarmas

En el capítulo 2 se describió los campos del ObjectServer, los campos NmosCauseType, Nmoserial y NmosObjInst son los tres campos importantes con los cuales Precision identificará a las alarmas raíz y síntoma, estos campos identifican el estado de las alarmas después de un análisis causa raíz.

Una alarma puede tener 3 estados, puede ser una alarma raíz, síntoma o informativa, la alarma informativa es aquella que luego del análisis no ha sido identificada como alarma raíz o síntoma.

Luego del análisis causa raíz una alarma puede ser identificada como una alarma raíz, si se da el caso, se procede a identificar sus alarmas síntomas. Si llegara una alarma resolución para la alarma raíz, se procede a clarear a la alarma raíz y a desinhibir a las alarmas síntoma.

La descripción de los estados de las alarmas se muestra en la Tabla 4.1

Tabla 4.1 Estados de las alarmas

Evento	Tipo de alarma	Severidad	NmosCauseType	NmosSerial
Llegada de alarmas	Desconocido	Severidad de sonda	0 – (Desconocido)	0
Se identifica alarma raíz y alarmas síntomas	Raíz	5 - (Crítica)	1 - (Raíz)	0
	Síntoma	1 - (Desconocido)	2 - (Síntoma)	Serial raíz
Llega alarma resolución de alarma raíz	Raíz	0 - (Limpio)	0 - (Desconocido)	0
	Síntoma	2 - (Advertencia)	0 - (Desconocido)	0

4.3.4 Descripción de Entidades

El modelamiento de partes de un equipo es representado por entidades dentro de Precision. Así por ejemplo tendremos 3 tipos de entidades las cuales podrán darnos un modelo de toda la topología de red que se está monitorizando.

El modelo ya fue descrito en el capítulo 2 en la sección 2.1.4 para mayor información.

4.3.5 Secuencia de Ejecución de Reglas

Como ya se comentó en el capítulo 3, el análisis causa raíz consiste en la correlación de eventos para identificar alarmas raíz y alarmas síntomas. La correlación de las alarmas se hace en base a la definición de reglas, las reglas se agrupan en archivos denominados rules. Se ha seguido un diseño de tal manera que un archivo de reglas defina reglas con un propósito determinado. El propósito de cada archivo de reglas es correspondiente a los tipos de análisis causa raíz definidos en la sección 4.3.1. La tabla 4.2 resume la correspondencia

Tabla 4.2 Estados de las alarmas

Archivo rule	Tipo RCA
EntityEventToAlert.rule	
TimedAlertTransition.rule	Indisponibilidad
SuppressConnectedAlerts.rule	Intranodo
SuppressDownstreamAlerts.rule	Internodo
EntityClearEvent.rule	Indisponibilidad clareo
ClearEventAwakenConnected.rule	Intranodo clareo
ClearEventAwakenDownstream.rule	internodo clareo

a) Regla EntityEventToAlert

Este es el archivo de reglas que verifica la llega de eventos a AMOS cada 30 segundos, si se trata de un evento nuevo que no tienen una alarma asociada, entonces escribe una alarma nueva dentro de AMOS y la llena con la información del evento. Si el evento ya tiene una alarma asociada, entonces incrementa el campo de conteo de la alarma.

b) Regla TimedAlertTransition

Realiza la correlación de alarmas con la finalidad de identificar a las alarmas raíz y síntomas dentro de una misma entidad, es decir que si llega una alarma y se identifica como una alarma raíz dentro de una entidad equipo, entonces se inhibirá solo las alarmas que pertenezcan a la entidad equipo y no a las alarmas de las entidades que pudieran estar contenidas dentro de la entidad equipo. Un caso similar se realizaría si se detectara una alarma raíz dentro de una entidad tarjeta o puerto.

c) Regla SuppressConnectedAlerts

Con una alarma raíz detectada, dentro de una entidad, en el archivo de reglas TimedAlertTransition, realiza la identificación e inhibición de alarmas síntoma en las entidades contenidas dentro de la entidad de la alarma raíz, también identifica e inhibe a las alarmas síntoma de entidades puerto de otros equipos que estén conectadas a entidades puerto contenidas dentro de la entidad de la alarma raíz.

d) Regla SuppressDownstreamAlerts

Si la alarma raíz, identificada en TimedAlertTransition, tiene un identificador que denote la caída de un equipo, entonces este archivo de reglas identificará e inhibirá a las alarmas de los equipos que quedaron aislados como consecuencia de la caída del equipo en donde se identificó a la alarma raíz de caída de equipo. Un equipo se dice que quedó aislado, como consecuencia de la caída de otro equipo, si es que no hay conectividad desde el servidor de monitorización debido a la caída del otro equipo. Si llegan alarmas de caída de los equipos aislados, estas alarmas también serán inhibidas.

e) Regla EntityClearEvent

Si llega una alarma de resolución para una alarma raíz o problema dentro de una entidad, entonces las reglas de este archivo primero clarearán a la alarma que ha sido resuelta y luego desinhibirán a las alarmas que fueron inhibidas por el archivo de reglas TimedAlertTransition.

f) Regla ClearEventAwakenConnected

Este archivo de reglas desinhibe a todas las alarmas síntomas de entidades contenidas dentro de la entidad de donde fue clareada una alarma raíz, también desinhibe alarmas síntoma de entidades puerto, de otros equipos, que estén conectadas a entidades puerto que estén contenidas dentro de la entidad donde fue clareada una alarma raíz.

g) Regla ClearEventAwakenDownstream

Desinhibe las alarmas síntoma de equipos que quedaron aislados luego de que se clarea a la alarma raíz de caída de equipo que las inhibió. En otras palabras desinhibe a las alarmas que fueron inhibidas por el archivo de reglas SuppressDownstreamAlerts.

h) Secuencia de acción de archivos de reglas

Los archivos de regla se ejecutan unos después de otros, a esta propiedad se le denomina encadenamiento de archivos de reglas. La figura 4.4 muestra como es que se lleva acabo el encadenamiento y como es que se ejecuta un análisis causa raíz luego de la detección de una alarma raíz.

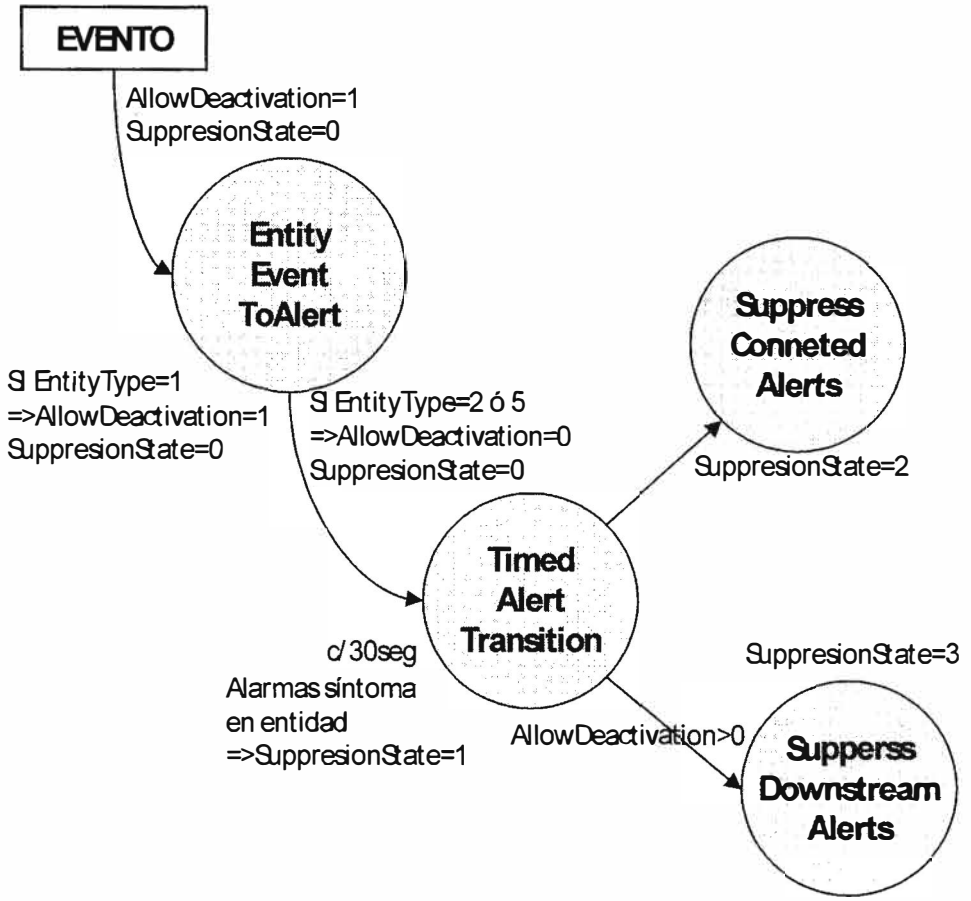

Figura 4.4 Diagrama de eventos inhibición de alarmas

En la figura se muestran dos etiquetas que permitirán un buen manejo del flujo en flujo del análisis causa raíz en la tabla 4.2 se describen todos los valores posibles para estas etiquetas.

Tabla 4.2 Etiquetas para el flujo RCA

Etiqueta	Valores
AllowDeactivation	0: no debe ejecutar reglas de nodos aislados 1: debe ejecutar reglas de nodos aislados
SuppresionState	0: Alarma no inhibida 1: inhibida en la misma entidad que alarma raíz 2: Alarma inhibida en entidad contenida por entidad de la alarma raíz 3: alarma inhibida de equipo aislado de equipo de alarma raíz

Con los posibles valores de las etiquetas pasamos a describir el flujo del análisis causa raíz en caso de supresión de eventos.

- Llegan eventos desde NCOGATE y se les escribe con dos etiquetas: AllowDeactivation=1 y SuppresionState=0 que son definidas por defecto.

- Si la entidad a la que pertenece la alarma es de entidad equipos (EntityType=1) entonces se podrán ejecutar las reglas de nodos aislados, en caso de que las entidades de la alarma sean una tarjeta o un puerto entonces, no podrán ejecutar las reglas de nodos aislados. El estado de la alarma es no inhibido por otra alarma.
- Una vez que se han escrito los eventos como alarmas, el archivo de reglas TimedAlertTransition reconoce a la alarma raíz e inhibe a las alarmas de la misma entidad poniéndoles SuppresionState=1.
- Una vez ejecutado TimedAlertTransition, se procede ejecutar SuppressConnecteAlerts que inhibe a las alarmas contenidas y conectadas a entidades contenidas, a todas las alarmas inhibidas les pone la etiqueta SuppresionState=1.
- Finalmente solo aquellas alarmas que tengan AllowDeactivation=1 podrán realizar el análisis causa raíz de nodos aislados. Solo las alarmas trigger de caída de equipo podrán desencadenar el análsiis causa raíz de nodos aislados.

La secuencia de desinhibición de alarmas también es controlada por archivos de reglas.

La secuencia de los eventos se muestra en la figura 4.5

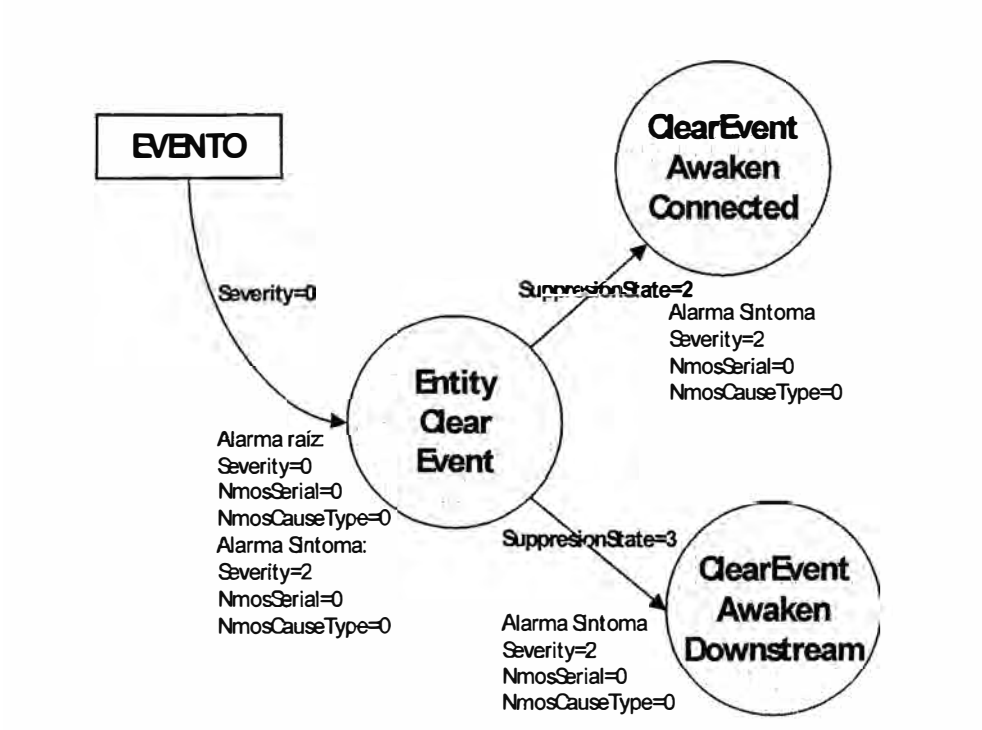

Figura 4.5 Diagrama de eventos desinhibición de alarmas

Para desinhibir las alarmas, se cuenta con 3 archivos de reglas, solamente las alarmas problema que fueron clareadas por su alarma resolución respectiva serán procesadas por estos archivos de reglas.

- Los eventos de aparición de alarmas problema clareadas son procesadas por el archivo de reglas EntityClearEvent, este hace el clareo de la alarma raíz y desinhibe a las alarmas síntomas de la misma entidad de la alarma raíz.
- Una vez ejecutado el archivo EntityClearEvent, se procede a ejecutar el archivo ClearEventAwakenConnected que desinhibe a las alarmas síntoma de de las entidades contenidas y conectadas a entidades puertos contenidos en la entidad de la alarma raíz que se acaba de clarear en EntityClearEvent. En esta regla solo las alarmas que tienen la etiqueta SuppresionState=2 son desinhibidas.
- El archivo ClearEventAwakenDownstream también se ejecuta después del EntityClearEvent y desinhibe todas las alarmas que fueron inhibidas por la alarma raíz de la caída de un equipo, solo las alarmas raíz que tienen la etiqueta AllowDeactivation podrán ejecutar este archivo de reglas.

Con estos pasos se ha descrito el proceso de detección de alarmas raíz, la inhibición de eventos de acuerdo a los tipos de análisis causa raíz y también se ha descrito el proceso de desinhibición de alarmas inhibidas.

CAPITULO IV
IMPLEMENTACION

El sistema de análisis causa raíz es un sistema que complementa a un sistema de supervisión de alarmas de una red, el sistema en conjunto conforma así una arquitectura de servidores, los cuales son una plataforma mas para el operador de telecomunicaciones. En este capítulo se describe las características de los servidores y el software utilizado para la implementación de los dos sistemas.

5.1 Arquitectura del Sistema de Supervisión y del Sistema de Análisis Causa Raíz

El sistema de supervisión esta conformado por dos servidores, En uno de ellos se encuentra instalado el ObjectServer, el sistema centralizado de recolección de eventos. Todas las alarmas de la red son enviadas y recolectadas en este servidor.

En el otro se encuentra el servidor Webtop que tiene la interfaz de presentación. El servidor Webtop se conecta al ObjectServer para poder obtener información de las alarmas en tiempo real, procesa la información de las alarmas y las prepara para su presentación a los clientes. Todas las interfaces de presentación de alarmas se hacen a través de servicios web, la información se muestra a través de listas, mapas topológicos y gráficos de estadísticas. Todos los clientes se conectan a éste servidor para poder ver el estado de las alarmas y el estado de la red a través de los mapas topológicos.

El servidor Precision se encuentra instalado en un tercer servidor del sistema de supervisión, todos los procesos se levantan en este servidor y desempeñan sus funciones respectivas. El servidor Precision se conecta al servidor del ObjectServer para obtener y actualizar las alarmas en tiempo real, a su vez la información de topología de red es enviada al servidor Webtop para su presentación a los clientes.

La arquitectura de servidores se muestra en la figura 5.1

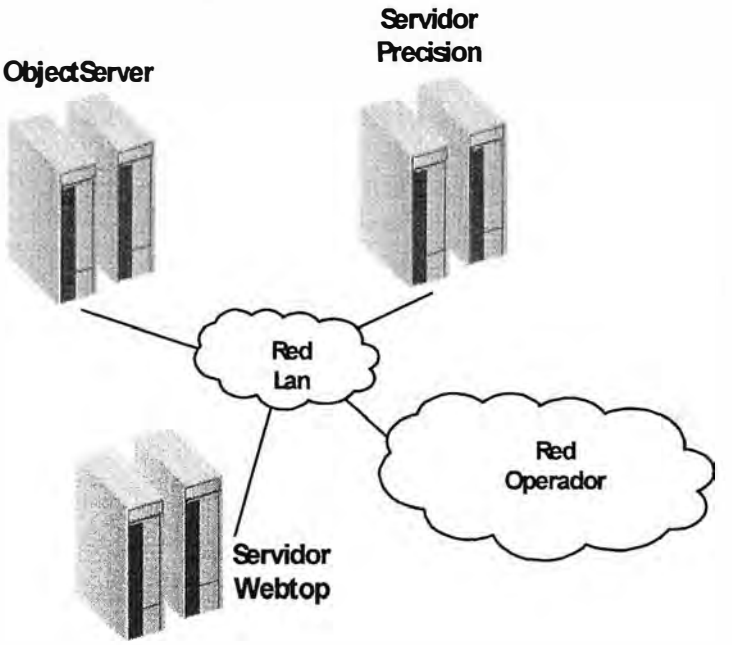

Figura 5.1 Arquitectura de servidores del sistema de supervisión.

El hardware y el software utilizado en los servidores se detallan en la tabla 5.1

Modelo Servidor	Software	Sistema Operativo	DiscoDuro	Memoria Ram	CPU
SunFire V440	ObjecServer	Solaris9	4x73GB(RAID 1)	4GB	2X1.5GHz
SunFire V440	Servidor Precision	Solaris9	4x73GB(RAID 1)	4GB	2X1.5GHz
SunFire V440	Servidor Webtop	Solaris9	2x73GB(RAID 1)	4GB	2X1.34GHz

Tabla 5.1 Hardware y software de plataforma de supervisión de alarmas

Los servidores del ObjectServer y Precision son los que tienen mayor carga de procesamiento, razón por la cual tienen mayor capacidad comparados con el servidor Webtop. El servidor Webtop al ser de interfaz de presentación no necesita muchos recursos de procesamiento.

5.2 Interconexión del Sistema con la Red a Monitorizar

El operador tiene varias plataformas, así los servidores de monitorización conformarán otra plataforma adicional que debe cumplir con las normas de seguridad e interconectividad del operador.

La topología del operador obliga a conectar la plataforma con dos firewall que permiten el paso de paquetes que solo la plataforma de monitorización necesita, se utilizan dos firewall con la finalidad de tener redundancia en la conectividad.

Los firewalls están conectados con dos switches que a su vez están conectados entre sí y que son la interfaces de redundancia de conectividad a la red multiservicio del operador, estos switches son transparentes para la interconectividad con la red multiservicio.

Finalmente la plataforma de supervisión se interconecta con la red multiservicio para poder hacer monitorización de la misma y también para tener conectividad con el resto de equipos de la red del operador.

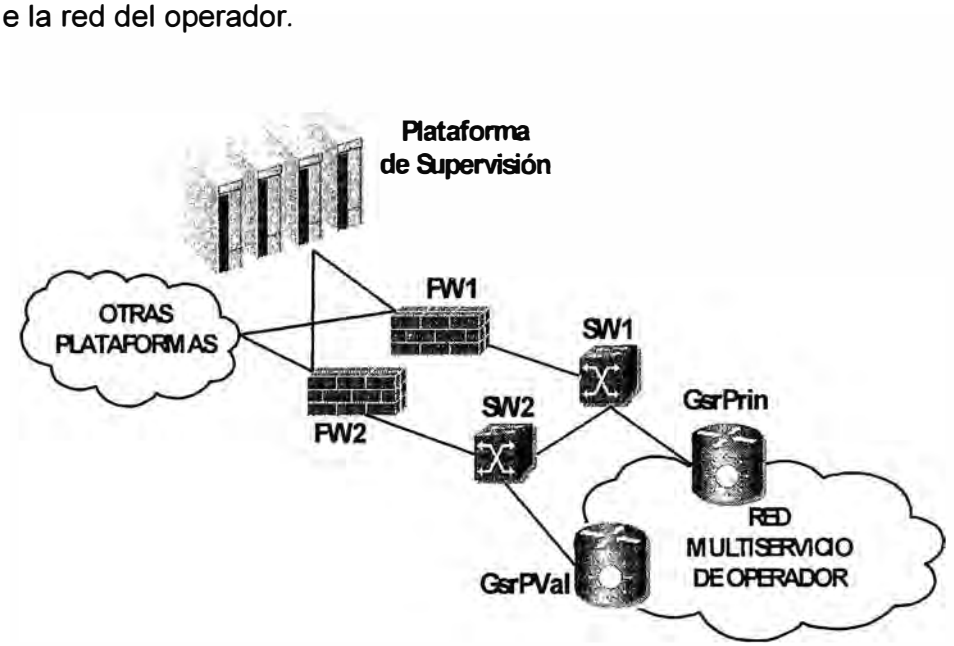

Figura 5.2 Interconexión de plataforma de supervisión con la red

5.3 Resultados de Análisis Causa Raíz

Luego de instalado el hardware, software e configurado el servido Precision, se obtuvieron buenos los resultado deseados para la detección de alarmas raíz y síntoma. En la figura 5.3 mostramos una lista en la que se visualizan los eventos de las alarmas identificadas como alarmas raíz y alarmas síntoma.

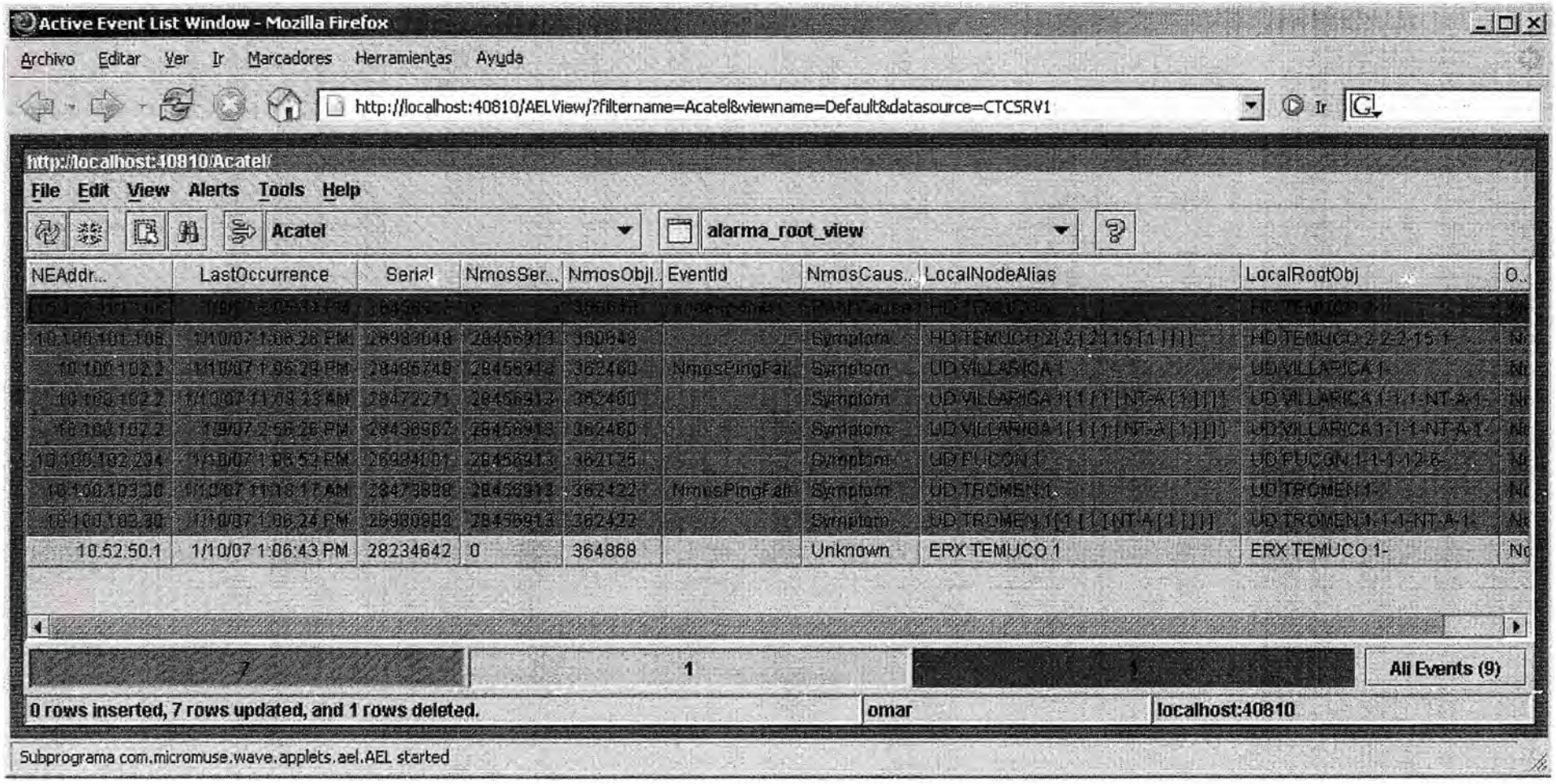

Figura 5.3 Lista de Eventos luego de análisis causa raíz

En la lista de eventos de la figura 5.3 se puede observar la identificación de una alarma raíz, la alarma raíz sucedió en el equipo de nombre HD TEMUCO 2, Serial=28456913 y Severidad=5 (color rojo), también se puede ver que ha inhibido las alarmas de equipos que quedaron aislados luego de la caída del equipo HD TEMUCO 2, las alarmas de equipos aislados tienen NmosSerial=28456913 (serial de la alarma raíz), NmosCauseType=2 (alarmas síntomas) y Severidad=1 (indeterminado).

Una mejor representación de la información obtenida luego del análisis causa raíz, se puede ver en un mapa topológico que se muestra en la figura 5.4

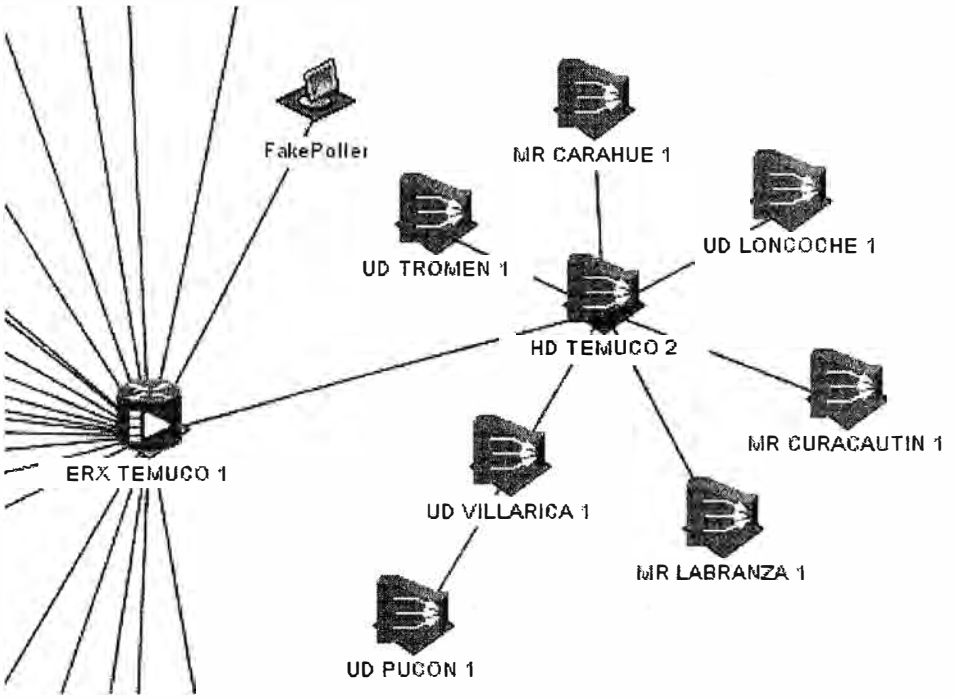

Figura 5.4 Topología de red luego de análisis causa raíz.

El mapa topológico muestra como la caída del equipo HD TEMUCO 2, inhibe las alarmas de los equipos que quedaron aislados. El equipo HDTEMUCO 2 tiene una manta roja que indica que tiene una alarma crítica y que ha sido identificada como una alarma raíz. Los equipos aislados tienen una manta de color púrpura que indica que las alarmas de esos equipos han sido inhibidas. También hay un ordenador que simboliza el servidor Precision desde donde se hace la monitorización de todos los equipos de red a través del protocolo ICMP.

CONCLUSIONES

1. La integración de alarmas de muchos modelos y varios fabricantes hace difícil la correcta asignación del campo LocalNodeAlias que identifica la parte afectada del equipo. Esta asignación incorrecta se traduce en alarmas que no son procesadas correctamente por el análisis causa raíz.
2. Se debe hacer un inventario detallado de todas las posibles alarmas que pueden generar los equipos de red. Esto con la finalidad de hacer una asignación exacta del campo EventId que determina la precedencia de las alarmas.
3. Se ha presentado el caso de alarmas de nodos de entrada intermitentes que hicieron que el sistema inhibiera la totalidad de las alarmas de la red. Tal caso se presentó por la mala conectividad que presentan los firewalls de entrada a la red multiservicio. Se debe de elegir otros nodos que no tengan problemas de conectividad para no mostrar resultados erróneos.
4. El funcionamiento del análisis es correcto para los tres tipos de análisis que se consideraron. El análisis más importante es el de nodos aislados que determina el origen de la pérdida de conectividad de varios equipos a la vez.
5. Puesto que la topología de la red es muy dinámica en la red de acceso, se hace indispensable tener el inventario de equipos de red actualizado.
6. El sistema de análisis causa raíz fue integrado en el sistema de apertura de tickets de casos de red, automatizando mas aún el proceso de detección y solución de alarmas críticas dentro de la red del operador.

BIBLIOGRAFIA

1. Douglas Mauro, Kevin Shmidt, "Essential SNMP"
   Editorial O'Reilly, Julio 2001
2. Martin C. Brown, "Perl: The Complete Reference Second Edition"
   Editorial Osborne / McGraw-Hill, 2001
3. "Perl 5.8.8 documentation"
   http://perldoc.perl.org/, 2007
4. Tim Bunce, "Programing the Perl DBI"
   Editorial O'Reilly, Febrero 2000
5. Netcool/OMNIbusTM 3.6, "Administration Guide"
   Micromuse Inc., 2003
6. Netcool/OMNIbusTM , "SNMP Probe"
   Micromuse Inc., Diciembre 2005
7. Netcool/OMNIbusTM, "Probe for HP OpenView NNM"
   Micromuse Inc., Febrero 2005
8. Netcool/OMNIbusTM 3.6, "Syslog Probe"
   Micromuse Inc., Julio 2002
9. Netcool/Precision for IP NetworksTM 3.5.1, "Server Guide"
   Micromuse Inc., 2005
10. Netcool/Precision for IP NetworksTM 3.5.1, "Monitoring and RCA Guide"
    Micromuse Inc., 2005
11. Netcool/Precision for IP NetworksTM 3.5.1, "TopoViz Guide"
    Micromuse Inc., 2005
12. Janice Winsor, "Solaris Advanced System Administrator's Guide, 2nd edition"
    Editorial New Riders Publishing, 1997