Universidade Federal Fluminense
TET – Departamento de Engenharia de Telecomunicações
Orientador: Luiz Cláudio Schara Magalhães

Projeto Final de Curso

Integração de Gerências SNMP em um Ambiente Distribuído

Autores: Cesar Henrique Pereira Ribeiro Matrícula: 100.41.040-8
Livia Ferreira Gerk Matrícula: 100.41.021-8

Integração de Gerências SNMP em um Ambiente Distribuído

Luiz Cláudio Schara Magalhães

Cesar Henrique Pereira Ribeiro

Livia Ferreira Gerk

Resumo

Neste projeto será apresentada uma solução para a integração de gerência SNMP considerando uma topologia de gerenciamento hierárquica e heterogênea, com sites remotos possuindo gerentes locais, os quais se conectam a um gerente central, responsável por realizar uma gerência macro da rede de toda a empresa.

Palavras Chave – Gerenciamento Integrado, SNMP.

Índice

Índice de Figuras

Índice de Tabelas

1. Introdução

No surgimento das redes de computadores, não existia a idéia de gerenciamento de rede. Quando um problema aparecia, normalmente eram feitos testes simples para localizar a fonte do problema e a partir daí se modificavam os ajustes do sistema ou reinicializava-se o software ou o hardware.

A necessidade da gerência de rede surgiu em grande parte em função do grau de importância que as redes adquiriram dentro do ambiente das empresas, alcançando maiores extensões e tornando-se mais complexas e mais heterogêneas em termos de tecnologias e fabricantes. Muitas vezes, o quadro de pessoal das empresas não avançou em conhecimento na mesma proporção de modo a garantir o bom funcionamento dessas redes em constante evolução.

T. Saydam em seu artigo "From Networks and Network Management into Service and Service Management" [1] fornece uma definição breve do que seria o gerenciamento de rede:

"Gerenciamento de rede inclui a disponibilização, a integração e a coordenação de elementos de hardware, software e humanos, para monitorar, testar, consultar, configurar, analisar, avaliar e controlar os recursos da rede, e de elementos, para satisfazer às exigências operacionais, de desempenho e de qualidade de serviço em tempo real a um custo razoável".

A idéia de gerenciamento de rede é fornecer ferramentas para que um administrador de rede seja capaz de monitorar equipamentos remotos, analisando os dados de modo a garantir que estejam funcionando e operando dentro dos limites especificados, controlar reativamente o sistema fazendo ajustes de acordo com as modificações ocorridas no sistema ou em seu ambiente e gerenciar pró-ativamente o sistema detectando tendências ou comportamentos anômalos que permitam tomar uma ação antes que surjam problemas mais sérios.

As tarefas típicas de gerenciamento dos elementos de uma rede envolvem a configuração de dispositivos, a administração de endereços IP, os serviços de diretório, a monitoração de tráfego, o diagnóstico de falhas, o tratamento de alarmes, a restauração de serviços, a análise de dados e relatórios, a abertura de trouble ticketing, a segurança de rede e o inventário dos elementos.

O objetivo do presente trabalho visa caracterizar um ambiente de rede geograficamente distribuído, que representa a realidade de grande parte das empresas atualmente, e criar uma solução para a integração da gerência dos diversos sites da corporação, proporcionando uma visão macro da saúde da rede como um todo a um administrador de rede central.

Como base para o entendimento do objetivo principal, alguns conceitos se mostram importantes e são abordados nos capítulos seguintes.

No capítulo 2, será feita uma breve introdução sobre o gerenciamento focado no modelo criado pela ISO, caracterizando suas divisões e funções e descrevendo a arquitetura geral do gerenciamento de rede.

O capítulo 3 dará uma visão do padrão SNMP, o mais amplamente utilizado atualmente, mostrando suas operações, arquitetura, estruturas de dados e outros aspectos mais relevantes.

No capítulo 4, são apresentados os softwares de gerência, destacando as características fundamentais das ferramentas utilizadas no projeto, HP Open View, LorriotPro e Trapgen.

Dando prosseguimento, o capítulo 5 fala sobre os conceitos de Gerenciamento Distribuído, que motivou este trabalho.

O capítulo 6 caracteriza o cenário e escopo do projeto, detalhando a implementação da solução escolhida como alternativa para os desafios apresentados.

O trabalho é encerrado com uma breve conclusão, apresentando as vantagens do método escolhido e os pontos fracos de outras abordagens possíveis.

2. Introdução ao Gerenciamento

A International Organization for Standardization (ISO) criou um modelo de gerenciamento de rede que é útil para situar os cenários apresentados em um quadro mais estruturado. Este modelo define cinco áreas de gerenciamento e é conhecido como "modelo FCAPS" – "fault, configuration, accounting, performance and security" (falha, configuração, contabilidade, desempenho e segurança) [2].

7.2 Áreas do Modelo FCAPS

2..1. Gerenciamento de Desempenho

A meta desta área de gerenciamento é quantificar, medir, analisar e controlar o desempenho, como, por exemplo, "throughput" (vazão) e utilização, dos diferentes componentes de rede, tanto dispositivos individuais como abstrações fim a fim (trajeto pela rede). Pode ser visto como o fornecimento de níveis aceitáveis de desempenho em face de demandas variáveis e de ocasionais falhas na rede.

O desempenho corrente da rede deve se basear em indicadores, tais como atraso, throughput, disponibilidade, utilização, taxa de erros, recuperabilidade (através das variáveis MTBF – "Mean Time Between Failure" ou Tempo Médio entre Falhas – e MTTR – "Mean Time to Repair" ou Tempo Médio de Reparo), entre outros. A monitoração desses indicadores permite estabelecer um "baseline" (comportamento normal) da rede. Com isso, podem ser definidos limiares para gerar eventos ou alarmes e registros históricos permitem a análise de desempenho e o planejamento de capacidade, inclusive indicando a necessidade de mudanças na rede.

O planejamento de capacidade consiste na análise de tendências, balanceamento da carga da rede entre recursos e planejamento de expansões ou mudanças de configuração da rede.

Algumas estatísticas de desempenho interessantes são mostradas abaixo de acordo com o elemento monitorado:

- Interfaces: utilização dos enlaces por unidade de tempo, utilização por protocolo, qualidade dos enlaces por unidade de tempo (fração de erros na entrada e na saída, número total de erros, disponibilidade, piores interfaces diariamente, etc);
- Roteadores: utilização de CPU e memória, disponibilidade, taxa de descarte, taxa total de pacotes comutados por segundo;
- LANs Ethernet: percentagem de colisões;
- Hosts: taxa de retransmissão TCP.

2..2. Gerenciamento de Falhas

Seu objetivo é registrar, detectar, isolar e reagir às condições de falha na rede. Corresponde ao tratamento imediato de falhas transitórias da rede, como interrupção de serviço em enlaces, hosts ou em hardware e software de roteadores, por exemplo.

O Gerenciamento de Falhas pode ser dividido nas seguintes tarefas básicas:

- Coleta de dados e detecção de faltas;
- Diagnóstico de problemas ou Isolação de falhas;
- Soluções emergenciais para evitar a paralisação da rede;
- Resolução de problemas;
- Notificação e registro ou Supervisão de alarmes.

A detecção de faltas envolve a manutenção e monitoração do estado de cada um dos elementos gerenciados de modo a perceber a ocorrência de algum problema. Através de técnicas desenvolvidas, pode-se diagnosticar a localização e a razão da falha percebida. Dentre essas técnicas, estão a correlação de eventos e os testes de diagnósticos.

Essa área do gerenciamento permite a antecipação de falhas através da monitoração de indicadores como taxas crescentes de erro ou atrasos de transmissão. Podem ser usados limiares (thresholds) para gerar alarmes.

A supervisão de alarmes é uma interface do usuário que indica quais elementos estão funcionando, quais estão parcialmente funcionando e quais estão fora de operação. Ela inclui níveis de severidade e pode indicar possíveis causas, além de incluir a possibilidade de avisos externos, como via e-mail e SMS (“Short Message System” ou Sistema de Mensagens Curtas).

Dentro desse contexto, são realizados as ações necessárias ao restabelecimento dos elementos com problemas, que podem até ser sugeridas automaticamente, e os testes para permitir a verificação do funcionamento de recursos da rede em condições normais ou artificiais.

Todas as ocorrências são registradas e podem prover relatórios para posterior análise.

2..3. Gerenciamento de Configuração

É responsável pela descoberta, manutenção e monitoração de mudanças nas estruturas física e lógica da rede. Permite a descoberta dos dispositivos da rede, no que diz respeito a topologias física e lógica, e o controle das configurações de hardware e software dos mesmos.

Suas funções básicas são:

- Coleta de informações de configuração através do descobrimento dos elementos e da interconectividade entre os mesmos;
- Geração de eventos a partir da adição ou remoção de recursos, permitindo a manutenção de um inventário atualizado;
- Atribuição de valores iniciais aos parâmetros dos elementos gerenciados;
- Registro de informações de configuração, permitindo a emissão de relatórios, a partir dos dados coletados nas funções anteriores;
- Alteração de configuração dos elementos gerenciados de modo a corrigir uma falha ou problema de segurança ou mesmo redimensionar a alocação de recursos para melhorar o desempenho, onde há uma relação com a área de Gerenciamento de Desempenho;
- Início e Encerramento de operação dos elementos gerenciados.

Há vários tipos de topologia dando enfoque a determinadas visões da rede. Dentre elas, destacam-se:

- Visão de Conectividade Física, que indica os dispositivos fisicamente conectados, podendo mostrar domínios de colisão;
- Visão de Conectividade Lógica, que indica apenas as conexões IP, evidenciando os domínios de broadcast;
- Visão Administrativa, que agrupa os dispositivos de rede de acordo com uma definição administrativa sem relação com aspectos de conectividade física ou lógica;
- Visão de Serviços, que evidencia os dispositivos de rede utilizados pelos vários serviços, facilitando o diagnóstico de problemas.

A conectividade lógica pode ser levantada automaticamente utilizando algumas técnicas, porém as outras visões devem ser construídas manualmente. A conectividade física pode ser levantada automaticamente apenas se elementos transparentes, como hubs, bridges e switches, forem gerenciáveis.

O descobrimento de dispositivos e de sua interconexão pode ser feito de forma ativa, com o envio de informação através da rede, ou de forma passiva, "escutando" a comunicação entre os elementos. Dentre os protocolos usados nesse processo de descobrimento, estão o ARP ("Address Resolution Protocol" ou Protocolo de Resolução de Endereços), ICMP ("Internet Control Message Protocol" ou Protocolo de Mensagens de Controle da Internet), RIP ("Routing Information Protocol" ou Protocolo de Informação de Roteamento), DNS ("Domain Name Service" ou Serviço de Nomes de Domínio) e SNMP ("Simple Network Management Protocol" ou Protocolo de Gerenciamento de Rede Simples).

A forma ativa deve ser cuidadosa para não sobrecarregar de tráfego a rede. Alguns protocolos só permitem o descobrimento de dispositivos em um segmento de rede local, como é o caso do ARP, e outros podem causar problemas na rede, como tempestades de broadcast.

A Gerência de Topologia também envolve, além do descobrimento de dispositivos e construção de mapas com visões adequadas da rede, a definição de LANs virtuais (VLANs) para facilitar alterações e adições na rede e a configuração do protocolo Spanning Tree para definir a escolha da bridge raiz em uma rede comutada, por exemplo.

Essa área de gerenciamento também permite a atualização de software dos diversos dispositivos "descobertos" e o registro dessas atualizações a partir de uma aplicação de inventário.

2..4. Gerenciamento de Contabilização

Corresponde à especificação, registro e controle do acesso de usuários e dispositivos aos recursos da rede. Também fazem parte deste gerenciamento quotas de utilização, cobrança por utilização e alocação de acesso privilegiado a recursos.

Dentre as suas funções, estão:

- Coleta de informações de utilização, monitorando quais recursos e quanto desses recursos estão sendo utilizados por que entidade;
- Estabelecimento de quotas de utilização com limites de uso de recursos por usuários ou grupos de usuários;
- Estabelecimento de escalas de tarifação, que podem também servir apenas para estatísticas;
- Aplicação das tarifas e faturamento.

2..5. Gerenciamento de Segurança

Seu objetivo é o controle do acesso aos recursos da rede de acordo com alguma política definida. Alguns de seus componentes podem ser as centrais de distribuição de chaves, as autoridades certificadoras e os firewalls.

Através dela, os elementos são protegidos, monitorando-se e detectando-se possíveis violações da política de segurança estabelecida, podendo, nesse caso, disparar alarmes. Mantém logs de segurança tanto para a posterior análise e geração de relatórios como para detectar violações não óbvias manualmente.

7.2 Arquitetura do Gerenciamento de Rede

Basicamente a arquitetura do gerenciamento de rede envolve quatro componentes principais: uma entidade gerenciadora, os dispositivos gerenciados, o protocolo de comunicação entre estes elementos e as informações de gerência.

A entidade gerenciadora é uma aplicação que roda em uma estação central de gerência, responsável pela coleta, processamento, análise e/ou apresentação de informações de gerenciamento de rede. Também a partir dela são iniciadas ações para controlar o comportamento da rede e é onde há a interação com os dispositivos gerenciados.

Normalmente, a entidade gerenciadora é construída como uma plataforma com aplicações sobre a mesma. A plataforma oferece funções básicas (comuns) de gerência e um ambiente para o desenvolvimento e a integração de aplicações que estarão usando essas funções básicas fornecidas.

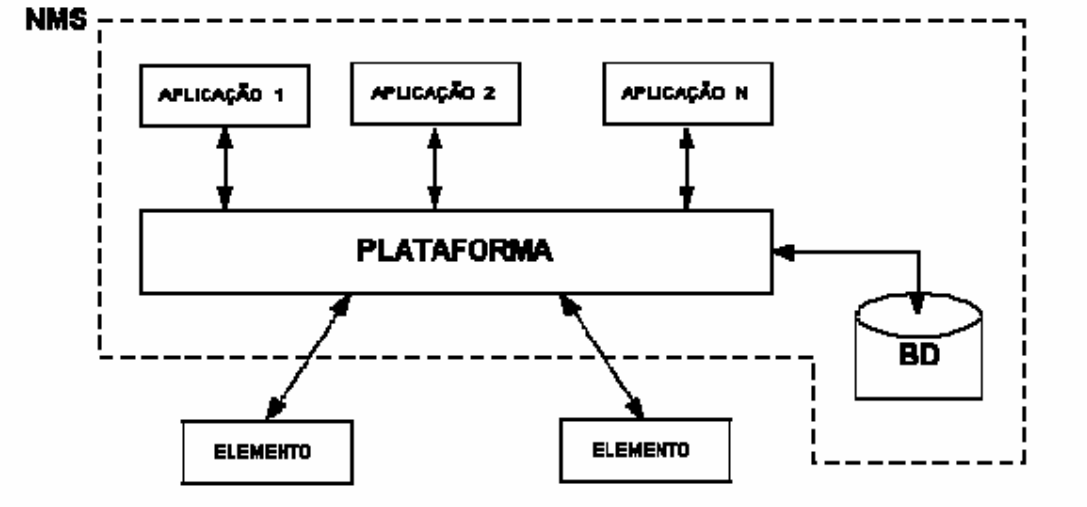


Figura 2.1: Diagrama Simplificado de uma Estação de Gerenciamento de Rede (NMS)

Os dispositivos gerenciados são os diversos equipamentos de rede, incluindo seu software. Em seu interior, estão os vários objetos gerenciados, como hardware e parâmetros de configuração de

hardware e software. Estes objetos são coletados dentro de uma base de informações de gerenciamento (“management information base” – MIB). Além da definição dos objetos gerenciados, nos dispositivos gerenciados, existe um processo que se comunica com a entidade gerenciadora e executa ações locais sobre os dispositivos de acordo com as instruções da mesma, chamado de agente de gerenciamento de rede.

O protocolo de gerenciamento de rede fornece um canal através do qual a entidade gerenciadora pode investigar o estado dos dispositivos gerenciados (monitoramento) e, indiretamente, tomar ações sobre eles mediante seus agentes (controle). Os agentes usam este protocolo para informar a entidade gerenciadora sobre a ocorrência de eventos excepcionais.

As informações de gerência definem os dados que podem ser referenciados em operações do protocolo e a forma como são acessados.

Esta análise é genérica e se aplica a uma série de padrões que vêm sendo propostos ao longo dos anos para o gerenciamento de rede. Os padrões mais importantes, projetados para serem independentes de produtos de fabricantes específicos ou de redes, são o OSI CMISE/CMIP (“common management service element/common management information protocol” ou elemento de serviço de gerenciamento comum/protocolo de informação de gerenciamento comum) e o SNMP (“simple network management protocol”), porém, devido à rápida disponibilização deste último, ele é o modelo de gerenciamento de rede mais amplamente usado e disseminado.

3. Introdução ao SNMP

As raízes da atual estrutura de gerenciamento de rede da Internet (Internet standard management framework) estão no SGMP ("simple gateway monitoring protocol" ou protocolo de monitoramento de gateway simples) projetado por um grupo de pesquisadores, usuários e administradores universitários de rede, cuja experiência com esse protocolo permitiu a definição do SNMP. A partir daí, o SNMP evoluiu por três versões, tendo a última sido lançada em abril de 1999.

A filosofia do SNMP é a de que o impacto de se adicionar gerência de rede aos elementos gerenciados deve ser mínimo, daí a solução básica de gerenciamento de rede e a versão 1 do protocolo SNMP serem muito simples, com a complexidade estando nas poucas estações de gerência e não nos diversos elementos gerenciados.

Dessa forma, pode-se definir o SNMP (Simple Network Management Protocol) como sendo um protocolo padrão da Internet para gerenciamento de dispositivos em uma rede IP, tais como roteadores, switches, servidores, estações de trabalho, impressoras, racks de modems e UPSs.

O uso do SNMP permite monitorar a "saúde" de dispositivos de rede, além de controlar estes dispositivos, tomando ações automaticamente mediante o surgimento de algum problema.

A estrutura de gerenciamento da Internet é constituída de quatro partes: os objetos de gerenciamento de rede, a linguagem de definição de dados, o protocolo de comunicação e as capacidades de segurança e administração[1].

Nessa estrutura, as informações de gerenciamento são representadas como uma coletânea de objetos gerenciados, que, em conjunto, formam um banco virtual de informações virtuais conhecido como

[1] Todos estes itens serão detalhados nas seções seguintes deste capítulo.

MIB. Entre estes objetos, estão contadores, informações descritivas, de estado e informações específicas sobre um protocolo. Os objetos MIB relacionados são organizados em módulos MIB.

A linguagem de definição dos objetos MIB é conhecida como SMI ("structure of management information" – estrutura de informação de gerenciamento) e define os tipos de dados, um modelo de objeto e as regras para escrever e revisar informações de gerenciamento.

O protocolo SNMP fornece o transporte das informações e dos comandos entre uma entidade gerenciadora e um agente que os executa em nome dessa entidade no dispositivo gerenciado.

O tópico de segurança e administração só foi aprimorado e adequadamente implementado no SNMPv3.

Dessa forma, a arquitetura de gerenciamento de rede da Internet é modular por projeto, com uma linguagem de definição de dados independente de protocolo e um protocolo independente de MIB. Esta modularidade permitiu a evolução independente de cada uma das quatro partes do SNMP.

7.2 Versões do SNMP

Existem três versões deste protocolo definidas em RFCs ("Request for Comments") pelo IETF ("Internet Engineering Task Force"), que é o responsável por definir os padrões dos protocolos que governam o tráfego da Internet. São elas:

- SNMPv1 – é o padrão atual do protocolo, definido na RFC 1157. Sua segurança é baseada em "communities", que são como senhas, cadeias de caracteres ("strings") sem criptografia que permitem qualquer aplicação baseada em SNMP que conheça tal senha ter acesso às informações de gerenciamento do dispositivo. Os tipos de acesso garantidos pelas

"communities" são somente leitura (read-only), leitura e escrita (read-write) e envio de notificações (trap);

- SNMPv2 – definida nas RFCs 1905, 1906 e 1907, corrige vários problemas da versão 1. Esta versão permite a especificação de variáveis com mais detalhes, inclusive com o uso de uma tabela de estrutura de dados para recuperação de dados mais facilmente.Também foram incluídos novos tipos de acesso, entre eles Inform-Request e Get-Bulk-Request, que serão detalhados nas próximas seções;
- SNMPv3 – é a próxima versão a se tornar padrão pelo IETF, sendo ainda uma proposta definida nas RFCs 1905, 1906, 1907, 2571, 2572, 2573, 2574 e 2575. Acrescenta o suporte a uma autenticação mais avançada e comunicação privada entre agentes e gerentes. Suas funcionalidades serão melhor apresentadas em seção próxima.

7.2 Gerentes e Agentes

A arquitetura do SNMP se baseia em duas entidades: gerentes e agentes.

Um gerente é um software que roda em um servidor normalmente referenciado como "Network Management Station" (NMS ou Estação de Gerenciamento de Rede) cuja função é requisitar o status dos dispositivos ("polling") e receber os traps dos agentes na rede. A requisição, nesse contexto, é o ato de pedir ao agente, que está rodando em algum dispositivo, alguma informação. Essa informação poderá ser usada para determinar se ocorreu algum problema na rede.

Um trap é a forma do agente informar à NMS que algum problema ocorreu. São enviados assincronamente, não em resposta a requisições da NMS. Baseado nas informações recebidas nos traps, a NMS é responsável por tomar alguma ação previamente configurada, como, por exemplo, enviar um e-mail ao administrador da rede.

O agente é um software que roda no dispositivo de rede que está sendo gerenciado, podendo ser um programa separado, um “daemon” na linguagem Unix, ou incorporado ao sistema operacional, como o Cisco IOS em um roteador ou o sistema operacional de baixo nível que controla um UPS. O agente provê informações de gerenciamento à NMS monitorando vários aspectos operacionais do dispositivo. Quando o agente descobre que algo não está funcionando corretamente, ele envia um trap à NMS, que tomará as ações necessárias. Alguns equipamentos enviam um trap “all clear” correspondente quando o estado volta ao normal. Isso é útil para determinar quando um problema foi resolvido.

A figura abaixo resume a comunicação entre o agente e o gerente.

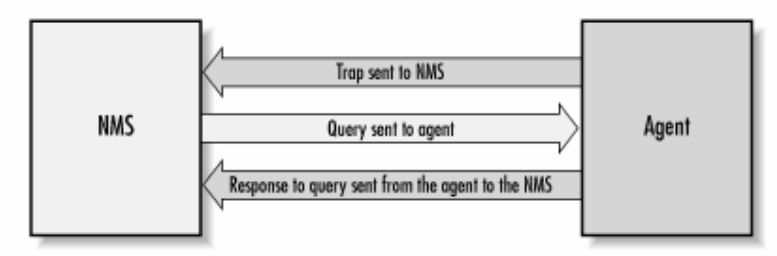


Figura 3.1: Diagrama da Comunicação entre Agentes e Gerentes

7.2 Estrutura de Informações de Gerenciamento (SMI)

A Estrutura de Informações de Gerenciamento (“Structure of Management Information” - SMI) provê a forma de se definir os objetos gerenciados e seus comportamentos. A lista formada define coletivamente as informações que a NMS pode usar para determinar a saúde geral do dispositivo onde o agente reside.

A linguagem de definição de informações de gerenciamento assegura que a sintaxe e a semântica dos dados de gerenciamento de rede sejam bem definidas e não apresentem ambigüidade. Ela

baseia-se na linguagem de definição de objetos ASN.1 (“Abstract Syntax Notation One” ou Notação de Sintaxe Abstrata Um).

A definição dos objetos gerenciados é dividida em 3 atributos:

- Nome – também conhecido como Object Identifier (OID), define exclusivamente um objeto gerenciado. Aparece sob a forma numérica ou textual. De qualquer maneira, é longo e inconveniente;

- Tipo e Sintaxe – o tipo de dados usado é definido por um subconjunto do Abstract Syntax Notation One (ASN.1);

- Codificação – cada instância de um objeto gerenciado é codificada em uma cadeia de bytes usando o Basic Encoding Rules (BER) de forma que possa ser transmitido sobre um meio de transporte como Ethernet, por exemplo, de forma independente da máquina, logo eliminando dificuldades como ordem dos bytes.

Os tipos de dados básicos são mostrados na tabela abaixo. Além desses objetos escalares, é possível impor uma estrutura tabular sobre um conjunto ordenado de objetos MIB usando a construção SEQUENCE OF.

Tipo de Dados	Descrição
INTEGER	Um número de 32 bits normalmente usado para especificar tipos enumerados no contexto de um único objeto gerenciado. Por exemplo, o estado operacional da interface de um roteador pode ser “up”, “down” ou em teste. Com tipos enumerados, 1 poderia representar “up”, 2 “down” e 3 em teste. O valor zero não deve ser usado como um tipo enumerado de acordo com a RFC 1155.
OCTET STRING	Uma cadeia de caracteres (string) de zero ou mais octetos geralmente usada para representar cadeias de texto, mas também algumas vezes usada para representar endereços físicos.
Counter	Um número de 32 bits com valor mínimo zero e máximo de 2^{32}-1 (4.294.967.295). Quando o valor máximo é alcançado, ele retorna a zero e começa novamente. É tipicamente usado para registrar informações, como o número de octetos enviados e recebidos em uma interface ou o número de erros e descartes vistos em uma

	interface. Um Counter é incrementado monotonamente de modo que seus valores nunca devem diminuir durante uma operação normal. Quando um agente é reiniciado, todos os valores Counter devem ser zerados. Deltas são usados para determinar se alguma informação útil pode ser obtida através de requisições sucessivas de valores Counter. Um delta é computado requisitando-se um valor Counter pelo menos duas vezes em uma instância e fazendo-se a diferença entre os resultados sobre um determinado intervalo de tempo.
OBJECT IDENTIFIER	Uma cadeia de decimais separados por pontos que representa um objeto gerenciado dentro da estrutura em árvore das MIBs[2]. Por exemplo, .1.3.6.1.4.1.9 representa a OID de fabricante privada da Cisco Systems.
NULL	Não usado atualmente no SNMP.
SEQUENCE	Define listas que contêm zero ou mais outros tipos de dados ASN.1
SEQUENCE OF	Define um objeto gerenciado que é composto de uma seqüência (SEQUENCE) de tipos ASN.1.
IpAddress	Representa um endereço Ipv4 de 32 bits. Nem o SMIv1 ou o SMIv2 discutem endereços IPv6 de 128 bits. Este caso será tratado pelo grupo de trabalho do IETF "SMI Next Generation" (SMING).
NetworkAddress	Mesmo que o tipo IpAddress, porém pode representar outros tipos de endereços de rede.
Gauge	Um número de 32 bits com valor mínimo zero e máximo 2^{32}-1 (4.294.967.295). Diferentemente do Counter, um Gauge pode aumentar ou diminuir durante sua operação normal, mas nunca pode exceder seu valor máximo. A velocidade de uma interface de um roteador é medida através de um Gauge.
TimeTicks	Um número de 32 bits com valor mínimo zero e máximo 2^{32}-1 (4.294.967.295). Mede o tempo em centésimos de segundos. O tempo desde a última interrupção do sistema (uptime) de um dispositivo é medido usando esse tipo de dados.
Opaque	Permite que qualquer outra codificação ASN.1 seja inserido em um OCTET STRING.

Tabela 3.1: Tipos de Dados Básicos definidos na SMIv1

Em um nível mais alto, a SMI fornece algumas construções.

A construção OBJECT-TYPE é usada para especificar o tipo de dado, o status e a semântica de um objeto gerenciado e contém quatro cláusulas. A cláusula de SYNTAX especifica os tipos de dados básicos associados ao objeto. A ACCESS especifica se o objeto gerenciado pode ser lido ou escrito. A cláusula STATUS indica se a identificação do objeto é obrigatória (mandatory), obsoleta (obselete – caso em que não deve ser implementada, pois sua definição está incluída apenas por motivos históricos) ou opcional. A DESCRIPTION contém uma definição textual do objeto, identificando a finalidade do objeto gerenciado e fornecendo todas as informações semânticas necessárias para implementá-lo. Os nomes dos objetos são definidos usando letras maiúsculas e minúsculas, mas a primeira letra é sempre minúscula.

O formato dessa construção é dado a seguir.

[2] A estrutura hierárquica em árvore das MIBs será vista na seção sobre MIBs.

```
<name> OBJECT-TYPE
    SYNTAX <datatype>
     ACCESS  <either  read-only,  read-write,  write-only,  or  not-
accessible>
  STATUS <either mandatory, optional, or obsolete>
    DESCRIPTION
        "Textual  description  describing  this  particular  managed
object."
    ::= { <Unique OID that defines this object> }
```

Uma seqüência é simplesmente uma lista de objetos e seus tipos de dados que formam uma tabela conceitual, através da construção SEQUENCE OF. Um exemplo é mostrado abaixo. A tabela pode ter tantas linhas quanto se queira e é responsabilidade do agente gerenciá-las. A NMS também pode inserir novas linhas. Cada linha de uma tabela é indexada com uma sentença do tipo INDEX. O agente é responsável por garantir que cada índice seja único no contexto da tabela.

```
ifTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF IfEntry
    ACCESS  not-accessible
    STATUS  mandatory
    DESCRIPTION
        "A list of interface entries. The number of entries is given
by the value of ifNumber."
    ::= { interfaces 2 }

IfEntry ::=
    SEQUENCE {
        ifIndex
            INTEGER,
        ifDescr
            DisplayString,
        ifType
            INTEGER,
        ifMtu
            INTEGER,
        .
        .
        .
        ifSpecific
            OBJECT IDENTIFIER
    }
```

A construção MODULE-IDENTITY permite que objetos relacionados entre si sejam agrupados, como conjunto, dentro de um “módulo”. Além das definições OBJECT-TYPE, contém cláusulas

para documentar as informações de contato do autor do módulo, a data da última atualização, um histórico de revisões e uma descrição textual do módulo.

A SMIv2 expande a lista de objetos SMI adicionando novos tipos de dados resumidos na tabela abaixo e adicionando os novos conceitos introduzidos pela versão 2 do SNMP.

Tipo de Dados	Descrição
Integer32	Mesmo que um INTEGER.
Counter32	Mesmo que um Counter.
Gauge32	Mesmo que um Gauge.
Unsigned32	Representa valores decimais no intervalo de 0 a 2^{32}-1 inclusive.
Counter64	Similar ao Counter32, porém seu valor máximo é 18.446.744.073.709.551.615. Counter64 é ideal para situações em que um Counter32 retornaria a zero num espaço muito curto de tempo.
BITS	Uma enumeração não negativa chamada bits.

Tabela 3.2: Novos Tipos de Dados introduzidos pela SMIv2

A forma de se definir um objeto na SMIv2 recebeu novos campos opcionais, dando mais controle sobre a maneira como um objeto é acessado, permitindo aumentar uma tabela acrescentando mais colunas e fornecendo descrições mais abrangentes.

```
<name> OBJECT-TYPE
    SYNTAX <datatype>
    UnitsParts <Optional, see below>
    MAX-ACCESS <See below>
    STATUS <See below>
    DESCRIPTION
        "Textual description describing this particular managed
object."
    AUGMENTS { <name of table> }
    ::= { <Unique OID that defines this object> }
```

As partes alteradas são descritas na tabela abaixo.

Melhoramento na Definição de um Objeto	Descrição
UnitsParts	Uma descrição textual da unidade (segundos, milissegundos, etc) usada para representar um objeto.
MAX-ACCESS	O ACCESS de um OBJECT-TYPE pode ser um MAX-ACCESS do SNMPv2. As opções válidas são read-only (somente leitura), read-write (leitura e escrita), read-create (leitura e criação), not-accessible (não acessível) e accessible-for-notify

	(acessível para notificações).
STATUS	Essa sentença pode ser estendida para permitir as palavras-chave current (atual e válida), obsolete (mantida apenas por motivos históricos não devendo ser implementada) e deprecated (depreciada, obsoleta, mas implementável por causa de sua interoperabilidade com implementações mais antigas). current em SNMPv2 tem o mesmo significado que mandatory numa MIB SNMPv1.
AUGMENTS	Em alguns casos, pode ser útil inserir colunas em uma tabela já existente. A sentença AUGMENTS permite que uma tabela seja estendida adicionando-se uma ou mais colunas, representadas por algum outro objeto. Essa sentença requer o nome da tabela que o objeto estará estendendo.

Tabela 3.3: Alterações na Definição de um Objeto Gerenciado na SMIv2

A construção NOTIFICATION-TYPE é usada para especificar informações referentes a mensagens SNMPv2-Trap e Information-Request, descritas em seção mais a frente, geradas por um agente ou por uma entidade gerenciadora. Entre essas informações, estão uma DESCRIPTION, descrição textual que especifica quando tais mensagens devem ser enviadas, bem como uma lista de valores que devem ser incluídos na mensagem gerada.

Um exemplo é mostrado a seguir.

```
linkDown NOTIFICATION-TYPE
    OBJECTS { ifIndex, ifAdminStatus, ifOperStatus }
    STATUS  current
    DESCRIPTION
        "A linkDown trap signifies that the SNMPv2 entity, acting in
an agent role, has detected that the ifOperStatus object for one of
its communication links left the down state and transitioned into
some other state (but not into the notPresent state). This other
state is indicated by the included value of ifOperStatus."
    ::= { snmpTraps 3 }
```

Além destas, a construção MODULE-COMPLIANCE define o conjunto de objetos gerenciados dentro de um módulo que um agente deve implementar e a AGENT-CAPABILITIES especifica as capacidades dos agentes relativas às definições de notificação de objetos e de eventos.

A SMIv2 também introduz novas convenções textuais que permitem que objetos sejam criados de forma mais abstrata.

7.2 Linguagem de Descrição de Objetos ASN.1 e Codificação BER

Devido ao fato de diferentes arquiteturas de computadores armazenarem e apresentarem seus dados de forma diferente, o SNMP adotou um método independente de máquina, sistema operacional e linguagem para descrever os dados transmitidos em suas mensagens e regras que estabelecem como cada um desses tipos de dados deve ser transmitido pela rede.

A SMI, que é baseada na linguagem ASN.1, é a linguagem usada para definir a descrição dos objetos num modelo que, pelos termos da ISO, é o serviço de apresentação.

Além dessa linguagem de descrição de dados, a ASN.1 oferece regras básicas de codificação ("basic encoding rules" – BER), que especificam como instâncias de objetos devem ser enviadas pela rede. A BER adota a abordagem TLV ("type, length, value" – tipo, comprimento, valor) para a codificação de dados para a transmissão. Cada tipo de dado tem um código associado que é transmitido como o tipo do código TLV. Os inteiros são transmitidos na representação "big-endian", ou seja, enviando os bytes (ou bits) mais significativos primeiro.

7.2 Base de Informações de Gerenciamento (MIB)

Os objetos gerenciados são especificados com o uso da construção OBJECT-TYPE da SMI e agrupados em módulos MIB ("Management Information Base") que utilizam a construção MODULE-IDENTITY. Esses valores, em conjunto, refletem o "estado" atual da rede.

A MIB pode ser encarada como um banco de dados de objetos gerenciados que um agente observa. Qualquer tipo de informação estatística ou de status que pode ser acessado pela NMS é definido em uma MIB. A SMI provê a forma de definir os objetos gerenciados enquanto que a MIB é a definição propriamente dita usando a sintaxe da SMI dos objetos.

No esforço de padronizar os diversos módulos MIB existentes, a IETF adotou uma estrutura padronizada de identificação de objetos que já tinha sido publicada pela ISO dentro da premissa de identificar todo e qualquer objeto padronizado possível, incluindo formato de dados, protocolo ou informação, em qualquer rede, independente das organizações dedicadas à padronização das redes (IETF, ISO, IEEE ou ANSI), do fabricante do equipamento ou do proprietário da rede. A estrutura de identificação de objeto adotada é parte da linguagem de definição de objetos ASN.1.

Os objetos são nomeados hierarquicamente numa estrutura em árvore. Cada ponto da árvore possui um nome e um número e pode ser identificado pela seqüência de nomes (ou números) que especificam o trajeto da raiz até o ponto em questão.

No topo da hierarquia, estão a antiga CCITT (agora ITU-T), a ISO e um ramo para o esforço conjunto realizado por essas duas organizações. No ramo da ISO, estão um ramo para os registros de todos os padrões ISO, um para os padrões emitidos por entidades padronizadoras de vários países membros e um para os padrões emitidos por entidades reconhecidas pela ISO. Embaixo deste último, está o Departamento de Defesa dos Estados Unidos, sob o qual está o ramo dos padrões da Internet, além de várias outras organizações.

Sob Internet, estão sete categorias, dentre elas, management sob a qual estão as MIBs padronizadas e private sob a qual as empresas privadas posicionam suas MIBs com nomes e códigos registrados na Internet Assigned Numbers Authority (Iana ou Autoridade de Atribuição de Números da Internet).

A figura a seguir mostra a estrutura em árvore que define os objetos gerenciados.

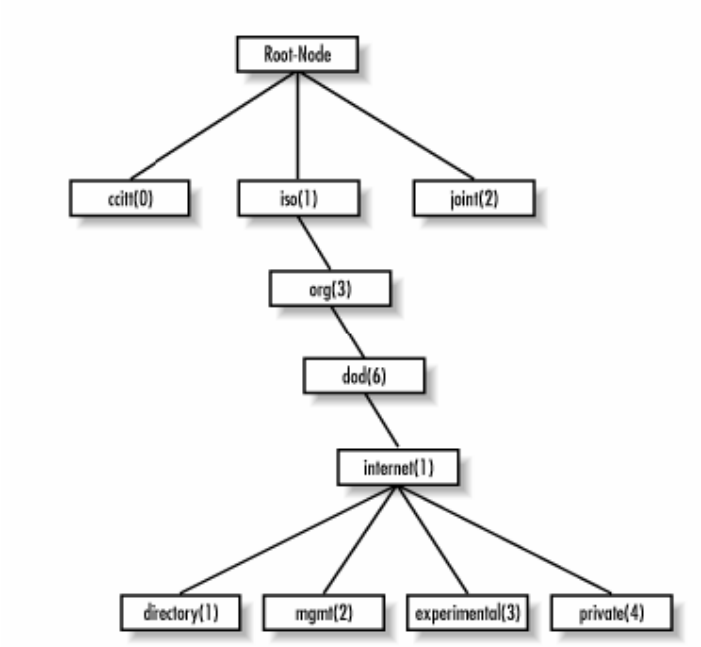


Figura 3.2: Estrutura Hierárquica de Definição de Objetos

Um agente pode rodar várias MIBs mas todos os agentes rodam uma MIB em particular chamada MIB-II (RFC 1213). Essa MIB padrão define variáveis para informações como estatísticas de interface (velocidade da interface, MTU, octetos enviados, octetos recebidos, etc...) assim como variáveis pertinentes ao sistema, como a localização do sistema, contato do administrador, entre outras. O principal objetivo da MIB-II é prover informação geral para o gerenciamento TCP/IP.

Além dessa, cada fabricante pode implementar sua MIB proprietária de modo a prover informações específicas de seu equipamento.

Outras MIBs importantes são ATM MIB (RFC 2515), Frame Relay DTE Interface Type MIB (RFC 2115), BGP Version 4 MIB (RFC 1657), RDBMS MIB (RFC 1697), RADIUS Authentication Server MIB (RFC 2619), Mail Monitoring MIB (RFC 2249), DNS Server MIB (RFC 1611), dentre muitas outras definidas.

7.2 Protocolo de Transporte

Apesar de o SNMP poder ser transportado sobre qualquer protocolo, é tipicamente usado o UDP. Essa escolha se deve ao fato dele ser um protocolo não-orientado a conexão. Sua vantagem é o baixo overhead, reduzindo o impacto na performance da rede, além de que em uma rede congestionada e com problemas, um protocolo que tenta entregar a mensagem mas desiste se não consegue é uma melhor escolha do que um que inunde a rede com retransmissões em busca de confiabilidade (TCP).

A confiabilidade é provida pelo protocolo de aplicação através do campo Request ID presente nas mensagens de requisição e de resposta. Este campo assume o mesmo valor na mensagem de resposta correspondente a uma requisição, permitindo que a entidade gerenciadora detecte requisições ou respostas perdidas. Ela é responsável por decidir pela retransmissão das requisições na ausência de uma resposta dentro de um intervalo de tempo especificado.

No caso de traps, entretanto, a NMS não confirma o recebimento dos mesmos, logo o agente não tem conhecimento se a mensagem foi entregue.

O protocolo SNMP usa, por padrão, as portas 161 para enviar e receber requisições e 162 para receber traps dos agentes.

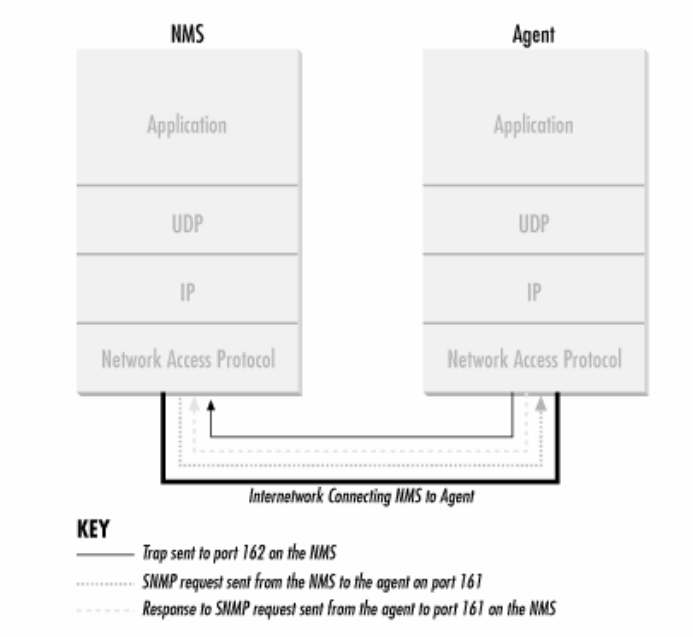


Figura 3.3: SNMP e a pilha de protocolos TCP/IP

7.2 Communities SNMP

O SNMPv1 e SNMPv2 usam o conceito de communities, que são essencialmente senhas, para estabelecer autenticação entre gerentes e agentes. Um agente pode ser configurado com três communities: read-only, read-write e traps, cada qual controlando diferentes tipos de acessos.

A read-only permite que dados sejam lidos mas não permite que seus valores sejam modificados. A read-write permite que dados sejam lidos e modificados. A trap permite que o agente envie traps para o gerente.

A vulnerabilidade das communities é que elas são enviadas sem criptografia. Para reduzir os riscos relacionados a elas, podem ser usados firewalls para restringir a comunicação SNMP entre os dispositivos, VPNs para garantir que o tráfego esteja criptografado, ou alterar as communities regularmente, inclusive usando scripts se a rede for muito grande.

7.2 Operações do Protocolo SNMP

Os dois modos de operação mais comuns do SNMP são o modo comando-resposta e as mensagens trap.

No primeiro, a entidade gerenciadora envia uma requisição a um agente, que a recebe, realiza alguma ação e envia uma resposta à requisição. Em geral, uma requisição é usada para consultar (recuperar) ou modificar (estabelecer) valores de objetos MIB associados ao dispositivo gerenciado.

As mensagens trap são usadas para notificar uma entidade gerenciadora de uma situação excepcional que resultou em mudança nos valores dos objetos da MIB.

São definidos sete tipos de mensagens ou PDUs (“Protocol Data Units”) a partir do SNMPv2. São elas:

- Get Request (versões 1, 2 e 3);
- Get-Next Request (versões 1, 2 e 3);
- Get-Bulk Request (versões 2 e 3);
- Set Request (versões 1, 2 e 3);
- Inform Request (versões 2 e 3);
- Trap / Notification (versões 1, 2 e 3);
- Get-Response (versões 1, 2 e 3);

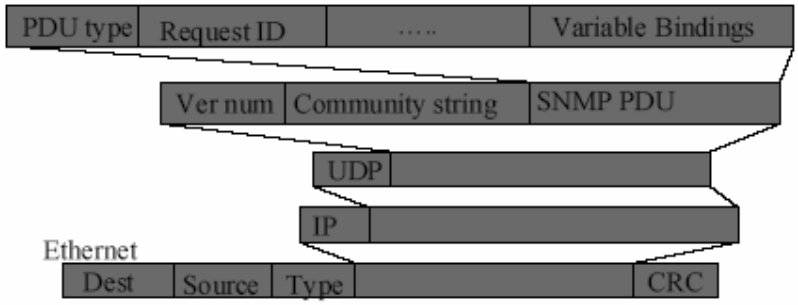


Figura 3.4: Formato de uma PDU SNMP

A requisição get é iniciada na NMS e enviada ao agente. Este a recebe, processa e, caso consiga coletar a informação solicitada, responde com um get-response à NMS, onde este é processado. Alguns dispositivos podem não responder à requisição get, descartando o pacote, por estarem sob uma carga de processamento excessiva, como pode ser o caso de roteadores, por exemplo.

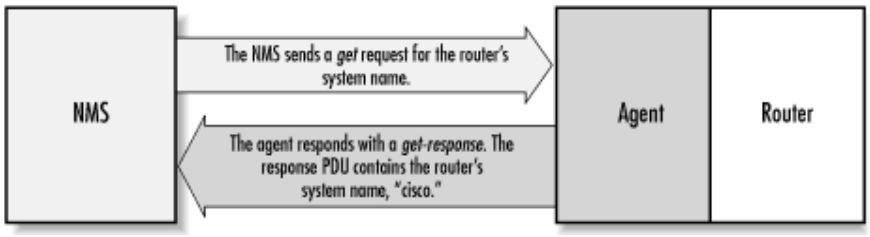


Figura 3.5: Diagrama da Seqüência de uma Requisição Get

Um dos itens contidos na requisição get é a chamada “variable binding” ou “varbind”, que é uma lista de objetos de MIB que permite ao agente saber qual informação retornar ao gerente e a este retirar tal informação do pacote de resposta. Uma variable binding é um par OID = valor, onde o agente preenche o valor correspondente à OID solicitada.

Na solicitação do valor de um objeto MIB, é necessário especificar a instância. Para objetos escalares, ou seja, que não estão definidos como uma linha em uma tabela, a instância é zero. Para uma tabela, a instância permite selecionar uma específica linha na tabela.

A requisição get permite apenas que um objeto seja devolvido a cada vez que o comando é executado. Para requisitar múltiplos objetos, foi criado o comando get-next, que, na verdade, é uma seqüência de comandos que devolvem um grupo de valores da MIB. Para cada objeto desejado, um comando get-next separado e um get-response correspondente são gerados. O get-next atravessa um ramo em ordem lexicográfica. Assim que a NMS recebe a resposta de um get-next, ela executa um novo get-next até que o agente retorne um erro, significando que o fim da MIB foi alcançado.

A versão 2 do SNMP define a operação get-bulk, que permite buscar grandes seções de informação de uma tabela de uma única vez. Isso porque se o agente não consegue preencher todas as informações solicitadas (com um get-next, por exemplo) em um único pacote de resposta, ele retorna uma mensagem de erro sem dados.

A operação get-bulk permite mensagens incompletas. Ela usa dois campos: “nonrepeaters” e “max-repetitions”. “Nonrepeaters” define os primeiros N objetos que podem ser devolvidos com um simples comando get-next; são objetos escalares. “Max-repetitions” indica o número de operações get-next que devem ser tentadas para buscar a informação restante; na prática representa o número de linhas de uma tabela cuja informação solicitada deve ser devolvida.

O comando set é usado para modificar o valor de objetos gerenciados definidos como read-write ou write-only e criar novas linhas em uma tabela, sendo mandatória a informação do tipo de dado do valor a ser inserido presente na definição do objeto na MIB. Permite que sejam alterados mais de um objeto por vez, porém, se a operação falhar em um dos objetos, ela não será realizada para nenhum dos demais.

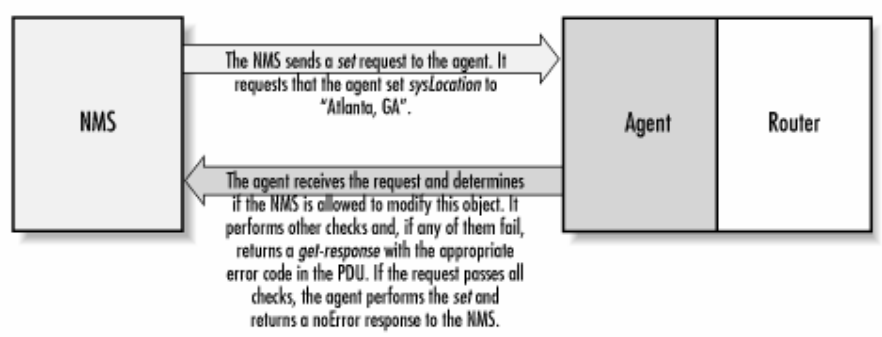


Figura 3.6: Diagrama da Seqüência de uma Requisição Set

Para cada falha de processamento nas operações descritas acima, existe uma mensagem de erro com código correspondente. A versão 2 do SNMP provê respostas de erro mais robustas, desde que tanto o agente quanto o gerente a suportem.

Um trap é originado no agente a partir de algum evento e enviado ao endereço de destino configurado no próprio agente (normalmente da NMS). A definição de quais traps um agente pode gerar é feita nas MIBs que ele suporta. A NMS deve ser capaz de interpretar o trap recebido, conhecendo o que ele significa e interpretando a informação contida nele.

O trap é primeiramente identificado por seu número de trap genérico, que são sete (0-6) conforme tabela abaixo. O trap genérico 6 é usado para traps específicos de cada fabricante ou usuário individual que estão fora do escopo dos 6 outros números. Os traps específicos dos fabricantes são identificados por um ID da empresa (no ramo private.enterprises da estrutura hierárquica em árvore das MIBs) e um número de trap específico definido pela empresa.

Nome e Número de Trap Genérico	Definição
coldStart (0)	Indica que um agente foi reiniciado. Todas as variáveis de gerenciamento serão reiniciadas; especificamente Counters e Gauges serão zerados. Uma vantagem desse trap é a possibilidade de se determinar quando um novo hardware foi adicionado à rede, pois, quando um dispositivo é ligado, ele envia este trap ao receptor de traps. Se o endereço deste destino dos traps estiver definido corretamente (ou seja, com o endereço IP da NMS), a NMS receberá o trap e determinará se deve gerenciar o dispositivo.
warmStart (1)	Indica que um agente se reinicializou. Nenhuma das variáveis de gerenciamento serão reiniciadas.
linkDown (2)	Enviado quando a interface de um dispositivo cai. O primeiro par de OID-valor identifica

	qual interface caiu.
linkUp (3)	Enviado quando uma interface volta ao funcionamento normal. O primeiro par OID-valor identifica qual interface voltou ao seu funcionamento normal.
authenticationFailure (4)	Indica uma tentativa de requisição ao agente com a community incorreta. É útil para determinar ataques de acesso não-autorizado aos dispositivos.
egpNeighborLoss (5)	Indica que um vizinho de um protocolo de roteamento esterno (Exterior Gateway Protocol – EGP) caiu.
enterpriseSpecific (6)	Indica que o trap é específico de um fabricante. Os fabricantes de dispositivos com suporte a SNMP e os usuários definem seus próprios traps sob o ramo "private-enterprise" da estrutura em árvore dos objetos SMI. Para tratar corretamente este trap, a NMS deve decodificar o número específico do trap que é parte da mensagem SNMP.

Tabela 3.4: Números Genéricos de Traps

A informação contida no trap está na forma de "variable bindings". Para os traps genéricos de 0 a 5, a forma como a informação contida no trap deve ser interpretada já está construída na NMS ou receptor de traps. Para os específicos de fabricantes, as MIBs correspondentes devem ser carregadas (ou compiladas) na NMS.

Um exemplo de definição de um trap é mostrado a seguir, com o ID da empresa (rdbmsTraps) e o número específico do trap (2). A definição da "variable binding" enviada como informação no trap é dada em outra parte da MIB.

```
rdbmsOutOfSpace TRAP-TYPE
    ENTERPRISE  rdbmsTraps
    VARIABLES   { rdbmsSrvInfoDiskOutOfSpaces }
    DESCRIPTION
        "An rdbmsOutOfSpace trap signifies that one of the database
servers managed by this agent has been unable to allocate space for
one of the databases managed by this agent. Care should be taken to
avoid flooding the network with these traps."
    ::= 2
```

Na intenção de padronizar o formato das PDUs, o SNMPv2 definiu uma nova construção já vista em seção anterior: o NOTIFICATION-TYPE. Dessa forma, o formato da PDU dos traps (ou notificações) passou a ser idêntico ao das PDUs de gets e sets. Essa versão elimina o conceito de traps genéricos, definindo várias notificações em MIBs públicas. O SNMPv3 usa os traps (ou "notificações") da versão 2, acrescentando apenas as capacidades de autenticação e privacidade aos mesmos.

Os programas usados para o envio de traps basicamente precisam dos mesmos argumentos:

- a porta UDP para a qual deve-se enviar o trap (a padrão é 162);
- a versão do SNMP a ser utilizada;
- o hostname ou endereço IP da NMS ou receptor de traps;
- a community, já que muitas NMSs estão configurados para tratar apenas traps com uma community específica;
- a OID da "empresa" completa sem o número específico do trap a ser enviado;
- o hostname ou endereço IP do originador do trap. Esta informação é útil quando existe um proxy entre o agente e o gerente, pois identifica o dispositivo que realmente gerou o trap, independente do endereço presente no cabeçalho IP;
- o número de trap genérico (6 no caso de trap privado específico);
- o número específico do trap;
- o timestamp, que é o intervalo de tempo entre a última inicialização do elemento de rede e o momento de geração do trap. Corresponde à hora atual do sistema se nada for informado;
- os conjuntos de OIDs, tipos e valores a serem incluídos nos traps.

O SNMPv2 fornece um mecanismo de informe (Inform-Request) que permite comunicação entre gerentes, com confirmação de recebimento. Também pode ser usado para substituir o envio de traps por um agente. Dessa forma, o agente seria notificado do recebimento do trap pelo gerente.

A operação de report foi definida no draft do SNMPv2 mas nunca implementada. Agora é parte da especificação do SNMPv3. A finalidade é permitir que os dispositivos SNMP se comuniquem entre si reportando principalmente problemas com o processamento de mensagens SNMP.

No contexto do SNMP, podem ser feitos monitoramentos internos e externos. O monitoramento interno é feito a partir de scripts ou aplicativos locais, normalmente agendados na cron para rodar periodicamente. Este tipo de monitoramento pode monitorar variáveis pré-definidas e gerar traps caso algum limite estabelecido seja ultrapassado, evitando excesso de tráfego desnecessário na rede. O externo é realizado por uma NMS.

Além disso, podem ser definidos limites que se forem ultrapassados para cima ou para baixo desencadeiam eventos pré-configurados. Um parâmetro importante no planejamento é o intervalo entre as requisições de informação para efetuar este monitoramento. Para se chegar a este valor, deve-se considerar o impacto no processamento em CPU no dispositivo gerenciado, o consumo de banda e os tipos de valores solicitados.

7.2 Segurança e Administração

Essas capacidades foram desenvolvidas a partir do SNMPv3. O SNMPv3 fornece criptografia, autenticação, proteção contra ataques de reprodução e controle de acesso. Sua segurança é conhecida como segurança baseada no usuário, pois utiliza o conceito tradicional de um usuário, identificado por um nome de usuário, ao qual as informações de segurança, dentre elas uma senha, um valor de chave e acessos privilegiados, são associadas.

As PDUs podem ser criptografadas usando o DES no modo encadeamento de blocos de cifras, desde que ambas as partes conheçam a chave compartilhada.

A autenticação e a proteção contra adulterações são obtidas a partir do uso combinado de uma função de hash como o algoritmo MD5 e um valor de chave secreta. A chave secreta, que não precisa ser a mesma usada na criptografia, é anexada à mensagem e o código de integridade de mensagem (MIC) é calculado sobre essa combinação. Assim, se o valor obtido no receptor

combinar com o transmitido, garante-se a integridade dos dados e a identidade confiável do remetente.

Para garantir que uma mensagem recebida não seja uma reprodução de alguma mensagem anterior, o receptor exige que o remetente inclua em cada mensagem um valor baseado em um contador do receptor. Esse contador reflete o período de tempo decorrido entre a última reinicialização do software de gerenciamento de rede do remetente e o número total de reiniciaçizações desde a última vez que o software de gerenciamento de rede do receptor foi configurado. Contanto que o contador de uma mensagem recebida esteja dentro de uma determinada margem de erro em relação ao próprio valor do receptor, a mensagem é aceita como uma mensagem original e a partir daí pode ser autenticada e/ou decriptografada.

O SNMPv3 fornece um controle de acesso baseado em visões que controla quais das informações de gerenciamento de rede podem ser consultadas e/ou definidas por quais usuários. Tais informações de controle de acesso e políticas são armazenadas em um banco de dados de configuração local (“local configuration datastore” – LCD), que é acessível em parte como objetos gerenciados definidos na MIB de Configuração do Modelo de Controle de Acesso Baseado em Visões (RFC 2575).

Para todas as questões práticas, no entanto, segurança é a única característica que a versão 3 do SNMP acrescenta. Todas as operações definidas nas versões anteriores são suportadas e não apresentam modificações. Foram introduzidos novas convenções textuais, que na verdade são simplesmente formas mais precisas de se interpretar os tipos de dados definidos nas outras versões, conceitos e terminologias.

Estes novos conceitos definem uma arquitetura, ao invés de um simples conjunto de mensagens. Nessa arquitetura, não existe mais a noção de agentes e gerentes, todos são apenas entidades SNMP. Cada entidade consiste de uma "máquina" SNMP ("engine") e uma ou mais aplicações SNMP.

A "máquina" SNMP ("engine") é constituída de 4 partes: Despachante ("Dispatcher"), o Subsistema de Processamento de Mensagens, o Subsistema de Segurança e o Subsistema de Controle de Acesso.

A função do Despachante é enviar e receber mensagens. Ele verifica a versão da mensagem recebida e, caso seja de uma versão suportada, encaminha ao Subsistema de Processamento de Mensagens. Também encaminha mensagens a outras entidades SNMP.

O Subsistema de Processamento de Mensagens prepara as mensagens que serão enviadas e extrai os dados das mensagens recebidas. Um sistema de processamento de mensagens pode conter mais de um módulo de processamento, por exemplo um para cada versão de SNMP e um para outros modelos de processamento ainda a serem definidos.

O Subsistema de Segurança provê serviços de autenticação, integridade e privacidade, conforme explicado acima.

O Subsistema de Controle de Acesso tem a função de controlar o acesso aos objetos das MIBs, tanto indicando quais objetos um usuário pode acessar como que tipo de operações ele pode executar sobre esses objetos.

A versão 3 divide o que era conhecido por SNMP em algumas aplicações, a saber:

- Gerador de Comandos – gera as requisições get, get-next, get-bulk e set e processa as respostas. É implementada por uma NMS;
- Respondedor de Comandos – responde às requisições get, get-next, get-bulk e set, sendo implementada numa entidade que equivale ao agente das versões anteriores;
- Originador de Notificação – gera traps e notificações SNMP. É implementada em entidades que equivalem aos agentes das versões anteriores, porém utilitários individuais geradores de traps também estão incluídos;
- Receptor de Notificações – recebe traps e mensagens inform, sendo implementada em uma NMS;
- Encaminhador Proxy – facilita a passagem de mensagens entre as entidades.

Uma vantagem significativa da versão 3 sobre as outras é a capacidade de se estender a estrutura ("framework"), adicionando-se novas aplicações como definido no RFC 2571.

A figura abaixo mostra os componentes de uma entidade em seu conjunto.

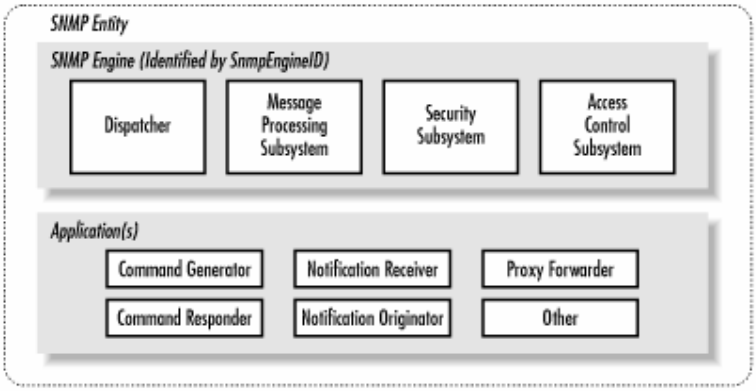


Figura 3.7: Entidade SNMPv3

A tabela de usuários, senhas e outras informações de autenticação é outra tabela SNMP definida em MIB e acessível pelas operações SNMP. É chamada de usmUser.

7.2 MIB-II

A MIB-II é um grupo de gerenciamento muito importante pois todo dispositivo que suporta SNMP deve suportá-la também. Sua posição na árvore da internet é mostrada na figura abaixo e seus objetos-base são descritos brevemente na tabela.

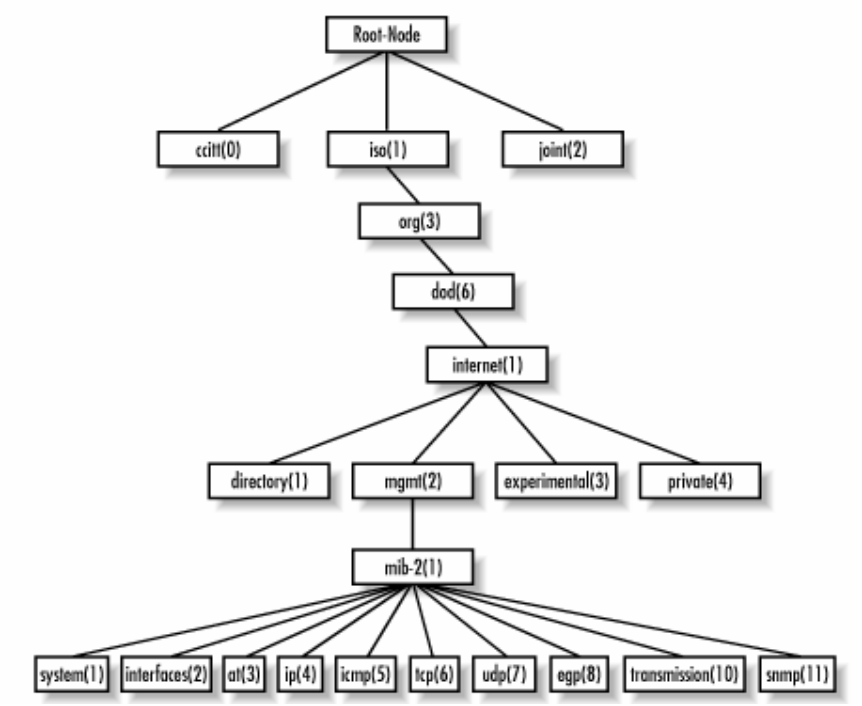


Figura 3.8: Posição da MIB-II na estrutura hierárquica em árvore das MIBs

Nome do Ramo	OID	Descrição
system	1.3.6.1.2.1.1	Define uma lista de objetos que pertence à operação do sistema, como o tempo desde a última interrupção (uptime), o contato e o nome do sistema.
interfaces	1.3.6.1.2.1.2	Mantém registro do estado de cada interface de um objeto gerenciado. O grupo interfaces monitora quais interfaces estão funcionando normalmente e quais não estão e registra o número de octetos enviados e recebidos, erros e descartes, entre outros.
at	1.3.6.1.2.1.3	O grupo tradução de endereços (at – address translation) é depreciado e é fornecido apenas para compatibilidade com implementações anteriores. Ele provavelmente será retirado da MIB-III.
ip	1.3.6.1.2.1.4	Mantém registro de vários aspectos do protocolo IP, incluindo roteamento IP.
icmp	1.3.6.1.2.1.5	Registra estatísticas do protocolo ICMP, como erros, descartes, entre outros.
tcp	1.3.6.1.2.1.6	Registra, entre outras coisas, o estado das conexões TCP.
udp	1.3.6.1.2.1.7	Registra estatísticas UDP, como datagramas enviados e recebidos, etc.
egp	1.3.6.1.2.1.8	Registra várias estatísticas sobre o EGP e mantém uma tabela de vizinhos EGP.
transmission	1.3.6.1.2.1.10	Não existem objetos definidos para este grupo atualmente, mas outras MIBs específicas para um meio de transmissão são definidas sob este ramo.
snmp	1.3.6.1.2.1.11	Mede a performance da implementação SNMP de base da entidade gerenciada e registra estatísticas como o número de pacotes SNMP enviados e recebidos.

Tabela 3.5: Descrição dos Objetos-Base da MIB-II

7.2 Gerenciamento de Hosts

Para o gerenciamento de recursos dos hosts de uma rede, foi definida uma MIB específica chamada Host Resources MIB (RFC 2790), que define um conjunto de objetos para ajudar a gerenciar aspectos críticos de um sistema operacional.

Esta MIB, incluída dentro da MIB-II, define seis grupos:

```
hrSystem        OBJECT IDENTIFIER ::= { host 1 }
hrStorage       OBJECT IDENTIFIER ::= { host 2 }
hrDevice        OBJECT IDENTIFIER ::= { host 3 }
hrSWRun         OBJECT IDENTIFIER ::= { host 4 }
hrSWRunPerf     OBJECT IDENTIFIER ::= { host 5 }
hrSWInstalled   OBJECT IDENTIFIER ::= { host 6 }
```

Os grupos fornecem informações sobre o sistema. O hrSystem define objetos que pertencem ao próprio sistema, incluindo o tempo desde a última interrupção (“uptime”), data e hora do sistema, usuários e processos. O hrDevice e hrStorage definem objetos relacionados a sistemas de arquivos e armazenamento, tais como memória total do sistema, utilização de disco e percentagem de ocupação de CPU. O hrSWRun, hrSWRunPerf e hrSWInstalled definem objetos relacionados aos vários aspectos dos softwares rodando ou instalados no sistema.

7.2 Monitoramento Remoto (RMON)

O “Remote Network Monitoring” (RMON) foi desenvolvido para ajudar a entender o funcionamento da rede como um todo assim como o efeito que os dispositivos individuais provocam nessa rede. O RMONv1 está definido na RFC 2819 e uma versão aprimorada chamada RMONv2 foi definida na RFC 2021. A primeira provê à NMS informações estatísticas a nível de pacote sobre toda a LAN ou WAN e a segunda provê informações a nível de rede e de aplicação.

Essas estatísticas podem ser colhidas de diversas formas, dentre elas utilizando-se um ponto de prova (“probe”) em cada segmento de rede que se deseja monitorar. A MIB RMON foi criada de modo que este ponto de prova fosse capaz de colher as informações em modo “offline” sem a necessidade de a NMS estar constantemente requisitando essas informações, que podem ser requisitadas no momento apropriado.

Uma característica que a maioria destes pontos de prova implementam é a capacidade de definir limiares para várias condições de erro e, quando um desses limiares é ultrapassado, alertar a NMS enviando um trap SNMP ou registar em um arquivo de log.

A MIB RMONv1 define 9 grupos:

```
statistics        OBJECT IDENTIFIER ::= { rmon 1 }
history           OBJECT IDENTIFIER ::= { rmon 2 }
alarm             OBJECT IDENTIFIER ::= { rmon 3 }
hosts             OBJECT IDENTIFIER ::= { rmon 4 }
hostTopN          OBJECT IDENTIFIER ::= { rmon 5 }
matrix            OBJECT IDENTIFIER ::= { rmon 6 }
filter            OBJECT IDENTIFIER ::= { rmon 7 }
capture           OBJECT IDENTIFIER ::= { rmon 8 }
event             OBJECT IDENTIFIER ::= { rmon 9 }
```

O grupo statistics contém estatísticas sobre todas as interfaces Ethernet monitoradas pelo ponto de prova, cujas amostras periódicas são registradas e armazenadas pelo grupo history. O grupo alarm permite que o usuário defina um intervalo entre requisições e um limiar para cada objeto registrado pelo ponto de prova RMON. O grupo hosts registra estatísticas do tráfego de cada host da rede, que são usadas para gerar relatórios com os hosts ordenados por algum parâmetro deifinido na tabela de hosts acessados pelo grupo hostTopN. O grupo matrix armazena informações de erro e utilização entre conjuntos de dois endereços. O grupo filter define uma equação de filtro através do qual os pacotes são correlacionados de modo que seja capturado ou um evento seja gerado. O grupo capture permite essa captura e o event controla a definição desses eventos.

4. Softwares de Gerência

Os softwares para gerenciamento de rede englobam cinco categorias: agentes SNMP, NMS suites, gerenciadores de elementos (“element managers”, específicos para cada fabricante), analisadores de tendências e softwares de suporte.

Os agentes SNMP normalmente já vem implementados no próprio dispositivo não requerendo qualquer tipo de instalação, porém para alguns dispositivos devem ser instalados como pacotes adicionais. Alguns dos pacotes atualmente disponíveis são o HP Extensible SNMP Agent, o da Sun Microsystems, o da Microsoft, o Concord SystemEDGE e os gratuitos Net-SNMP e SNMP Research.

O termo suite para os softwares de NMS significa que o pacote engloba múltiplas aplicações em um único e conveniente produto, que determinam o preço e o grau de dificuldade de configuração e operação. As soluções para gerenciamento mais conhecidas são: Tivoli Netview, da IBM; Unicenter TNG, da Computer Associates; System Management Server (SMS) da Microsoft e HP OpenView, da HP.

Os gerenciadores de elementos são focados em um determinado produto ou fabricante. Exemplos são o Cisco Works 2000 e o 3Com Total Control.

Os softwares de análise de tendência são fundamentais para obter proatividade no diagnóstico de problemas antes que eles ocorram, na medida em que fornecem estatísticas que permitem determinar o que é o comportamento normal da rede. Pode ser integrado com os pontos de prova do monitoramento remoto (RMON) para melhores resultados. O mais usado é o MRTG (“Multi Router Traffic Grapher”), que é livre, fácil de configurar e usar e muito bem documentado.

Os softwares de suporte permitem escrever aplicações SNMP e rotinas que trabalham em conjunto com os softwares apresentados anteriormente. A linguagem de programação mais usada para este fim é o Perl “(Practical Extraction and Report Language”). Pacotes com sub-rotinas que acessam funções do núcleo SNMP, como o SNMP Support for Perl e o Net-SNMP Perl Module, facilitam o trabalho de programação.

A seguir, serão apresentados com mais detalhes os softwares utilizados no presente trabalho.

7.2 HP Open View Network Node Manager (NNM)

4..1. Visão Geral

Network Node Manager [5] é a base da maioria dos produtos de gerenciamento (HP Open View) da HP. Estes produtos podem ser encarados como funcionalidades adicionais atuando sobre dados coletados pelo NNM.

O NNM provê uma ferramenta integrada para controle e gerenciamento de múltiplos sistemas ligados em rede e aplicações a partir de uma única representação gráfica da rede.

No objetivo de um gerenciamento proativo, algumas funcionalidades do NNM podem ser vantajosas, dentre elas:

- Requisição de informação e correlação de eventos – permite descobrir o estado atual da rede, ou seja, os dispositivos presentes, como estão configurados, quais seus níveis de performance e operação e quais os problemas atuais da rede;
- Coleta de históricos da rede – permite identificar tendências e determinar como otimizar a rede em termos de alteração de configurações, substituição de dispositivos, etc;

- Monitoração de limiares em dispositivos de missão crítica – permite antecipar a ocorrência de uma falha e determinar como prevenir sua ocorrência.

O NNM continuamente solicita informações dos dispositivos quanto aos estados dos objetos gerenciados e busca alterações de topologias de rede, incluindo a descoberta de novos nós, e de configuração. De posse desses dados, podem ser configuradas ações a serem tomadas quando alguma situação específica ocorrer.

O NNM trabalha com o conceito de eventos classificados em categorias e associados a níveis de severidade. O usuário é ativamente notificado sobre a ocorrência de eventos, que são usados no diagnóstico de problemas. A funcionalidade de redução de eventos do NNM monitora os alarmes, identificando padrões de problemas de rede comuns e, a partir deste, envia um único e significativo alarme que representa todos os alarmes relacionados. Existe a possibilidade de customização dos eventos de modo a automatizar as tarefas de gerenciamento de falha pela tomada de ações automáticas em função do alarme específico recebido. Os eventos podem também ser filtrados ou apenas enviados a um arquivo de log.

O NNM usa os seguintes protocolos para manter a comunicação com os dispositivos gerenciados: SNMPv1 e SNMPv2, ICMP, IPX, UDP e ARP.

A quantidade de objetos gerenciados determina a quantidade de memória e espaço em disco requerida na estação de gerência. Objetos podem ser incluídos nos mapas, porém estando num estado não gerenciado, onde o NNM continua a receber traps e gerar alarmes para o dispositivo sem contudo lhe enviar requisições.

O software é capaz de realizar o autodescobrimento dos nós conectados ao mesmo segmento que ele. As informações obtidas na descoberta são armazenadas no banco de dados do NNM e permite gerar um mapa lógico que representa a conexão entre os elementos.

O processo que faz a descoberta dos elementos e o envio de requisições aos mesmos se chama “netmon” e é iniciado automaticamente quando o sistema inicia. Este processo descobrirá roteadores adjacentes ao seu segmento de rede (“gateways”) e suas redes conectadas (desde que eles sejam compatíveis com SNMP) e os colocará em um estado de não-gerenciado, modificado manualmente de modo a permitir a descoberta e a gerência dos dispositivos no interior das redes descobertas.

O “netmon” fornece duas funcionalidades para auxiliar no processo de descobrimento de elementos. A primeira é um arquivo de base (“seed file”), onde são fornecidos endereços IP individuais, ranges de IPs e nomes de domínio para estreitar o escopo de hosts que o NNM deve procurar. A outra funcionalidade é o comando “loadhosts” (carregar hosts) que permite que sejam adicionados hosts ao mapa individualmente.

4..2. Funcionalidades do NNM por Área de Gerenciamento

4..2.1. Gerenciamento de Falha

Dentro deste contexto, o NNM tenta identificar problemas e erros, reconhecer tendências e agir proativamente, através dessas características:

- Descoberta automática de dispositivos IP e IPX da rede;
- Monitoramento automático do estado da rede através do mapa e do navegador de eventos;

- Gerenciamento de dispositivos que suportem SNMP, monitorando objetos definidos tanto em MIBs padrões como em específicas de fabricantes, bastando para isso carregá-las no NNM, e de dispositivos que não suportem SNMP porém que usem os protocolos IP ou IPX;
- Construção de novas aplicações MIB padrões ou específicas;
- Definição de limiares em eventos para objetos MIB;
- Definição de ações a serem tomadas automaticamente em função da recepção de um evento específico;
- Estratégias de redução de eventos de modo a prevenir eventos indesejados e redundantes e adicionar informações mais significativas aos eventos;
- Integração com outras aplicações de detecção de falhas;
- Diagnóstico de falhas e problemas de performance, automatizando a resposta a eventos.

Outros produtos que podem ser adicionados para estender essas funcionalidades são o NNM Extended Topology, que descobre e apresenta informações adicionais de conectividade entre os elementos gerenciados, e o HP Open View Operations, que é uma solução de gerenciamento de problemas e eventos que permite identificar, localizar, correlacionar e resolver falhas de sistema e de rede.

4..2.2. Gerenciamento de Desempenho

O NNM coleta e cria relatórios a partir de informações de performance, disponibilidade, inventário e exceções, podendo enviá-las a outros produtos HP Open View para a análise estatística dos dados.

Dentre as funcionalidades existentes, estão:

- Monitoramento automático do estado da rede;

- Coleta de histórico, armazenamento de dados para análise de tendência e criação de gráficos a partir das informações coletadas;
- Definição automática de limiares baseados em desvios padrões dos dados históricos coletados;
- Geração e gerenciamento de relatórios, dentre eles o de Disponibilidade Geral e o Inventário Geral;
- Integração de outras aplicações de monitoramento de performance;
- Customização e automatização do monitoramento da rede e da resposta da NMS a eventos.

4..2.3. Gerenciamento de Configuração

O NNM mantém um registro dos dispositivos na rede, suas configurações e interconexões, de modo a criar um banco de dados das configurações, inventário e topologia da rede facilitando uma reconfiguração ou recuperação de desastres mais rápida e eficiente.

7.2 LorriotPro

O LorriotPro [6] é uma ferramenta de gerência baseada no protocolo SNMP para o monitoramento da rede. Além deste, também são usados os protocolos ICMP e http, este último para o monitoramento de servidores WEB, por exemplo.

A ferramenta é implementada no sistema operacional Windows e faz uso de interfaces gráficas e navegadores para mostrar o estado dos recursos de rede e de sua performance sob a forma de ícones codificados em cores e mensagens de alerta.

As formas de displays disponíveis são:

- Topologias em mapas geográficos ou diagramas de estado;
- Árvore hierárquica customizada pelo administrador;
- Filtros para os dispositivos de missão critica;
- Lista de eventos informando as mudanças de estado na rede.

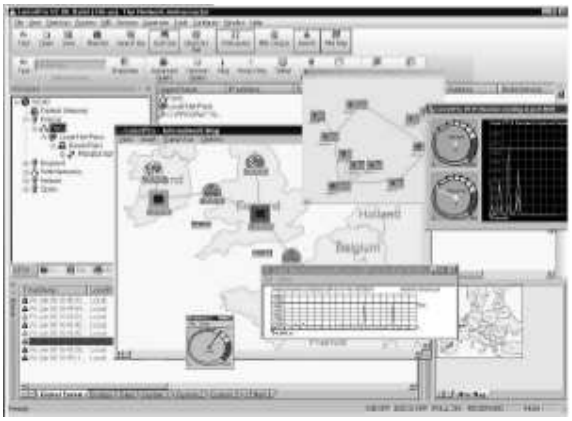


Figura 4.1: Interfaces gráficas do LorriotPro

Ele apresenta a possibilidade de se desenvolver scripts para automatizar tarefas repetitivas ou recuperar dados definidos por algum critério de uma grande quantidade de elementos de uma única vez.

A capacidade de organizar hierarquicamente os elementos gerenciados permite também que parte da tarefa de gerenciamento da rede seja delegada a terceiros sem a necessidade de revelar a totalidade da rede.

Todo o controle exercido sobre os dispositivos gerenciados é feito a partir da árvore hierárquica dos elementos e as alterações entram em funcionamento automaticamente.

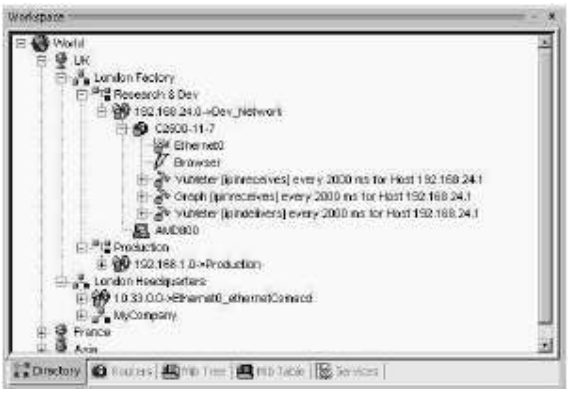


Figura 4.2: Estrutura hierárquica de organização dos elementos

O sistema de busca de um determinado dispositivo dentro da estrutura hierárquica é baseado em múltiplos critérios que podem ser de caráter técnico, como seu endereço IP, ou organizacional, como sua localização física.

Os roteadores são considerados de forma especial, sendo-lhes reservada uma árvore hierárquica dedicada, além de mais recursos de gerenciamento e extração de informações.

O processo de descobrimento dos elementos utiliza os protocolos SNMP e ICMP e pode ser disparado a partir de qualquer nó da rede, rodando em background e alertando a cada descoberta de um novo elemento.

A análise de tendências também é possível através do banco de dados integrado que armazena dados relacionados às coletas de informações, eventos recebidos, volume de tráfego e estados operacionais.

A tela de alarmes permite que sejam configurados filtros para quais alarmes devem ser mostrados ao administrador, além de ações mediante a chegada de algum evento, dentre os quais o envio de um e-mail, um alerta sonoro e a execução de um programa.

Figura 4.3: Tela de alarmes

O LorriotPro já vem com alguns modelos de relatórios pré-definidos, porém também permite que sejam criados de acordo com as necessidades do administrador de rede. Esses relatórios são feitos em formato HTML e acessíveis de qualquer navegador.

Outras funcionalidades são a criação de requisições sobre objetos MIB específicos, o acesso direto a esses objetos a partir da estrutura hierárquica em árvore das MIBs, o compilador de MIBs e um processo de definição de scripts para automatizar a requisição sobre objetos MIB.

O LorriotPro é modular e admite a expansão de suas funcionalidades através da adição de “plug-ins”. Dentre estes, estão o monitoramento de serviços WEB, o coletor de recursos NetBIOS, o agendador de envio de emails baseado em alarmes, o coletor de dados NetFlow da Cisco e o supervisor de portas TCP que estejam no estado “listening”. O código fonte dos “plug-ins” é fornecido também, de modo que possam ser adaptados às necessidades do administrador. Acompanha um passo a passo (“wizard”) em Visual C que facilita a criação da estrutura dos códigos das novas aplicações.

7.2 Trapgen

O trapgen é uma ferramenta de linha de comando que gera traps a partir dos parâmetros fornecidos como argumentos e os envia ao destino indicado. Ele entende apenas um subconjunto dos tipos de

dados dos objetos gerenciados definidos pela SMI, que compreende string, counter, gauge, integer, address, OID e timeticks.

Os parâmetros podem ser fornecidos diretamente da linha de comando um após o outro em seqüência ou a partir de um arquivo que os contenha. Podem ser associados parâmetros definidos no arquivo com parâmetros fornecidos na linha de comando. Os argumentos da linha de comando sobrescrevem os contidos no arquivo, com exceção das “variable bindings” que são cumulativas.

Alguns dos parâmetros são:

-d <endereço_IP_de_destino>:<porta>

-c <community>

-o <enterprise_ID>

-i <endereço_IP_do_agente>

-g <número_genérico_de_trap>

-s <número_específico_de_trap>

-t <timestamp>

-r <ID_da_requisição>

-m <timeout>

-v <OID> <TIPO> <valor>

O único parâmetro obrigatório é o endereço IP de destino dos traps. Usando-se apenas ele, é gerado um trap com as seguintes informações:

community: public

Endereço IP do agente: endereço IP do host a partir de onde o trap está sendo gerado

enterprise ID: 1.3.6.1.4.1.2854

Número Genérico de Trap: 6 (enterprise specific)

Número Específico de Trap: 1

Timestamp: hora atual do sistema

A sintaxe para que o trapgen utilize os argumentos contidos em um arquivo é:

trapgen –f <nome_do_arquivo>

5. Gerenciamento Distribuído

Redes de computadores são infra-estruturas indispensáveis em um ambiente de economia global, baseado na distribuição da carga de trabalho. Para o sucesso das corporações, é essencial que esses sistemas ofereçam segurança e rapidez. Com o desenvolvimento cada vez mais rápido da tecnologia e com o crescimento da Internet, as redes se tornam mais complexas a cada dia, exigindo serviços de comunicações confiáveis e novas formas para o seu gerenciamento. [7]

A função do gerenciamento de redes é garantir que as redes de computadores operem com eficiência e com os parâmetros pré-definidos de qualidade de serviço. As soluções para gerenciamento mais conhecidas são: Tivoli, da IBM; Unicenter TNG, da C.A; SMS da Microsoft e HP OpenView, da HP.

Em uma rede complexa, diferentes departamentos ou até mesmo aplicações possuem diferentes necessidades, sendo responsabilidade do elemento gerenciador da rede coletar informações que possibilitem ao administrador monitorar o comportamento da rede e garantir que essas necessidades sejam atendidas. Este monitoramento é feito através da troca constante de informações entre o elemento gerenciador e os elementos gerenciados.

7.2 Conceitos

As redes de computadores modernas adotam cada vez mais o conceito de processamento distribuído. Realizar uma tarefa de forma distribuída significa dividi-la em sub-tarefas que serão realizadas em sistemas diferentes, mas de maneira orientada e de forma cooperativa visando atingir o objetivo final. O processamento distribuído permite um grande grau de flexibilidade ao administrador da rede.

O conceito de processamento distribuído aplicado ao gerenciamento de redes permite um sistema de gerenciamento distribuído onde as unidades de gerenciamento são conectadas umas as outras através de um sistema de transporte, e interagem de forma a permitir diferentes níveis de gerenciamento.

Uma solução de gerenciamento distribuído possui as seguintes vantagens:

- Aumento da escalabilidade da rede. Redes de computadores são entidades dinâmicas cujas mudanças em suas estruturas ocorrem de acordo com as necessidades das organizações as quais elas servem. Assim como uma rede cresce e muda, o sistema de gerenciamento deve ser flexível a ponto de acompanhar de forma rápida e simples essas mudanças;

- O gerenciamento distribuído pode reduzir drasticamente o tráfico de gerenciamento, diminuindo congestionamento e evitando tráfego desnecessário através de links WAN, que são links relativamente lentos e caros. Sem o gerenciamento distribuído seria impossível gerenciar milhares de nós através de uma única estação de gerenciamento A distribuição minimiza a quantidade de tráfego de gerenciamento nesses links, liberando largura de banda para os propósitos da empresa.

- O gerenciamento distribuído torna possível compartilhar a responsabilidade pelo gerenciamento da rede e os recursos necessários através de múltiplos sites. Com isso, o risco de perder todo o gerenciamento da rede no caso de falha em uma máquina ou sistema é eliminado.

- Uma solução de gerenciamento escalável apropriadamente configurada resulta em uma maior velocidade de resposta da rede e melhor performance. Com isso se gasta menos tempo

esperando por uma resposta e mais atuando para resolver o problema. Além disso, também ocorre o aumento da confiabilidade do gerenciamento, permitindo gerenciamento local mesmo quando a conexão ao site central falhar.

5..1. Arquiteturas de Gerenciamento Distribuído

Existem diversas topologias de acordo com as necessidades de cada rede. As principais arquiteturas estão listadas abaixo:

G - Gerente
GG – Gerente dos Gerentes
RG – Recurso Gerenciado

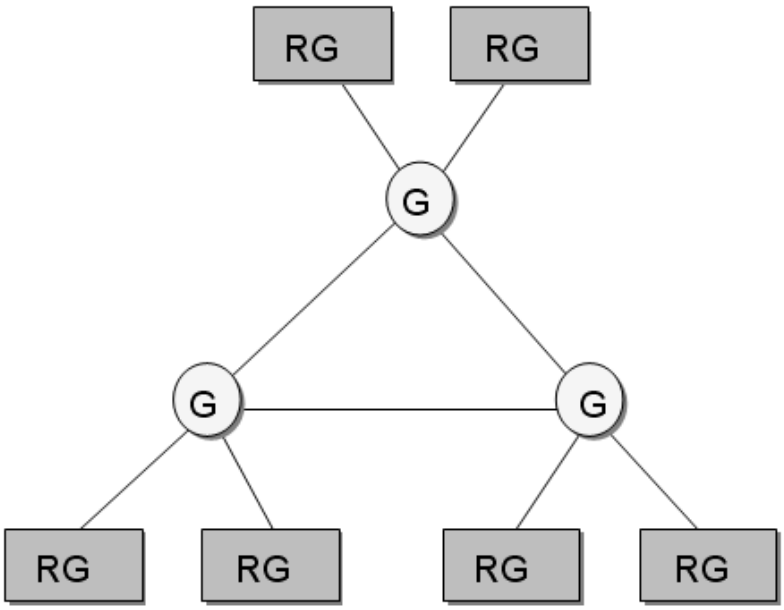


Figura 5.1: Controle Multicentralizado

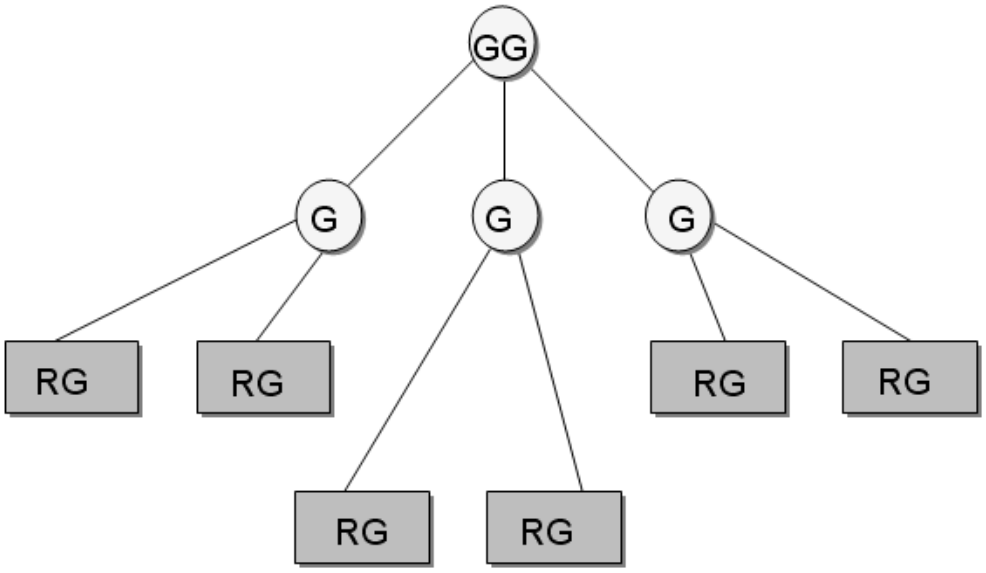


Figura 5.2: Gerenciamento Hierárquico

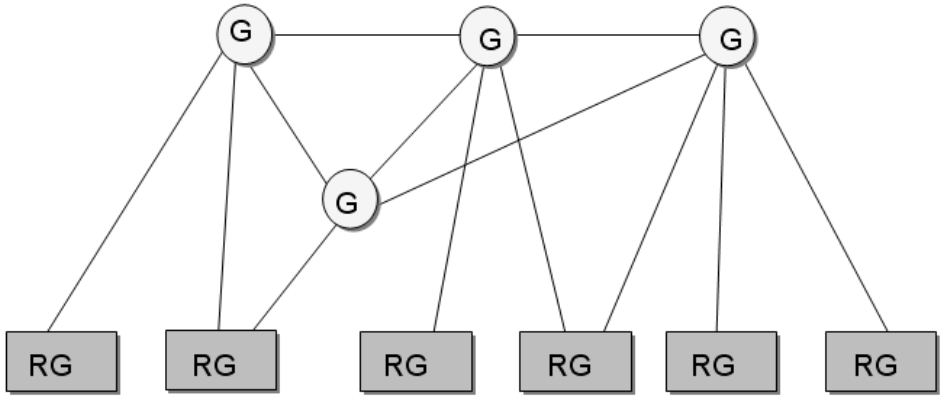


Figura 5.3: Redes de Gerentes

As figuras acima mostram algumas das diversas topologias e arranjos funcionais existentes para os sistemas de gerenciamento distribuído. Com o controle multiponto, os recursos gerenciados são combinados em grupos (domínios) de acordo com certos aspectos (topológicos, funcionais ou organizacionais) e um gerente é associado a cada grupo. Se se deseja coordenar os gerentes na topologia controle multiponto para que trabalhem de forma cooperativa, então a topologia lógica é alterada para controle multicentralizado ou gerenciamento hierárquico, dependendo do esquema de cooperação adotado. A topologia mais complexa é a rede de gerentes, onde um recurso pode ser

gerenciado por mais de um gerente. Por exemplo, um gerente responsável pelo gerenciamento de falhas e outro pelo gerenciamento de segurança.

6. Escopo do Projeto

Independentemente do tamanho de uma empresa ou instituição, cada vez mais se torna imprescindível a conexão desta à Internet, a filiais, a fornecedores, a clientes e a instituições governamentais. Em uma rede geograficamente distribuída as dificuldades se tornam ainda maiores, pois o uso de um único sistema de gerência se torna inviável devido à quantidade de informações que ele teria que processar e com a interrupção do gerenciamento a cada vez que o link com as redes remotas caísse. Com isso, torna-se necessária a utilização de uma topologia hierárquica, com cada site remoto possuindo um gerente local fornecendo ao administrador local todas as informações necessárias para o monitoramento de sua rede.

Algumas vezes também é necessário que um administrador sênior possua uma visão geral da saúde da rede inteira, monitorando os aspectos mais importantes. A integração de gerências visa à integração de sistemas heterogêneos complexos distribuídos de forma a possibilitar uma visão macro da saúde da rede como um todo.

No contexto acima, existem dois grandes desafios para a integração de gerências:

- Definição de uma forma para que somente as informações de gerenciamento realmente importantes sejam exportadas do gerente local para o gerente central, uma vez que a exportação de toda a base de dados dos gerentes locais consumiria uma grande quantidade de recursos da rede, sem contar o tempo e os recursos necessários para o processamento de toda essa informação pelo gerente central;
- Implementação da comunicação entre os gerentes locais e o gerente central, que podem ser de diferentes desenvolvedores e não possuir ferramentas que possibilitem esta integração.

Para enfrentar o primeiro desafio é necessário um conhecimento profundo sobre as topologias físicas e lógicas da rede, assim como das aplicações envolvidas. Com isso, é possível filtrar a base de dados e enviar ao gerente central apenas o que for realmente relevante. Para esta tarefa também pode ser usada a correlação de eventos. A correlação, como o nome sugere, estuda a relação entre os eventos. No SNMP, os eventos podem ser classificados de acordo com o seu grau de criticidade.[4] Com isso, uma estratégia pode ser que somente eventos graves devam ser retransmitidos ao gerente central. O uso da correlação [8] pode permitir que um evento grave possa ser descoberto indiretamente, associando a chegada de dois eventos menos críticos específicos a um evento crítico, que deve ser informado ao gerente central. A forma de implementação da correlação foge do escopo deste trabalho.

7.2 Cenário

A topologia de gerenciamento distribuído adotada para o trabalho é a de gerenciamento hierárquico, explicada em detalhes na seção Arquiteturas de Gerenciamento do capítulo 5.

A figura abaixo ilustra o cenário adotado:

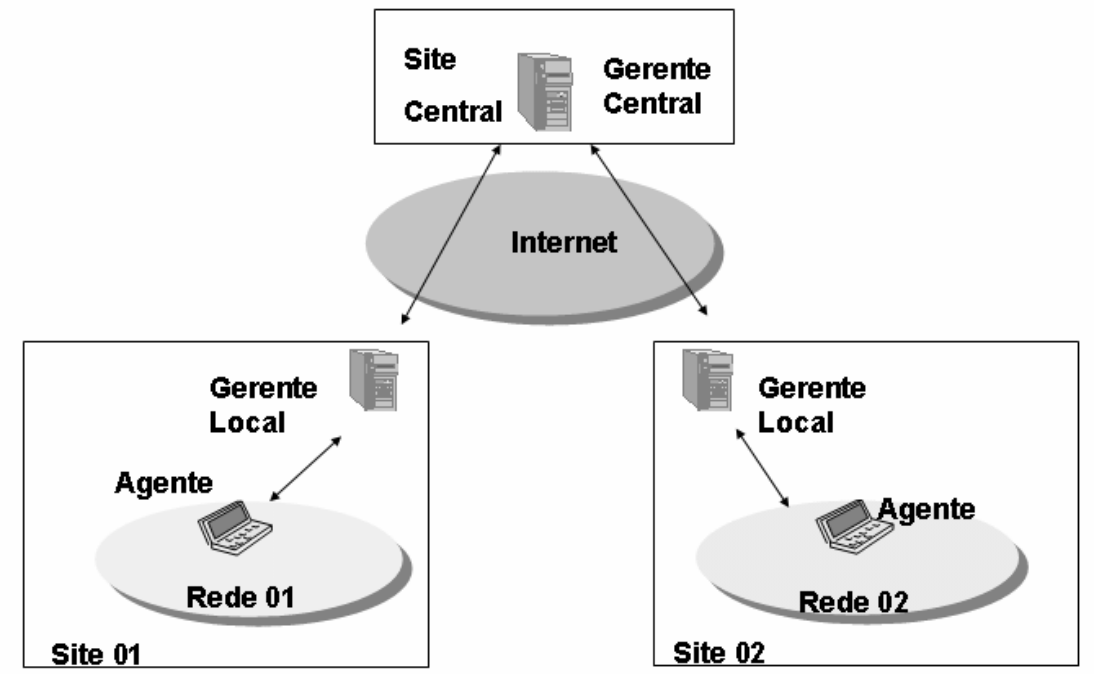


Figura 6.1: Arquitetura

7.2 Implementação

A integração entre os gerentes é realizada através de duas rotinas denominadas interceptador de eventos e sincronizador de banco de dados, respectivamente.

O interceptador de eventos é uma rotina implementada em cada gerente local, que intercepta os traps recebidos e os copia em um novo arquivo que servirá como uma base de dados independente, que posteriormente será tratada de acordo com as políticas adotadas e exportada para o gerente central. A aplicação de gerência local recebe simultaneamente os pacotes capturados pelo interceptador, já que ambos “escutam” a mesma porta, 162, padrão para a recepção de traps no protocolo SNMP, permitindo também o gerenciamento local de todos os eventos.

A rotina é composta de uma ferramenta de captura de pacotes que repassa essas informações a uma linguagem de programação de processamento de texto de modo a formar os campos da base de dados independente. A princípio, foi utilizado, como ferramenta de captura de pacotes, o tcpdump

[9], que é capaz de decodificar os pacotes até a camada de aplicação. O awk [10] foi a linguagem de programação de processamento de texto escolhida. Ele recebe em tempo real os pacotes capturados pelo tcpdump, extraindo as informações relevantes e formatando-as de modo a estarem prontas para posterior tratamento.

O sincronizador de banco de dados garante o sincronismo entre as bases de dados criadas pelo inteceptador de eventos e a base de dados do gerente central. É uma rotina disparada pela Crontab [11] da plataforma de gerência local, que verifica, em períodos definidos pelo administrador, o tamanho da base de dados criada pelo interceptador de eventos e, caso o tamanho do arquivo tenha sido incrementado, indicando que novas informações foram inseridas, ele trata essas novas informações e, caso coincidam com os critérios pré-definidos pelo administrador, a rotina exporta esses novos dados para o gerente central sob a forma de envio de traps.

A implementação dessa rotina baseia-se na manutenção do estado da última vez que a rotina foi executada em um arquivo texto, que é consultado no início de cada execução. A informação guardada é o número de linhas que a base de dados continha então. Dessa forma, toda a base de dados é mantida, permitindo posteriores sincronismos em caso de falha de algum elemento do processo. Ainda assim, evita-se que o sincronizador reenvie informações já processadas.

A partir de um critério definido, os eventos considerados importantes são filtrados e seus campos de informação vão constituir os argumentos da função geradora de traps, responsável pela exportação dos dados ao gerente central. Para essa função, foi usado o programa trapgen, de fácil instalação e disponível para a maioria dos sistemas operacionais.

Nessa rotina, optamos por uma simplificação devido aos parâmetros requeridos pela função trapgen. Ela espera receber as “variable bindings” que complementam as informações dos traps no formato

“OID tipo valor”, porém o tipo de dados de uma OID é extraído da MIB onde ela foi definida. De modo a simplificar o script para que não fosse necessária a inclusão de um código para consulta das MIBs (até mesmo pela existência de diversas MIBs privadas no ambiente de gerenciamento de uma rede), adotamos as seguintes premissas de acordo com a observação da freqüência de recepção de cada tipo de dado:

- Valores contidos entre aspas foram considerados do tipo STRING;
- Valores que continham pontos foram considerados do tipo ADDRESS. O outro tipo de dado que também contém pontos é o OID, porém raras vezes é utilizado em traps;
- Os demais valores foram considerados do tipo COUNTER. Estes valores são números e podem ser de um desses tipos: INTEGER, COUNTER ou GAUGE, porém o tipo INTEGER é usado tipicamente para definir tipos enumerados, não tão freqüentes de acordo com as observações, e não foi possível fazer uma diferenciação apenas visual sem consultas a MIBs entre os tipos COUNTER e GAUGE, optando-se pelo primeiro tipo.

A figura abaixo ilustra a implementação das rotinas:

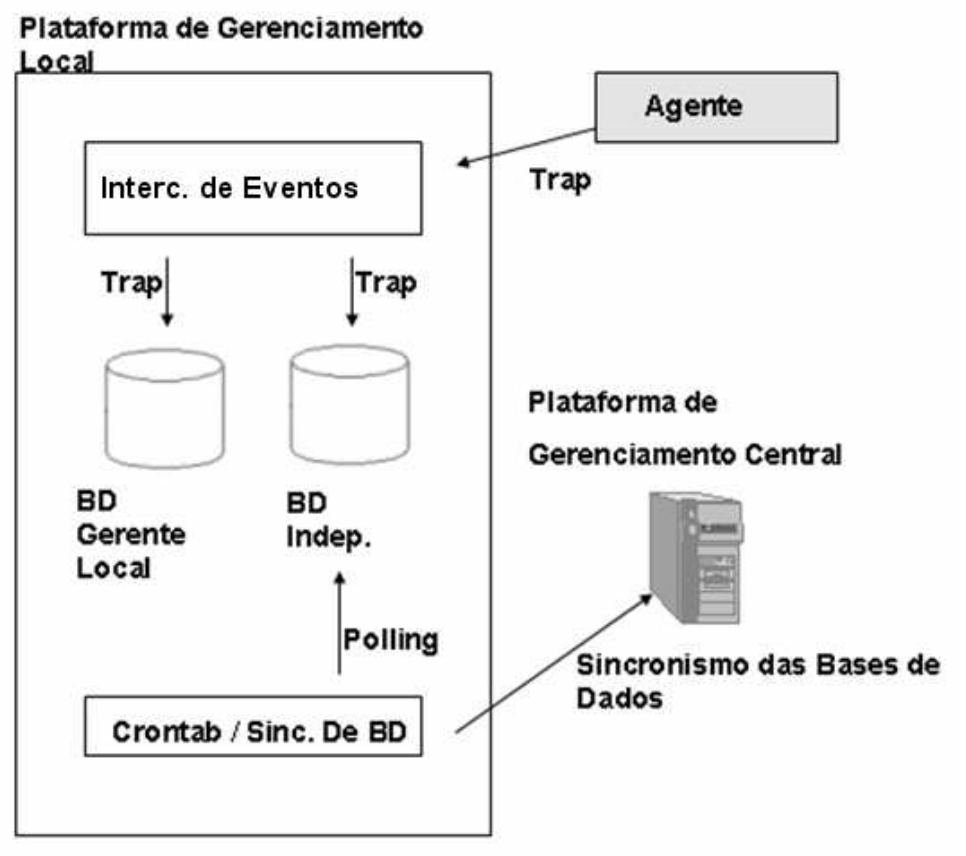


Figura 6.2: Implementação de Rotinas

Seguem os scripts criados em shell, implementados e testados no sistema operacional Linux.

Interceptador

intercepta.sh */* roda em background no gerente local */*

```
#!/bin/sh

/* O tcpdump captura os pacotes até os seus 1000 bytes (suficiente pra obter todas as informações contidas
nos traps) filtrando pela porta 162. Os pacotes alimentam, em tempo real, a linguagem de processamento de
texto awk para formar a base de dados banco.txt */

/usr/sbin/tcpdump -s 1000 port 162 | awk -f ./formata.awk > ./banco.txt
```

formata.awk */* Arquivo com o código de execução do awk. */*

```
{
 /* retira os [] do campo 9 que contém o endereço IP do agente que enviou o trap */
 split ($9, temp, "[");
 split(temp[2], agente, "]");
```

```
/* separa o campo 10 de modo a extrair o número de trap genérico e o número de trap específico */
 split ($10, num_trap, "[");
 split (num_trap[2], spec_trap, "(");
 num_spec = substr (spec_trap[2], 1, 1);

/* transforma a descrição de trap genérico no número de trap genérico correspondente */
if(num_trap[1] == "coldStart") {num_gen = 0;}
else if(num_trap[1] == "warmStart") {num_gen = 1;}
else if(num_trap[1] == "linkDown") {num_gen = 2;}
else if (num_trap[1] == "linkUp") {num_gen = 3;}
else if(num_trap[1] == "authenticationFailure") {num_gen = 4;}
else if(num_trap[1] == "egpNeighborLoss") {num_gen = 5;}
else if(num_trap[1] == "enterpriseSpecific") {num_gen = 6;}

/* imprime as informações relevantes (a partir do campo 8 e com as formatações realizadas) no arquivo de
base de dados */
printf ("%s\t%s\t%s\t%s\t", $8,agente[1] , num_gen, num_spec);

/* formata os campos que representam as "variable bindings" que complementam as informações do trap de
modo que as strings formem um único campo com cada palavra unida por um "_" */
inicio = 0;
for (i = 12; i <= NF; i++) {
 if (($i ~/\"/) && (inicio==0)) {inicio = 1;}
 if ($i ~/\"$/) {inicio = 0;}
 if (inicio==1) {printf("%s_", $i);}
 else {printf("%s\t", $i)}};

/* separa cada pacote recebido em uma linha da base de dados */
printf("\n");}
```

Sincronizador

filtro.sh */* disparado pela Crontab do gerente local em intervalos definidos pelo administrador de rede */*

```
#!/bin/sh

/* verifica se o arquivo que contém a informação do número de linhas da base de dados na última execução
da rotina existe ou se essa é a primeira vez que a rotina é invocada */
if [ -e ./last.txt ]
then
        ultima=`cat ./last.txt`
else
        ultima="0"
fi

/* obtém o número de linhas atual da base de dados */
linhas=`wc -l ./banco.txt |cut -c1-8`

/* verifica se a base de dados foi incrementada sendo necessário processamento dos novos dados */

if (test $linhas -gt $ultima)
then
```

```
/* seleciona a parte da base de dados que ainda não foi processada, filtra de acordo com o critério estabelecido pelo administrador e alimenta o código de execução da linguagem de processamento de texto awk para o envio dos traps com as informações escolhidas */

        temp=`expr $ultima + 1`
        tail +$temp < /home/livia/banco.txt | grep <criterio> | awk -f ./envia_trap.awk

/* armazena o número de linhas atual da base de dados para consulta na próxima execução da rotina */
        echo $linhas > ./last.txt
fi

exit 0
```

envia_trap.awk /* *Arquivo com o código de execução do awk para o envio de traps* */

```
/* monta o arquivo com os argumentos da função trapgen, geradora de traps */
{

/* coloca em cada linha do arquivo os argumentos: enterprise ID, endereço IP do agente, número de trap genérico e número de trap específico, respectivamente */

 printf ("-o %s\n-i %s\n-g %s\n-s %s\n", $1, $2, $3, $4) > "./trapdir/input_trapgen.txt";


/* formata as "variable bindings" que complementam as informações do trap de acordo com o esperado pela função trapgen */

for (i=5; i<=NF; i++){

/* extrai o número da OID e seu valor que na base de dados se apresentam no formato "OID=valor" */

        split($i, oid, "=");

/* preenche o argumento de tipo de dado da OID requerido pela função trapgen, considerando STRING todos os valores que contiverem '"', ADDRESS todos aqueles que contiverem '.' e COUNTER, os demais valores */

        if(oid[2] ~/\"/) {type = "STRING";}
        else if (oid[2] ~/\./) {
                type = "ADDRESS";
                split(oid[2],temp, "[");
                split(temp[2],tmp,"]");
                oid[2]=tmp[1];}
        else {type = "COUNTER";}

/* armazena em cada linha subseqüente do arquivo os argumentos que correspondem às OIDs no formato -v OID tipo valor */

        printf ("-v %s %s %s", oid[1], type, oid[2]) >> "./trapdir/input_trapgen.txt";
        if(i != NF) {printf("\n") >> "./trapdir/input_trapgen.txt";}
};

/* fecha o arquivo de modo que ele possa ser lido desde o seu início */
```

close ("./trapdir/input_trapgen.txt");

/ faz uma chamada ao sistema para que ele invoque a função trapgen com os parâmetros definidos no arquivo enviando os traps para o endereço IP e porta definidos na linha de comando */*

system("./trapdir/trapgen -f ./trapdir/input_trapgen.txt -d <IP>:162");
}

Como resultados dos scripts, apresentamos uma amostra da base de dados formada pelo interceptador e o arquivo que armazena os parâmetros requeridos pela função trapgen em um dado momento de execução para um pacote específico.

banco.txt */* as colunas são separadas por tabulação e cada pacote forma uma nova linha do arquivo */*

.1.3.6.1.4.1.4922.130.1.2 10.35.14.19 6 9
.1.3.6.1.4.1.4922.130.2.1.1.1.4.0="TM14DP01"
.1.3.6.1.4.1.4922.130.2.1.1.1.6.0=[10.35.14.19]
.1.3.6.1.4.1.4922.130.2.1.1.1.7.0=[0.0.0.0] .1.3.6.1.4.1.4922.130.2.1.1.1.8.0=[0.0.0.0]
.1.3.6.1.4.1.4922.130.2.1.1.1.9.0=[0.0.0.0] .1.3.6.1.4.1.4922.130.2.1.1.1.12.0="2005-05-31,10:39:22,GRNLNDST" .1.3.6.1.4.1.4922.130.2.1.1.1.13.0=10
.1.3.6.1.4.1.4922.130.2.1.1.1.14.0=1177
.1.3.6.1.4.1.4922.130.1.2 10.36.14.51 6 9
.1.3.6.1.4.1.4922.130.2.1.1.1.4.0="SCP02BOLAND"
.1.3.6.1.4.1.4922.130.2.1.1.1.6.0=[10.36.14.51]
.1.3.6.1.4.1.4922.130.2.1.1.1.7.0=[10.36.15.51]
.1.3.6.1.4.1.4922.130.2.1.1.1.8.0=[10.36.14.52]
.1.3.6.1.4.1.4922.130.2.1.1.1.9.0=[10.36.15.52]
.1.3.6.1.4.1.4922.130.2.1.1.1.12.0="2005-05-31,10:39:23,GRNLNDST"
.1.3.6.1.4.1.4922.130.2.1.1.1.13.0=10 .1.3.6.1.4.1.4922.130.2.1.1.1.14.0=10644
.1.3.6.1.4.1.4922.130.1.2 10.35.14.40 6 3
.1.3.6.1.4.1.4922.130.2.1.1.1.1.0=282855 .1.3.6.1.4.1.4922.130.2.1.1.1.2.0=-1
.1.3.6.1.4.1.4922.130.2.1.1.1.3.0=1117546763
.1.3.6.1.4.1.4922.130.2.1.1.1.4.0="TM08SV01"
.1.3.6.1.4.1.4922.130.2.1.1.1.6.0=[10.35.14.40]
.1.3.6.1.4.1.4922.130.2.1.1.1.7.0=[10.35.15.40]
.1.3.6.1.4.1.4922.130.2.1.1.1.8.0=[0.0.0.0] .1.3.6.1.4.1.4922.130.2.1.1.1.9.0=[0.0.0.0]
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.1.0="SINGLEAPP"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.2.0="TM08SV01:tm08sv01:815456"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.3.0=7 .1.3.6.1.4.1.4922.130.2.1.1.3.2.1.4.0="2005-05-31,10:39:23,GRNLNDST"
.1.3.6.1.4.1.4922.130.1.2 10.35.14.40 6 1
.1.3.6.1.4.1.4922.130.2.1.1.1.1.0=282856 .1.3.6.1.4.1.4922.130.2.1.1.1.2.0=-1
.1.3.6.1.4.1.4922.130.2.1.1.1.3.0=1117546764
.1.3.6.1.4.1.4922.130.2.1.1.1.4.0="TM08SV01"
.1.3.6.1.4.1.4922.130.2.1.1.1.6.0=[10.35.14.40]
.1.3.6.1.4.1.4922.130.2.1.1.1.7.0=[10.35.15.40]
.1.3.6.1.4.1.4922.130.2.1.1.1.8.0=[0.0.0.0] .1.3.6.1.4.1.4922.130.2.1.1.1.9.0=[0.0.0.0]
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.1.0="SINGLEAPP"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.2.0="TM08SV01:tm08sv01:815560"

.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.3.0=3 .1.3.6.1.4.1.4922.130.2.1.1.3.2.1.4.0="2005-05-31,10:39:23,GRNLNDST" .1.3.6.1.4.1.4922.130.2.1.1.3.2.1.5.0="tm08sv01"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.6.0=710700
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.7.0="isa" .1.3.6.1.4.1.4922.130.2.1.1.3.2.1.8.0="isa_main"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.9.0=1
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.10.0="Communications"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.11.0="communications_sub-system_failure"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.12.0="All_connections_to_RIM_have_disconnected"
.1.3.6.1.4.1.4922.130.2.1.1.3.2.1.13.0="All_connections_to_RIM_have_disconnected"

input_trapgen.txt */* parâmetros do último pacote enviado nos testes realizados */*

```
-o .1.3.6.1.4.1.4922.130.1.2
-i 10.35.14.8
-g 6
-s 9
-v .1.3.6.1.4.1.4922.130.2.1.1.1.4.0 STRING "TM03SP01"
-v .1.3.6.1.4.1.4922.130.2.1.1.1.6.0 ADDRESS 10.35.14.8
-v .1.3.6.1.4.1.4922.130.2.1.1.1.7.0 ADDRESS 0.0.0.0
-v .1.3.6.1.4.1.4922.130.2.1.1.1.8.0 ADDRESS 0.0.0.0
-v .1.3.6.1.4.1.4922.130.2.1.1.1.9.0 ADDRESS 0.0.0.0
-v .1.3.6.1.4.1.4922.130.2.1.1.1.12.0 STRING "2005-05-31,10:39:32,GRNLNDST"
-v .1.3.6.1.4.1.4922.130.2.1.1.1.13.0 COUNTER 10
-v .1.3.6.1.4.1.4922.130.2.1.1.1.14.0 COUNTER 1390
```

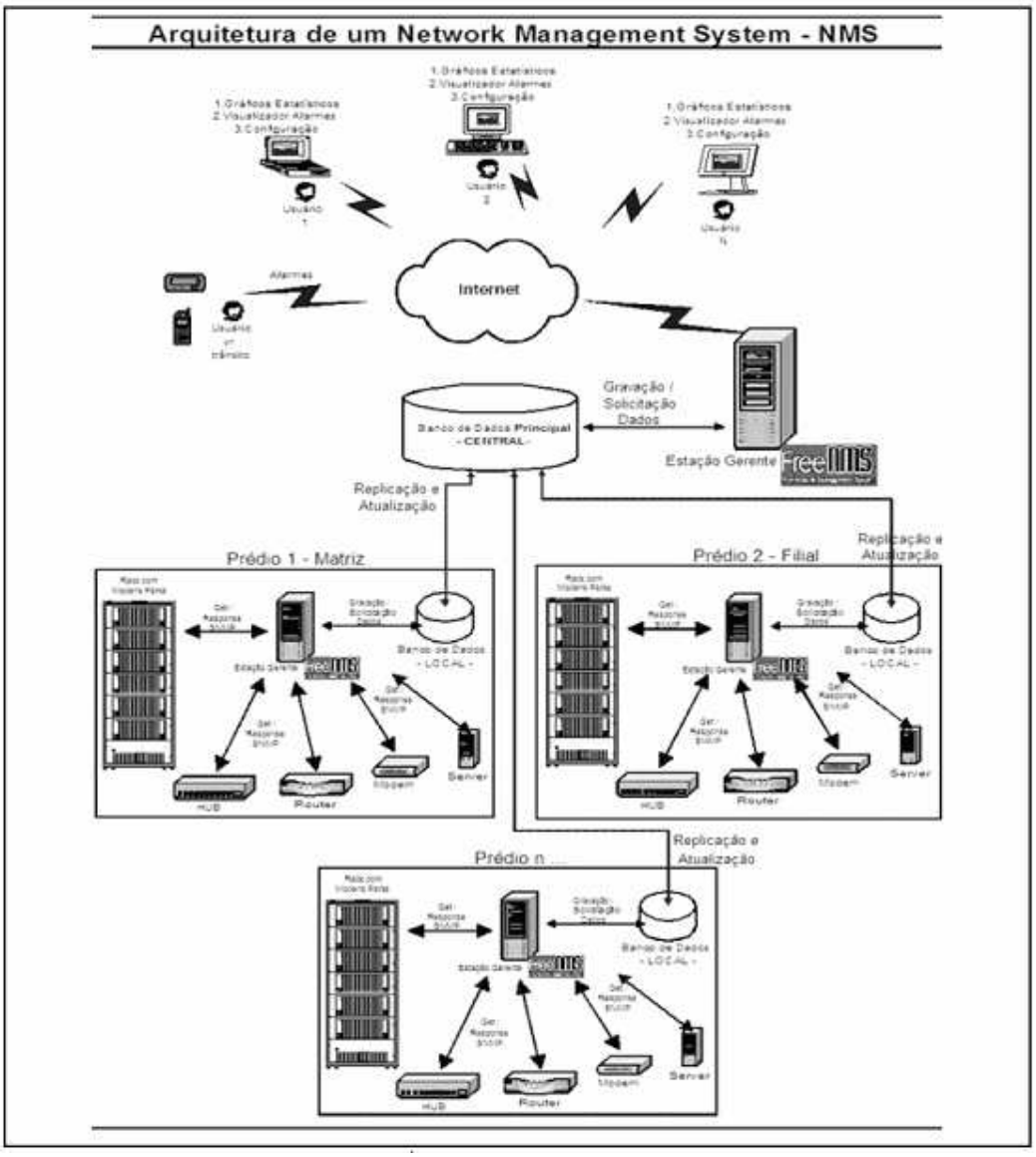
7. Trabalhos Relacionados

7.1 FreeNMS

O FreeNMS é um projeto realizado pelo Departamento de Pos-Graduação da Universidade Pontificia Universidade Catolica do Rio Grande do Sul - PUCRS (PUCMGMT – Faculdade de Engenharia Elétrica, Laboratório METROPOA, Grupo de Gerência de Redes).

O FreeNMS (Free Network Management System) é uma Plataforma de Gerência de Redes. Possui entre suas funcionalidades, características de gerência de Falhas, Performance, Contabilização, Configuração e Segurança. Como diferencial, possui além das características tradicionais e esperadas de uma Plataforma de Gerência de Redes, funcionalidade de acompanhamento de QoS contratados (SLA – Service Level Agreement): configura-se uma qualidade de serviço esperada, e o sistema monitora a qualidade deste serviço ininterruptamente, gerando alarmes quando SLAs são “quebrados”. Desta forma, o FreeNMS realiza Gerência de Nível de Serviço – SLM (Service Level Agreement).

No FreeNMS existe o módulo de gerenciamento distribuído, com gerentes locais enviando informações para o gerente central, conforme ilustrado abaixo:



7.1.1 Relação entre FreeNMS e Nosso Projeto

No nosso projeto é gerado uma base de dados independente que será tratada (filtros e correlação de eventos) e o resultado será enviado para o gerente central.

Na abordagem do FreeNMS toda a base de dados dos gerentes locais é exportada para o gerente central, desperdiçando recursos das máquinas e da rede.

Link: http://200.132.73.81/mkt/index.php

7.2 Trap Fowarder

Utilitário de linha de comando que recebe traps na porta 162 e as replica, enviando-os para múltiplos destinatários.

7.2.1 Relação entre Trap Fowarder e Nosso Projeto

Com o Trap Fowarder é necessário colocar mais uma máquina, que servirá de proxy para os gerentes. Esta máquina receberá os traps e as replicará, enviando-as para o gerente local e para o gerente central.

Novamente, nenhum tratamento é realizado nas informações enviadas ao gerente central, desperdiçando recursos.

Link: http://www.ncomtech.com/download.htm

Conclusão

Uma das grandes vantagens da solução apresentada é o fato do evento ser interceptado antes de ser tratado pelo gerente local. Isto torna o processo de integração transparente aos gerentes locais.

Uma outra abordagem seria analisar a própria base de dados dos gerentes locais a procura dos eventos mais importantes que seriam exportados para o gerente central. O ponto fraco desta abordagem é que para cada diferente solução (HP, IBM, etc...), seria necessário o estudo de como os eventos são armazenados na base de dados em cada implementação, o que aumentaria muito o trabalho e tornaria necessário o uso de rotinas específicas para cada solução adotada.

Referências Bibliográficas

[1] T. Saydam e T. Magedanz, “From Networks and Network Management into Service and Service Management”, *Journal of Networks and System Management*, vol. 4, n. 4, dez/1996, p. 345-348.

[2] James F. Kurose e Keith W. Ross, “Redes de Computadores e a Internet: Uma Nova Abordagem”, Addison Wesley, 2003.

[4] Douglas R. Mauro e Kevin J. Schmidt, “Essential SNMP”, O’Reilly, 2001.

[5] “Managing your Network with HP Open View Network Node Manager”, Hewlett-Packard Development Company, 2003.

[6] Lorriot Pro, “LorriotPro”, Luteus Sarl, 2004

[7] Hegering, Abeck e Neumair, “Integrated Management of Networked Systems”, Morgan Kaufmann Publishers, 1999

[8] Silvia Emiko Shimakura, http://www.est.ufpr.br/~silvia/CE003/node71.html

[9] Van Jacobson, Craig Leres e Steven McCanne, “Unix man pages: tcpdump (1)”, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 1997.

[10] Greg Goebel, “An Awk Tutorial”, public domain, out/2004, http://www.vectorsite.net/tsawk.html.

[11] Andrew M. Ross, “Getting Started with Awk”, HMC Computer Science Department, 2001, http://www.cs.hmc.edu/qref/awk.html.

[12] Sakari Mattila, “Guide to Awk”, University of Canberra, 2002, http://www.canberra.edu.au/~sam/whp/awk-guide.html.

[13] Gleydson Mazioli da Silva, “Guia Foca GNU/Linux”, Capítulo 26 – Manutenção do Sistema, 2003, http://focalinux.cipsga.org.br/guia/intermediario/ch-manut.htm.