Universidade Federal de Santa Catarina - UFSC
Curso de Pós-Graduação em Ciência da Computação

ANÁLISE DA SOBRECARGA DO PROTOCOLO SNMPv3 NO GERENCIAMENTO DE REDES CORPORATIVAS

por

Vilson Montagna

Dissertação submetida à Universidade Federal de Santa Catarina para a obtenção do grau de mestre em Ciência da Computação

Prof. Alexandre Moraes Ramos
Orientador

Florianópolis, Fevereiro de 2003.

ANÁLISE DA SOBRECARGA DO PROTOCOLO SNMPv3 NO GERENCIAMENTO DE REDES CORPORATIVAS

Vilson Montagna

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação na Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Fernando Álvaro Ostuni Gauthier, Dr.
Coordenador do Curso

BANCA EXAMINADORA:

_____

Prof. Alexandre Moraes Ramos, Dr.
Orientador

_____

Profª. Elizabeth Sueli Specialski, Dra.

_____

Prof. Vitório Bruno Mazzola, Dr.

"Tanto a ciência como a religião têm poder.
Mas, a ciência sem religião é manca e a religião sem ciência é cega."
Albert Einsten

Este trabalho é dedicado aos meus pais e irmãos que sempre deram-me apoio e incentivo permitindo assim a sua realização.

AGRADECIMENTOS

Ao meu orientador, professor Alexandre Moraes Ramos, pelo conhecimento técnico e entusiasmo transmitidos para a elaboração deste trabalho.

Ao amigo e companheiro Luciano Nakano, pelo apoio, pela amizade, pelo incentivo e pelas horas de estudo e idéias trocadas durante esses dois anos, assim como pelo apoio nos momentos difíceis, bem como, a todos os meus amigos, que estiveram juntos nessa jornada e fizeram as minhas viagens e estada em Cascavel agradáveis.

A meus irmãos, Vilmar L. Montagna, Valdor A. Montagna, Ivaneide Montagna da Silva, Valmor Antônio Montagna, Ivone Montagna Colombelli e Valmor Domingos Montagna, pelo apoio e amizade.

A minha namorada, Carmen Ramirez Alvarenga (That´s odd ball), por ter estado comigo nos bons e maus momentos.

A meus pais, Sestivo A. Montagna e Lúcia Arpini Montagna, por terem me dado a vida, educação e principalmente amor.

À Universidade Federal de Santa Catarina e à CAPES pelos recursos disponíveis para a realização deste trabalho.

A Deus, por nunca ter me desamparado nos momentos mais difíceis.

RESUMO

Este trabalho propõe uma análise da sobrecarga (*overhead*) no desempenho de redes corporativas, quando da utilização do protocolo SNMPv3 na gerência de redes, pois as redes atuais precisam, não somente serem gerenciadas em sua infra-estrutura física, mas também, nas aplicações e protocolos que são executados sobre elas, de uma forma integrada e segura, possibilitando com isso uma monitoração contínua e uma análise de tendências e desempenho. Para tal, é apresentada uma análise da sobrecarga do protocolo SNMPv3, o qual implementa elementos de segurança – autenticação, criptografia e controle de acesso, que não haviam sido definidos nas versões anteriores (SNMPv1 e SNMPv2c), o que o torna mais robusto, constituindo assim, a base para a segurança no gerenciamento de redes de computadores.

ABSTRACT

This work proposes an analysis of the overload (overhead) in the performance of the corporate networks, when of the use of the protocol SNMPv3 in the management of networks, because, the current networks need, not only they be managed in physical infrastructure, but also, of the applications and protocols that they are executed on them, in an integrated way and security, making possible with that a continuous monitoring and an analysis of tendencies and acting. For such, an analysis of the overload of the protocol is presented SNMPv3, which implements elements of security - authentication, cryptography and access control, that had not been defined in the previous versions (SNMPv1 and SNMPv2c), what turns it more robust, constituting like this, the base for the security in the management of networks of the computers.

SUMÁRIO

LISTA DE ABREVIATURAS

API	Application Program Interface
ASCII	American National Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
CBC	Cipher Block Chaining
CCITT	Consultative Commitee on International Telephony and Telegraphy
CMIP	Common Management Information Protocol
CRC	Cyclic Redundancy Code
DES	Data Encryption Standard
DOD	Department of Defense of the United States of America
DNS	Domain Name System
FTP	File Transfer Protocol
HMAC-MD5	Hashing Message Authentication – Message Digest-5
HMAC-SHA	Hashing Message Authentication – Secure Hash Algorithm
HTTP	Hypertext Trasnfer Protocol
IAB	Internet Activities Board
IETF	Inernet Engineering Task Force
ISO	International Organization Standardization
MAC	Media Control Layer
MIB	Management Information Base
MSM	Message Stream Modification
NMS	Network Management System
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RFC	Request for Comments
RMON	Remote Monitoring
RMON1	Remote Monitoring Version 1
RMON2	Remote Monitoring Version 2
SMI	Structure Management Information
SNMP	Simple Network Management Protocol

SNMPv1	Simple Network Management Protocol Version 1
SNMPv2	Simple Network Management Protocol Version 2
SNMPv3	Simple Network Management Protocol Version 3
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol / Inernet Protocol
TLS	Transport Layer Security
TMN	Telecommunication Management Networks
UDP	User Datagram Protocol
USM	User-Based Security Model
VACM	View-Based Access Control
XML	Extended Markup Language

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO I

1. INTRODUÇÃO

A aplicação da gerência de redes de computadores é muito importante no atual contexto das redes de computadores, pois surgem constantemente novos dispositivos e novas aplicações distribuídas que tornam a rede mais complexa. Devido à diversidade desses recursos, torna-se necessária à busca pela eficiência e eficácia na gerência de redes.

À medida que a tecnologia de redes evolui é necessária à busca por soluções efetivas de gerenciamento que permitam reduzir custos e prover altas disponibilidades, desempenho e segurança dessas redes, garantindo a qualidade dos serviços e a satisfação dos usuários.

O modelo de gerenciamento OSI – *Open Systems Interconnection* da ISO - *International Organization for Standardization* subdividiu a gerência de redes em cinco grandes áreas funcionais: Gerência de Falhas, de Configuração, de Desempenho, de Segurança e de Contabilização (KLERER, 1988). Essas funções têm sido comumente aplicadas para desenvolver uma gerência reativa de redes, ou seja, estão sendo utilizadas na detecção de problemas na rede e na busca de uma solução quando esses problemas ocorrem (ARTOLA, 1996).

Como foi mencionado anteriormente, as tecnologias de redes estão evoluindo a cada dia e o seu uso está-se expandindo em grandes proporções. Nesse sentido, a segurança do gerenciamento de redes é um fator crítico. É necessário adicionar segurança nos sistemas de gerência de redes para que eles próprios sejam seguros.

O protocolo de gerenciamento SNMPv3 implementa elementos de segurança - autenticação, criptografia e controle de acesso que não haviam sido definidos nas versões anteriores (SNMPv1 e SNMPv2c), o que o torna mais robusto, constituindo assim a base para a segurança no gerenciamento de redes de computadores (STALLINGS, 1998b).

Para implementar as novas características de segurança, o SNMPv3 requer um novo formato de mensagem (para maiores detalhes, ver apêndice B). O novo formato de mensagem é consideravelmente mais longo que o formato das mensagens de SNMPv1 e SNMPv2c. Por isso, o objetivo deste trabalho é apresentar uma análise da sobrecarga (*overhead*), da utilização do protocolo SNMPv3, no gerenciamento de redes corporativas. No contexto deste trabalho, considerou-se como sobrecarga, o adicional de *octetos*, acrescido a cada mensagem nas diferentes versões do protocolo SNMP.

1.1 Objetivos

O objetivo geral deste trabalho é realizar uma análise da sobrecarga no gerenciamento de redes corporativas, quando da utilização do protocolo SNMPv3 na gerência de redes.

Tem-se como objetivos específicos:

- Analisar, por meio de simulações, se a sobrecarga do protocolo SNMPv3 é excessiva ou não em aplicações de gerenciamento de rede baseadas em SNMP;

- Comparar, por meio de simulações, a sobrecarga do SNMPv3, com e sem os serviços de segurança oferecidos pelo protocolo em relação ao SNMPv1 e SNMPv2, que não contemplavam tais serviços;

- Realizar simulações de rede para projetar como operações SNMPv1, SNMPv2 e SNMPv3 (com e sem os serviços de segurança) afetam no desempenho da gerência de redes de computadores baseadas em SNMP;

- Analisar, por meio de simulações, quanta capacidade de rede é consumida por operações SNMPv1, SNMPv2 e SNMPv3 (com e sem os serviços de segurança);

- Prover uma análise quantitativa, para ajudar na decisão de gerentes e administradores de redes, na utilização de soluções de gerenciamento, que contemplam o protocolo SNMP, levando-se em conta os níveis de segurança desejados bem como o desempenho da rede.

1.2 Justificativa

Atualmente, o gerenciamento e manutenção das redes de computadores constitui-se em desafio para gerentes e administradores de redes e técnicos operacionais. A crescente complexidade das redes e o surgimento de novas tecnologias mobilizam pessoal especializado na busca de soluções efetivas de gerenciamento que permitem reduzir custos e prover altas disponibilidades, desempenho, segurança e qualidade de serviços aos usuários finais.

Enquanto as soluções para o gerenciamento do suporte físico das redes encontram-se bem sedimentadas, torna-se necessário investigar formas de prover o gerenciamento efetivo das aplicações e protocolos que são executados sobre essas redes. Levando-se em consideração o crescente número de protocolos e a importância das

aplicações executadas sobre as redes, a simples manutenção dos equipamentos da rede em operação (“no ar”) já não garante o seu funcionamento.

À medida que as empresas mais se comprometem com as redes de computadores e com o uso das novas tecnologias de rede (*hardware e software*), mais importância devem dar à sua segurança. Quanto mais importantes forem as informações transferidas através da rede, maior será o prejuízo em caso de ocorrência de algum incidente de perda ou destruição (RAMOS, 1994). Nesse sentido, a segurança do gerenciamento de redes é um fator crítico. É necessário adicionar segurança nos sistemas de gerência de redes para que eles próprios sejam seguros.

A gerência de redes é uma aplicação que torna o ambiente mais vulnerável, visto que é mais um ponto passível de ser usado para a invasão de sistemas. Por isso, o protocolo SNMPv3 assume uma grande importância, pois ele apresenta mecanismos de segurança que não haviam sido definidos nas versões SNMPv1 e SNMPv2, que são: autenticação (autenticação entre agentes e gerentes), criptografia (tráfego entre agentes e gerentes pode ser todo criptografado) e controle de acesso (controle dos níveis de acesso diferenciados entre gerentes e agentes). O SNMPv3 é significativamente mais complexo que SNMPv1 e SNMPv2 e o propósito primário para o seu desenvolvimento foi corrigir a falha mais óbvia das versões anteriores, qual seja, a falta de segurança.

Por outro lado, o ambiente de negócios requer alto desempenho e, ao se introduzir recursos de segurança, estes podem ou não afetar o desempenho das redes corporativas. Por meio do gerenciamento de desempenho, que tem como meta medir e armazenar vários aspectos do desempenho da rede (vazão, utilização da linha de comunicação, tempo de resposta para o usuário final, entre outros), é possível estabelecer limiares de forma que sejam mantidos níveis aceitáveis de desempenho entre as redes (MLAK, 2001).

Sendo assim, avaliar a sobrecarga do protocolo SNMPv3 é importante, porque se pode afirmar se o mesmo é excessivo ou não, em aplicações de gerenciamento de redes baseadas em SNMP. Ou seja, é preciso verificar se a gerência de redes com e sem os serviços de segurança, impacta ou não no desempenho das redes.

1.3 Metodologia

Entre os métodos de avaliação de desempenho, existem três técnicas para avaliação de desempenho de sistemas: **modelo analítico, simulação e medição.** Neste trabalho optou-se pela técnica de simulação por se tratar de uma técnica que freqüentemente está mais próxima da realidade; por permitir imitar o funcionamento de um sistema real e por permitir a análise de diversas alterações no cenário virtual (laboratório virtual), sem custo, em um prazo menor, e, devido ao risco e impossibilidade de atuar no cenário real, por se tratar de um ambiente corporativo.

A métrica considerada na avaliação da sobrecarga do protocolo SNMP foi o adicional de *octetos*, acrescido a cada mensagem nas diferentes versões do protocolo.

As operações realizadas consistiram na obtenção de variáveis SNMP da MIB 1213, bem como a geração de *traps*. Com a utilização desta métrica o trabalho mantém a homogeneidade com (DU, 2002) e (HIA, 2002) que também avaliaram a sobrecarga no protocolo SNMP em diferentes cenários (TLS e IPSec respectivamente).

1.4 Estrutura do Trabalho

Este trabalho está organizado em seis capítulos. No primeiro capítulo são apresentados uma introdução, os objetivos (geral e específicos), a justificativa, a metodologia e a estrutura do trabalho.

No segundo capítulo, apresenta-se os aspectos e importância do desempenho em redes de computadores bem como as métricas de desempenho, os métodos de avaliação dando-se ênfase à técnica de simulação e encerrando-se com a descrição de duas ferramentas de simulação SNMP.

No terceiro capítulo, apresenta-se a evolução da arquitetura de gerenciamento Internet/SNMP, descrevendo-se brevemente os conceitos básicos do SNMP, o modelo de gerenciamento e as características das diferentes versões do protocolo – SNMPv1, SNMPv2 e SNMPv3.

O quarto capítulo descreve a MIB 1213 (Base de Informações Gerenciáveis) e as características de desempenho do protocolo SNMP, abordando temas acerca do desempenho de sistemas finais, sobrecarga do protocolo SNMP e mapeamentos de transporte do SNMP.

No quinto capítulo é apresentado um estudo de caso, realizado por meio de simulação, da análise da sobrecarga quando da utilização do protocolo SNMPv3 na gerência de redes, provendo uma análise quantitativa, para auxiliar na decisão de gerentes e administradores de redes na utilização de soluções de gerenciamento baseadas em SNMP, levando-se em conta os níveis de segurança desejados bem como o desempenho da rede.

Por fim, o sexto capítulo apresenta as conclusões, principais dificuldades encontradas e trabalhos futuros.

CAPÍTULO II

2. DESEMPENHO EM REDES DE COMPUTADORES

2.1 Introdução

Devido à crescente evolução tecnológica, ao aumento do uso das aplicações em redes de computadores e conseqüentemente à dependência de usuários e negócios nas organizações, o ambiente torna-se complexo, o que exige soluções efetivas de gerenciamento que permitam reduzir custos e prover altas disponibilidades, desempenho e segurança dessas redes, garantindo a qualidade dos serviços e a satisfação dos usuários. Por isso deve-se levar em conta a capacidade de planejamento, que é o processo de determinar as futuras exigências de prováveis recursos de rede para prevenir o bom desempenho e o impacto de disponibilidade em aplicações críticas de negócio.

Na área de capacidade de planejamento, a *baseline*[1] (perfil) de rede (CPU, memória, *buffers*, *octetos* de entrada e saída, etc.) pode ser implementada para medir tempo de resposta. Então, é importante lembrar que problemas de desempenho

correlatam freqüentemente com capacidade. Em redes, estes são, tipicamente, a largura de banda e dados que precisam esperar em filas antes de serem transmitidos pela rede. Por exemplo, em aplicações de voz, este tempo de espera certamente impacta para os usuários porque fatores como estes afetam a qualidade de transmissão de voz.

Neste contexto, gerenciar este ambiente complexo e estratégico é fundamental para garantir níveis aceitáveis de desempenho e alta disponibilidade de rede, os quais estão se tornando missão crítica para grandes empreendimentos e provedores de serviços, pois a tendência é buscar ganhos econômicos em curto prazo ao risco de incorrer custos mais altos no final das contas (freqüentemente imprevistos). Durante todos os ciclos de orçamento, administradores de rede e pessoal de implementação de projetos enfrentam o dilema de achar um equilíbrio entre desempenho e implementação rápida.

Em geral, gerenciamento de redes é um serviço que emprega várias ferramentas, aplicações e dispositivos para ajudar os gerentes de redes no monitoramento e manutenção de suas redes. Por volta dos anos 80 (oitenta), algumas companhias enfrentavam grandes problemas, experimentando o surgimento de muitas e diferentes tecnologias de rede (muitas vezes incompatíveis). Cada nova tecnologia de rede requeria sua própria equipe de peritos. Em meados de 1980, as exigências para provimento de pessoal, somente para gerenciar e administrar grandes redes heterogêneas, criaram uma crise para muitas organizações. Era preciso a automatização integrada para o gerenciamento de redes (inclusive para o planejamento da capacidade e o desempenho das redes) em diferentes ambientes (MLAK, 2001).

Gerência de desempenho é uma das áreas funcionais da gerência de redes que tem como meta medir e armazenar vários aspectos do desempenho (vazão, utilização da linha de comunicação, tempo de resposta para o usuário final entre outros) da rede, de forma que sejam mantidos níveis aceitáveis de desempenho entre as redes (MLAK, 2001).

---

[1] É o perfil da rede, que descreve uma caracterização estatisticamente válida do comportamento normal da

2.2 Importância do Desempenho em Redes de Computadores

Desempenho de redes é, atualmente, um dos assuntos críticos no gerenciamento de redes cliente/servidor. Futuramente, aprimoramentos em desempenho de comunicação, continuarão melhorando o desempenho das aplicações.(MLAK, 2001).

Ainda segundo (MLAK, 2001), a análise de desempenho requer três passos principais:

- dados de desempenho devem ser reunidos em variáveis de interesse para que possam ser transmitidos aos administradores e gerentes de rede;

- os dados devem ser analisados para determinar níveis normais de utilização (*baseline*);

- os limiares de desempenho apropriados são determinados para cada variável importante, de forma que exceder estes limiares indica um problema de rede, merecedor de atenção.

Cada um dos passos faz parte do processo para configurar um sistema reativo. Quando o desempenho fica inaceitável por causa de um usuário que excedeu o limiar definido, o sistema reage enviando uma mensagem (por exemplo, o usuário está efetuando um *download* de um arquivo e este faz com que a taxa de utilização do canal externo de comunicação da organização ultrapasse o limiar de 90% (noventa por cento) de utilização, o qual foi previamente configurado pelo gerente de rede).

Análise de desempenho inclui (RABINOVITCH, 2002):

- **análise de tráfego de rede** - congestionamento é um dos problemas mais significativos em um contexto IP aparecendo na rede

---

rede sobre um período estendido de tempo (FRANCESCHI, 1996).

basicamente porque os recursos da rede são insuficientes ou porque são utilizados inadequadamente para acomodar a carga oferecida, fazendo com que determinadas partes da rede sejam sub-utilizadas, enquanto outras estão sobrecarregadas (AWDUCHE *et al.*, 1999; AWDUCHE, 1999). A engenharia de tráfego que tem como principal objetivo prover garantia de qualidade de serviço engloba quatro problemas básicos: o controle de admissão de novas conexões; o roteamento de pacotes, dadas algumas restrições; o re-roteamento de conexões já estabelecidas; e o planejamento dos recursos da rede (Girish *et al.* 2000);

- **análise da largura de banda** - é uma medida de capacidade de transmissão de dados, normalmente expressa em *kilobits* por segundo (Kbps) ou *megabits* por segundo (Mbps). A largura de banda indica a capacidade máxima de transmissão teórica de uma conexão, mas na medida em que a utilização se aproxima da largura de banda teórica máxima, fatores negativos como atraso de transmissão podem causar deterioração em qualidade (OLIVEIRA, 2001);

- **análise de tendências** - é a área que trata da coleta de dados sobre o consumo de recursos para propósitos de análises de capacidade e tendências, alocação de custos, auditoria e cobrança. Na análise de tendências e no planejamento de capacidade, o objetivo é tipicamente prever utilizações futuras de recursos com base em dados estatísticos atuais;

- **análise de protocolos** – determinar padrões de utilização da rede dos usuários requer do administrador um conhecimento pontual sobre os protocolos e aplicações que cada um deles executa, o instante em que isto ocorre e as estações com as quais mais se comunicam, tanto

local como remotamente. Por exemplo, determinar o volume de tráfego HTTP, FTP, Telnet, etc (GASPARY, 1998);

- **análise de vazão** - é o montante de tráfego de dados movidos de um nó de rede para outro em um dado período de tempo, expresso em kilobits por segundo (Kpbs) ou megabits por segundo (Mbps) (OLIVEIRA, 2001).

- **análise de utilização** - utilização mede o uso de um recurso particular com o passar do tempo. A medida normalmente é expressa na forma de uma porcentagem na qual o uso de um recurso é comparado com sua máxima capacidade operacional. Por medidas de utilização, pode-se identificar congestionamento (ou congestionamento potencial) ao longo da rede e recursos sub-utilizados (CISCO, 2002);

- **análise de mudanças e atualizações** - ter uma *baseline* da rede existente antes de uma nova solução (Aplicações ou Sistemas Operacionais que mudam) torna necessário para medir as expectativas de desempenho para a nova solução. Isto ajudará inicialmente determinar se a solução apresenta o desempenho e objetivos de disponibilidade e capacidade conforme o planejado. Também é possível executar uma comparação de *baselines* entre ambientes antigos e novos ambientes para verificar as exigências de solução (CISCO, 2002);

Vazão <= Média de Largura de Banda

A vazão é afetada pelas seguintes situações:

- **o computador cliente e o servidor** - pode ser valioso uma avaliação das exigências das aplicações de rede. Estas informações podem ser usadas para o planejamento do ciclo de atualizações. *Baselines* de

aplicação também podem ser importantes na área de disponibilidade de aplicação em relação a serviços ou qualidade de serviços por aplicação. A *baseline* de aplicação consiste principalmente em largura da banda usada por aplicações por período de tempo (CISCO, 2002);

- **outros usuários na LAN** – com quem os usuários da rede se comunicam e que protocolos estão envolvidos bem como o volume de tráfego imposto por eles (GASPARY, 1998);

- **roteamento** - pode-se relacionar este item com o tempo de resposta, que é o tempo requerido para o tráfego "viajar" entre dois pontos. O tempo de resposta cronometrado mais lento que normal pode ser visto por uma comparação de uma *baseline* ou excedendo um limiar, pode indicar congestionamento ou uma falha de rede (CISCO, 2002).

- **a topologia de todas as redes envolvidas** – está diretamente relacionado com disponibilidade, que é a medida de tempo para qual um sistema de rede ou aplicação está disponível a um usuário. De uma perspectiva de rede, disponibilidade representa a confiança dos componentes individuais em uma rede. Redundância de rede é outro fator para considerar ao medir disponibilidade. Perda de redundância indica degradação de serviço em lugar de fracasso total da rede. O resultado pode ser tempo de resposta mais lento e uma perda de dados devido a pacotes perdidos (CISCO, 2002);

- **tipos de dados que são transferidos** - é importante notar que com aplicações mais novas como voz e vídeo, o desempenho é a variável fundamental para o sucesso e se não for possível alcançar um desempenho consistente, então o serviço será considerado de baixo valor e falho (CISCO, 2002);

- **tempo de resposta do dia (tráfego)** - por exemplo, ao implementar tráfego de voz sobre IP, os pacotes de voz devem ser entregues na hora certa a uma taxa constante para manter boa qualidade de voz. Gerando tráfego classificado como tráfego de voz, é possível medir o tempo de resposta do tráfego junto aos usuários (CISCO, 2002).

O número de interconexões e o uso das mesmas, latência[2] e o uso de cada nó e demora de propagação representam os principais fatores que podem afetar o desempenho da rede.

Estatísticas de desempenho são necessárias para o planejamento, administração e manutenção de grandes redes. Tais informações servem para indicar situações de gargalos, execução de ações corretivas (como por exemplo, o balanceamento e redistribuição de carga do tráfego da rede) (SPECIALSKI, 2001).

O objetivo deste trabalho é realizar uma análise da sobrecarga no desempenho das redes de computadores, quando da utilização do protocolo SNMPv3 na gerência de redes. Para isso a métrica considerada na avaliação da sobrecarga do protocolo SNMP, foi o adicional de *octetos,* acrescido a cada mensagem nas diferentes versões do protocolo.

2.3 Métodos de Avaliação de Desempenho

A avaliação do desempenho de sistemas computacionais pode ser requerida a partir do momento em que se apresente a necessidade de se obter a melhor performance do sistema, ou mesmo na busca de como manter a melhor performance na rede. Em (JAIN, 1991) são apresentadas três técnicas para avaliação de desempenho de sistemas: **modelo analítico, simulação e medição**. As características dessas técnicas são mostradas na tabela 2.1 a seguir:

---

[2] **Latência –** é o tempo em que um pacote leva para trafegar de um ponto para outro, fim-a-fim, em uma rede (SHURAN, 1999). Para maiores detalhes ver item 5.3.

Critério	Modelo Analítico	Simulação	Medição
1. Estágio	Qualquer	Qualquer	Depois do Protótipo
2. Tempo Necessário	Pequeno	Médio	Variado
3. Ferramentas	Analistas	Linguagem de Simulação	Instrumentação
4. Confiabilidade	Baixa	Moderada	Variada
5. Custo	Baixo	Médio	Alto

Tabela 2.1 – Critérios de Seleção da Técnica de Avaliação de Desempenho

O modelo analítico é uma técnica que exige muito do analista e pode ser criado por meio de equações matemáticas, teoria das filas, etc. A simulação é uma técnica que freqüentemente está mais próxima da realidade, possibilitando procurar combinações mais adequadas para os parâmetros de um determinado sistema. A técnica de medição é a realidade propriamente dita; com os dados coletados é possível concluir sobre o comportamento do ambiente, detectar falhas e prevenir futuros problemas.

No escopo deste trabalho, a técnica de simulação é a que mais se adequa para a análise da sobrecarga do protocolo SNMPv3, por tratar-se de uma técnica que freqüentemente está mais próxima da realidade; por permitir imitar o funcionamento de um sistema real e por permitir a análise de diversas alterações no cenário virtual (laboratório virtual), sem custo, em um menor prazo, e, devido ao risco e impossibilidade de atuar no cenário real, por tratar-se de um ambiente corporativo.

2.4 Aspectos de Simulação

A análise de desempenho também permite métodos pró-ativos podendo-se usar simulações de rede para projetar como a transmissão de dados em rede afetará as métricas de desempenho. Tal simulação pode alertar os administradores e gerentes sobre problemas iminentes de forma que possam ser tomadas medidas pró ativas.

2.4.1 Seleção da Técnica de Avaliação

A escolha de uma técnica de avaliação baseia-se em alguns critérios bastante simples, mas de vital importância. Estes critérios estão relacionados, principalmente, ao tempo disponível para os trabalhos e os custos envolvidos, mas é claro que outros fatores são determinantes, tais como: validação, precisão e aceitabilidade. Os critérios que mais são considerados são: o custo e o tempo, estes são determinantes na escolha da técnica de avaliação. Na tabela 2.2 é apresentada uma relação entre as técnicas de avaliação baseada nos critérios de custo e tempo (OLIVEIRA, 2001).

Técnica de Avaliação / Critério	Modelagem Analítica	Simulação	Medição
Custo	Pequeno	Médio	Alto
Tempo	Pequeno	Médio	Variável

Tabela 2.2 – Seleção da Técnica de avaliação

Neste trabalho, utilizou-se uma ferramenta de simulação direcionada à gerência de redes *(AdventNet Simulation Toolkit)* que possibilita uma maior agilidade no processo de modelagem dos agentes e gerentes, podendo-se com isso analisar e comparar os resultados das projeções do modelo criado em diferentes cenários. A seguir é apresentada uma breve descrição sobre o uso da técnica de simulação, assim como uma metodologia para desenvolvimento de modelos de simulação de sistemas.

2.4.2 Modelagem e Simulação de Cenários de Redes de Computadores

Conforme já mencionado anteriormente, a simulação se mostrou como uma técnica que permite imitar o funcionamento de um sistema real. A grande vantagem da simulação reside no fato de permitir a análise de diversas alterações no cenário virtual (laboratório virtual), sem custo, em um menor prazo, e, o risco de atuar no cenário real. Além disso, a simulação proporciona a análise de desempenho, que no escopo deste trabalho trata especificamente da sobrecarga do protocolo SNMP (em suas diferentes versões), podendo-se com isso, afirmar se o mesmo é viável ou não, em diferentes cenários.

Através do uso de técnicas estatísticas e modelagem matemática aplicada a uma ferramenta de simulação específica, pode-se estimar com um certo grau de certeza o desempenho de um sistema, permitindo realizar testes e suposições sem interferir no funcionamento deste, considerando e avaliando seus inúmeros componentes e aplicações e determinando o comportamento e otimização do sistema modelado.

A simulação constitui-se de uma técnica que permite modelar, avaliar e prever a capacidade dos sistemas (Exemplo: redes de computadores, gerenciamento de redes de computadores) (OLIVEIRA, 2001). No diagrama da figura 2.5, é apresentada uma metodologia (FREITAS, 1999) que mostra as etapas a serem seguidas para o desenvolvimento de um modelo de simulação de um sistema.

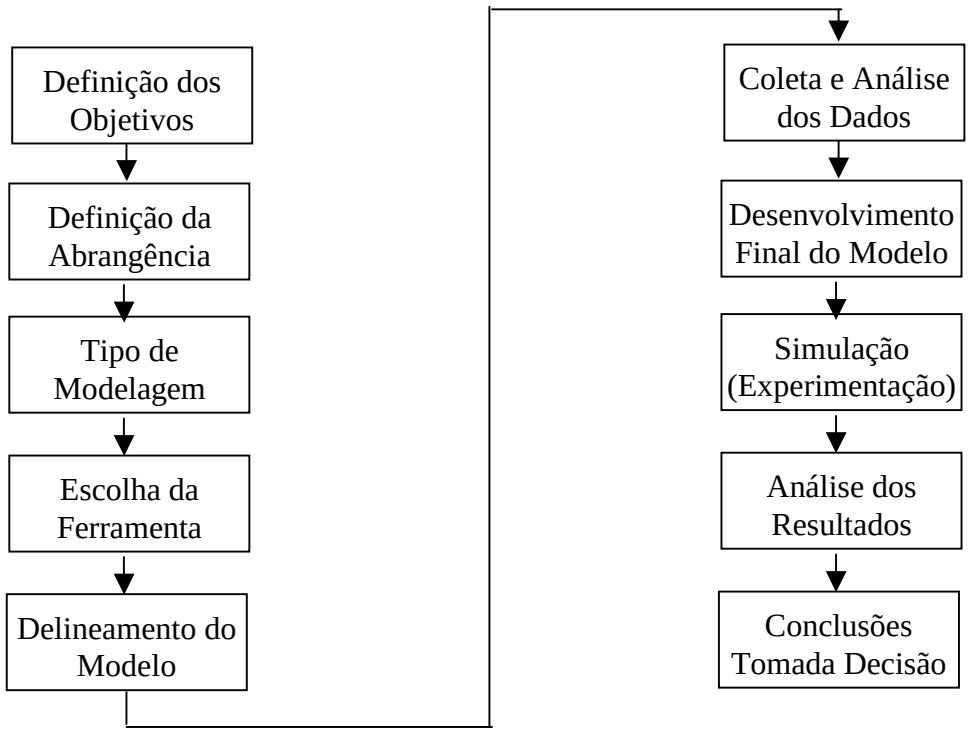

Figura 2.1 – Diagrama do processo de abordagem e etapas do projeto

A abordagem de um projeto inicia com a definição de objetivos razoáveis e realizáveis. Além das questões relativas à abrangência e ao nível de detalhes que será considerado, é necessário escolher a ferramenta a ser utilizada no processo de modelagem. O nível de detalhamento adotado no modelo deverá ser compatível com os objetivos estabelecidos.

As técnicas de avaliações, baseadas em simulações, constituem-se de ferramentas imprescindíveis no planejamento, validação e análise de projetos. Normalmente estas ferramentas disponibilizam recursos para trabalhar com:

- Funções estatísticas e distribuições de probabilidade para a geração dos dados;
- Funções para obtenção precisa de resultados estatísticos;
- Capacidade gráfica interativa para construção de modelos;
- Um depurador interativo para controle e monitoria da execução da simulação.

Feita a escolha da ferramenta e esboçado um perfil preliminar do modelo, é preciso traçar um plano para a aquisição dos dados necessários. Esta tarefa constitui-se de uma boa precisão (através de monitorações ou fontes de dados – arquivos), considerando que “dados ruins fornecerão resultados péssimos”. Depois a simulação poderá ser realizada. Todas as conclusões devem ser baseadas em análises estatísticas procedentes.

2.5 Ferramentas de Simulação SNMP

O mercado está inundado em larga escala de produtos para ajudar no gerenciamento de redes. Por isso, antes de adquirir um produto de gerenciamento de redes, dever-se-ía realizar testes em um simulador SNMP com o objetivo de verificar se o produto atende as expectativas desejadas.

MIMIC Simulator (GAMBIT, 2002) da *Gambit Communications* é um conjunto modular de simuladores direcionado para a redução do custo de propriedade e do empreendimento das aplicações de gerenciamento de redes, ao longo do ciclo de vida delas, do desenvolvimento, da garantia da qualidade, avaliação, customização,

marketing, vendas e treinamento. *MIMIC* oferece complementos que eram desenvolvidos em laboratórios físicos, tais como: avaliação, suporte, testes, demonstrações e o provimento de treinamento para aplicações de gerenciamento de redes. É possível criar um laboratório virtual com dispositivos de rede como: roteadores, *switches*, *hubs*, *cable modems*, estações de trabalho, servidores, etc.

MIMIC SNMP Agent Simulator permite simular até 10.000 dispositivos gerenciáveis através do SNMP, em um PC baseado na tecnologia *Intel* ou em uma *Sun Sparc*, bem como pode-se simular um número ilimitado de dispositivos, distribuindo-os sobre múltiplas estações de trabalho. Com suporte para qualquer dispositivo baseado em SNMP é possível inicializar uma variedade de configurações de dispositivos com a aplicação de gerenciamento. Configurações são completamente inicializadas em tempo de customização, ambas em uma base individual ou coletiva. O *MIMIC* responde consultas SNMP realizadas por qualquer um dos endereços IP´s configurados, e "enxerga" a aplicação de gerenciamento de rede, como se estivesse "conversando" com dispositivos reais. *MIMIC IOS Simulator* suporta o *Cisco IOS* software e SNMP v1, v2, v3. É um ambiente extensível que permite acrescentar novas MIBs e novos dispositivos à simulação. Está disponível para as plataformas: *Linux, Solaris, Windows*.

AdventNet Simulation Toolkit (ADVENTNET, 2002) é outra ferramenta de simulação do protocolo SNMP o qual é utilizado no gerenciamento de redes. A estrutura deste software é dividida em API de baixo nível, que contém a funcionalidade básica do SNMP, e um conjunto de APIs de alto nível, organizados em *JavaBeans*, e, apresenta as seguintes características:

- Configuração dos agentes SNMP nas versões SNMPv1, SNMPv2 e SNMPv3;
- Geração de *Traps*, com a possibilidade de visualização e captura das mensagens geradas;
- Geração de gráficos;

- Ferramenta para a modelagem de redes integrada com simulação de agentes SNMP e TL1;
- Armazenamento de dados disponível em banco de dados SQL e XML;
- Excelentes características gráficas para uma melhor produtividade;
- A rede projetada pode ser portada para outras plataformas;
- Disponível para as plataformas: *Linux, Solaris, Windows.*

É uma ferramenta específica para modelar e simular sistemas de gerenciamento de redes de computadores. Esta ferramenta fornece uma série de funcionalidades, como: simular o desempenho de aplicações considerando diferentes tecnologias de redes; predizer o comportamento e o desempenho exato de sistemas de gerenciamento de redes antes que eles sejam implantados, proporcionando desta forma redução de custos no caso de existirem falhas. Também podem ser realizados experimentos buscando uma melhor utilização da rede, bem como fazer um planejamento da mesma. É possível fazer uma análise do tráfego (gerência) simulando diferentes condições de funcionamento da rede, realizando as operações detalhadas, produzindo relatórios que descrevem o desempenho da rede e da aplicação.

Diante deste contexto, deve-se optar por ferramentas de simulação específicas para o sistema que será modelado. Neste trabalho optou-se por utilizar o *AdventNet Simulation Toolkit*, ferramenta específica para modelagem e simulação do protocolo SNMP, estando atualmente na versão 3.0, utilizado no gerenciamento de redes de computadores.

O uso da simulação e a escolha da ferramenta *AdventNet Simulation Toolkit* foram considerados neste trabalho, devido:

- a ferramenta possuir um ambiente de modelagem totalmente gráfico e intuitivo;
- o agente real SNMP não estar presente;
- todos os agentes reais SNMP não estarem implementados em sua totalidade (que não é o objetivo deste trabalho);
- realização de testes em diferentes MIBs;
- simulação de equipamentos de rede, do ponto de vista da gerência.

Pelas descrições das potencialidades do uso de simulação e pela impossibilidade e risco de atuar no cenário real (devido tratar-se de um ambiente corporativo), sem custo, em um menor prazo, e, devido ao risco e impossibilidade de atuar no cenário real, por tratar-se de um ambiente corporativo, é que se optou pelo uso de um simulador para validar o trabalho proposto. Embora, também poderia ser utilizada a técnica de modelagem analítica ou medição para fins de comparação com a simulação, conforme características já descritas na tabela 2.1. Indiferentemente da ferramenta usada para modelagem e simulação convém salientar que, da mesma forma, podem ser desenvolvidos modelos matemáticos (modelagem analítica) utilizando-se de recursos computacionais e obter-se resultados semelhantes.

Neste capítulo, descreveu-se a importância da análise do desempenho de redes corporativas, que é atualmente um dos assuntos críticos no gerenciamento de redes cliente/servidor, pois é preciso prover uma análise contínua das exigências da capacidade da rede que deverá assegurar largura de banda suficiente e estar disponível para atender aos objetivos de desempenho. Também foram apresentados métodos de avaliação de desempenho (modelo analítico, simulação e medição); métricas de desempenho. Além disso, foram apresentadas algumas características e os principais aspectos de simulação levando-se em conta a seleção da técnica de avaliação e a modelagem e simulação de cenários de redes de computadores e, por fim foram

apresentadas duas ferramentas de simulação SNMP, entre as quais o *AdventNet Simulation Toolkit* foi utilizado no estudo de caso proposto.

Portanto, a análise de desempenho e disponibilidade devem ser capazes de executar diagnósticos do desempenho da rede, analisar a eficiência do tráfego, operando para auxiliar na capacidade de planejamento e prover apoio essencial para a equipe de suporte técnico. O capítulo a seguir apresenta a evolução da arquitetura de gerenciamento Internet/SNMP bem como os conceitos básicos de SNMP, modelo de gerenciamento e as características das diferentes versões do protocolo.

CAPÍTULO III

3. EVOLUÇÃO DA ARQUITETURA SNMP

A crescente expansão do número de dispositivos e, conseqüentemente, da complexidade das redes TCP/IP incentivou o surgimento de mecanismos que permitissem seu efetivo gerenciamento (TANEMBAUM, 1997). A primeira proposta real nesse sentido foi lançada em 1988, com a definição do protocolo SNMP – ou SNMPv1. Esse protocolo tornou-se rapidamente o esquema de gerenciamento mais utilizado, independentemente de fabricantes (STALLINGS, 1998a).

Outras características muito importantes, oferecidas pelo SNMP, são a extensibilidade, que permite aos fornecedores de equipamentos de rede adicionarem novas funções de gerência aos seus produtos, e a independência do *hardware* utilizado.

Em 1992 foram iniciados os trabalhos para estender a versão original do protocolo SNMP. Essa iniciativa recebeu o nome de SNMPv2.

O primeiro conjunto de documentos com a definição de SNMPv2 foi lançado em 1993, no formato de *Proposed Internet Standard* (STALLINGS, 1998a). O protocolo não recebeu a aceitação esperada por seus criadores. Enquanto os aprimoramentos funcionais foram muito bem vindos, os desenvolvedores consideraram a solução criada para prover segurança demasiadamente complexa. Diante dessa

questão, o grupo de trabalho foi reativado. Alguns aspectos funcionais do protocolo foram aprimorados, mas o problema da segurança permaneceu sem solução. Essa revisão do protocolo SNMPv2 o fez progredir para *Draft Internet Standard* em 1996. Os documentos mantiveram os aprimoramentos funcionais intactos mas removeram os aspectos relacionados à segurança (STALLINGS, 1998b).

A versão 2 (dois) do protocolo SNMP acabou não definindo mecanismos de segurança . Para remediar essa deficiência, um conjunto de grupos independentes começou pesquisar formas de aprimorar a segurança do protocolo. Duas abordagens se destacaram: SNMPv2u e SNMPv2*. Essas duas propostas deram início a um novo grupo de trabalho criado para definir o SNMPv3 (STALLINGS, 1998b). Em janeiro de 1998, o grupo produziu um conjunto de *Proposed Internet Standards* (ver tabela 3.3). Esses documentos definem uma metodologia para incorporar características de segurança a SNMPv1 ou SNMPv2.

3.1 Conceitos Básicos de SNMP

3.1.1 Modelo de Gerenciamento

O modelo de gerenciamento de rede adotado pelo protocolo SNMP possui quatro elementos chave: estação de gerenciamento, agentes de gerenciamento, base de informação de gerenciamento – MIB – e protocolo para troca de informações de gerenciamento (ver figura 3.1) (MILLER, 1997; STALLINGS, 1998a; TANEMBAUM, 1997).

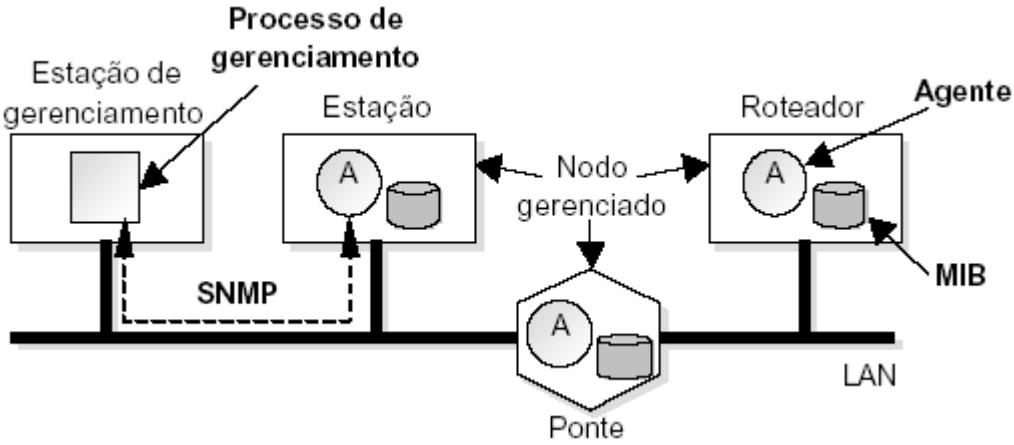

Figura 3.1 – Arquitetura de Gerenciamento TCP/IP

Segundo (STALLINGS, 1998a), a **estação de gerenciamento** deverá ter, no mínimo:

- Um conjunto de aplicações de gerenciamento para análise de dados, recuperação de falhas, entre outras;
- Uma interface através da qual o gerente possa monitorar e controlar a rede, com um poderoso mas amigável conjunto de comandos, para executar a maioria ou todas as tarefas de gerenciamento da rede;
- Um protocolo através do qual a estação de gerenciamento e as entidades gerenciadas possam trocar informações de gerenciamento e controle;
- Uma base de dados com informações extraídas a partir das bases de informação de gerenciamento de todas as entidades gerenciadas.

O **agente de gerenciamento** pode estar em roteadores, impressoras, estações de trabalho, *hubs*, etc, os quais podem ser monitorados pela estação de gerenciamento (STALLINGS, 1998a). O agente deve:

- Responder a pedidos de informações oriundos da estação de gerenciamento;

- Realizar ações requisitadas pela estação de gerenciamento;

- Enviar informações importantes para a estação de gerenciamento, embora não solicitadas.

Os dispositivos gerenciáveis mantêm uma ou mais variáveis que descrevem seu estado. Essas variáveis são denominadas objetos (TANENBAUM, 1997). A coleção desses objetos é denominada MIB (Base de Informações de Gerenciamento). A estação de gerenciamento monitora os equipamentos recuperando valores de objetos da MIB, podendo realizar ações no agente ou modificar a configuração de seus parâmetros alterando valores de variáveis específicas.

O **protocolo de gerenciamento** é responsável pela troca de informações entre a **estação de gerenciamento** e os **agentes**. O **protocolo de gerenciamento** provê as seguintes funcionalidades (STALLINGS, 1998a):

- *Get:* permite que a estação de gerenciamento recupere o valor de objetos do agente;

- *GetNext:* permite que a estação de gerenciamento recupere o próximo valor de objetos do agente;

- *GetBulk:* introduzido no SNMPv2; a operação G*etBulk* foi introduzida para recuperar grandes quantidades de informações, sem a necessidade de utilizar-se repetidas operações G*etNext*. *GetBulk* foi projetado para eliminar virtualmente a necessidade de utilização de operações G*etNext*;

- *Set:* permite que a estação de gerenciamento altere o valor de objetos do agente;

- *Trap:* permite que o agente notifique a estação de gerenciamento sobre ocorrência de eventos significativos;

- *Inform:* introduzido no SNMPv2; a operação *Inform* foi adicionada para possibilitar a notificação gerente-gerente.

O protocolo SNMP atua no nível de aplicação da arquitetura TCP/IP. Na figura 3.2, é possível observar que o protocolo SNMP opera sobre o protocolo UDP. Portanto, tanto a estação de gerenciamento como os agentes devem implementar os protocolos SNMP, UDP e IP.

Uma estação de gerenciamento pode gerar os seguintes tipos de mensagens SNMP: *GetRequest*, *GetNextRequest* e *SetRequest*. As duas primeiras são variações da função G*et*, citada anteriormente. As mensagens são confirmadas pelo agente através de uma mensagem *GetResponse*, que é repassada para a aplicação de gerenciamento. Além disso, um agente pode gerar uma mensagem *Trap* em resposta a um evento que afete a MIB e os recursos monitorados por ele (por exemplo, finalização abrupta do agente e seu reinício imediato, falha de um canal de comunicação, sobrecarga na rede – baseada em uma *baseline* pré estabelecida).

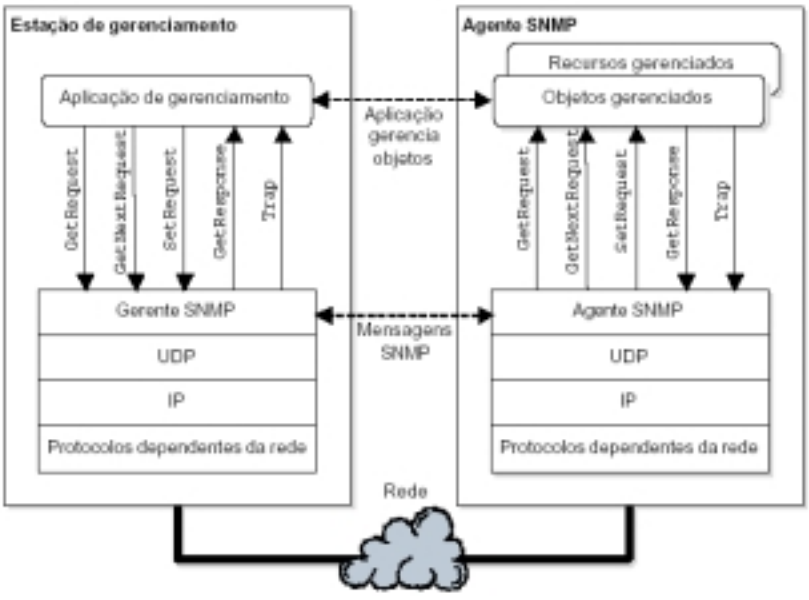

Figura 3.2 – Protocolos e Mensagens no Gerenciamento via SNMP

Segundo (SPECIALSKI, 2001), a arquitetura do *software* de gerenciamento residente no gerente e nos agentes varia de acordo com a funcionalidade da plataforma adotada. Genericamente, o *software* pode ser dividido em três grandes categorias:

- *Software* de apresentação (interface)
- *Software* de gerenciamento (aplicação)
- *Software* de suporte (base de dados e comunicação)

3.1.2 *Software* de Apresentação

A interface de usuário, em um sistema de gerenciamento, permite que o usuário monitore e controle a rede. Normalmente ela está localizada no sistema gerente. Em alguns casos é comum existir uma interface em alguns agentes a fim de permitir a execução de testes e também a visualização ou alteração de alguns parâmetros localmente. A figura 3.3 (a) mostra os dois blocos que representam o *software* de apresentação das informações de gerência.

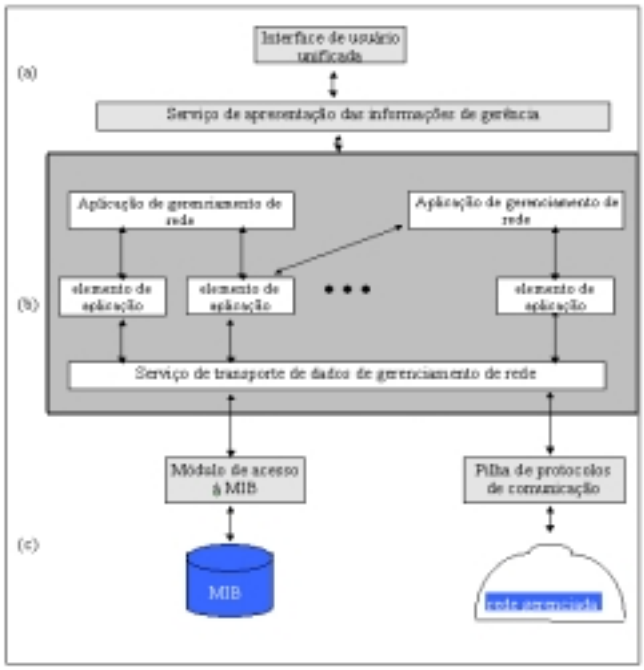

Figura 3.3 – Arquitetura de um Sistema de Gerenciamento de Rede

O principal, em qualquer sistema de gerenciamento, é que a interface seja unificada, isto é, que ela seja a mesma em qualquer nodo, permitindo que o usuário gerencie uma rede heterogênea com um mínimo de treinamento.

3.1.3 *Software* de Gerenciamento

O *software* que fornece a aplicação de gerenciamento pode ser muito simples, como é o caso do modelo SNMP, ou muito complexo, como o modelo OSI. A figura 3.3 (b) mostra uma estrutura genérica de um *software* de gerenciamento, organizado em três níveis: aplicação de gerenciamento de rede, elementos de serviço da aplicação e serviço de transporte de dados de gerenciamento da rede.

A aplicação de gerenciamento de rede provê os serviços de interesse do usuário como, por exemplo, gerenciamento de falhas, de configuração, de segurança, de desempenho e contabilização. Os elementos de serviço da aplicação implementam funções de propósito geral que servem de suporte à diversas aplicações, tais como, alarmes genéricos ou sumarização de dados. O serviço de transporte de dados de gerenciamento consiste de um protocolo usado para a troca de informações entre gerentes e agentes e de uma interface de serviço para os elementos de serviço de aplicação.

3.1.4 *Software* de Suporte ao Gerenciamento

Para executar suas tarefas, o *software* de gerenciamento necessita acessar uma base de informações de gerenciamento (MIB) e agentes e gerentes remotos.

A MIB localizada em um nodo agente contém informações de gerenciamento que refletem a configuração e o comportamento do nodo e parâmetros que podem ser usados para controlar a operação do nodo. A MIB localizada no gerente contém informações específicas do nodo onde está localizada e informações resumidas sobre os agentes sob o seu controle.

O módulo de acesso à MIB, mostrado na figura 3.3 (c), inclui *software* de gerenciamento de arquivos que habilitam o acesso à MIB. Adicionalmente, o módulo de acesso à MIB pode converter o formato local das informações para um formato padronizado do sistema de gerenciamento.

O modelo de informação de um sistema de gerenciamento fornece a estrutura para representação, armazenamento e transferência das informações de gerenciamento. Esta estrutura é denominada SMI (*Structure Management Information*) e, dependendo do sistema, poderá apresentar maior ou menor complexidade.

No modelo OSI, a SMI é baseada no paradigma de orientação a objetos, enfatizando as hierarquias de classe, de *containment* e de registro. Já o modelo SNMP, utiliza conceitos de Tipos de Dados, embora a sua nomenclatura refira-se a objetos.

A figura 3.4 abaixo, mostra exemplos de definição de objeto em cada um dos modelos.

Modelo SNMP	Modelo OSI

abcObjectType OBJECT-TYPE
SYNTAX INTEGER {
choicelabel1 (1),
choicelabel2 (2)
}
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Description Text"
::= { pqr 3 }
	network MANAGED OBJECT CLASS
	DERIVED FROM top;
	BEHAVIOR network-behavior;
	CHARACTERIZED BY
	networkPackage PACKAGE
	ATTRIBUTES
	networkID GET,
	networkType GET;
	
	REGISTERED AS (exemplo MObjectClass 2);

Figura 3.4 – Exemplos de Especificação de Objetos Gerenciados

A comunicação entre gerentes e agentes é suportada por uma pilha de protocolos, tais como a pilha OSI ou a pilha TCP/IP. A arquitetura de comunicação suporta o protocolo de gerenciamento de rede que está localizado na camada de aplicação.

Os serviços básicos de um sistema de gerenciamento de rede são os serviços de monitoração e controle. Estes serviços são obtidos através de primitivas para a leitura e escrita nos valores dos objetos gerenciados. Outros serviços adicionais são: criação e destruição de objetos gerenciados, execução de ações sobre objetos gerenciados e emissão de relatórios de eventos.

A comunicação entre agentes e gerentes, a fim de alcançar estes serviços, deve seguir um conjunto de regras básicas, como em qualquer outra aplicação distribuída. Este conjunto de regras pode apresentar maior ou menor complexidade, dependendo do modelo adotado. O modelo OSI utiliza o protocolo CMIP (*Common Management Information Protocol*) que apresenta funcionalidades para a execução de operações sobre vários objetos a partir da definição de um escopo e de um filtro para a seleção de objetos. O modelo Internet utiliza o protocolo SNMP (*Simple Network Management Protocol*), cuja funcionalidade reside basicamente na leitura e alteração de valores de variáveis e em alguns relatórios de evento para situações específicas.

3.1.5 O que é uma MIB (Base de Informações de Gerenciamento)

Antes de definir o que é uma MIB, será introduzido o conceito de **objetos gerenciados**.

Um objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas de dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso.

Dessa forma, a MIB (*Management Information Base*) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, possibilitando assim, a automatização de grande parte das tarefas de gerência.

É a estrutura de dados básica de um sistema de gerenciamento. Consiste basicamente numa tabela onde se encontram os dados relevantes ao gerenciamento de um sistema. Seu formato é definido pela SMI (*Structure of Management Information*), que é descrita na linguagem ASN.1 (*Abstract Syntax Notation One*).

Informações que formam uma coleção de objetos gerenciáveis , residentes em um repositório virtual de informações. (CASE, 1996).

Segundo (PEITER, 2000), embora já existam MIBs prontas com agentes e gerentes implementados que monitoram uma grande parte de informações de interesse de uma arquitetura de rede e até sistemas, é difícil achar uma que se adapte as nossas necessidades, pois existem várias implementações SNMP, sendo difícil conseguir uma aplicação que resolva os problemas específicos de cada empresa. Por este motivo, muitas empresas acabam criando uma MIB própria que colete e armazene as informações desejadas. A MIB criada torna-se então não válida, pela dificuldade e pelo tempo dispendido para a definição, compilação e implementação.

Neste trabalho, optou-se pela utilização da MIB-II, mais especificamente a MIB 1213 devido ao fato desta MIB ter sido projetada para monitorar a mídia de rede, em vez de um dispositivo especifico. Portanto, ela é útil no monitoramento do tráfego de rede como um todo. A MIB 1213 é uma Base de Informações padrão que contém um conjunto de objetos bem definidos, conhecidos e aceitos pelos grupos padrão da Internet. Duas versões da MIB padrão estão disponíveis hoje em dia, e são chamadas MIB-I (RFC 1156/1066) e MIB-II (RFC 1213/1158).

No apêndice A é apresentado um estudo completo do que é uma MIB e os dois principais padrões de MIB, a MIB da OSI e a MIB da Internet, aprofundando mais neste último, no qual são apresentados todos os objetos gerenciados e suas possíveis utilizações. Também são apresentados o status e evolução das MIBs RMON1 e RMON2.

3.1.6 SNMPv1

A tabela 3.1 lista as RFCs chave que definem o protocolo SNMPv1 que, como já mencionado, tornou-se rapidamente o esquema de gerenciamento de rede mais utilizado no mercado. Sua disseminação, no entanto, tornou suas deficiências visíveis (SATALLINGS, 1998a). (STALLINGS, 1996) apresenta algumas delas:

- SNMP pode não ser adequado no gerenciamento de redes muito grandes devido às limitações de desempenho impostas pelas consultas. Com SNMP, é necessário enviar um pacote para receber outro com informação. Esse tipo de consulta resulta em grande volume de mensagens de gerenciamento e acarreta problemas de tempo de resposta que podem degradar o desempenho da rede;

- SNMP não é adequado para recuperar grandes volumes de dados, como os existentes em tabelas;

- As *traps* SNMP não são confirmadas. No caso típico onde UDP/IP é usado para entregar as *traps*, o agente não tem certeza se um alerta crítico chegou ou não à estação de gerenciamento;

- O protocolo SNMP provê apenas um mecanismo trivial de autenticação. Assim, SNMP é mais adequado para monitoração do que para controle;

- SNMP não suporta comandos imperativos. A única forma de disparar um evento em um agente é indiretamente, determinando o valor de um objeto da sua MIB. Essa solução é menos flexível e poderosa do que uma baseada em algum tipo de chamada de procedimento remoto, onde é possível passar parâmetros, definir condições e obter resultados oriundos da execução do procedimento;

- O modelo da MIB é limitado e não suporta aplicações que façam consultas de gerenciamento sofisticadas baseadas em valores de objetos ou tipos;

- SNMP não suporta comunicações do tipo gerente-gerente. Por exemplo, não há um mecanismo que permita a um sistema de gerenciamento aprender sobre dispositivos e redes gerenciadas por outro sistema de gerenciamento.

RFC	Título	Data
1155	Structure and identification management information for TCP/IP-based internets	Maio de 1990
1157	A Simple Network Management Protocol (SNMP)	Maio de 1990
1212	Concise MIB definitons	Março de 1991
1213	Management information base for network management of TCP/IP-based internets: MIB-II	Março de 1991

Tabela 3.1 – RFCs do Protocolo SNMPv1

3.1.7 SNMPv2

Os principais objetivos do SNMPv2 eram:

- Criar soluções para aprimorar a recuperação de grandes volumes de dados dos agentes;
- Propor mecanismos para a realização de gerenciamento distribuído e;
- Definir mecanismos de segurança.

A tabela 2.2 lista as RFCs resultantes das definições do protocolo SNMPv2.

RFC	Título	Data
1901	Introduction to community-based SNMPv2	Janeiro de 1996
1902	Structure of management information for SNMPv2	Janeiro de 1996
1903	Textual conventions for SNMPv2	Janeiro de 1996
1904	Conformance statements for SNMPv2	Janeiro de 1996
1905	Protocol operations for SNMPv2	Janeiro de 1996
1906	Transport mappings for SNMPv2	Janeiro de 1996
1907	Management information base for SNMPv2	Janeiro de 1996
1908	Coexistence between version 1 and version 2 of the Internet standard network management framework	Janeiro de 1996

Tabela 3.2 – RFCs do Protocolo SNMPv2

3.1.7.1 Aprimoramentos Funcionais na Transferência de Dados

SNMPv1 pode gerar tráfego considerável quando gerentes se comunicam com agentes. Isso ocorre porque com SNMPv1 apenas uma quantidade limitada de dados pode ser recuperada em uma única transação, forçando gerentes e agentes a gerarem várias transações. Essa limitação pode resultar em sobrecarga da rede.

SNMPv2 define o comando *GetBulk*, criado para resolver a limitação citada acima. Ele é destinado à recuperação de informações presentes em tabelas. Com SNMPv1, é possível recuperar uma única linha por vez de uma tabela. Se o gerente precisa visualizar uma tabela de roteamento completa, por exemplo, várias transações *Get/Response* precisam ser executadas (STALLINGS, 1998 a). Com o comando *GetBulk*, o gerente pode recuperar uma tabela por inteiro e, adicionalmente, objetos simples em uma única consulta.

Outra característica do SNMPv2 para aprimorar a transferência de dados é o comando *Get* não atômico. No SNMPv1, se um comando *Get* solicita o valor de diversas variáveis e o agente não é capaz de retornar o valor de uma delas, o comando é rejeitado por inteiro. O comando *Get* não atômico do SNMPv2 permite que resultados parciais sejam recuperados, ou seja, o agente retornará os valores que puder e ignorará as outras variáveis no comando.

Outro aprimoramento segundo (STALLINGS, 1998a), é o gerenciamento de rede hierárquico fracamente distribuído, onde podem existir diversas estações de gerenciamento, conforme mostradas na figura 3.5, como servidores de gerenciamento. Cada um desses servidores pode gerenciar diretamente um subconjunto do número total de agentes da rede. No entanto, a monitoração de muitos desses agentes é delegada pelo servidor de gerenciamento a gerentes intermediários. Esses atuam como gerentes para monitorar e controlar os agentes que estão sob sua responsabilidade e, ao mesmo tempo, como agentes, para fornecer informações a um servidor de gerenciamento de nível hierárquico mais alto.

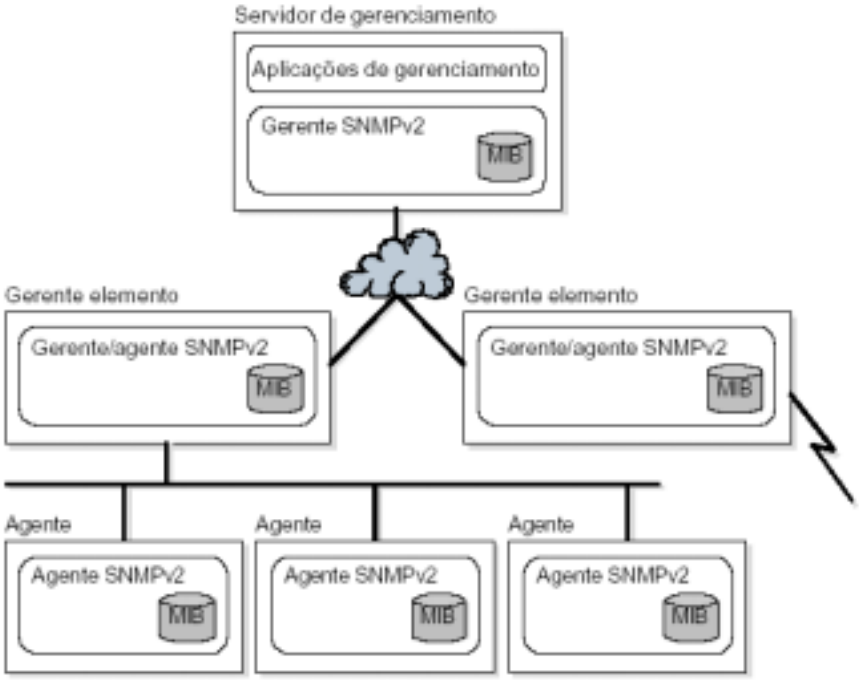

Figura 3.5 – Gerenciamento Descentralizado com SNMPv2

Para suportar a cooperação gerente-gerente, SNMPv2 possui duas novas características: o comando *Inform* e uma MIB gerente-gerente (STALLINGS, 1998a). Um gerente utiliza o comando *Inform* para enviar informações não solicitadas a outro gerente. Por exemplo, um gerente pode notificar outro quando eventos especiais ocorrem, como por exemplo, queda de um canal de comunicação ou taxa excessiva de tráfego em algum nó da rede. Essas informações são armazenadas na MIB gerente-gerente.

As notificações não solicitadas são essenciais na configuração de um ambiente de gerenciamento distribuído. Gerentes que estão em um alto nível não precisam se preocupar sempre com os detalhes de partes remotas da rede; quando um evento local requer a atenção da estação central, o gerente local pode usar o comando *Inform* para alertar o gerente central.

3.1.8 SNMPv3

A tabela 3.3 ilustra as RFCs do protocolo SNMPv3. Tais documentos definem uma metodologia para incorporar características de segurança a SNMPv1 ou SNMPv2.

RFC	Título	Data
2271	An architecture for describing SNMP Management frameworks	Janeiro de 1998
2272	Message processing and dispatching for SNMP	Janeiro de 1998
2273	SNMPv3 applications	Janeiro de 1998
2274	User-based security model for SNMPv3	Janeiro de 1998
2275	View-based access control model (VACM) for SNMP	Janeiro de 1998
Internet Draft	Introduction to version 3 of the Internet network management framework	Agosto de 1998

Tabela 3.3 – RFCs do Protocolo SNMPv3

O protocolo SNMPv3 não substitui o SNMPv1 ou SNMPv2, mas define mecanismos de segurança que devem ser utilizados em conjunto com um ou outro – SNMPv2 preferencialmente (ver figura 3.6). As características de segurança definidas em SNMPv3 são: **autenticação, criptografia** e **controle de acesso** (STALLINGS, 1998b).

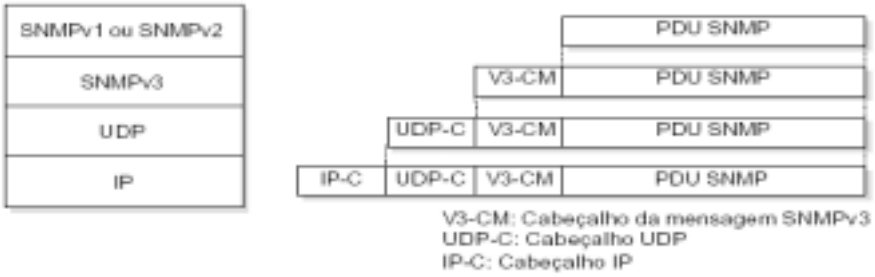

Figura 3.6 – Funcionamento do Protocolo SNMPv3

A **autenticação** permite a um agente verificar se um comando é procedente de um gerente autorizado e se seu conteúdo não foi alterado. Para obter essa funcionalidade, o gerente e o agente que desejam se comunicar, devem compartilhar uma chave secreta. O gerente utiliza essa chave para calcular um código de autenticação, que é uma função da mensagem a ser transmitida, e a anexa à mensagem a ela. Quando o agente recebe a mensagem ele usa a mesma chave para calcular o código de autenticação novamente. Se o código resultante for equivalente ao código adicionado à mensagem recebida, então o agente sabe que a mensagem é oriunda do gerente autorizado e que não foi alterada em trânsito.

O mecanismo de **criptografia** permite a gerentes e agentes embaralharem o conteúdo das mensagens trocadas entre si para evitar que outras pessoas que estejam ligadas de algum modo no caminho entre eles possam ler e interpretar as informações de gerenciamento que por ali trafegam. Essa funcionalidade também é implementada com a utilização de chave secreta; nesse caso, se gerente e agente forem configurados para usar uma chave secreta, todo o tráfego entre eles é criptografado.

Por fim, o **controle de acesso** permite a configuração dos agentes para prover níveis de acesso diferenciados a gerentes distintos. O acesso pode ser limitado de duas maneiras: comandos que o agente aceitará de um determinado gerente e porção da MIB que um dado gerente pode acessar. A política de controle de acesso a ser usada por um agente deve ser pré-configurada e consiste essencialmente em uma tabela que detalha os privilégios de acesso dos vários gerentes autorizados.

Para implementar as novas características de segurança, o SNMPv3 requer um novo formato de mensagem (para maiores detalhes, ver apêndice B). O novo formato de mensagem é consideravelmente mais longo que o formato das mensagens de SNMPv1 e SNMPv2c. Por isso, o objetivo deste trabalho é apresentar uma análise da sobrecarga (*overhead*), da utilização do protocolo SNMPv3, no gerenciamento de redes corporativas, podendo-se afirmar se o mesmo afeta ou não no desempenho das redes gerenciadas.

No apêndice B é apresentado um estudo completo sobre o modelo de segurança e as funcionalidades do protocolo SNMPv3.

3.1.9 Resumo da Evolução Cronólogica do Protocolo SNMP

A figura 3.7, mostra resumidamente a evolução cronológica do protocolo SNMP.

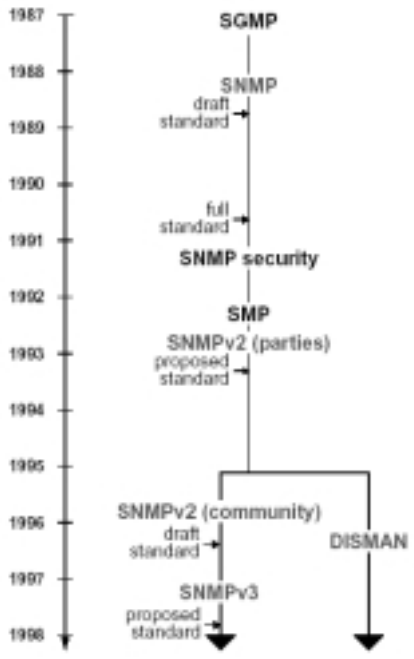

Figura 3.7 – Evolução Cronológica do Protocolo SNMP

Portanto, com esse estudo da evolução da arquitetura de gerenciamento Internet/SNMP e os conceitos básicos de SNMP, modelo de gerenciamento, definição

do que é uma MIB e as características das diferentes versões do protocolo – SNMPv1, SNMPv2 e SNMPv3, é possível afirmar que SNMPv2 foi uma melhoria significativa em relação ao SNMPv1, mantendo as suas características essenciais de fácil compreensão e implementação. A versão 2 fornece melhores suportes para uma arquitetura de gerenciamento de rede descentralizada, melhora o desempenho e fornece mais algumas “ferramentas” do interesse dos programadores. Já o SNMPv3 não substitui o SNMPv1 ou SNMPv2 e o propósito primário para o seu desenvolvimento foi: corrigir a falha mais óbvia das versões anteriores, a falta de segurança. No próximo capítulo é apresentado um estudo dos objetos da MIB padrão (MIB 1213) e características de desempenho inerentes ao protocolo SNMP.

CAPÍTULO IV

4. OBJETOS DA MIB PADRÃO (MIB 1213) E CARACTERÍSTICAS DE DESEMPENHO DO PROTOCOLO SNMP

4.1 O que é a MIB 1213

Uma MIB (MILLER, 1999) é uma estrutura que contém as variáveis necessárias para monitorar, gerenciar ou administrar os componentes em redes Internet.

O RFC 1158 propôs a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

A MIB 1213 é uma Base de Informações padrão que contém um conjunto de objetos bem definidos, conhecidos e aceitos pelos grupos padrão da Internet. Duas versões da MIB padrão estão disponíveis hoje em dia, e são chamadas MIB-I (RFC 1156/1066) e MIB-II (RFC 1213/1158).

Abaixo da subárvore MIB II estão os objetos usados para obter informações específicas dos dispositivos da rede. A **MIB II** fornece informações sobre o

equipamento gerenciado como por exemplo, o estado da *interface*, informações sobre os protocolos de rede, número de pacotes com erros, número de pacotes enviados e recebidos etc. Esses objetos são divididos em 11 grupos Para maiores informações sobre a MIB II ver Apêndice A.

Neste trabalho, optou-se pela utilização da MIB-II, mais especificamente a MIB 1213 devido ao fato desta MIB ter sido projetada para monitorar a mídia de rede, em vez de um dispositivo especifico. Portanto, ela é útil no monitoramento do tráfego de rede como um todo.

Para implementar um agente da MIB 1213, em um dispositivo, a interface de rede no dispositivo deve ser capaz de operar no modo promíscuo em que poderá aceitar pacotes não endereçados especificamente para ela. No escopo deste trabalho, não foi necessário configurar o dispositivo de rede para o modo promíscuo, porque os testes de simulação foram realizados na interface *loopback* do dispositivo.

Dentre os grupos de objetos da MIB 1213 que foram utilizados pelas simulações destacam-se o grupo *System* e o grupo *Interfaces*, os quais são descritos nas seções seguintes. Também como complemento, apresenta-se uma descrição do grupo SNMP.

4.1.1 Grupo *System*

O grupo *System* contém informações sobre o sistema no qual se encontra a entidade gerenciada. Muitos destes objetos são usados no gerenciamento de configuração e gerenciamento de falhas.

A tabela 4.1 apresenta os objetos do grupo *System* para Gerenciamento de Configuração.

Objeto	Informação usada no gerenciamento de configuração
SysDescr	descrição do sistema
SysLocation	localização física do sistema
SysContact	pessoa responsável pelo sistema
SysName	nome do sistema

Tabela 4.1 – Objetos do Grupo System para Gerenciamento de Configuração

O *SysDescr* informa a descrição do sistema. Este dado pode ser útil tanto para gerenciar a configuração do dispositivo como para diagnosticar falhas.

Os objetos *sysLocation*, *sysContact* e *sysName* são úteis quando há necessidade de contactar com alguém para um acesso físico a um dispositivo remoto.

Na tabela 4.2 são apresentados os objetos do grupo *System* para o Gerenciamento de Falhas.

Objeto	Informação usada no gerenciamento de falhas
SysObjectID	fabricante do sistema
SysServices	qual camada de protocolo o sistema serve
SysUpTime	quanto tempo o sistema está operacional

Tabela 4.2 – Objetos do Grupo System para Gerenciamento de Falhas

O *SysServices* informa quais os níveis do modelo de referência da ISO, o dispositivo serve. Ele retorna a soma dos números de cada camada, usando, para cada camada, a fórmula 2(L-1), onde L é o número da camada. Esta informação é útil para rastrear problemas quando a funcionalidade do dispositivo é desconhecida.

4.1.2 Grupo *Interfaces*

O Grupo *Interfaces* oferece dados sobre cada interface de um dispositivo gerenciável da rede. Essas informações são úteis para o gerenciamento de falhas, de configuração, de performance, e de contabilização.

O objeto *ifTable* contém informações sobre todas as interfaces de uma entidade.

Na tabela 4.3 são apresentados os objetos para o Gerenciamento de Falhas.

Objeto	Informação usada no gerenciamento de falhas
ifAdminStatus	indica se a interface esta administrativamente up/down/test
ifOperStatus	indica o status operacional da interface (up/down/test)
ifLastChange	indica quando a interface mudou seu estado operacional

Tabela 4.3 – Objetos do Grupo Interfaces para o Gerenciamento de Falhas

A combinação dos objetos *ifAdminStatus* e *ifOperStatus* determina o status da interface. A tabela 4.4 abaixo, apresenta as possíveis combinações.

IfAdminStatus / IfOperStatus	Up(1)	Down(2)	Testing(3)
Up(1)	Operacional	N/A	N/A
Down(2)	Falha	Down	N/A
Testing(3)	N/A	N/A	em teste

N/A - não aplicável.

Tabela 4.4 – Combinação de Objetos do Grupo Interfaces para Determinar o Status da Interface

Na tabela 4.5 a seguir, são apresentados os objetos para o Gerenciamento de Configuração.

Objeto	Informação usada no gerenciamento de Configuração
ifDescr	nome da interface
ifType	tipo de interface
ifMtu	tamanho máximo do datagrama suportado pela interface
ifSpeed	largura de banda da interface
ifAdminStatus	indica se a interface está administrativamente up/down/test

Tabela 4.5 – Objetos do Grupo Interfaces para o Gerenciamento de Configuração

O objeto *ifSpeed* é um medidor da velocidade da interface em *bits* por segundo. Ele é útil quando se deseja saber a velocidade atual de uma interface que aloca banda passante de acordo com a demanda de tráfego.

O objeto *ifAdminStatus* permite que, através do comando SNMP *Set-Request*, se configure remotamente a interface para *on/off*.

A tabela 4.6 apresenta os objetos para o Gerenciamento de Performance.

Objeto	Informação usada no gerenciamento de Performance
ifInDiscards	taxa de descartes de entrada
ifOutDiscards	taxa de descartes de saída
ifInErrors	taxa de erros de entrada
ifOutErros	taxa de erros de saída
ifInOctets	taxa de bytes recebidos
ifOutOctets	taxa de bytes enviados
ifInUcastPkts	taxa de pacotes unicast recebidos
ifOutUcastPkts	taxa de pacotes unicast enviados
ifInNUcastPkts	taxa de pacotes no-unicast recebidos
ifOutNucastPkts	taxa de pacotes no-unicast enviados
ifInUnknownProtos	taxa de pacotes de protocolos desconhecidos recebidos
ifOutQLen	total de pacotes na fila de saída

Tabela 4.6 – Objetos do Grupo Interfaces para o Gerenciamento de Performance

Com os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts*, *ifInErrors*, *ifOutErros* , pode-se calcular as porcentagens de erro de entrada/saída.

- porcentagem de erro de entrada = *ifInErrors*/(*ifInUcastPkts* + *ifInNUcastPkts*)
- porcentagem de erro de saída = *ifOutErrors*/(*ifOutUcastPkts* + *ifOutNUcastPkts*)

Da mesma forma, com os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts*, *ifInDiscards*, *ifOutDiscards* , pode-se calcular as porcentagens de descartes de entrada/saída.

O objeto *ifInUnknownProtos* informa o número de descartes realizados devido ao recebimento de pacotes de protocolo desconhecido. Portanto, não há a detecção de nenhum problema, caso o valor dos objetos *ifInUnknownProtos* e *ifInDiscards* estiverem crescendo proporcionalmente.

Com os objetos *ifInOctets* e *ifOutOctets* pode-se calcular a taxa de utilização de uma interface. Para isso, primeiro calcula-se o total de *bytes* recebidos e enviados em um intervalo de tempo entre x e y:

- total de *bytes* = (*ifInOctets*_y - *ifInOctets*_x)+(*ifOutOctets*_y - *ifOutOctets*_x)

Depois calcula-se o total de *bytes* e *bits* por segundo:

- total de *bytes* por segundo = total de *bytes*/(y - x)
- total de *bits* por segundo = total de *bytes* por segundo * 8

E, finalmente, a taxa de utilização:

- taxa de utilização = (total de *bits* por segundo)/*ifSpeed*

Detalhes das variáveis de MIB II usados na fórmula acima:

.1.3.6.1.2.1.2.2.1.10
ifInOctets OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Counter
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "The total number of octets received on the interface, including framing characters."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 10 }

.1.3.6.1.2.1.2.2.1.16
ifOutOctets OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Counter
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "The total number of octets transmitted out of the interface, including framing characters."
::= { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 16 }

.1.3.6.1.2.1.2.2.1.5
ifSpeed OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Gauge
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth."
::= { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 5 }

O objeto *ifOutQLen* indica se o dispositivo está tendo problemas em enviar dados para fora. Seu valor aumenta de acordo com o aumento do número de pacotes esperando para deixar a interface.

Os objetos *ifOutOctets* e *ifOutDiscards* juntos podem sinalizar um congestionamento na rede. Isto ocorre se, caso houver um aumento no valor do *ifOutDiscards* devido ao descarte de muitos pacotes que tentam deixar a interface, e uma diminuição do número total de *bytes* de saída, indicado pelo objeto *ifOutOctets*.

A tabela 4.7 apresenta os objetos utilizados para o Gerenciamento de Contabilização.

Objeto	**Informação utilizada no gerenciamento de contabilização**
IfInOctets	taxa de bytes recebidos
IfOutOctets	taxa de bytes enviados
IfInUcastPkts	taxa de pacotes unicast recebidos
IfOutUcastPkts	taxa de pacotes unicast enviados
IfInNUcastPkts	taxa de pacotes no-unicast recebidos
IfOutNucastPkts	taxa de pacotes no-unicast enviados

Tabela 4.7 – Objetos do Grupo Interfaces para o Gerenciamento de Contabilização

Com os objetos *ifInOctets* e *ifOutOctets*, uma aplicação de gerenciamento de contabilização pode determinar o número de *bytes* enviados e recebidos em uma interface. Se a unidade de contabilização utilizada for pacotes ao invés de *bytes*, são utilizados os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts* para calcular o número de pacotes recebidos e enviados.

4.1.3 O Grupo SNMP

Os objetos do grupo SNMP podem ser aplicados em todas as cinco áreas de gerenciamento. Aplicações de gerenciamento de falhas observando os problemas SNMP podem achar útil conhecer o número de erros SNMP e sua freqüência, enquanto aplicações de gerenciamento performance podem calcular a taxa de pacotes SNMP entrando e deixando a entidade. Já aplicações de gerenciamento de contabilização

podem usar os objetos SNMP para encontrar o número de pacotes SNMP enviados ou recebidos pela entidade. ,E por fim, alguns objetos do grupo SNMP podem ajudar no gerenciamento de configuração e segurança .

Na tabela 4.8 são apresentados os objetos para Gerenciamento de Falhas.

Objeto	**Informação usada para gerenciamento de falhas**
SnmpInASNParseErrs	total de mensagens recebidas com erros ASN
SnmpInTooBigs	total de mensagens recebidas com erro "too big"
SnmpInNoSuchNames	total de mensagens recebidas com erro "noSuchName"
SnmpInBadValues	total de mensagens recebidas com erro "badValue"
SnmpInReadOnlys	total de mensagens recebidas com erro "readOnly"
SnmpInGenErrs	total de mensagens recebidas com erro "genErr"
SnmpOutTooBigs	total de mensagens enviadas com erro "too big"
SnmpOutNoSuchNames	total de mensagens enviadas com erro "noSuchName"
SnmpOutBadValues	total de mensagens enviadas com erro "badValue"
SnmpOutGenErrs	total de mensagens enviadas com erro "genErr"

Tabela 4.8 – Objetos do Grupo SNMP para Gerenciamento de Falhas

Os objetos listados informam erros referentes a mensagens SNMP. Esses erros não indicam erros na rede em si, mas pode informar que a entidade não está manipulando os pacotes SNMP apropriadamente. O número e tipos de erros também podem indicar que a entidade está recebendo pacotes SNMP com erros dos dispositivos da rede. A solução para esses erros geralmente está na configuração do gerente ou agente SNMP. Se a reconfiguração não diminuir o número de erros, o problema provavelmente residirá na implementação do gerente ou agente SNMP.

A tabela 4.9 apresenta os objetos para Gerenciamento de Performance.

Objeto	Informação usada para gerenciamento de performance
SnmpInPkts	taxa de pacotes SNMP recebidos
SnmpOutPkts	taxa de pacotes SNMP enviados
SnmpInTotalReqVars	taxa de Get/Get-Next-Requests recebidas
SnmpInTotalSetVars	taxa de Set-Requests recebidas
SnmpInGetRequests	taxa de Get-Requests recebidas
SnmpInGetNexts	taxa de Get-Next-Requests recebidas
SnmpInSetRequests	taxa de Set-Requests recebidas
SnmpInGetResponses	taxa de Get-Responses recebidas
SnmpInTraps	taxa de Traps recebidas
SnmpOutGetRequests	taxa de Get-Requests enviadas
SnmpOutGetNexts	taxa de Get-Next-Requests enviadas
SnmpOutSetRequests	taxa de Set-Requests enviadas
SnmpOutGetResponses	taxa de Get-Responses enviadas
SnmpOutTraps	taxa de Traps enviadas

Tabela 4.9 – Objetos do Grupo SNMP para Gerenciamento de Performance

Como qualquer outra atividade da entidade, o SNMP pode afetar a performance do sistema. Se deseja-se conhecer a porcentagem de recursos que uma entidade está usando para manipular o SNMP, pode-se calcular a taxa de pacotes SNMP recebidos ou enviados, usando os objetos snmpInPkts e snmpOutPkts.

Os demais objetos listados na tabela acima permite que se conheça os tipos de pacotes SNMP que a entidade está manipulando.

Na tabela 4.10 são apresentados os objetos para Gerenciamento de Contabilização.

Objeto	Informação usada para gerenciamento de contabilização
SnmpInPkts	taxa de pacotes SNMP recebidos
SnmpOutPkts	taxa de pacotes SNMP enviados
SnmpInTraps	taxa de traps recebidas
SnmpOutTraps	taxa de traps enviadas

Tabela 4.10 – Objetos do Grupo SNMP para Gerenciamento de Contabilização

A tabela 4.11 apresenta os objetos para Gerenciamento de Segurança.

Objeto	Informação para gerenciamento de segurança
SnmpInBadCommunityNames	total de pacotes com uma community string incorreta
SnmpInBadCommunityUses	total de pacotes com community string que não permite a operação requisitada

Tabela 4.11 – Objetos do Grupo SNMP para Gerenciamento de Segurança

O objeto *snmpInBadCommunityNames* conta o número de vezes que um usuário ou aplicação, na tentativa de comunicar-se com o SNMP de uma entidade, não informou a *community string* correta.

Na tabela 4.12 é apresentado o único objeto para Gerenciamento de Configuração.

Objeto	Informação usada para gerenciamento de configuração
SnmpEnableAuthenTraps	indica se o agente SNMP pode enviar traps

Tabela 4.12 – Objetos do Grupo SNMP para Gerenciamento de Configuração

No contexto deste trabalho, para a análise da sobrecarga do protocolo SNMPv3, foram utilizados os objetos *ifInOctets* e *ifOutOctets* para a obtenção do número de *bytes* recebidos e o número de *bytes* enviados, respectivamente, quando da execução das operações SNMP, conforme descrito no estudo de caso que é apresentado no capítulo V.

No Apêndice C encontra-se a descrição de todos os objetos pertencentes aos grupos da MIB 1213, classificados por área funcional.

4.2 Desempenho dos Sistemas Finais

Experimentos realizados para verificar o desempenho dos sistemas finais são de suma importância para que se possa ter informações sobre a capacidade e as limitações dos mesmos. Estas informações contribuem, por exemplo, para confirmar se um dado sistema final está apto ou em condições de participar de determinados experimentos (SIQUEIRA, 2000).

4.2.1 Vazão da Interface de *Loopback*

A interface de *loopback* é um endereço reservado em que os dados retornam para o usuário sem sair do computador. Desta forma, um pacote pode ser transmitido até a interface de rede e retornar para a aplicação que o gerou como se tivesse trafegado de um sistema final a outro (SIQUEIRA, 2000)..

4.3 Sobrecarga do protocolo SNMP

O protocolo de comunicação de uma determinada camada transmite uma informação dela mesma, junto com os dados recebidos da camada de nível superior. Este protocolo, com a informação adicionada, é chamado sobrecarga do protocolo. Tal sobrecarga é usualmente anexado ao cabeçalho e/ou no decorrer do trajeto da informação recebida da camada superior. A sobrecarga do protocolo não faz parte dos dados propriamente ditos e, desta forma, entende-se como sendo perda de largura de banda, na perspectiva do usuário (OLIVEIRA, 2001).

A transmissão eficiente é, particularmente, importante para meios de comunicação de longas distâncias, onde a largura de banda é considerada muito cara. Para ilustrar isso, podemos considerar uma linha alocada de 100 Kbps, entre o Brasil e os EUA, para uma corporação multinacional com escritórios em ambos. Se a sobrecarga do protocolo contribuir com 20% no total do tráfego, isto significa que somente 80 Kbps estarão disponíveis para o transporte de dados; o restante da largura de banda é consumida pela comunicação da própria rede (endereçamento). Se a corporação necessita de 100 Kbps de banda disponível, terá que super dimensionar a capacidade do *link* para 125 Kpbs (80% de 125 é 100). Este aumento do custo efetivo da largura de banda disponível, requer um custo adicional.

4.3.1 Objetos Gerenciados para a Medição do Tráfego SNMP

Utilizando o protocolo SNMP, o gerente pode então acessar as variáveis de uma MIB, através do envio de uma solicitação de *get*, *get-next* ou *getbulk* ao processo agente e, dependendo das variáveis, pode até modificar seu valor (utilizando a primitiva *set*). Uma variável SNMP é composta pela concatenação do identificador do objeto gerenciado com o identificador de uma instância deste objeto. Um objeto gerenciado pode ter múltiplas instâncias. Na figura 5.1 abaixo, é apresentado um exemplo de obtenção do valor de duas instâncias do objeto gerenciado *ifInOctets* (pertencente a MIB-II). O valor de uma instância do objeto gerenciado *ifInOctets* é um número inteiro representando um contador de *bytes* que é incrementado sempre que ocorre a entrada de um *byte* numa dada interface do equipamento de rede gerenciado (ou seja, se o valor da variável *ifInOctets.1* é 1000, sabe-se que chegaram 1000 *bytes* nesta interface). No exemplo apresentado, a instância 1 do objeto gerenciado *ifInOctects* (*ifInOctets.1*) fornece o contador de *bytes* que entram na interface E1 do roteador monitorado. A instância 2 do objeto *ifInOctets* (*ifInOctects.2*) fornece o contador de bytes que entram na interface S1 do roteador. Como será visto a seguir, os objetos gerenciados *ifInOctects* podem ser extremamente úteis para medir o tráfego de entrada em um *link* TCP/IP.

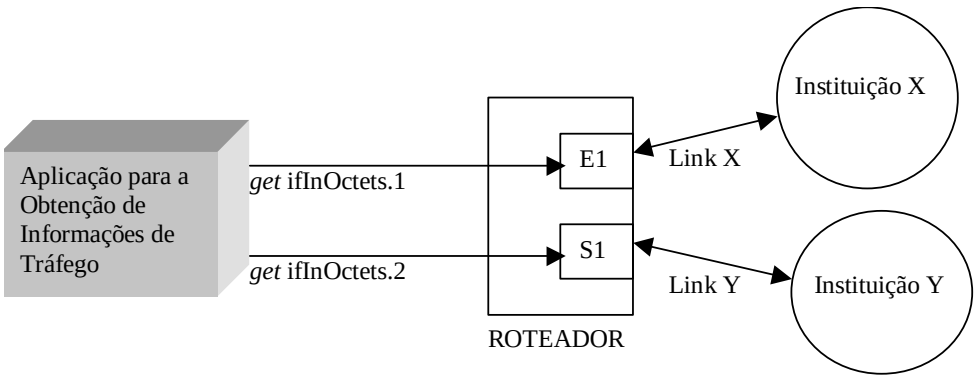

Figura 4.1 – Obtendo o Contador de Bytes que entram em uma Interface

Para o tráfego de saída, a situação é basicamente a mesma. A única variante para este caso, é que deve ser obtido o valor do objeto *ifOutOctets*, ao invés de *ifInOctets*.

4.4 Mapeamentos de Transporte SNMP

O SNMP é independente do protocolo de transporte. Existem mapeamentos já definidos para ele, mas em todos eles o processo de serialização das informações é o mesmo. Isso permite que uma estrutura de dados seja codificada como uma seqüência de *bytes* para a transmissão entre dois processos SNMP. Quando os *octetos* são recebidos eles são, novamente, convertidos na estrutura de dados com semântica idêntica.

Todas as implementações SNMP devem aceitar mensagens que são serializadas. Quando esta serialização é feita sobre o protocolo de transporte UDP a entidade SNMP gerente envia uma mensagem SNMP como um único datagrama UDP para o endereço de transporte da entidade agente SNMP.

O endereço de transporte consiste de um endereço IP e uma porta UDP. Todos os agentes SNMP "escutam" a porta UDP 161. Se a mensagem contém um *trap*, o processo receptor "escuta" na porta 162.

A especificação do protocolo SNMP informa que apenas o protocolo UDP pode ser utilizado para trocar mensagens SNMP, mas nenhum protocolo deve ser considerado como a melhor solução para todas as circunstâncias.

Entretanto, no momento em que a interoperabilidade é desejada, o SNMP deve ser utilizado sobre a pilha de protocolos UDP/IP (BRONZATTI, 1993).

A figura 4.2 mostra a interface do SNMP com as camada inferiores.

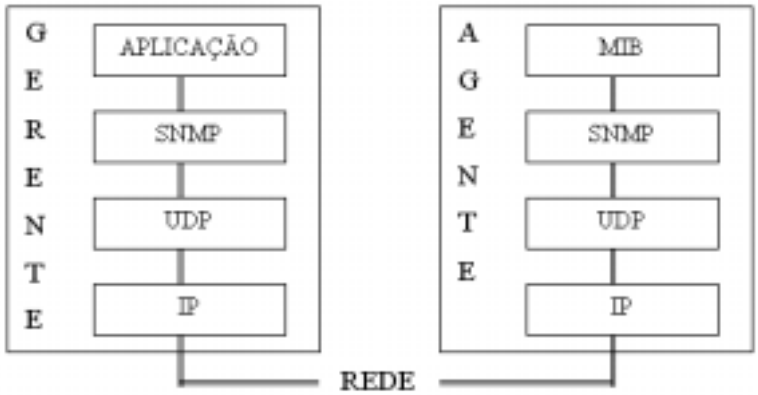

Figura 4.2 – O protocolo SNMP sobre a pilha UDP/IP

4.4.1 Protocolo SNMP sobre uma Camada de Transporte e de Rede

A utilização do protocolo SNMP sobre uma camada de transporte e de rede tem cinco razões básicas:

a) roteamento: o nível de rede provê funções de roteamento, facilitando a escolha de uma nova rota no caso de uma falha. Isso permite que o gerenciamento de uma rede continue a operar durante possíveis falhas;

b) independência do meio de transmissão: o nível de rede provê um alto grau de independência dos meios de transmissão;

c) *checksum* fim-a-fim: os *checksum* providos pelos protocolos de transporte garantem confiabilidade à transferência de dados;

d) multiplexação e demultiplexação: os serviços de transporte provêm multiplexação e demultiplexação. Estes serviços facilitam o relacionamento muitos-para-muitos que o SNMP permite;

e) fragmentação e agrupamento: o IP permite aos pacotes SNMP transitarem pelo meio, em diferentes tamanhos. Isso está relacionado à independência do meio. Esta capacidade não está disponível para enlace de dados específicos. A fragmentação e agrupamento reduzem a

robustez do gerenciamento da rede uma vez que, se qualquer fragmento da mensagem for perdido, a operação falhará. Desta forma, enquanto a rede opera normalmente a fragmentação e agrupamento não são problemas, mas na possibilidade de acontecer o contrário, é recomendável que sejam utilizados pacotes pequenos para prevenir que sejam fragmentados.

4.4.2 Vantagens do Protocolo SNMP sobre um Serviço de Transporte Sem Conexão

Os recursos necessários para um objeto gerenciado suportar UDP/IP são mínimos. Os recursos de CPU são apenas necessários quando há transmissão ou recepção de um pacote.

O maior e único recurso necessário para o UDP/IP é o cálculo de *checksum* do UDP, que é muito pequeno se comparado com o exigido para codificação e decodificação ASN.1, reconhecimento de um identificador de objeto e assim por diante.

A sobrecarga de rede, ao se utilizar o UDP/IP, é relativamente pequeno. Um cabeçalho UDP/IP requer 28 *octetos* (não assumindo nenhuma opção IP). Desde que o UDP é sem conexão, ele não gerará nenhuma sobrecarga de tráfego dele próprio (tais como TCP *Syns* e *Acks*).

Para o SNMP o serviço de transporte sem conexão é especificado no protocolo. A primeira questão que leva a esta escolha é a necessidade que o SNMP tem em continuar operando mesmo quando houver problemas na rede. Para outras aplicações, tais como *TELNET* e *FTP*, o usuário sempre pode realizar uma tentativa em outro momento, mas em aplicações de gerência acontece justamente o contrário. São os piores momentos da rede que o protocolo de gerência deve reportar os seus problemas e gargalos.

4.4.3 Desvantagens do Protocolo SNMP Utilizar um Serviço de Transporte Orientado à Conexão

Se o SNMP utilizar um serviço de transporte orientado à conexão, ele passa a possuir a responsabilidade da transmissão confiável dos dados. Desta forma de organizar o serviço de transporte decorrem as seguintes exigências:

- manter uma conexão aberta para cada agente da rede;
- estabelecer e cancelar conexões entre pares de gerentes e agentes;
- manter um número fixo de conexões abertas e, quando outra for necessária, utilizar algum algoritmo de seleção para encerrar uma conexão de forma a liberá-la para um novo agente.

Todas estas exigências introduzem sérios inconvenientes e, assim, são indesejáveis. A primeira opção reduz a quantidade de recursos necessários para realizar uma única operação após o estabelecimento da conexão. O custo das operações é amortizado pelo número que se fizer delas durante a conexão. Por outro lado, manter a conexão aberta implica na estação gerente manter um grande número de registros de conexão. Além do mais, uma grande quantidade de tráfego adicional será gerada para manter as conexões, consumindo uma boa quantidade de recursos de rede.

A segunda exigência aumenta a quantidade de recursos necessários para realizar uma operação. Uma conexão precisa ser estabelecida, a mensagem ser enviada, a resposta ser retornada e, então, a mesma deve ser cancelada. A terceira exigência requer que a estação gerente mantenha as conexões implementando um algoritmo de seleção. Isso complica excessivamente a estação gerente. Enfim, um protocolo de transporte orientado a conexão pode introduzir aspectos indesejáveis ou desnecessários ao SNMP (ROSE, 1995).

Portanto, neste capítulo apresentou-se o conceito da MIB padrão (MIB 1213); descrição dos grupos *System*, *Interfaces* e SNMP com seus respectivos objetos,

os quais fazem parte do contexto deste trabalho e servem para a coleta de dados dos dispositivos gerenciados. Também foram apresentadas as fórmulas para calcular a taxa de utilização de uma interface utilizando-se os objetos *ifInOctets* e *ifOutOctets*. Outros tópicos abordados foram os aspectos de desempenho em sistemas finais dando-se ênfase à sobrecarga do protocolo SNMP (que é o objeto desse estudo), apresentado-se para isso os objetos gerenciados para a medição do tráfego SNMP. E, por fim, descreveu-se brevemente sobre as vantagens do protocolo SNMP sobre um serviço de transporte sem conexão e as desvantagens do protocolo SNMP utilizar um serviço de transporte orientado à conexão. Diante desse contexto, o capítulo a seguir apresenta um estudo de caso onde é feita uma análise da sobrecarga do protocolo SNMPv3 no desempenho da gerência de redes corporativas, utilizando-se para isso o método de avaliação de desempenho através de simulação, podendo-se prover, uma análise quantitativa que poderá ajudar na decisão de gerentes e administradores de redes, na utilização de soluções de gerenciamento seguro.

CAPÍTULO V

5. ESTUDO DE CASO: ANÁLISE DA SOBRECARGA DO PROTOCOLO SNMPV3 NO GERENCIAMENTO DE REDES CORPORATIVAS USANDO SIMULAÇÃO

5.1 Introdução

Ao se fazer uma análise dos trabalhos correlatos, observa-se que em (DU *et al.*, 2002), é apresentada uma implementação de SNMP em uma base de TLS/TCP e experiências realizadas para determinar se o adicional de sobrecarga é aceitável. Os resultados indicam que ambas as soluções configuradas com mensagens seguras não apresentam sobrecarga excessiva e que, por conseguinte, SNMP/TLS/TCP, parece ser uma escolha válida para a gerência de rede segura que tem a vantagem da eficiência de TCP. As comparações de SNMPv3/TLS/TCP com SNMPv3/TCP (UDP) indicam que SNMPv3/TLS/TCP e SNMPv1/TLS/TCP são mais eficientes que SNMPv3 (com USM)/UDP e SNMPv3 (com USM)/TCP, para níveis semelhantes de segurança. Porém, no momento não está claro até que ponto esta vantagem aparente é estrutural e até que ponto pode refletir diferentes graus de otimização de código. Em experimentos futuros, os autores pretendem realizar experimentos com implementações de software diferentes para verificar a generalidade dos resultados observados.

Em (HIA, 2002), é apresentado um estudo qualitativo e quantitativo do gerenciamento de redes seguras que usam o Simple Network Management Protocol (SNMP) em redes que contêm ligações de baixas transferências de dados, cuja rede deve prover: (i) segurança usando autenticação e encriptação, (ii) custo efetivo de interoperabilidade baseado em tecnologias de padrão comercial, (iii) baixa sobrecarga da rede em termos de tamanho das mensagens e a freqüência das mensagens, e (iv) baixa sobrecarga dos agentes SNMP em termos de tempo de computação. São examinadas duas situações básicas. O primeiro usa uma versão não-segura de SNMP usando Segurança de IP (IPSec). Foi examinado a utilização do SNMPv2c como o protocolo de gerenciamento não-seguro, embora a maioria dos resultados também aplicam o uso de SNMPv1. IPSec provê autenticação e privacidade. Também foi examinado o uso do SNMPv3 que inclui capacidades de segurança inerentes. Segundo os autores, a solução de SNMPv3 consome até 24% a mais da capacidade de rede do que a solução SNMPv2c sobre IPSec. Porém, a vantagem mostrada pela solução SNMPv2c sobre IPSec deteriora-se com o tamanho dos aumentos de carga útil da camada de aplicação. Muito da ineficiência da solução de SNMPv3 está devido às Regras de Codificação Básicas (BER) usadas para a codificação de aplicações de dados SNMP. Contudo, a solução SNMPv2c sobre IPSec e a solução de SNMPv3 impõem quantias semelhantes de sobrecarga computacional sobre dispositivos de rede. Adicionando os serviços de autenticação e encriptação sobre um dispositivo com SNMPv2c-habilitado, e então atualizando o dispositivo para SNMPv3 ou instalando IPSec no dispositivo, efetivamente dobra a sobrecarga imposta por operações de SNMP que são executadas sobre aquele dispositivo. A solução SNMPv2c sobre IPSec pode aliviar este problema separando o processamento de segurança SNMP em dispositivos diferentes. Para a maioria das operações de SNMP que usam a solução SNMPv2c sobre IPSec, o gateway de segurança é distinto do dispositivo de rede que é hospedeiro do agente de SNMP. Reciprocamente, a solução de SNMPv3 integra o processamento de segurança e o processamento SNMP sobre a camada de aplicação que rodam em um único dispositivo de rede.

Em (HIA, 2002a), é apresentado outro estudo semelhante ao anteriormente citado, porém, considerando o problema de como prover gerenciamento de rede seguro eficaz sobre um *backbone*.

Este trabalho, diferencia-se dos citados anteriormente devido basear-se em um estudo de caso o qual utiliza a técnica de simulação para apresentar uma análise da sobrecarga de desempenho, exclusivamente, do próprio protocolo utilizado no gerenciamento de redes, no caso, o SNMPv3. Ou seja, não foram feitas comparações de desempenho em relação a outros protocolos, como por exemplo IPSec e/ou TLS/TCP.

O método estudo de caso é caracterizado por ser um estudo intensivo onde leva-se em consideração, principalmente, a compreensão, como um todo, do assunto investigado. Todos os aspectos do caso são investigados. Quando o estudo é intensivo, podem até aparecer relações que de outra forma não seriam descobertas.

O direcionamento desse método é dado na obtenção de uma descrição e compreensão completas das relações dos fatores em cada caso, sem contar o número de casos envolvidos. Conforme o objetivo da investigação, o número de casos pode ser reduzido a um elemento *caso* ou abranger inúmeros elementos como grupos, subgrupos e outros. Às vezes, uma análise detalhada desses casos selecionados pode contribuir para a obtenção de idéias sobre possíveis relações.

Além de ser importante para detectar novas relações, o estudo de caso pode ser auxiliado pela formulação de hipóteses e com o apoio da estatística. Sua principal função é a explicação sistemática das coisas (fatos) que ocorrem no contexto do estudo e geralmente se relacionam com uma multiplicidade de variáveis. Quando assim ocorre, os dados devem ser representados sob a forma de tabelas, quadros, gráficos estatísticos e por meio de uma análise descritiva que os caracterizam.

No método de estudo de caso, não se pode prescindir da analogia e do procedimento analítico. Suas principais características auxiliares para o levantamento de dados são:

a) características que são comuns a todos os casos no grupo como um todo;

b) características que não são comuns a todos os casos, porém não são comuns em certos subgrupos; e

c) características que são únicas de determinado caso.

A partir disso, pode-se chegar a uma correlação entre semelhanças e diferenças. Contudo, esse método sempre é baseado nos objetivos específicos do estudo, levando em conta a amostragem estatística.

A principal vantagem do método do estudo de caso, está no fato de que se pode obter inferência do estudo de todos os elementos que envolvam uma entidade completa, em vez do estudo de vários aspectos selecionados. Um *estudo* é uma descrição analítica de um evento ou de uma situação. Se bem apreciado, atinge a expressão máxima, sendo de grande valia.

A análise de desempenho de redes, tem como meta a análise contínua das exigências da capacidade da rede que deverá assegurar largura de banda suficiente e estar disponível para atender os objetivos de desempenho (MLAK, 2001).

A importância da análise da sobrecarga do protocolo SNMPv3 deve-se ao fato da implementação de novas características de segurança do protocolo, o qual requer um novo formato de mensagem SNMPv3 conforme mostrado no Apêndice B (figura b.4).

Este estudo de caso, apresenta as experiências que foram realizadas através da simulação de agentes SNMP, atendendo-se com isso, os objetivos propostos neste trabalho.

5.2 As Experiências e os Resultados

O ambiente de testes das simulações é composto por 01 (uma) estação de trabalho com a ferramenta de simulação do protocolo SNMP (*AdventNet Simulation Toolkit*) instalada (laboratório virtual).

5.2.1 Ambiente de Hardware

A estação de trabalho onde foram executadas todas as simulações, apresenta as seguintes características de hardware:

Processador

Processador…….....: Intel Pentium(r) III

Velocidade do Processador….: 870 MHz

Quantidade de Processadores..: 1

Cache Primário…….....: 32 KB

Cache Secundário….....: 256 KB

Memória

Total de Memória Física….: 512 MB

Total de Memória Virtual ..: 1246 MB

Disco Rígido

Capacidade do Disco Rígido…: 18.64 GB

Espaço Livre no Disco Rígido..: 11.37 GB (61%)

Adaptardor de Rede

Tipo do Adaptador…….....: Intel(R) PRO/100+ Management Adapter

Barramento da Arquitetura…….....: PCI

Tipo de Conector…….....: Cat5, 2pr, RJ-45

Taxa de Transferência de Dados.....: 10 & 100Mbps

Memória On-board …….: 6KB cache

Suporte IEEE …….....: 802.2 & 802.3

Largura de Banda…….....: 32-bit

Mode de Transferência de Dados.....: Bus-master DMA

Driver rate…….....: 100

Topologia de Rede Standard IEEE …: 10BASE-T, 100BASE-TX

Gerenciamento de Rede……..: ACPI Support, Wake on LAN*, DMI 2.0 support, Board and network diagnostics, 3-pin auxiliary power cable, Wired for Management support

NOS Software Support……..: Microsoft Windows NT* 3.51, 4.0, MicrosoftWindows* 95, 98, 2000, Novell NetWare* 3.11, 3.12 & 4.1x Server, SunSoft Solaris*,SCO UnixWare*, OpenDesktop*, OpenServer*, Boot Socket

Sistema Operacional

Sistema Operacional..: Microsoft Windows 2000

Versão……...: 5.0.2195 Service Pack 3

5.2.2 Ambiente de Software

Para análise da sobrecarga do protocolo SNMP, foi realizada uma seqüência de testes que consistem em gerar tráfego de dados SNMP para a interface de *loopback* (127.0.0.1). O objetivo deste experimento é comparar a sobrecarga do protocolo SNMPv1, SNMPv2 e SNMPv3 bem como avaliar se a sobrecarga adicional do protocolo SNMPv3 é excessiva ou não em aplicações de gerenciamento de redes.

O tráfego e simulação dos agentes, foi gerado utilizando-se a ferramenta *AdventNet Simulation Toolkit* (ADVENTNET, 2001). É importante ressaltar que um teste de *loopback* é uma simulação de transferência de dados da aplicação até a interface de rede, conforme já descrito no capítulo anterior.

5.2.3 A Metodologia e os Objetos Gerenciados para a Medição da Sobrecarga do SNMP

As operações realizadas consistiram na obtenção de variáveis SNMP do grupo sistema da MIB 1213, bem como a geração de *traps*. Com a utilização desta métrica o trabalho mantém a homogeneidade com (DU, 2002) e (HIA, 2002) que

também avaliaram a sobrecarga no protocolo SNMP em diferentes cenários (TLS e IPSec respectivamente).

Uma variável SNMP é composta pela concatenação do identificador do objeto gerenciado com o identificador de uma instância deste objeto. Na figura 5.1 abaixo, é apresentado um exemplo genérico para a obtenção do valor de uma instância do objeto gerenciado *ifInOctets* (pertencente a MIB-II). O valor de uma instância do objeto gerenciado *ifInOctets* é um número inteiro representando um contador de *bytes* que é incrementado sempre que ocorre a entrada de um *byte* numa dada interface do equipamento de rede gerenciado (ou seja, se o valor da variável *ifInOctets.1* é 1000, sabe-se que chegaram 1000 *bytes* nesta interface). No exemplo apresentado, a instância 1 do objeto gerenciado *ifInOctects* (*ifInOctets.1*) fornece o contador de *bytes* que entram na interface de rede.

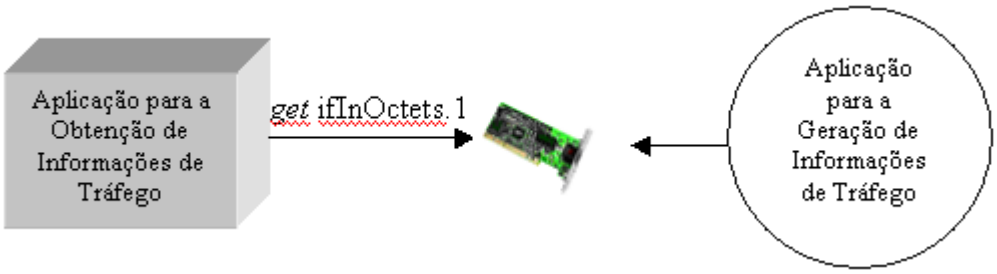

Figura 5.1 – Obtendo o Contador de Bytes que entram em uma Interface

Para o tráfego de saída, a situação é basicamente a mesma. A única variante para este caso, é que deve ser obtido o valor do objeto *ifOutOctets*, ao invés de *ifInOctets*.

No contexto deste trabalho, a ferramenta *AdventNet Simulation Toolkit* foi usada para analisar a sobrecarga do protocolo SNMP. Para esta análise foram desenvolvidas as seguintes situações:

- Análise da sobrecarga do protocolo SNMPv1;
- Análise da sobrecarga do protocolo SNMPv2 e;

- Análise da sobrecarga do protocolo SNMPv3.

Enquanto permanece aberto à incorporar outros protocolos, SNMPv3 usa atualmente o MD5 ou SHA para a geração de mensagens seguras assegurando autenticidade e integridade das mensagens, e encriptação DES-CBC para assegurar privacidade de mensagens. Estas características são usadas para prover três níveis distintos de segurança: nenhuma autenticação e sem privacidade (noAuthNoPriv), autenticação sem privacidade (authNoPriv), e autenticação com privacidade (authPriv).

Para implementar as novas características de segurança, o SNMPv3 requer um novo formato de mensagem (ver apêndice B). O novo formato de mensagem é consideravelmente mais longo que o formato das mensagens de SNMPv1 e SNMPv2c.

Neste trabalho, foi realizada uma análise de sobrecarga que assume operações de SNMPv3 que usam os três níveis de segurança acima mencionados. Todos os testes de SNMPv3 descritos neste trabalho usam MD5 para autenticação e DES para encriptação, dependendo do nível de segurança usado. A configuração dos níveis de segurança utilizados, são apresentados na tabela 5.1 como segue:

Context Name	Security Level	User Name	Auth Protocol	Priv Protocol	Auth Password	Priv Password
noAuth	noAuth, noPriv	noAuthUser	-	-	-	-
auth	Auth, noPriv	authUser	MD5	-	authUser	-
priv	Auth, Priv	privUser	MD5	CBC-DES	authUser	privUser

Tabela 5.1 – Configuração dos Níveis de Segurança do SNMPv3

Segundo (HIA, 2002), não pode ser declarado o comprimento de um campo em uma mensagem de SNMP em condições absolutas por causa das seguintes razões:

- Alguns campos, por definição, variam em comprimento. Por exemplo, a string da comunidade e nomes de usuários, ambos variam em comprimento;

- Antes de passar a mensagem à camada de transporte, as Regras de Codificação Básicas (BER) associadas com a Notação de Sintaxe Abstrata (ASN.1) são usadas para codificar cada campo em tipo, comprimento e valor, cujo comprimento é um valor dependente dessas operações.

Nas simulações executadas não foram considerados os elementos de tempo contidos no ciclo de leitura da rede, que são:

1) solicitação de leitura (*request)*

2) tempo para processar a resposta

3) resposta (*response)*

4) tempo para formular a próxima pergunta.

Também é importante salientar que as simulações realizadas não assumem enlaces[3] e nós[4] de rede, conseqüentemente não há carga de outros protocolos e não têm perda. Além disso, o tempo de processamento das mensagens (tempo necessário ao agente SNMP obter uma dada variável na MIB) e os detalhes do protocolo SNMP, como segmentação de pacotes e protocolo de transporte, não são considerados.

A seguir apresenta-se uma descrição dos procedimentos executados e os resultados das situações acima citadas.

[3] Circuito de comunicação ou via de transmissão conectando dois pontos.
[4] Ponto de conexão ou junção de uma rede.

5.2.4 Ambiente de Testes

Configurando o ambiente de simulação conforme ilustrado na figura 5.2 abaixo, foi possível simular as situações do estudo proposto, executando tráfego SNMP entre a estação A (gerente) e a estação B (agente), podendo ser colecionados dados da simulação para comparar quantitativamente a sobrecarga das diferentes versões do protocolo SNMP.

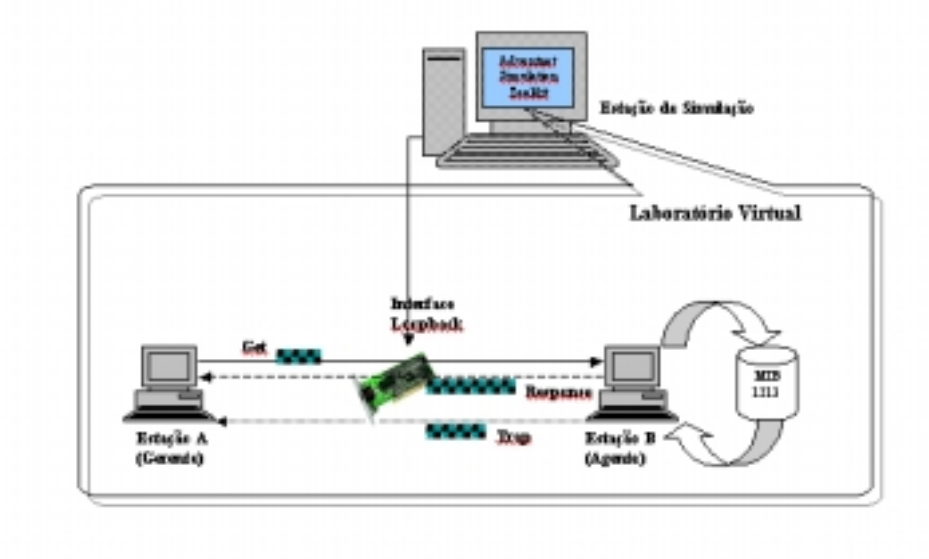

Figura 5.2 – Representação do Ambiente de Testes

Passos básicos das configurações realizadas no simulador:

a) Carregamento da MIB 1213 no *AdventNet Simulation Toolkit*;

b) Configuração do agente SNMPv1, SNMPv2 e SNMPv3, respectivamente, sendo esta a ordem da execução dos testes;

c) Configuração das mensagens (SNMP-*Get*, SNMP-*Response* e *Traps*) SNMPv1, SNMPv2 e SNMPv3 para o grupo de sistema da MIB-II;

d) Inicialização dos agentes SNMPv1, SNMPv2 e SNMPv3, no *Adventnet SNMP Agent Simulator*. A inicialização foi executada individualmente para cada agente;

e) Inicialização do *Adventnet MibBrowser*, carregamento da MIB 1213 e inicialização das mensagens SNMP-*Get*, SNMP-*Response*, para o grupo de sistema da MIB-II , para as diferentes versões do protocolo;

f) Captura de *octetos* nos nós *ifInOctets* e *ifOutOctets* (total de *octetos* recebidos e total de *octetos* enviados, respectivamente, na interface de rede) da MIB 1213. A coleta dos *octetos* de entrada e saída foi executada separadamente para cada cenário e para cada versão do protocolo.

Foram feitas várias experiências, cada uma projetada para responder as perguntas previamente citadas nos objetivos específicos deste trabalho. Para tal, foram definidos 02 (dois) cenários com o propósito de comparar a sobrecarga do protocolo SNMP em suas diferentes versões (SNMPv1 versus SNMPv2 versus SNMPV3) e a sobrecarga do protocolo SNMP versus a utilização de largura de banda, respectivamente.

As seções seguintes explicam as experiências projetadas e executadas para responder as perguntas e então documentar os resultados dessas experiências.

5.2.4.1 Cenário I – Sobrecarga do SNMPv1 versus SNMPv2 versus SNMPv3

A seguir são apresentados três experimentos da sobrecarga do protocolo SNMP nas suas diferentes versões. Para tal, considerou-se o tamanho das mensagens SNMP em octetos e os dados resultantes estão organizados em tabelas e gráficos comparativos.

5.2.4.1.1 Experimento I

A primeira experiência foi realizada com uma operação SNMP-*Get* para obter a variável sysContact.0 do grupo de sistema da MIB-II. A tabela 5.2 mostra os tamanhos (em *octetos*) da mensagem de SNMP-*Get*, os tamanhos associados às mensagens de SNMP-*Response*, e o tamanho total da troca de *Get/Response* para as diferentes versões de SNMP que usam vários esquemas de segurança.

Versão do Protocolo SNMP / Nível de Segurança	***Get***	***Response***	**Total**
SNMPv1	86	108	194
SNMPv2c	110	124	234
SNMPv3 NoAuthNoPriv	136	158	294
SNMPv3 AuthNoPriv	148	171	319
SNMPv3 AuthPriv	161	185	346

Tabela 5.2 – Tamanho das Mensagens SNMP em *Octetos* para 01 (uma) Variável da Operação SNMP-*Get*

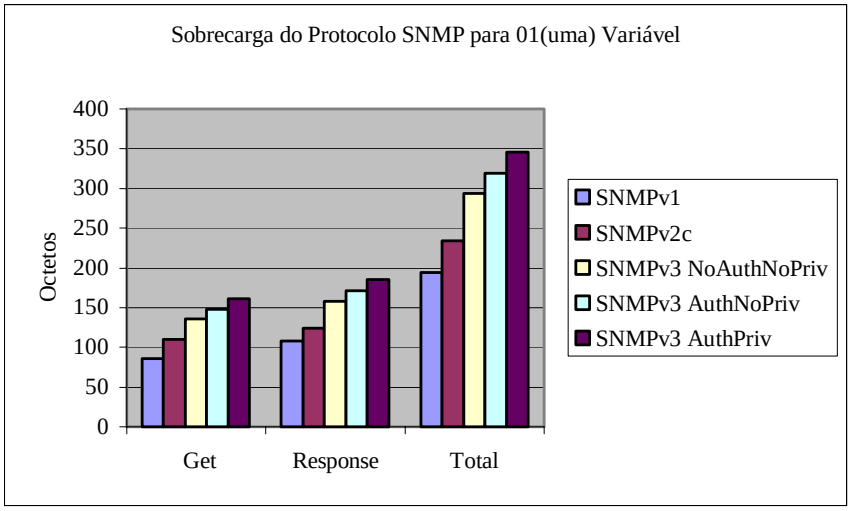

Figura 5.3 – Tamanho das mensagens SNMP em *octetos* para 01 (uma) variável das operações SNMP-*Get* e SNMP-*Response*

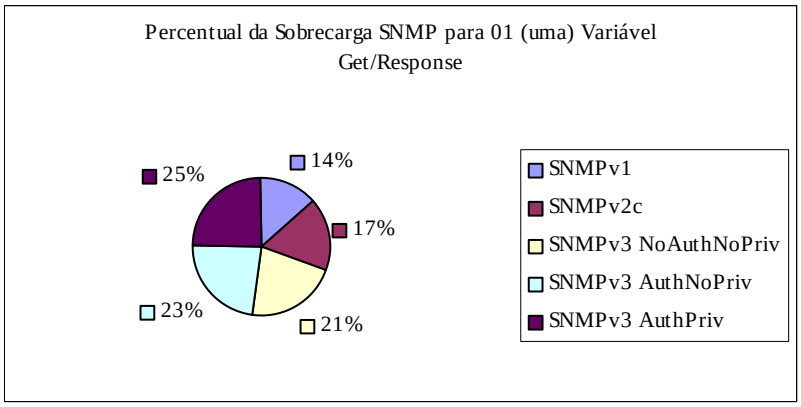

Figura 5.4 – Percentual da Sobrecarga SNMP para 01 (uma) Variável Get/Response

5.2.4.1.2 Experimento II

A segunda experiência é idêntica à primeira, porém foi usado uma operação SNMP-*Get* para adquirir 7 (sete) variáveis do grupo sistema da MIB-II. Comparando os resultados da segunda experiência com os resultados da primeira experiência, é ilustrado como o tamanho da carga útil afeta a eficiência da mensagem. A tabela 5.3 mostra os resultados desta segunda experiência.

Versão do Protocolo SNMP / Nível de Segurança	***Get***	***Response***	**Total**
SNMPv1	176	218	394
SNMPv2c	196	283	479
SNMPv3 NoAuthNoPriv	237	355	592
SNMPv3 AuthNoPriv	252	396	648
SNMPv3 AuthPriv	269	436	705

Tabela 5.3 – Tamanho das Mensagens SNMP em *Octetos* para 07 (sete) Variáveis da Operação SNMP-*Get*

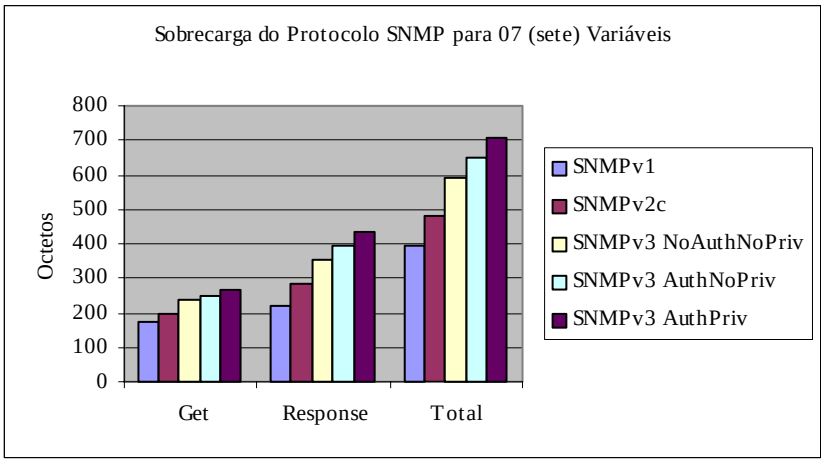

Figura 5.5 – Tamanho das mensagens SNMP em *octetos* para 07 (sete) variáveis das operações SNMP-*Get* e SNMP-*Response*

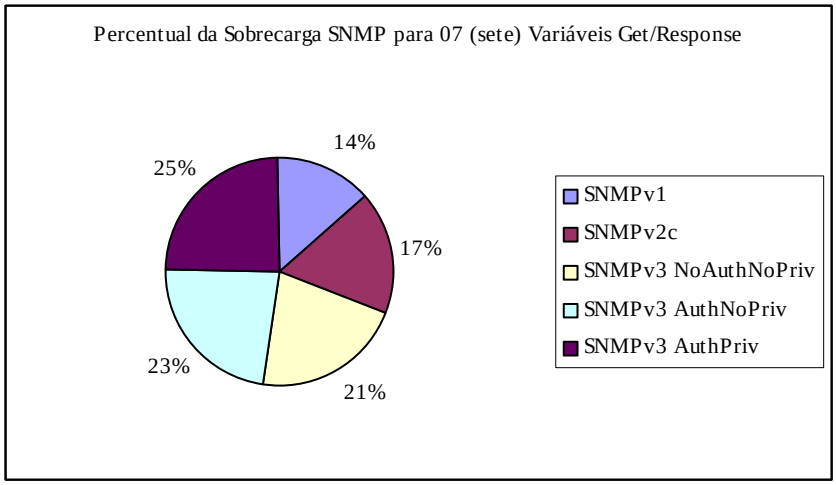

Figura 5.6 – Percentual da Sobrecarga SNMP para 07 (sete) Variáveis Get/Response

5.2.4.1.3 Experimento III

Na terceira experiência foi usado uma operação SNMP-*Trap* para enviar a variável sysContact.0 do grupo de sistema da MIB-II. Foram configurados 180 *Traps* com intervalo de 1 segundo entre cada *trap*, com intervalo de captura a cada 5 segundos. A tabela 5.4 mostra os tamanhos (em *octetos*) da mensagem de SNMP-*Trap* para as diferentes versões de SNMP que usam vários esquemas de segurança.

Versão do Protocolo SNMP / Nível de Segurança	Total
SNMPv1	88
SNMPv2c	113
SNMPv3 NoAuthNoPriv	138
SNMPv3 AuthNoPriv	151
SNMPv3 AuthPriv	164

Tabela 5.4 – Tamanho das Mensagens SNMP em *Octetos* para 01 (uma) Variável da Operação SNMP-*Trap*

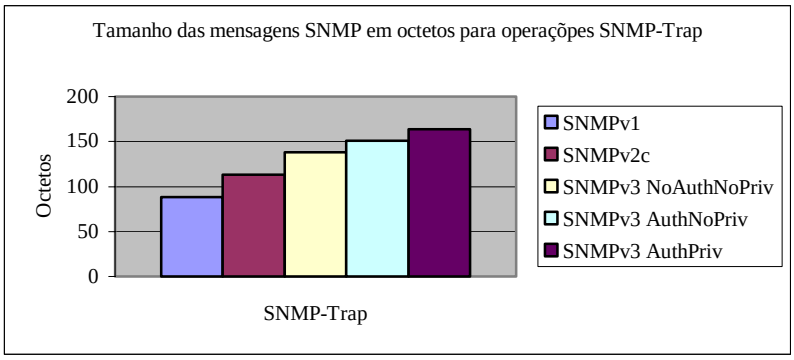

Figura 5.7 – Tamanho das mensagens SNMP em *octetos* para operações SNMP-*Trap*

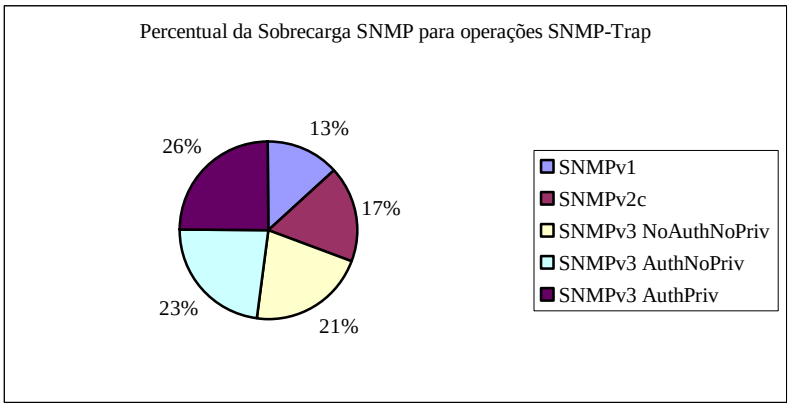

Figura 5.8 – Percentual da Sobrecarga SNMP operações SNMP-*Trap*

Analisando-se os resultados dos experimentos acima, pode-se verificar que nas simulações realizadas, a solução SNMPv3 (com os parâmetros de autenticação e privacidade - AuthPriv) consome 13% a mais da capacidade de rede que a solução SNMPv1 e 9% a mais que o SNMPv2c.

Também há indícios que, nos três cenários apresentados, a variação percentual da sobrecarga do protocolo, entre as diferentes versões, manteve-se praticamente constante e que, o SNMPv3 impõem quantias semelhantes de sobrecarga nas operações realizadas.

Outra indicação é que uma operação SNMP-*Trap* para enviar a variável sysContact.0 do grupo de sistema da MIB-II apresenta pouca variação de tamanho em relação à operação SNMP-*Get* para obter a variável sysContact.0 do grupo de sistema da MIB-II.

5.2.4.2 Cenário II – Sobrecarga do Protocolo SNMP versus Utilização de Largura de Banda

O cenário II visa demonstrar a taxa de utilização de largura de banda do protocolo SNMP em suas diferentes versões em redes de 10 Mpbs e 100 Mbps.

Com os objetos *ifInOctets* e *ifOutOctets* pode-se calcular a taxa de utilização de uma interface. Para isso, primeiro calcula-se o total de *bytes* recebidos e enviados em um intervalo de tempo entre x e y:

- total de *bytes* = (*ifInOctets*_y - *ifInOctets*_x)+(*ifOutOctets*_y - *ifOutOctets*_x)

Depois calcula-se o total de *bytes* e *bits* por segundo:

- total de *bytes* por segundo = total de *bytes*/(y - x)
- total de *bits* por segundo = total de *bytes* por segundo * 8

E, finalmente, a taxa de utilização:

- taxa de utilização = (total de *bits* por segundo)/*ifSpeed*

Com os resultados obtidos no cenário I pode-se analisar a sobrecarga do protocolo SNMP em relação à utilização de largura de banda da rede.

5.2.4.2.1 Experimento I – Taxa de Utilização em 10 Mbps e 100 Mbps para uma variável

Os valores totais de octetos foram obtidos com uma operação SNMP-*Get* para obter a variável sysContact.0 do grupo de sistema da MIB-II. A tabela 5.5 mostra o total de *bytes* por segundo (foi gerado 01 (uma) mensagem SNMP por segundo), o total de bits por segundo e a taxa de utilização da largura de banda em 10 Mbps e a taxa de utilização em 100 Mbps, para as diferentes versões de SNMP que usam vários esquemas de segurança.

Versão do Protocolo SNMP / Nível de Segurança	Total de Bytes por Segundo (01 mensagem SNMP por segundo)	Total de Bits por Segundo (Total de Bytes por segundo * 8)	% Taxa de Utilização em 10 Mbps	% Taxa de Utilização em 100 Mbps
SNMPv1	194	1552	0,1515625	0,01515625
SNMPv2c	234	1872	0,1828125	0,01828125
SNMPv3 NoAuthNoPriv	294	2352	0,2296875	0,02296875
SNMPv3 AuthNoPriv	319	2552	0,24921875	0,024921875
SNMPv3 AuthPriv	346	2768	0,2703125	0,02703125

Tabela 5.5 – Taxa de Utilização de Largura de Banda do Protocolo SNMP em 10Mbps e 100Mbps para 01 (uma) variável das operações SNMP-*Get* e SNMP-*Response*

5.2.4.2.2 Experimento II – Taxa de Utilização em 10 Mbps para sete variáveis

O segundo experimento é idêntico ao primeiro, porém, os valores totais de octetos foram obtidos com uma operação SNMP-*Get* para obter sete variáveis do grupo sistema da MIB-II. A tabela 5.6 mostra os resultados desta segunda experiência.

Versão do Protocolo SNMP / Nível de Segurança	Total de Bytes por Segundo (01 mensagem SNMP por segundo)	Total de Bits por Segundo (Total de Bytes por segundo * 8)	% Taxa de Utilização em 10 Mbps	% Taxa de Utilização em 100 Mbps
SNMPv1	394	3152	0,3078125	0,03078125
SNMPv2c	479	3832	0,37421875	0,037421875
SNMPv3 NoAuthNoPriv	592	4736	0,4625	0,04625
SNMPv3 AuthNoPriv	648	5184	0,50625	0,050625
SNMPv3 AuthPriv	705	5640	0,55078125	0,055078125

Tabela 5.6 – Taxa de Utilização de Largura de Banda do Protocolo SNMP em 10Mbps e 100Mbps para 07 (sete) variáveis das operações SNMP-*Get* e SNMP-*Response*

Analisando-se os resultados dos experimentos acima, é possível observar que nas simulações realizadas, nos dois experimentos, a taxa de utilização da largura de banda, das operações SNMP em suas diferentes versões, ficou abaixo de 1% tanto para a velocidade de 10 Mbps como para a velocidade de 100 Mbps.

Comparando os resultados obtidos no cenário I e no cenário II, observa-se que o protocolo SNMP manteve quantias semelhantes de sobrecarga em suas diferentes versões e que, tal sobrecarga não afeta na utilização global de largura de banda das redes de computadores.

5.3 Tendências

Nem todas as futuras necessidades de infra-estrutura são previsíveis, mas sim, um investimento em pró-atividade, capacidade cíclica para o planejamento do crescimento de uma plataforma, produtividade e rentabilidade.

A fibra óptica se tornou o meio de transporte de dados fundamental para as próximas gerações de redes. A confiabilidade, alcance, alto desempenho e o custo versus desempenho da fibra, fizeram para que esta fosse a tecnologia escolhida como meio de transmissão na Internet e o backone das LAN´s, MAN´S de empreendimentos em expansão.

Com isso, segundo (MLAK, 2001), a partir de 2003, sistemas de gerenciamento serão capazes de identificar e possivelmente resolver automaticamente, a causa fundamental dos problemas de desempenho dos serviços. Isto baixará o custo total de TI e proverá melhora nos níveis de serviços empresariais. Nos dias de hoje, os empreendimentos de redes estão sendo usados com capacidades, para as quais eles nunca foram projetados, um fato que até mesmo no futuro complicará o gerenciamento de redes.

As necessidades de futuras aplicações empreendedoras estão crescendo e está faltando planejamento, o que produzirá aumento de custos em termos de tempo de manutenção de rede e retardo nos projetos de mudanças depois das implementações iniciais. As redes são essenciais nos dias de hoje, como por exemplo, para o *e-commerce* e o planejamento da capacidade das redes é uma necessidade para todos os profissionais que trabalham com redes.

“ Em 2005, o maior sucesso para garantir a capacidade de processamento nos negócios em e-*commerce* e *m-commerce* será através de um processo com capacidade de planejamento baseado em *workflow*” (GARTNER GROUP, 2002).

CAPÍTULO VI

6. CONCLUSÕES

Neste trabalho, foram apresentados os principais aspectos de desempenho em redes de computadores; o funcionamento da arquitetura SNMP e sua evolução (SNMPv1, SNMPv2 e SNMPv3); características da MIB padrão (MIB 1213), dos seus grupos e respectivos objetos gerenciáveis; características de desempenho do protocolo SNMP e, por fim, um estudo de caso acerca da análise da sobrecarga quando da utilização do protocolo SNMPv3 na gerência de redes usando simulação.

Com os estudos realizados foi possível verificar que:

- Como muitos outros protocolos utilizados hoje em dia na internet, o SNMP também se debateu com problemas de segurança e privacidade. Todos estavam conscientes que era necessário uma solução contundente. Para isso, foi criado um grupo de pessoas que se reuniram para realizar uma série de propostas dando como resultado o protocolo SNMPv3, versão que certamente incrementará grandes avanços nas capacidades de autenticação, privacidade e controle de acesso, transmitindo, assim, maior confiança aos seus utilizadores;

- O SNMPv2 foi uma melhoria significativa em relação ao SNMPv1, mantendo as suas características essenciais. A versão 2 fornece melhores suportes para uma arquitetura de gerenciamento de rede descentralizada, melhora o desempenho e fornece mais algumas “ferramentas” do interesse dos programadores.

O SNMPv3 é significativamente mais complexo que SNMPv1 e SNMPv2 e o propósito primário para o seu desenvolvimento foi: corrigir a falha mais óbvia das versões anteriores, a falta de segurança. O grupo de melhoramentos introduzido por SNMPv3 pode ser dividido em três categorias:

- Uma nova arquitetura que impõe uma estrutura de entidades em SNMPv3 que efetivamente promovem extensibilidade e agilizam desenvolvimento futuro;

- Novas mensagens despachando e processando especificações que ditam como entidades SNMPv3 devem operar interiormente;

- Novas características de segurança, como autenticação, encriptação, administração de chaves de segurança, e controle de acessos.

Destas três categorias, as capacidades de segurança e os custos delas foram muito pertinentes a esta análise. Porém, as outras categorias de melhoramentos, provêem razões para adotar SNMPv3 que se igualam às características de segurança, porém não são sempre necessárias. Por exemplo, SNMPv3 incorpora um mecanismo muito flexível para configurar acesso de usuário remotamente às capacidades/funcionalidades do SNMP, um atributo que poderia ser muito útil em muitas situações.

Apesar de muitas mudanças incorporadas em SNMPv3, uma mensagem SNMPv3 ainda carrega uma PDU SNMPv2 dentro dela, porém as mensagens de SNMPv3 invocam as mesmas operações como mensagens de SNMPv2, só que em um

modo altamente seguro. As novas capacidades de segurança podem ser categorizadas em duas áreas: características criptográficas e características de controle de acesso.

As principais contribuições deste trabalho resumem-se nos seguintes itens:

- Nas simulações realizadas foi possível verificar que, a solução SNMPv3 (com os parâmetros de autenticação e privacidade - AuthPriv) acrescenta 13% a mais de sobrecarga em relação ao SNMPv1 (quase o dobro do que SNMPv1) e 9% a mais que o SNMPv2c. Porém, o excesso de sobrecarga, está associado ao acréscimo dos parâmetros de segurança e à ineficiência das Regras de Codificação Básicas da solução de SNMPv3 (HIA, 2002);

- É possível verificar que, nos três experimentos apresentados no cenário I, a variação percentual da sobrecarga do protocolo, entre as diferentes versões, manteve-se praticamente constante e que, o SNMPv3 impõem quantias semelhantes de sobrecarga nas operações realizadas;

- A taxa de utilização da largura de banda, das operações SNMP em suas diferentes versões, ficou abaixo de 1%, tanto para a velocidade de 10 Mbps como para a velocidade de 100 Mbps, ou seja, por apresentar baixa porcentagem de utilização, a sobrecarga do protocolo SNMP não influencia no desempenho global da rede.

É importante destacar que usualmente não é implementado a segurança na maioria dos dispositivos de rede com SNMP habilitado, assim, provavelmente é significante o tráfego SNMP “atravessando” as redes sem nenhuma segurança. Isto pode ou não representar um problema, dependendo das políticas de segurança adotadas pelos administradores/gerentes de redes bem como a maneira que as mesmas são empregadas. Não obstante, este assunto deve ser considerado desde que a segurança da rede pudesse ser facilmente comprometida (por exemplo, por um usuário que instala um modem sem autorização).

A análise quantitativa provida neste trabalho, poderá ajudar na decisão de gerentes e administradores de redes, na utilização de soluções de gerenciamento, que contemplam o protocolo SNMP, levando-se em conta os níveis de segurança desejados bem como o desempenho da rede.

Por fim, as melhorias na segurança do SNMP, fornecem a chave para os aspectos de segurança que faltavam no SNMP: privacidade, autenticação e controle de acesso. Existe agora um sucessor válido para o SNMPv1, e, o novo "*standard*" terá sucesso no mercado. Pode-se esperar que MIB´s adicionais venham a ser definidas no datagrama do SNMPv3 para expandir o seu campo de suporte para diversas aplicações de gerenciamento de rede.

Sistemas de gerenciamento de redes (*NMS – Network Management Systems*) incluem os serviços tradicionais FCAPS (falha, configuração, contabilização, desempenho, segurança) em suas soluções. Diferentes áreas de redes, têm diferentes exigências do gerenciamento de desempenho. A chave para tudo isso é o estado da arte no gerenciamento de redes, mas nos dias de hoje, redes heterogêneas e multi-plataformas requerem uma base de conhecimentos e perícia para administrá-las e controlá-las.

6.1 Trabalhos Futuros

Alguns possíveis trabalhos futuros que seriam interessantes e relevantes inerentes à sobrecarga do protocolo SNMP e o desempenho em redes de computadores seriam:

- Configuração de uma rede para poder coletar e analisar dados reais e comparar quantitativamente a eficácia de cada solução em se tratando da capacidade de uso da largura de banda da rede e da sobrecarga no protocolo SNMP em suas diferentes versões;

- Realizar experimentos e/ou simulações para verificar quanto tempo de processamento é consumido por operações SNMP seguras.

O conhecimento adquirido com estas experiências proverá perspectivas de como melhorar o desempenho e desdobrar a administração e gerência seguras, de redes de computadores, utilizando-se o protocolo SNMP.

Dentre as principais dificuldades encontradas, pode-se destacar:

- Complexidade da teoria de gerência de redes, existindo muitas informações sobre o assunto, o que torna uma tarefa árdua a organização de quais tópicos são realmente relevantes para a execução de um determinado trabalho;

- Poucos trabalhos correlacionados;

- Aprendizado sobre ferramentas de gerenciamento, sendo o primeiro contato aprofundado com a ferramenta *AdventNet Simulation Toolkit*, o que exigiu boa parte do tempo para estudo de suas características, particularidades e funcionamento, bem como a observação e estudos de exemplos já existentes;

- Distância geográfica, dificultando o contato pessoal com o professor orientador para a troca de idéias bem como, não ter dedicação exclusiva para a realização da pesquisa devido às atividades profissionais.

REFERÊNCIAS BIBLIOGRÁFICAS

ADLEX, URL:http://www.adlex.com; obtido em 22 de janeiro de 2003.

ADVENTNET, **AdventNet SNMP**. URL: http://www.adventnet.com/products/; obtido em 22 de novembro de 2001.

ALCATEL, URL:http://www.alcatel.com; obtido em 24 de janeiro de 2003.

ARTOLA, Esmilda Sáenz. **Olho Vivo – Sistema Especialista para Gerência Pró-Ativa Remota.** Dissertação de Mestrado. Porto Alegre: PGCC da UFRGS, 1996.

ATESTO, URL:http://www.atesto.com; obtido em 28 de janeiro de 2003.

AWDUCHE, D.; MALCOLM, J.; AGOGBUA, J.; O´DELL, M.; MCMANUS, J. **Requirements for Traffic Engineering over MPLS**, September, 1999.

AWDUCHE, D.; REKHTER, Y. **Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnets**. IEEE Communications Magazine, March, 2001.

AWDUCHE, D. **MPLS and Traffic Engineering in IP Networks**. IEEE Communications Magazine, December, 1999.

AWDUCHE, D.; CHA.; ELWALID, A.; WIDJAJA, I.; XIAO, X. **A Framework for Internet Traffic Engineering**. (draft-ietf-tewg-framework-04.txt), April, 2001.

BIERMAN, Andy Et All. **RFC 2074 – Remote Network Monitoring MIB Protocol Identifiers**. Network Working Group. January 1997.

BLUMENTHAL, Uri Et All. **RFC 2574 – User-based Security Model (USM) for SNMPv3.** Network Working Group. April 1999.

BLUMENTHAL, Uri Et All. **RFC 2274 – User-based Security Model (USM) for SNMPv3.** Network Working Group. January 1998.

BMC, URL:http://www.bmc.com; obtido em 23 de janeiro de 2003.

BOARDMAN, Bruce. **The Survivor´s Guide to 2002.** URL:http://www.netcomputing.com; obtido em 03 de setembro de 2002.

BRADNER, S. **Benchmarking terminology for network interconnection devices.** Network Working Group, Request for Comments: 1242, July, 1991.

BRAUNSCHWEIG, Technical University. **SNMPv3 Web site**. URL: http://www.ibr.cs.tu-bs.de/projects/snmpv3; obtido em 30 de maio de 2002.

BRONZATTI, Reges Antônio. **Um modelo de Gerência de Rede Baseado no Protocolo SNMP**. Porto Alegre: UFRGS, 1993. (Dissertação de Mestrado).

CASE, Jeffrey D. Et All. **RFC 2570 – Introduction to Version 3 of the Internet-Standard Network Management Framework**. Network Working Group. April 1999.

CASE, Jeffrey D. Et All. **RFC 2572 – Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)**. Network Working Group. April 1999.

CASE, Jeffrey Et All. **RFC 2272 – Message Processing and Dispatching for SNMP.** Network Working Group. January 1998.

CASE, J.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, S. **RFC 1907 - Management Information Base for Versio 2 of The Simple Network Management Protocol (SNMPv2).**, Janeiro de 1996.

CASE, Jeffrey D. Et All. **RFC 1908 – Coexistence Between Version 1 and Version of the Internet Standard Network Management Framework**. Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1907 – Management Information Base for SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1906 – Transport Mappings for SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1905 – Protocol Operations for SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1904 – Conformance statements for SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1903 – Textual Conventions for SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1901 – Introduction to Community-based SNMPv2.** Network Working Group. January 1996.

CASE, Jeffrey D. Et All. **RFC 1157 – A Simple Network Management Protocol (SNMP).** Network Working Group. May, 1990.

CASTLEROCK, URL:http://www.castlerock.com; obtido em 29 de janeiro de 2003.

CAVALCANTE, M. D.; LIMA, T. M. R.; COSTA, V. L.; MAGALHÃES, M. F.; MENDES, R. S. **Algoritmo Adaptativo para Balanceamento de Carga para Tráfego Tipo Melhor Esforço em Redes IP/MPLS.** Anais SBRC 2002, Vol. 1, 2002.

COMNET, CACI Products Company. **COMNET III – Reference Guide.** Release 2.0, 1998.

CONCORD, URL:http://www.concord.com; obtido em 28 de janeiro de 2003.

CISCO, **Performance Management**. URL:http://www.cisco.com/warp/public/126/perfmgmt.htm; obtido em 21 de novembro de 2002.

DU, X.; SHAYMAN, M.; ROZENBLIT M. **Implementation and Performance Analysis of SNMP on a TLS/TCP Base.** URL: http://www.umiacs.umd.edu/docs/Du.pdf; obtido em 30 de setembro de 2002.

ERICSSON, URL:http://www.ericsson.com; obtido em 27 de janeiro de 2003.

EMPIRIX, URL:http://www.empirix.com; obtido em 26 de janeiro de 2003.

FRANCESCHI, Analúcia Schiaffino Morales De. **Aplicação de Desempenho para Validar a Gerência Pró-ativa de Redes.** Dissertação de Mestrado, CPGCC-UFSC, 1996.

FREITAS FILHO, Paulo José de. **A High Performance Core for OSI Management Agents: Implementation, Simulation and Performance Evaluation**. In: Symposium on Performance Evaluation of Computer and Telecommunication Systems, 1999, Chicago, IL, USA. Proceedings of the SPECTS'99. 1999. v.1.

GAMBIT, URL:http://www.gambit.com; obtido em 29 de janeiro de 2003.

GARTNER GROUP, URL: http://www4.gartner.com/Init; obtido em 20 de novembro de 2002.

GASPARY, Luciano P. **Estudo do Padrão RMON2.** Trabalho Individual n. 646. Porto Alegre: PGCC da UFRGS, 1998.

GIRISH, M. K.; ZHOU, B.; HU, J.Q. **Formulation of the Traffic engineering Problems in MPLS based IP Networks.** In: Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISC´00), Antibes-Juan les Pins, France: IEEE, Communications Society and Computer Society, 2000.

HARRINGTON, Dave Et All. **RFC 2571 – An Architecture for Describing SNMP Management Frameworks** . Network Working Group. April 1999.

HARRINGTON, Dave Et All. **RFC 2271 – An Architecture for Describing SNMP Management Frameworks**. Network Working Group. January 1998.

HIA, Herik H.; MIDKIFF, Scott F. **Deploying Secure SNMP in Low Data Rate Networks.** URL: http:// www.irean.vt.edu; obtido em 10 de outubro de 2002.

HIA, Herik H.; MIDKIFF, Scott F. **Securing SNMP Across Backbone Networks.** URL: http://fiddle.visc.vt.edu; obtido em 12 de outubro de 2002a.

IBM, URL:http://www.ibm.com; obtido em 30 de janeiro de 2003.

IESG, Internet Engineering Steering Group. **SNMP Version 3 (SNMPv3).** URL: http://www. ibr.cs.tu-bs.de/ietf/snmpv3/#Working Group; obtido em 13 de dezembro de 2001.

IETF, The Internet Engineering Task Force. **Working Groups, Internet-Drafts, RFC Pages, etc.** URL: http://www.ietf.org; obtido em 25 de setembro de 2001.

JAIN, R. **The Art of Computer Systems Performance Analisys**. EUA: John Wiley E Sons, 1991.

LUCENT, URL:http://www.lucent.com; obtido em 30 de janeiro de 2003.

KLERER, S. Mark. **The OSI Managements Architecture: An Overview.** IEEE Network. New York, v.2, n.2, p.20-29, March 1988.

KRAWCZYK, Hugo. **RFC 2104 - HMAC: Keyed-Hashing for Message Authentication.** Network Working Group. February 1997.

LEVI, David B. Et All. **RFC 2573 – SNMP Applications.** Network Working Group. April 1999.

LEVI, David B. Et All. **RFC 2273 – SNMPv3 Applications.** Network Working Group. January 1998.

LYCOS, **SNMPv3 - Everything About the Current Status**. URL:http://dir.lycos.com/Computers/Internet/Protocols/SNMP/News/; obtido em 13 de dezembro de 2001.

MCCLOGHRIE, Keith. **RFC 1909 – An Administrative Infrastructure for SNMPv2.** Network Working Group. February 1996.

MCCLOGHRIE, Keith Et All. **RFC 1902 – Structure of Management Information for SNMPv2.** Network Working Group. January 1996.

MCCLOGHRIE, Keith Et All. **RFC 1213 – Management Information Base for Network Management of TCP/IP-based Internets**. Network Working Group. March 1991.

MICROMOUSE, URL:http://www.micromouse.com; obtido em 27 de janeiro de 2003.

MILLER, Mark. **Managing Internetworks with SNMP.** Second Edition. USA: M&T Books, 1997.

MILLER, Mark. **Managing Internetworks with SNMP.** IETF Network Management Documents, M&T Books, 1999.

MLAK, Madalina. **Aspects in the Computers Network Management.** URL: http://www.csiease.ro/Catedre_files/IE_files/f_central_Prof_IE.htm; obtido em 17 de agosto de 2001.

MURPHY, Jeff. **The Network Management Web Server.** URL: http://netman.cit.buffalo.edu; obtido em 11 de Janeiro de 2001.

OLIVEIRA, Sandro Silva de. **Análise de Tráfego na Integração de Redes IP e ATM usando Simulação**, Dissertação de Mestrado, CPGCC-UFSC, 2001.

PEITER, Rui C. **Um Modelo de um Agente SNMP para Gerenciamento de Redes**. TCC-Univali – Ciência da Computação, 2000.

PERKINS, David T. **RMON – Remote Monitoring of SNMP-Managed LANs.** First Edition. USA: Prentice Hall, 1998.

RABINOVITCH, Eddie, "**Network Management Performance – Tips and Tools**" URL: http://www.uniforum.org/web/pubs/; obtido em 07 de agosto de 2002.

RAMOS, Alexandre Moraes. **Interface de Controle de Acesso para o Modelo de Gerenciamento OSI.** Dissertação de Mestrado. Departamento de Informática e Estatística - UFSC, 1994.

RMON, Methodology. **RMON Methodology**. URL: http://www.3com.com/nsc/500251.html; obtido em 06 de novembro de 2001

ROSE, Marshall, MCCLOGHRIE Keith. **How to Manage your network: using SNMP.** New Jersey: Prentice Hall, 1995.

ROSE, Marshall T. **RFC 1418 – SNMP over OSI.** Network Working Group. March 1993.

ROSE, Marshall T. Et All. **RFC 1212 – Concise MIB Definitions.** Network Working Group. March 1991.

ROSE, Marshall T. Et All. **RFC 1155 – Structure and Identification of Managemente Information TCP/IP-based Internets. Network** Working Group. May 1990.

SHURAN, Software. **IP product testing.** URL: http://www.tcpip.com/support/w_papers.htm, obtido em 12 de agosto de 2002.

SIMIER, Pierrick. **Welcome to SNMPLink.org.** URL: http://www.snmplink.org; obtido em 20 de setembro de 2001.

SNMPWORLD. **The World of SNMP and Network Management.** URL: http://www.snmpworld.com; obtido em 03 de janeiro de 2001.

SIQUEIRA, Walter Ferreira. **Multi-Protocolos sobre ATM, Interoperabilidade e Gerência – Um Estudo de Caso.** Dissertação de Mestrado, CPGCC-UFSC, 2000.

SNMPv3, Working Group. URL: http://ietf.org/html.charters/snmpv3-charter.html; obtido em 15 de julho de 2002.

SOCIETY, Internet. **RFC Editor Homepage.** URL: http://www.rfc-editor.org/; obtido em 6 de fevereiro de 2002.

SPECIALSKI, Elizabeth. **Gerência de Redes de Computadores e de Telecomunicações. Florianópolis**: UFSC, 2001.

STALLINGS, William. **SNMP, SNMPv2 and RMON: Practical Network Management.** Second Edition. USA: Addison Wesley, 1996.

STALLINGS, William. **SNMP and SNMPv2: The Infraestructure for Network Management.** IEEE Communications Magazine. New York, v.36, n.3, p.37-46, March 1998a.

STALLINGS, William. **SNMP, SNMPv2, SNMPv3, and RMON 1 and RMON2**. 3rd ed. Reading, Massachusetts: Addison Wesley Longman, 1999.

STALLINGS, William. **SNMPv3: A Security Enhancement for SNMP. IEEE Communications Survey.** New York, v.1, n.1, Fourth Quarter 1998b. URL:http://www.comsoc.org/pubs/surveys/4q98issue/pdf/Stallings.pdfobtido em 12 de abril de 2001.

STALLINGS, Willian. **SNMPv3: A Security Enhancement for SNMP.** URL: http://www.comsoc.org/livepubs/surveys/public/4q98issue/stallings.html; obtido em 03 de janeiro de 2001.

SUN, URL:http://www.sun.com; obtido em 23 de janeiro de 2003.

TANENBAUM, Andrew S. **Redes de Computadores.** Trad. da 3ª. edição Rio de Janeiro: Editora Campus, 1997.

TECHNOLOGY, Bandwith Management for Corporate Intranets. **Monitoring Intranet Traffic Flows with RMON/RMON2.** URL:

http://www.3com.com/nsc/50631.htm; obtido em 02 de agosto de 2001.

TIMES, The Simple. **Promotion of the Simple Network Management Protocol (SNMP).** URL: http://www.simple-times.org; obtido em 17 de dezembro de 2001.

TWENTE, University of. URL: http://wwwsnmp.cs.utwente.nl; obtido em 20 de dezembro de 2001.

WALDBUSSER, Steven. **RFC 2021 –Remote Network Monitoring Management Information Base Version 2 using SMIv2.** Network Working Group. January 1997.

WALDBUSSER, Steven. **RFC 1757 –Remote Network Management Information Base.** Network Working Group. February 1995.

WALDBUSSER, Steven. **RFC 1513 – Token Ring Extensions to the Remote Network Monitoring MIB.** Network Working Group. September 1993.

WALDBUSSER, Steven. **RFC 1271 – Remote Network Monitoring Management Information Base.** Network Working Group. November 1991.

WATERMAN, Richard Et All. **RFC 2613 – Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0.** Network Working Group. June 1999.

WATERS, Glenn W. **RFC 1910 – User-based Security Model for SNMPv2.** Network Working Group. February 1996.

WIJNEN, Bert Et All. **RFC 2275 – View-based Access Control Model (VACM) for SNMPv3.** Network Working Group. January 1998.

WIJNEN, Bert Et All. **RFC 2575 – View-based Access Control Model (VACM) for SNMP.** Network Working Group. April 1999.

XIAO, X.; NI, L. **Internet QoS: A Big Picture.** IEEE Network Magazine, March/April, 1999.

APÊNDICE A

7. MANAGEMENT INFORMATION BASE - MIB

7.1 Introdução

O conhecimento das estruturas de organização das MIB's (*Management Information Base* - Base de Informações Gerenciáveis), e principalmente, o conhecimento, de como utilizar estas informações, é de fundamental importância na Gerência de Redes.

Este capítulo apresenta o conceito de MIB e os dois principais padrões de MIB, a MIB da OSI e a MIB da Internet, aprofundando mais neste último, no qual serão apresentados todos os objetos gerenciados e suas possíveis utilizações. Também são apresentados o status e evolução das MIBs RMON1 e RMON2.

7.2 O que é uma MIB

Antes de definir o que é uma MIB, será introduzido o conceito de **objetos gerenciados**.

Um objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas de dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso.

Dessa forma, a MIB (*Management Information Base*) é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede, possibilitando assim, a automatização de grande parte das tarefas de gerência.

É a estrutura de dados básica de um sistema de gerenciamento. Consiste basicamente numa tabela onde se encontram os dados relevantes ao gerenciamento de um sistema. Seu formato é definido pela SMI (*Structure of Management Information*), que é descrita na linguagem ASN.1 (*Abstract Syntax Notation One*).

Informações que formam uma coleção de objetos gerenciáveis , residentes em um repositório virtual de informações. (CASE, 1996).

Segundo (PEITER, 2000), embora já existam MIBs prontas com agentes e gerentes implementados que monitoram uma grande parte de informações de interesse de uma arquitetura de rede e até sistemas, é difícil achar uma que se adapte as nossas necessidades, pois existem várias implementações SNMP, sendo difícil conseguir uma aplicação que resolva os problemas específicos de cada empresa. Por este motivo, muitas empresas acabam criando uma MIB própria que colete e armazene as informações desejadas. A MIB criada torna-se então não válida, pela dificuldade e pelo tempo dispendido para a definição, compilação e implementação.

7.2.1 Abstract Syntax Notation One – ASN.1

Para que o gerenciamento de dispositivos de diferentes fabricantes seja possível, é necessária, uma forma padronizada, para a descrição dos mesmos; por isso a linguagem ASN.1 (*Abstract Syntax Notation*), foi proposta com o objetivo de descrever esses objetos numa forma padronizada e independente de fabricante.

A tabela A.1, abaixo, apresenta os tipos primitivos de dados ASN.1, permitidos no SNMP.

Primitive Type	Meaning	Code
INTEGER	Arbitrary length integer	2
BIT STRING	A string of 0 or more bits	3
OCTET STRING	A string of 0 of more unsigned bytes	4
NULL	A place holder	5
OBJECT IDENTIFIER	An officially defined data type	6

Tabela 7.1 – Tipos Primitivos de dados ASN.1 permitidos no SNMP

7.2.1.1 ASN.1 – Declaração de Variáveis

Exemplos:

count INTEGER ::=100

Status ::= INTEGER

{up(1),down(2)}

PacketSize ::= INTEGER(0..1023)

7.2.1.2 ASN.1 – Identificação de Objetos

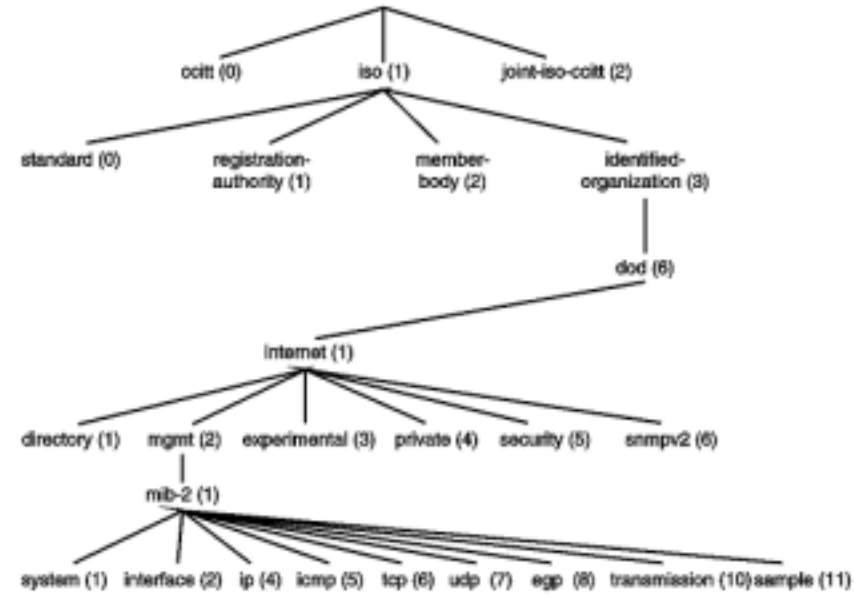

Figura 7.1 – Identificação de Objetos ASN.1

7.2.1.3 ASN.1 – Exemplos

{iso(1) identified-organization(3) dod(6) internet(1) ... }

{ 1 3 6 1 ... }

7.2.1.4 ASN.1 – Transfer Syntax

A sintaxe de transferência define como valores dos tipos definidos em ASN.1 são convertidos em *bytes* para transmissão pela rede.

A sintaxe de transferência usada pelo SNMP é chamada de **BER** (***Basic Encoding Rules***)

Os padrões de gerenciamento OSI e Internet definiram MIBs que representam os objetos necessários para a gerência de seus recursos. A seguir serão apresentadas considerações sobre a MIB da OSI e a MIB Internet, bem como as diferenças entre as MIBs desses dois padrões.

7.3 MIB da OSI

O padrão OSI define três modelos para gerência de redes: o modelo organizacional, o modelo informacional e o modelo funcional. O modelo organizacional descreve a forma pela qual a gerência pode ser distribuída entre domínios e sistemas dentro de um domínio. O modelo funcional descreve as áreas funcionais e seus relacionamentos. Já o modelo informacional provê a base para a definição de objetos gerenciados e suas relações, classes atributos, ações e nomes.

Na definição de objetos gerenciados é utilizada a orientação a objetos. Objetos com características semelhantes são agrupados em classes de objetos. Uma classe pode ser uma subclasse de outra, e a primeira herda todas as propriedades da segunda. Uma classe é definida pelos atributos da classe, pelas ações que podem ser invocadas, pelos eventos que podem ser relatados, pela subclasse a qual ela deriva e pela superclasse na qual ela está contida.

Para a definição dos objetos gerenciados deve-se considerar três hierarquias: hierarquia de herança, de nomeação e de registros usados na caracterização e identificação de objetos gerenciados.

A seguir é descrito cada uma das hierarquias mencionadas acima.

7.3.1 Hierarquia de Herança

Também denominada hierarquia de classe, tem como objetivo facilitar a modelagem dos objetos, através da utilização do paradigma da orientação a objetos.

Assim podem ser definidas classes, superclasses, subclasses. Trata-se de uma ferramenta para uma melhor definição de classes.

7.3.2 Hierarquia de Nomeação

A hierarquia de nomeação, também chamada hierarquia de *containment*, descreve a relação de "estar contido em" aplicado aos objetos. Um objeto gerenciado está contido dentro de um e somente um objeto gerenciado.

Um objeto gerenciado existe somente se o objeto que o contém existir, e dependendo da definição, um objeto só pode ser removido se aqueles que lhe pertencerem forem removidos primeiro.

7.3.3 Hierarquia de Registro

A hierarquia de registro é usada para identificar de maneira universal os objetos, independentemente das hierarquias de heranças e nomeação. Esta hierarquia é especificada segundo regras estabelecidas pela notação ASN.1 (*Abstract Syntax Notation. One*).

Assim, cada objeto é identificado por uma seqüência de números, correspondente aos nós percorridos desde a raiz, até o objeto em questão.

Esta hierarquia é também usada pelo padrão Internet, e será melhor explicada logo adiante.

7.4 MIB da Internet

O RFC 1066 apresentou a primeira versão da MIB para uso com o protocolo TCP/IP, a MIB-I. Este padrão explicou e definiu a base de informação necessária para

monitorar e controlar redes baseadas no protocolo TCP/IP. O RFC 1066 foi aceito pela IAB (*Internet Activities Board*) como padrão no RFC 1156.

O RFC 1158 propôs uma segunda MIB, a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

Uma MIB (MILLER, 1999) é uma estrutura que contém as variáveis necessárias para monitorar, gerenciar ou administrar os componentes em redes Internet. Basicamente, existem três tipos de MIBs (PEITER, 2000), pois a MIB I se tornou obsoleta quando foi acrescentada de alguns itens, tornando-se assim a MIB II.

- **MIB II –** fornece informações sobre o equipamento gerenciado como por exemplo, o estado da *interface*, informações sobre os protocolos de rede, número de pacotes com erros, etc.

- **MIBs Experimentais -** são aquelas que estão em fase de testes para que no futuro possam ser padronizadas.

- **MIBs Privadas –** são específicas dos equipamentos gerenciados, como *Hubs, Switches,* Roteadores, etc. Estas fornecem informações particulares de cada um destes equipamentos.

No padrão Internet os objetos gerenciados são definidos em uma árvore de registro, equivalente a hierarquia de registro do padrão OSI, e que será descrita com maiores detalhes a seguir.

7.4.1 A Árvore MIB II

A MIB II usa uma arquitetura de árvore (ver figura A.2), definida na ISO ASN.1, para organizar todas as suas informações. Cada parte da informação da árvore é um **nó rotulado** que contém:

- um identificador de objetos (OID): seqüência de números separados por pontos;

- uma pequena descrição textual: descrição do nó rotulado

Exemplo:

directory(1)

identificador de objetos: 1.3.6.1.1

descrição textual: {internet 1}

Um **nó rotulado** pode ter subárvores contendo outros **nós rotulados**. Caso não tenha subárvores, ou nós folhas, ele conterá um valor e será um **objeto**.

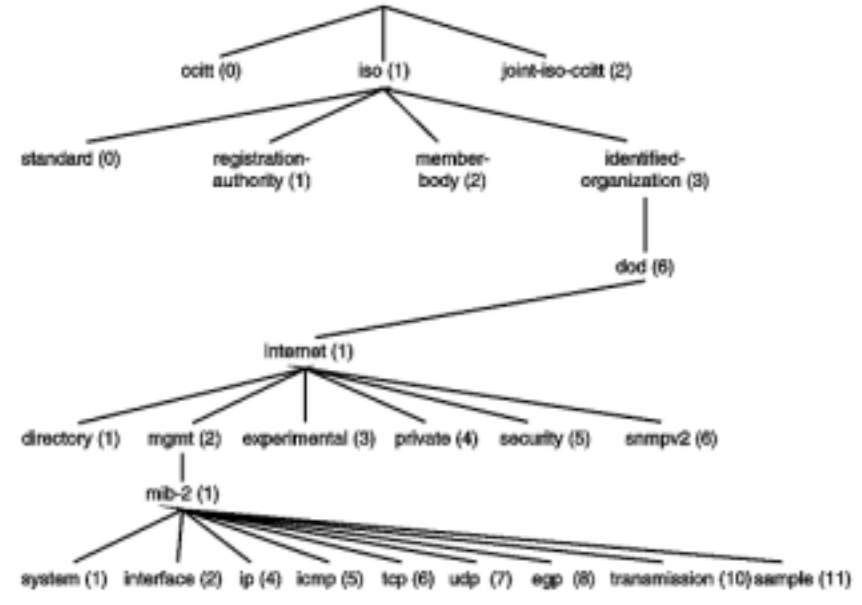

Figura 7.2 – Árvore da Hierarquia da MIB II

O nó raiz da árvore MIB não tem nome ou número, mas tem três subárvores:

- **ccitt(0)**, administrada pelo CCITT (*Consultative Commitee on International Telephony and Telegraphy*);

- **iso(1)**, administrada pela ISO (*International Organization for Standardization*)

- **joint-iso-ccitt(2**), administrada pela ISO juntamente com o CCITT.

Sob o nó **iso(1)**, estão outras subárvores, como é o caso da subárvore **org(3)**, definida pela ISO para conter outras organizações. Uma das organizações que está sob a subárvore **org(3)** é o Departamento de Defesa dos EUA (DOD), no nó **dod(6)**. A **Internet(1)** está sob o **dod(6)**, e possui quatro subárvores:

- **directory(1)**: contém informações sobre o serviço de diretórios OSI (X.500);

- **mgmt(2)**: contém informações de gerenciamento, é sob esta subárvore que está o nó da MIB II, com o identificador de objeto 1.3.6.1.2.1 ou { mgmt 1 }.

- **experimental(3)**: contém os objetos que ainda estão sendo pesquisados pela IAB (*Internet Activities Board*).

- **private(4)**: contém objetos definidos por outras organizações.

Abaixo da subárvore MIB II estão os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos são divididos em 11 grupos, que são apresentados na tabela A.2, a seguir:

Grupos	No. Objetos	Informações
system(1)	07	Sistema de operação dos dispositivos da rede. Nome, locação e descrição do equipamento
interfaces(2)	23	Interfaces da rede com o meio físico e medição do tráfego delas.
Address translation(3)	3	Mapeamento de endereços IP em endereços físicos
ip(4)	42	Protocolo IP. Estatísticas de pacotes IP
icmp(5)	26	Protocolo ICMP. Estatísticas sobre mensagem ICMP recebidas.
Tcp(6)	19	Protocolo TCP. Algoritmos TCP, parâmetros e estatísticas.
Udp(7)	6	Protocolo UDP. Estatísticas de tráfego UDP.
Egp(8)	20	Protocolo EGP (Exterior Gateway Protocol). Estatísticas de tráfego.
Cmot(9)	-	Protocolo CMOT
transmission(10)	0	Meios de transmissões.
Snmp(11)	29	Protocolo SNMP. Estatísticas de tráfego SNMP.

Tabela 7.2 – Grupos da MIB II

Cada objeto contido nos grupos apresentados na tabela acima é descrito no RFC 1213. A descrição dos objetos é dividida em cinco partes: o nome do objeto, a sintaxe abstrata do objeto, a descrição textual do significado do objeto, o tipo de acesso permitido ao objeto (*read-only, read-write, write-only* ou não acessível), e o estado do objeto (obrigatório, opcional, obsoleto).

O exemplo abaixo apresenta a descrição do objeto *sysDescr*{Internet 1}.

OBJECT

sysDescr{Internet 1}

Syntax:

DisplayString(SIZE(0..255))

Definition: descrição textual da entidade. Nome completo, versão, ...

Access: read-only

7.4.2 Exemplo de Informações Coletadas por uma MIB

No exemplo fictício que segue, os objetos usados para serem gerenciados e armazenados em uma MIB foram divididos da seguinte forma:

Computador: são informações a respeito do computador. Os atributos desta categoria são:

- *HostName*: nome que identifica o computador na rede;
- *Kernel*: versão do kernel (núcleo do sistema operacional).

CPU: são informações que identificam o processador usado no computador e sua utilização. Os atributos desta categoria são:

- Processador: tipo de processador (ex.: Pentium);
- Velocidade: velocidade em Mhz do processador;
- Fabricante: nome do fabricante do processador (ex.: Intel);
- OcupaUsuário: percentual de CPU ocupada pelo usuário;
- OcupadaSistema: percentual de CPU ocupada pelo sistema operacional;
- Ociosa: percentual de CPU livre para uso.

Memória: são informações que mostram a utilização da memória RAM (*Random Access Memory)* e memória de SWAP. Os atributos são:

- Total: quantidade de memória RAM total (em Kbytes);
- Usada: quantidade de memória RAM ocupada (em Kbytes);

- Livre: quantidade de memória RAM livre (em Kbytes);
- Swap Total: quantidade de memória SWAP total (em Kbytes);
- Swap Livre: quantidade de memória SWAP livre (em Kbytes).

Placa Rede: são informações que identificam as *interfaces* da rede disponíveis no computador. Os atributos são:

- Dispositivo: nome do dispositivo que identifica a *interface* da rede (ex.: eth0);
- Pacotes Recebidos: quantidade de pacotes recebidos pela *interface* de rede;
- Pacotes Transmitidos: quantidade de pacotes transmitidos pela *interface* de rede;
- Endereço IP: endereço IP usado pela *interface* de rede;
- IRQ: número da interrupção usada pela *interface* de rede;
- ES: endereço de entrada e saída usada pela *interface* de rede.

7.5 Comparação entre a MIB da OSI e a MIB da Internet

As MIB's da OSI e da Internet são modeladas através de técnicas de programação por objeto. Dentro deste contexto, os recursos a serem gerenciados são representados através de objetos gerenciados.

A diferença entre estas duas MIB's reside nas hierarquias usadas para representar os objetos. Na MIB da OSI são definidas três hierarquias: hierarquia de herança, hierarquia de nomeação e hierarquia de registro.

A hierarquia de herança ou de classes está relacionada às propriedades associadas a um determinado objeto. Dentro desta hierarquia diz-se que objetos da mesma classe possuem propriedades similares.

No caso da Internet não são usados os conceitos de classes de objetos e seus respectivos atributos. São definidos tipos de objetos. A definição de tipo de objetos contém cinco campos: nome textual com o respectivo identificador de objeto (OBJECT IDENTIFIER), uma sintaxe ASN.1, uma descrição do objeto, o tipo de acesso e o status.

A hierarquia de nomeação ou de *containment* é usada para identificar instâncias de objetos na MIB da OSI. Este tipo de hierarquia não é definido no caso da Internet.

Finalmente tem-se a hierarquia de registro que é especificada em ambos padrões.

7.6 Histórico, Status e Evolução das MIBs RMON1 e RMON2

O gerenciamento tradicional, SNMP e a MIB-II, ainda são amplamente utilizados em equipamentos de redes, onde agentes de *software* instalados nesses equipamentos coletam informações sobre tráfego de entrada e saída dos mesmos (como por exemplo, o número de pacotes enviados e recebidos). Com essas informações, o gerente de rede tem conhecimento sobre o volume de tráfego gerado por cada dispositivo monitorado. Para ter uma noção aproximada do tráfego total da rede, é preciso somar os valores obtidos de cada dispositivo. Porém, a distribuição cada vez maior, tanto geográfica como lógica, das redes de computadores tornou a utilização dessa abordagem complexa e ineficiente.

A alternativa que supre a deficiência citada acima, são os denominados monitores de rede ou *probes*. Geralmente um monitor opera em um segmento de rede local no modo *promíscuo*, observando cada pacote propagado nesse segmento. Os monitores podem produzir estatísticas com base nos pacotes observados (por exemplo, o

número de pacotes entregues por segundo, número de colisões, taxa de erros, etc). Podem também armazenar pacotes para submetê-los posteriormente, para algum tipo de análise, bem como é possível utilizar filtros para limitar o número de pacotes contabilizados ou capturados, baseado em seu tipo ou alguma outra característica. As estatísticas e o tráfego armazenado podem ser recuperados pelas aplicações de gerenciamento através do protocolo SNMP (PERKINS, 1998).

Segundo (STALLINGS, 1996), a maior contribuição ao conjunto de padrões SNMP foi a especificação de RMON1, MIB para monitoração remota, suplementar a MIB-II. Essa MIB foi criada para estabelecer funções e interfaces para a comunicação entre estações de gerenciamento e *probes*.

Os objetos RMON são organizados em 20 grupos. Os primeiros 10 grupos constituem a MIB RMON1, voltada ao gerenciamento das operações realizadas nos níveis físicos e de enlace em redes Ethernet. Ela compila estatísticas e informações históricas como número de colisões, erros de CRC, entre outras (MILLER, 1997). Os 10 grupos restantes constituem a MIB RMON2, proposta para viabilizar a coleta de estatísticas e informações para protocolos acima do nível de enlace. Ela permite a monitoração de padrões de uso da rede e a observação do tráfego de aplicações cliente servidor (ex.: HTTP, FTP, DNS, entre outras) e comunicações fim a fim (PERKINS, 1998) (ver figura A.3).

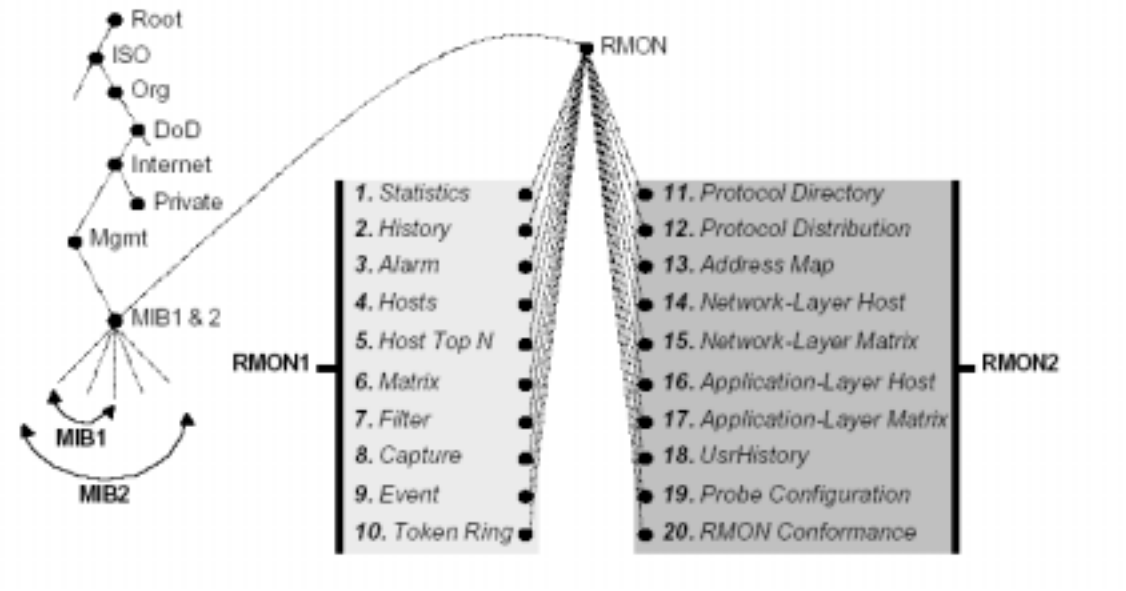

Figura 7.3 – Grupos das MIBs RMON1 e RMON2

Segundo (PERKINS, 1998), o passo inicial para a definição do padrão RMON ocorreu quando alguns fabricantes começaram a desenvolver *probes* que não possuíam interface. Esses dispositivos eram uma “caixa-preta” com uma interface de rede e outra serial. Na interface serial era conectado um terminal ASCII para configuração inicial do *probe*. Alguns destes dispositivos usavam um protocolo proprietário, outros SNMP, para recuperar informações de MIBs não padronizadas.

Em 1990, líderes do IETF (*Internet Engineering Task Force*) formaram o grupo de trabalho RMON. O primeiro documento gerado pelo grupo foi a RFC 1271, publicada em novembro de 1991. Este documento contém a definição da MIB RMON1 para redes Ethernet. Em 1993 o grupo de trabalho propôs extensões para redes *Token Ring* através da RFC 1513 (ver tabela A.3). O início dos trabalhos para a definição de RMON2 ocorreu em 1994. Paralelamente, o resultado das primeiras implementações e operações com RMON1 foi utilizado para aprimorar a versão original desta MIB. Esta iniciativa deu origem a RFC 1757, publicada em fevereiro de 1995, que substitui a RFC 1271. Os trabalhos na definição da MIB RMON2 resultaram na publicação das RFCs 2021 e 2074, em janeiro de 1997. Ainda em 1997 novos trabalhos foram iniciados: MIB RMON para redes de alta velocidade e aprimoramentos dos identificadores de protocolos.

RFC	Título	Data
1271 obsolet	Remote Network Monitoring Management Information Base	Novembro de 1998
1513	Token Ring Extensions to the Remote Network Monitoring MIB	Setembro de 1993
1757	Remote Network Monitoring Management Information Base	Fevereiro de 1995
2021	Remote Network Monitoring Management Information Base Version 2 using SMIv2	Janeiro de 1997
2074	Remote Network Monitoring MIB Protocol Identifiers	Janeiro de 1997
2613	Remote Network Monitoring MIB Extensions for Switched Networks Version 1.0	Junho de 1999
Internet Draft	Remote Network Monitoring Management Information Base for High Capacity Networks	-
Internet Draft	Remote Network Monitoring MIB Protocol Identifier Macros	-
Internet Draft	Remote Network Monitoring MIB Protocol Identifier Reference	-

Tabela 7.3 – RFCs da RMON

7.7 Objetivos da RMON

As metas definidas pelo grupo de trabalho para a definição das MIBs RMON são descritas nas RFCs 1757 e 2021 (PERKINS, 1998). São as seguintes:

- **Operação *off-line*:** nem sempre a estação de gerenciamento estará em constante contato com os *probes* RMON. Isto pode ser feito com a finalidade de reduzir os custos de comunicação (especialmente quando o enlace de comunicação é uma linha discada) ou por falha entre as estações de gerenciamento e os agentes. Por essa razão, a MIB RMON deve permitir que os agentes sejam configurados para realizar diagnósticos e coletar estatísticas continuamente, mesmo que a comunicação entre a estação de gerenciamento e os agentes não seja possível ou não seja eficiente. O agente deve tentar modificar a estação de gerenciamento sobre a ocorrência de algum evento importante. Se a comunicação de notificação falhar, a informação sobre essa ocorrência pode ser continuamente acumulada pelos

monitores e repassada às estações de gerenciamento da forma mais conveniente e eficiente possível.

- **Monitoração pró-ativa:** os recursos disponíveis nos monitores são potencialmente úteis para continuamente executar diagnósticos e manter *logs* de desempenho da rede. Como o monitor está sempre disponível desde o início de qualquer problema, ele deve poder notificar a estação de gerenciamento sobre a ocorrência de uma determinada falha e fazer um registro histórico de informações estatísticas sobre as falhas ocorridas. Esse histórico pode ser revisado pela estação de gerenciamento para, no futuro, realizar diagnósticos sobre as causas dos problemas.

- **Detecção e registro de problemas:** o monitor deve poder ser configurado para o reconhecimento de determinadas condições, realizando constantes averiguações. Quando uma dessas condições ocorre, o evento pode ser armazenado num *log* e as estações de gerenciamento podem ser notificadas de diferentes maneiras.

- **Dados com valor agregado:** o monitor de rede deve poder realizar análises específicas com os dados coletados na sub-rede onde atua, liberando a estação de gerenciamento dessa responsabilidade. Por exemplo, o *probe* pode analisar o tráfego da sub-rede para determinar que estações geram maior tráfego ou maior número de erros na sub-rede em questão (STALLINGS, 1996).

- **Múltiplos gerentes:** uma organização pode ter múltiplas estações de gerenciamento em diferentes unidades da organização e com diferentes funções, possibilitando assim uma fácil recuperação de falhas. Como os ambientes com múltiplas estações de gerenciamento são comuns, os monitores devem ter a capacidade de se relacionar com mais de uma estação de gerenciamento, usando assim seus recursos potencialmente.

7.8 Grupos da MIB RMON1

Os agentes RMON1 podem residir não apenas em dispositivos dedicados à tarefa de monitoração (*probes*), mas também em equipamentos como *switches*, roteadores, entre outros. Distribuídos em pontos remotos, eles coletam informações definidas na MIB RMON1, analisando os quadros que trafegam nos segmentos da rede. A figura A.3, lista os grupos da MIB RMON1 e ilustra onde ela se enquadra dentro dos padrões da ISO e do IETF (RMON, 1997). São eles (ARTOLA,1996; PERKINS,1998; STALLINGS,1996):

- ***Statistics*:** provê estatísticas medidas pelo monitor em cada uma de suas interfaces. As estatísticas incluem número de pacotes *unicast*, *broadcast* e *multicast*, número de colisões observadas no segmento, número de pacotes não contabilizados pelo agente, entre outras;

- ***History*:** registra amostras estatísticas periodicamente e armazena para uma posterior recuperação. Em cada amostra, são coletados dados pré-determinados. No caso de redes *Ethernet*, alguns dados coletados são: número de octetos e de pacotes observados, número de pacotes *broadcast* e *multicast*, número de erros de CRC, número de colisões, entre outros. Esta funcionalidade contribui tanto para a redução de tráfego SNMP na rede como para a diminuição do processamento realizado pela estação de gerenciamento (TECHNOLOGY, 1997), ver figura A.2;

- ***Alarm*:** periodicamente obtém amostras estatísticas de variáveis da MIB e as compara com limiares superior e inferior previamente configurados. Se o valor recuperado ultrapassa o limite superior ou fica abaixo do limite inferior, um evento é gerado. Para limitar a geração de alarmes, o grupo define *baseline*;

- ***Host*:** mantém estatísticas sobre cada *host* descoberto na rede. O termo *host* designa qualquer equipamento dotado de uma interface de rede;

- ***HostTopN*:** mantém relatórios que especificam os principais *hosts* de uma lista, ordenados por uma de suas estatísticas (por exemplo, os primeiros 20 *hosts* com maior número de pacotes enviados);

- ***Matrix*:** armazena estatísticas de tráfego e número de erros entre pares de *hosts*;

- ***Filter*:** provê um mecanismo para a estação de gerenciamento poder instruir a *probe* a observar pacotes selecionados. O critério para a seleção dos pacotes é definido no formato de um ou mais filtros conjugados;

- ***Capture*:** utilizado para configurar um esquema de armazenamento temporário para captura de pacotes, de acordo com um dos critérios de seleção definido no grupo *filter*;

- ***Event*:** controla a geração e notificação de eventos.

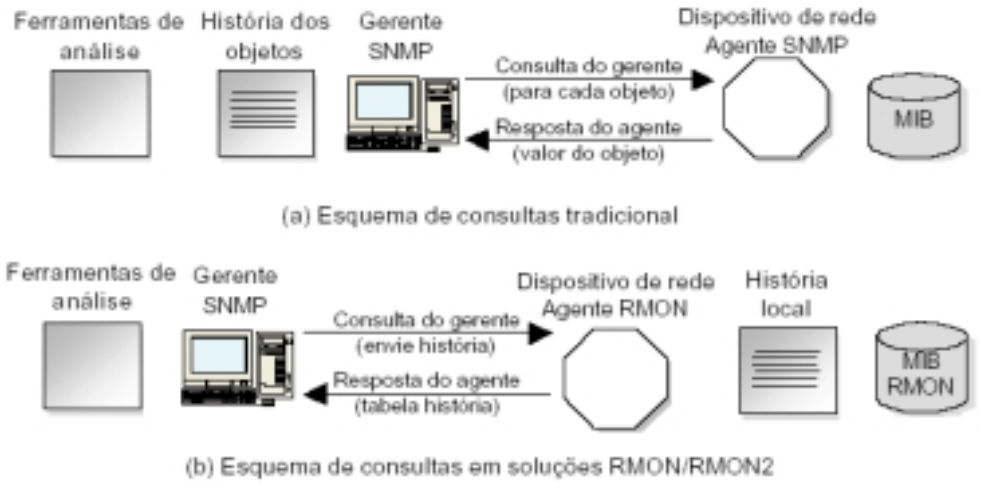

Figura 7.4 – Interações entre Agentes e Gerentes

7.9 Grupos da MIB RMON2

A MIB RMON2 é uma extensão da MIB RMON1 tradicional, criada para suportar a monitoração de protocolos de alto nível (GASPARY, 1998). Os grupos definidos por ela estão ilustrados na figura A.3. São:

- ***Protocol directory*:** repositório que indica todos os protocolos (encapsulamentos) que a *probe* é capaz de interpretar;

- ***Protocol distribution*:** agrega estatísticas sobre o volume de tráfego gerado por cada protocolo, por segmento de rede local;

- ***Address map*:** associa cada endereço de rede ao respectivo endereço MAC, armazenando-os em uma tabela. A tradução de endereços permite a geração de mapas topológicos aprimorados e a detecção de endereços IPs duplicados em uma rede;

- ***Network-layer host*:** mantém estatísticas sobre o volume de tráfego de entrada e saída das estações com base no endereço do nível de rede. Como conseqüência, o gerente pode observar além dos roteadores que interligam as sub-redes e identificar as reais estações

que estão se comunicando. Este grupo coleta estatísticas similares às do grupo *host* da MIB RMON1. A diferença é que o grupo *nlHost* faz esta coleta com base no endereço de rede e não no endereço MAC;

- ***Network-layer matrix*:** provê estatísticas sobre volume de tráfego entre pares de estações com base no endereço do nível de rede;

- ***Application-layer host*:** agrega estatísticas sobre o volume de tráfego de entrada e saída das estações com base em endereços do nível de aplicação. Consultas a este grupo permitem que o gerente trace um perfil sobre o volume de tráfego gerado ou recebido por aplicações específicas como por exemplo o *Lotus Notes*, o *Microsoft Mail*, entre outras;

- ***Application-layer matrix*:** coleciona estatísticas sobre o volume de tráfego entre pares de estações com base no endereço do nível de aplicação;

- ***User history collection*:** coleta amostras periodicamente de objetos especificados pelo usuário (gerente) e armazena as informações coletadas de acordo com parâmetros definidos pelo usuário. No padrão RMON1, esta funcionalidade é oferecida para um conjunto pré-definido de objetos;

- ***Probe configuration*:** defini parâmetros de configurações padrões para *probes* RMON. Deste modo, a estação de gerenciamento com *software* de um fabricante é capaz de configurar remotamente um *probe* de outro fabricante;

- ***Rmon conformance*:** especifica requisitos de conformidade para a MIB RMON2.

Portanto, com a MIB RMON1, é possível ter visibilidade no nível de rede, ou seja, um *probe* pode monitorar todo o tráfego da LAN a qual está ligado, podendo capturar todos os quadros relativos à sub-camada MAC do nível de enlace e ler os endereços MAC fonte e destino dos mesmos. O *probe* é capaz de prover informações detalhadas sobre o tráfego de quadros enviados e recebidos, por cada estação, em cada LAN associada. Porém, se um roteador está ligado a uma destas LANs, não há como determinar a fonte do tráfego que chega por ele, tampouco o destino de quadros que saem do segmento via este roteador (STALLINGS, 1996). Esta situação é ilustrada na figura A.5 (TECHNOLOGY, 1997).

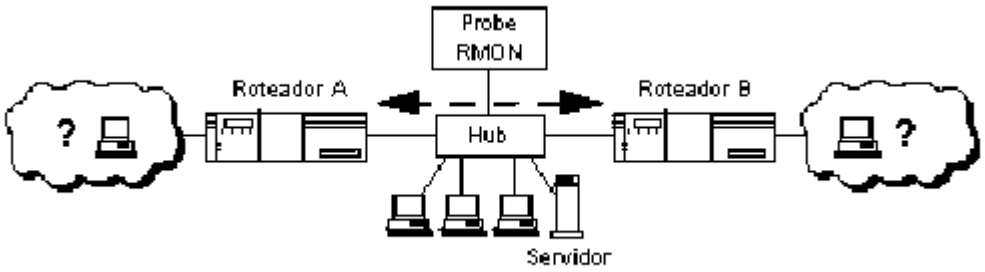

Figura 7.5 – Limitação da MIB RMON1

Com a MIB RMON2, um *probe* não está limitado à monitoração e decodificação de tráfego no nível de rede; é possível ter visibilidade no nível de aplicação (utilizando a terminologia apresentada na RFC que define o RMON2, qualquer protocolo acima do nível de rede é considerado um protocolo de aplicação), onde o *probe* é capaz de operar com protocolos localizados acima do nível de enlace, podendo, por exemplo, ler o cabeçalho do protocolo do nível de rede encapsulado no quadro, que é tipicamente o protocolo IP. Com isso é possível analisar o tráfego que passa através do roteador para determinar a fonte e destino reais dos pacotes. Com esta capacidade, o gerente de rede pode responder a uma série de questões como:

- Se há sobrecarga na LAN devido ao tráfego que chega via o roteador, que sub-redes ou estações são responsáveis por atrair este volume de tráfego?

- Se um roteador está sobrecarregado devido a um grande volume de tráfego de saída, que estações locais são responsáveis por este volume de tráfego e para que sub-redes ou estações destinos o tráfego é direcionado?

- Se existe um grande volume de dados que chega via um roteador e deixa o segmento através de outro, que redes ou estações são responsáveis por este volume de tráfego?

Obtendo as respostas acima, o gerente de rede pode ser capaz de fazer um melhor planejamento para identificar este tráfego e, melhorar o desempenho da rede. Por exemplo, ele pode verificar quais os clientes estão se comunicando com quais servidores e direcionar os sistemas em segmentos apropriados da rede, otimizando o fluxo de tráfego.

Um *probe* RMON2, por exemplo, é capaz de verificar acima do nível IP através da leitura e decodificação de protocolos de níveis superiores como TCP encapsulados no datagrama IP e, além disso, pode verificar os cabeçalhos dos protocolos do nível de aplicação. Isto permite ao gerente de rede monitorar tráfego com um alto grau de detalhamento (STALLINGS, 1996).

Uma aplicação de gerenciamento de rede que utiliza a RMON2, pode implementar a geração de gráficos apresentando porcentagens de tráfego por protocolo ou por aplicações, viabilizando a otimização da carga da rede e a manutenção do seu desempenho.

Resumindo-se, neste apêndice, descreveu-se sobre o que é uma MIB (Base de Informações Gerenciáveis) que serve para o armazenamento de dados dos objetos gerenciados bem como, das estruturas de organização destas. Também abordou-se sobre as MIBs RMON1 e RMON2 com seu *status* e evolução, as quais são utilizadas para a monitoração remota e servem para estabelecer funções e interfaces para a comunicação entre as estações de gerenciamento e os dispositivos gerenciados.

APÊNDICE B

8. MODELO DE SEGURANÇA E O PROTOCOLO SNMPV3

Nas seções seguintes, será descrito o modelo de segurança e as funcionalidades do protocolo SNMPv3 (STALLINGS, 2001).

8.1 Arquitetura SNMP

O protocolo SNMPv3 inclui os seguintes serviços:

- Autenticação
- Privacidade
- Controle de Acesso

Para oferecer estes serviços de uma forma eficiente, o protocolo SNMPv3 introduz um novo conceito chamado Principal, o qual é uma entidade na qual a maior parte dos serviços são fornecidos ou onde o processamento acontece. Um principal pode ser uma ação individual num caso particular; um conjunto de ações particulares, com um papel para cada ação; uma aplicação ou conjunto de aplicações ou a combinação

destes. Essencialmente um Principal opera a partir de uma estação de gerenciamento e envia comandos SNMP para os agentes. A identidade do Principal e a dos agentes determina as capacidades de segurança, incluindo autenticação, privacidade e controle de acesso.

A arquitetura modular como se apresenta (figuras B.2 e B.3) proporciona algumas vantagens:

- O papel da entidade SNMP é determinado por módulos que estão implementados nessa entidade;

- A estrutura modular das especificações permitem definir diferentes versões de cada módulo, o que permite definir capacidades inerentes ou alternativas do SNMP, sem que seja necessário definir um novo *standard* (SNMPv4 por exemplo), deste modo consegue-se que as diversas versões coexistam.

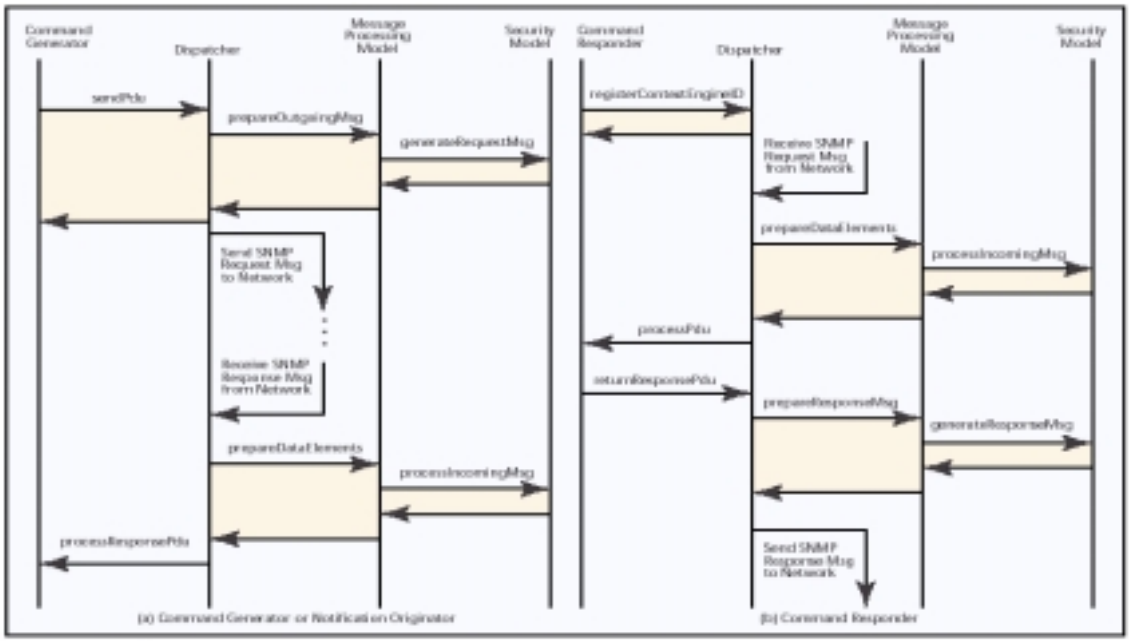

Figura 8.1 – Processamento de Entidades

8.1.1 Elementos de uma Entidade SNMP

A seguir, serão apresentados os elementos que constituem uma entidade SNMP.

8.1.1.1 Motor SNMP

Cada entidade SNMP inclui um *simple* SNMP *Engine*. Um SNMP *Engine* implementa funções para enviar/receber, autenticar e encriptar/desencriptar mensagens além de controlar o acesso aos objetos gerenciados. Estas funções são proporcionadas como serviços para uma ou mais aplicações que são configuradas com o SNMP *Engine* para assim formar a entidade SNMP.

O motor SNMP de uma estação de gerenciamento contém um ***Dispatcher***, um ***Message Processing Subsystem***, um ***Security SubSystem*** e um ***Access Control SubSytem***.

O ***Dispatcher*** é basicamente um gerenciador de tráfego de mensagens SNMP. O *Dispatcher* é responsável por:

- Aceitar as PDU´s (Protocolos de Unidades de Dados) das aplicações para que sejam transmitidas através da rede e enviar as PDU´s que chegam às aplicações;

- Passar as PDU´s que saem pelo *Message Processing Subsystem* para que sejam preparadas e passar as PDU´s que chegam pelo mesmo subsistema para que sejam extraídas;

- Enviar e receber mensagens SNMP sobre a rede e recolher estatísticas destas e do comportamento do motor SNMP no gerenciamento de objetos com o objetivo de as tornar acessíveis às entidades SNMP remotas.

O ***Message Processing Subsystem*** é responsável por preparar as mensagens para enviar e de extrair os dados da informação recebida:

- ***Security Subsystem*** - proporciona os serviços de autenticação e privacidade da mensagem. Este subsistema potencialmente possui múltiplos modelos de segurança;

- ***Access Control Subsystem*** - proporciona um conjunto de serviços de autorização que uma aplicação pode utilizar para verificação de acesso das mensagens.

8.1.1.2 Estação de Gerenciamento SNMP

A figura B.2 apresenta os vários blocos de processamento de uma entidade de uma estação de gerenciamento SNMP. No protocolo SNMPv3, uma estação de gerenciamento SNMP possui três categorias de aplicações:

- ***Command Generator*** – recebe as PDU´s SNMP *Get, Getnext, GetBulk* ou *SetRequest* e responde a um pedido que tenha sido gerado;

- ***Notification Orienter*** – recebe as PDU´s SNMP *Get, Getnext, GetBulk* ou *SetRequest* destinadas ao sistema local, e, logo opera os protocolos adequados, usando o controle de acesso e gera uma mensagem de resposta que é enviada à estação que fez o requerimento;

- ***Notification Receiver*** – monitora o sistema para uma condição ou evento particular e gera uma mensagem de *Trap*. Esta aplicação deve ter um mecanismo para determinar para onde enviar a mensagem e qual é a versão do SNMP e quais os parâmetros de segurança usar quando a mensagem é enviada.

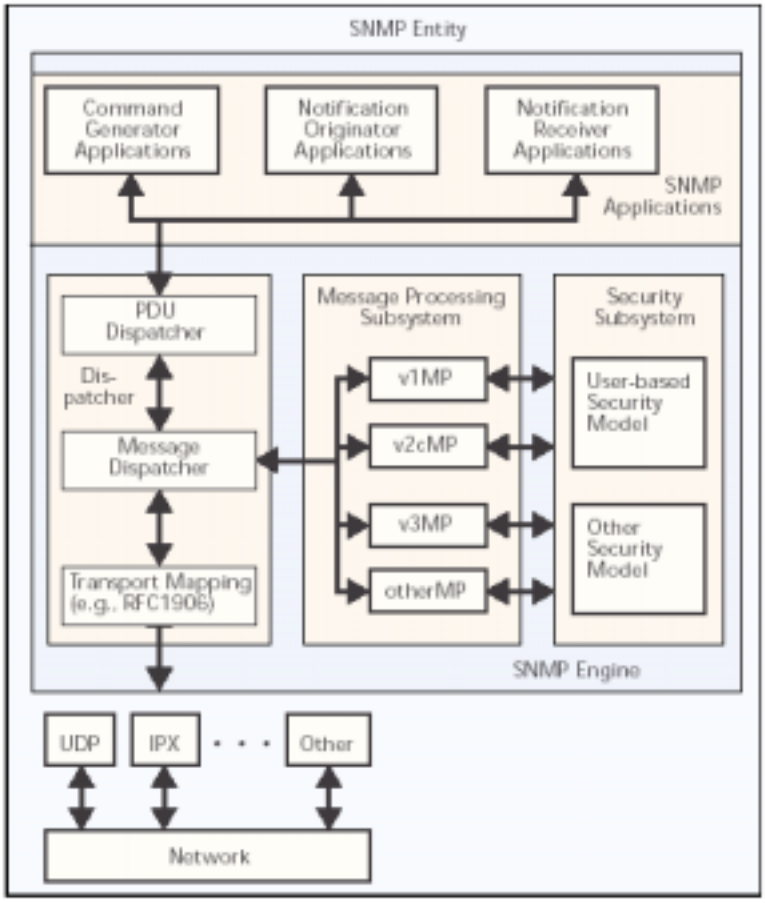

Figura 8.2 – Estação de Gerenciamento SNMP Tradicional

8.1.1.3 Agente SNMP

A figura B.3 mostra os vários blocos de processamento de um agente SNMP. Essencialmente um agente SNMP contém quatro tipos de aplicação:

- ***Command Responder*** – permite acesso à informação processada/gerenciada. Esta aplicação, no momento da chegada de um pedido, responde com a devolução e/ou ativação de objetos processados/gerenciados e em seguida atribui-lhes uma PDU de resposta;

- ***Notification Originator*** – é responsável por iniciar as mensagens assíncronas, o *trap* PDU dos protocolos SNMPv1 e SNMPv2 que é

utilizado pela aplicação. O encaminhamento das mensagens é realizado pelo *Proxy Forward Application*;

- ***Security Subsystem*** – trata da segurança e autenticação das mensagens SNMP. Em uma implementação desta aplicação pode-se encontrar um ou mais modelos distintos de controle de acesso;

- ***Access Control Subsystem*** – trata das questões relacionadas com o controle de acesso à informação de gerenciamento contida nas PDU´s.

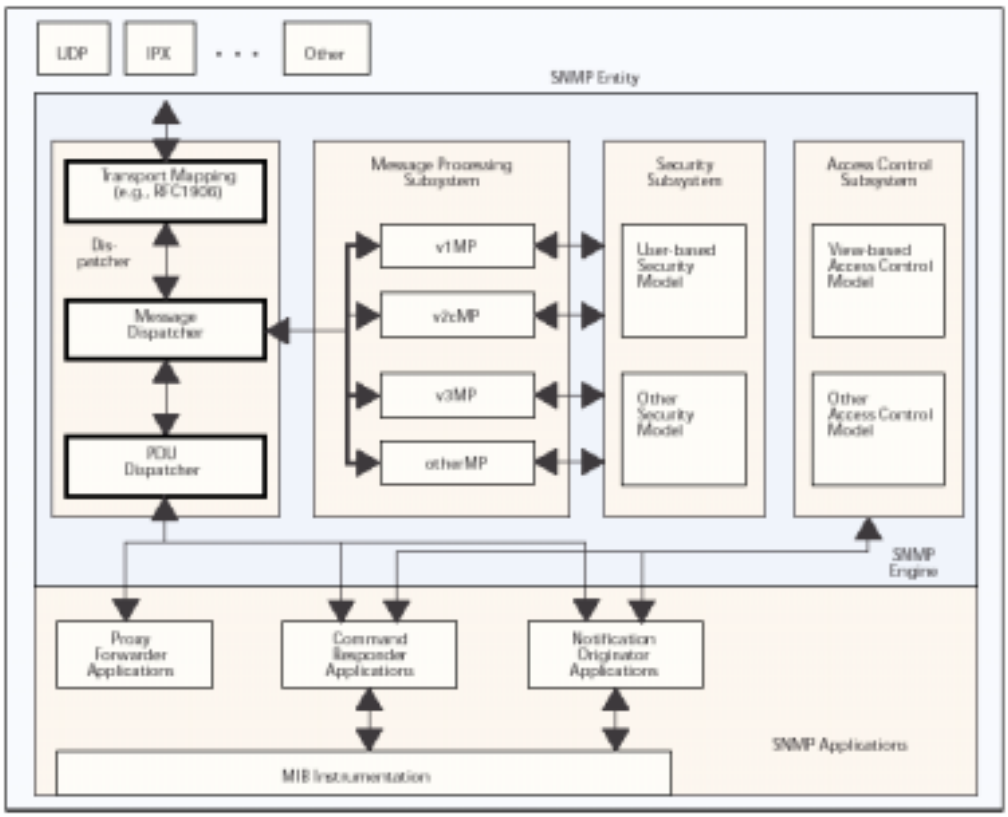

Figura 8.3 – Agente SNMP Tradicional

8.1.2 Terminologia

Antes de entrar em mais detalhes sobre o protocolo SNMPv3, convém apresentar alguns termos usados relativamente às entidades, conforme tabela B.1 a seguir:

snmpEngineID	Unique and unambiguous identifier of an SNMP engine, as well as the SNMP entity that corresponds to that engine.
contextEngineID	Uniquely identifies an SNMP entity that may realize an instance of a context with a particular contextName.
contextName	Identifies a particular context within an SNMP engine. It is passed as a parameter to the Dispatcher and Access Control Subsystem.
scopedPDU	A block of data consisting of a contextEngineID, a contextName, and an SNMP PDU. It is passed as a parameter to/from the Security Subsystem.
snmpMessageProcessingModel	Unique identifier of a message processing model of the Message Processing Subsystem. Possible values include SNMPv1, SNMPv2c, and SNMPv3.
snmpSecurityModel	Unique identifier of a security model of the Security Subsystem. Possible values include SNMPv1, SNMPv2c, and USM.
snmpSecurityLevel	A level of security at which SNMP messages can be sent or with which operations are being processed, expressed in terms of whether or not authentication and/or privacy are provided. The alternative values are noAuthnoPriv, authNoPriv, and authPriv.
principal	The entity on whose behalf services are provided or processing takes place. A principal can be an individual acting in a particular role; a set of individuals, with each acting in a particular role; an application or set of applications; and combinations thereof.
securityName	A human-readable string representing a principal. It is passed as a parameter in all of the SNMP primitives (Dispatcher, Message Processing, Security, Access Control)

Tabela 8.1 - Terminologia

- ***SNMP Engine*** – motor SNMP que implementa serviços para enviar e receber mensagens, controla o acesso aos objetos processados/gerenciados e autenticação e encriptação de mensagens;

- ***SNMPEngineID*** – é um identificador de um motor SNMP, assim como da entidade SNMP a que corresponde esse motor;

- ***ContextEngineID*** – identifica uma entidade SNMP que é capaz de executar uma instância de um contexto com um determinado *contextName*;

- ***ContextName*** – realiza a identificação de um contexto específico de um motor SNMP. É passado como parâmetro ao *Access Control Subsystem* e ao *Dispatcher*;

- ***ScopePDU*** – é um bloco de dados que é constituído por: um *contextEngineID*; um *contextName* e uma PDU SNMP;

- ***SnmpMessageProcessingModel*** – identifica um modelo de processamento de mensagem do *Message Processing Subsystem*;

- ***SnmpSecurityModel*** – identifica um sistema de segurança do subsistema de segurança;

- ***SnmpSecurityLevel*** – é um nível de segurança a quem as mensagens podem ser enviadas, ou com o qual as operações serão realizadas, expresso em função de ser ou não atribuída privacidade e autenticação.

A cada entidade SNMP está associado um único *SnmpEngineID*.

Relativamente à questão do controle de acessos, a entidade é capaz de gerenciar um determinado número de contextos de informação e que cada um tem apenas um *ContextName* na entidade.

Associado a cada entidade SNMP há um único *ContextEngineID*.

Resultante do fato de existir uma relação entre o motor de contexto e o motor SNMP, o *SnmpEngineID* tem o mesmo valor que *contextEngineID*.

O *SnmpMessageProcessingModel* determina qual o formato que a mensagem deve ter numa das versões SNMP para o seu processamento.

O *SnmpSecurityModel*, é o modelo de segurança que determina quais são os serviços de segurança necessários para que uma determinada operação se realize em segurança.

Por fim, o *Model Dependent Security ID* é a representação do modelo específico do *SecurityName* de um determinado modelo de segurança *Security Model*.

O protocolo SNMPv3 representa uma arquitetura mais complexa que as versões anteriores, mas, contudo, suportada nas versões anteriores. Porém a arquitetura do protocolo SNMPv3 está estruturada para suportar o desenvolvimento de novos módulos futuros.

8.1.3 Aplicações do Protocolo SNMPv3

De acordo com a arquitetura apresentada e para a melhor compreensão do funcionamento das aplicações do protocolo SNMPv3, segue abaixo, explicação sucinta do funcionamento de cada uma delas, podendo-se dividí-las em cinco tipos, respectivamente:

- *Command Generator*
- *Command Responder*
- *Notification Gen*erator
- *Notification Receiver*
- *Proxy Forwarder*

8.1.3.1 Command Generator

A aplicação *Command Generator* trata do envio de PDU´s, usando para tal as primitivas "*SendPdu*" e "*ProcessResponsePdu*". Analisando e interpretando a figura B.2, conclui-se que esta aplicação faz o envio e processamento de mensagens.

Neste contexto, a primitiva "*SendPdu*" envia ao *Dispatcher* a informação necessária para identificar o destino pretendido, os parâmetros de segurança estabelecidos, bem como a PDU que pretende. Seguidamente, o *Dispatcher* invoca o Modelo de Processamento de Mensagem (*Message Processing Model*) que efetuará em

seguida o chamamento do Modelo de Segurança (*Security Model*), responsável pelo modelo de segurança a ser aplicada na mensagem que será enviada.

No *Dispatcher* a mensagem é preparada com os campos necessários para poder ser enviada sobre a camada de transporte "UDP" para o encaminhamento. Quando a preparação da mensagem não for bem sucedida a primitiva de envio do *Dispatcher* "*SendPdu*" é encaminhada como se tratasse de um erro de transmissão. Se, por outro lado a preparação for bem sucedida, o *Dispatcher* desenvolve um identificador (designado por *SendPduHandle*) colocando-o na PDU sendo o seu valor devolvido ao *Command Generator*.

Por sua vez o *Command Generator* guarda o valor do identificador "*SendPdu Handle*" para que quando receber a PDU de resposta, possa identificar a que pedido esta resposta está associada.

Como já mencionado anteriormente, o *Dispatcher* envia para cada PDU de chegada, uma resposta à aplicação *Command Generator* correspondente utilizando para esse efeito, a primitiva "*ProcessResponsePdu*".

8.1.3.2 Command Responder

Analisando a figura B.3, é possível verificar a identificação da aplicação *Command Responder*, a qual utiliza três primitivas no sistema principal (*registerContextEngineID, unregisterContextEngineID, processPdu*), e, uma primitiva do Subsistema de Controle de Acessos (*isAccessAllowed*) (*returnResponsePdu*).

O registro "*registerContextEngineID*" contém uma primitiva que é responsável pela ativação da aplicação *Command Responder*. Esta aplicação vai acionar um motor SNMP com o objetivo de processar determinados tipos de PDU´s de forma previamente definida quando da sua constituição.

Logo que a aplicação *Command Responder* (desencadeada pela primitiva) registra todas as mensagens com as combinações "*contextEngineID*" e "*pduType*", estas estão disponíveis e acessíveis para o controle de gerenciamento.

Esta aplicação (*Command Responder*) tem capacidade de distinguir um motor SNMP que utiliza primitivas do tipo "*unregisterContextEngineID*".

O processamento e tratamento de mensagens (PDU), é inicializado no *Dispatcher* que utilizando a primitiva "*processPdu*", envia cada PDU de chegada à aplicação *Command Responder*. Seguidamente a aplicação *Command Responder* executará os seguintes passos:

- O conteúdo do formato da PDU de chegada é examinado pela aplicação *Command Responder*, devendo o tipo de operação pertencer aos tipos de operações antecipadamente registrados para esta aplicação;

- É ao *Command Responder* que pertence a decisão sobre o tipo de operação requerida na respectiva PDU, sendo esta permitida ou não. Para tal decisão é utilizado a primitiva "*isAccessAllowed*". O valor do parâmetro do Modelo de Segurança assegura que tipo de acesso é permitido, atendendo ao conteúdo do Subsistema de Controle de Acessos onde, para isso, são verificados os parâmetros "*securityName*", "*securityLevel*", "*viewType*", "*variableName*" e "*contextName*";

- Havendo permissão, o acesso é permitido e gerado então a PDU de resposta. Se não houver permissão, o *Command Responder* gera uma PDU de erro;

- Concluído este processo, o *Command Responder* invoca o *Dispatcher* através da primitiva "*returnResponsePdu*" para que a PDU de resposta seja enviada.

8.1.3.3 Notification Generator

Esta aplicação (*Notification Generator*) orienta a execução das suas primitivas de acordo com procedimentos semelhantes aos utilizados pela aplicação *Command Generator*.

No caso de ser necessário enviar uma PDU do tipo "*informRequest*", tanto a primitiva "*sendPdu*" como a primitiva "*processResponsePdu*" serão utilizadas, da mesma forma como são utilizadas pela aplicação *Command Generator*. Se pretende-se enviar um "*trap*" PDU, então é utilizada a primitiva "*sendPdu*".

8.1.3.4 Notification Receiver

Nesta aplicação (*Notification Receiver*), são utilizados também, os mesmos procedimentos da aplicação *Command Generator*, devendo o receptor da notificação ter registros memorizados do que recebe, assim como PDU´s do tipo "*inform*" ou "*trap*". Esses dois tipos de PDU´s vão ser recebidos pela mesma primitiva ("*processPdu*"). No caso de se tratar de uma resposta a uma PDU do tipo "*inform*", então é utilizada a primitiva "*returnResponsePdu*".

8.1.3.5 Proxy Forwarder

Esta aplicação, "*proxy forwarder*" necessita utilizar as primitivas do *Dispatcher* para gerar o encaminhamento das mensagens SNMP. Basicamente esta aplicação utiliza quatro tipos de mensagens:

- Mensagens contendo tipos de PDU´s pertencentes a aplicações *Command Generator*. Para tratamento destas mensagens a aplicação determina qual o motor SNMP irá utilizar, bem como, o motor SNMP mais próximo para o que se pretende atingir, enviando a PDU apropriada para cada caso;

- Mensagens contendo tipos de PDU´s pertencentes a aplicações que geram notificações. Tratando-se de notificação, a aplicação determina qual o motor SNMP que deve receber a notificação, e, envia as respectivas notificações apropriadas, PDU ou PDUs;

- Mensagens contendo o tipo de PDU de resposta. É nesta aplicação que se determina qual o pedido ou notificação previamente recebido, caso ele exista, e, envia a PDU correspondente de resposta;

- Mensagens contendo informações de relatório. Este conjunto de mensagens, formam relatórios do tipo PDU, que são comunicações SNMPv3, mecanismo a mecanismo, ou, motor a motor. A aplicação "*proxy forwarder*" seleciona também qual o pedido ou notificação previamente recebido, caso exista, é comparado com a indicação de relatório e, finalmente é tratada a informação que em seguida é enviada (relatório) de volta à entidade que desencadeou o pedido ou notificação.

8.2 Modelo de Processamento de Mensagens

O RFC que define o modelo de processamento de mensagens é o RFC 2572. Cabe a este modelo a responsabilidade de receber as PDU´s enviadas pelo *Dispatcher*, o qual as transforma em mensagens. Seguidamente é invocado o USM (*User Security Model*) para que no cabeçalho destas mesmas mensagens sejam inseridos os parâmetros de segurança, devolvendo em seguida a PDU ao *Dispatcher* com esses parâmetros definidos.

Observando a figura B.4 é possível verificar a estrutura da mensagem, na qual pode-se constatar que, para as mensagens a enviar (saída), os cinco primeiros campos (cabeçalho) são gerados pelo Modelo de Processamento de Mensagens. No caso das mensagens a receber (entrada), o processo é constituído de forma análoga, com estes cinco primeiros campos a serem também processados por este mesmo modelo.

Finalmente, a PDU conjuntamente com "*contextEngineID*" e o "*contextName*" constituem o corpo da mensagem usado para o processamento de cada PDU. Os parâmetros de segurança utilizados pelo USM são definidos nos próximos seis campos.

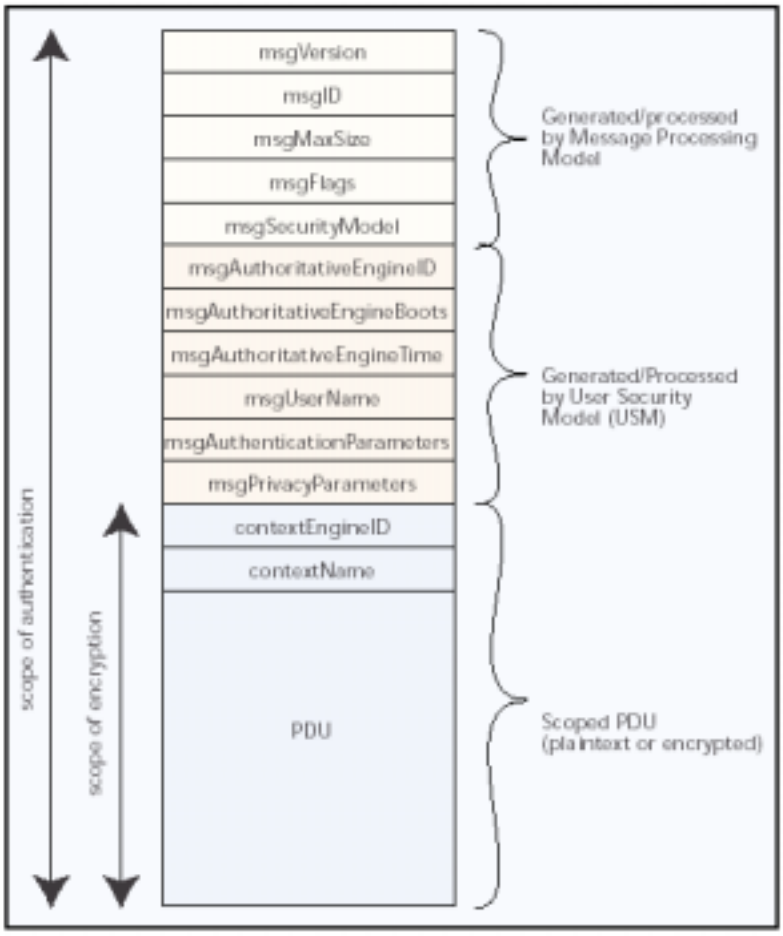

Figura 8.4 – Modelo de Processamento de Mensagem SNMPv3 com USM

Os primeiros cinco campos são:

- ***msgVersion*** – versão utilizada quando da construção da mensagem, que neste caso a mensagem será do tipo SNMPv3;

- ***msgID*** – é um identificador único utilizado entre duas entidades SNMP, de modo a coordenar as mensagens de pedidos e respostas. Este identificador é também utilizado pelo processador de mensagens para coordenar o processamento de mensagens que fluem através dos diferentes modelos do seu subsistema, de acordo com a

sua arquitetura. O tamanho deste ID pode variar desde 0 (zero) a 2^{31} –1.

- ***msgMaxSize*** – combina o tamanho máximo da mensagem suportado pelo emissor quando do seu envio, em conjuntos de octetos numerados de 484 a 2^{31} –1. Este valor é o máximo dos segmentos que o emissor pode receber vindo de um outro motor SNMP;

- ***msgFlags*** – define uma cadeia de caracteres (*string*) que definem *flags*, formada por um *octeto*, indicando aos três bits menos significativos um conjunto de três variáveis, caso de "*reportableFlag*", "*privFlag*" e "*authFlag*". Se a variável "*reportableFlag*" estiver a nível lógico alto (1), temos a indicação que o relatório PDU deverá ser enviado ao emissor; se esta variável estiver a nível lógico baixo (0), então esse mesmo relatório não será enviado. No entanto esta variável só é utilizada quando a parte PDU da mensagem não for descodificável. Para indicar o nível de segurança são inicializadas pelo emissor, as variáveis "*privFlag*" e "*authFlag*" que indicam o nível de segurança que está associado às mensagens;

- ***msgSecurityModel*** – campo que contém um identificador indicando qual o modelo de segurança utilizado pelo emissor quando do envio da mensagem, simultaneamente, fornece ao receptor indicações sobre qual o modelo de segurança que deve ser utilizado para o processamento da respectiva mensagem. É também neste modelo que se reservam valores para indicar qual a versão do SNMP em utilização, respectivamente ´1´ para SNMPv1, ´2´para o SNMPv2 e ´3´para o SNMPv3. Este identificador contém valores com um comprimento que varia desde 0 (zero) até 2^{31} –1.

8.3 Modelo de Segurança Baseado no Utilizador

Este modelo de segurança baseado no utilizador (*User-Based Security Model* – USM), caracterizado no RFC 2574, é um modelo que fornece serviços privados de autenticação a cada utilizador para o SNMP.

Especificamente este modelo é utilizado para garantir a segurança dos dados contra os seguintes aspectos:

- ***Modification of Information*** – sumariamente entende-se por alteração da informação. Esta alteração pode ocorrer quando uma mensagem se encontra a circular na rede, depois de ter sido emitida por uma estação autorizada. Esta mensagem pode ser alterada por uma outra entidade autorizada, obviamente esta alteração é suscetível de alterar as operações de gerenciamento da rede, assim como causar impacto nos valores de alguns objetos. A razão fundamental da existência desta proteção prende-se com o fato de se proteger os dados de configuração, operação e "*login*" de estações não autorizadas;

- ***Masquerade*** – quando uma estação não autorizada na rede, consegue autorização para efetuar operações de gerenciamento sobre dados, essa autorização só existe quando a estação não autorizada se encontra mascarada, assumindo a identidade de uma estação autorizada;

- ***Message Stream Modification*** – pelo que foi dito anteriormente, é necessário um mecanismo de proteção contra operações não autorizadas, assim o MSM (*Message Stream Modification*) protege os dados contra atrasos ou duplicação de informação. Facilmente se conclui que situações deste tipo ocorrem principalmente quando o SNMP utiliza um protocolo de transporte não orientado à conexão, tipo TCP;

- ***Disclosure*** – outra situação a se considerar, é que, qualquer entidade não autorizada pode observar a rede e "aprender" o conteúdo e os valores dos objetos de gerenciamento que circulam na rede, e, caso não exista qualquer tipo de proteção, poderia utilizar esses valores para efetuar operações de gerenciamento não autorizadas, desvirtuando completamente a rede.

Contudo, o USM não prevê segurança para os seguintes aspectos:

- ***Denial of Service*** – um "*hacker*" pode prever qual a troca de dados entre uma estação de gerenciamento e uma estação agente;

- ***Traffic Analysis*** – um "*hacker*" pode observar e analisar qual o modelo de tráfego existente entre uma estação de gerenciamento e um agente.

8.3.1 Encriptação

O USM define duas funções criptográficas necessárias para a transmissão segura de mensagens, que são: "Autenticação" e "Encriptação". Para suportar estas funções um motor SNMP requer dois valores:

- Uma chave privada

- Uma chave autenticada

A separação dos valores destas duas chaves, são mantidos com os seguintes "*users*" :

- ***Local users*** – qualquer motor principal SNMP para operações de gerenciamento está autorizado;

- ***Remote users*** – qualquer motor remoto SNMP para comunicação é desejado.

Para serem implementadas, estas duas funções necessitam da Chave Privada (*privKey*) e da Chave de Autorização (*authKey*). Estas chaves são mantidas pelos utilizadores locais e pelos utilizadores remotos. Cada utilizador relevante possui estes atributos e por uma questão de segurança, os valores dos atributos não estão acessíveis através do protocolo SNMP.

O USM permite utilizar dois tipos de protocolo de autenticação, HMAC-MD5-96 e o HMAC-SHA-96. Estes protocolos estão descritos no RFC 2104.

8.3.2 Motores Autoritários e Não Autoritários

Quando ocorre a comunicação, necessariamente existirão trocas de mensagens, e pelo menos uma das duas entidades transmite ou recebe. Para que esta troca de mensagens se verifique é designado um motor autoritário, sendo definido de acordo com as seguintes regras:

- Quando uma mensagem SNMP contém uma instrução que necessita de resposta (por exemplo, a *Get*, *GetNext*, *GetBulk*, *Set*, ou *Inform* PDU), então o receptor desta mensagem é um motor autoritário;

- Quando uma mensagem SNMP contém uma instrução que não necessita de resposta (por exemplo, um SNMPv2-*Trap*, *Response*, ou *Report* PDU), então o emissor desta mensagem é um motor não autoritário.

Desta forma, é possível afirmar que, o receptor de mensagens transmitidas utilizando o *Commnand Generator* e o *Notification Generator* é um motor autoritário, analogamente é também um motor autoritário o receptor das mensagens enviadas

através do *Command Responder* ou do *Notification Originator*. Esta designação atinge dois propósitos importantes:

- O tempo de vida de uma mensagem é calculado com base no relógio do motor autoritário. Quando um motor autoritário envia uma mensagem (*Trap, Response, Report*), é introduzido nesta, o valor do relógio do emissor na intenção de que o receptor possa sincronizar o seu relógio com o relógio do emissor. Quando um motor não autoritário envia uma mensagem (*Get, GetNext, GetBulk, Set, Inform*), lhe é etiquetado o tempo estimado que demora a atingir o receptor, permitindo ao receptor ter informação acerca do tempo de vida da mensagem;

- O processo de localização da chave (como será mostrado mais adiante), possibilita a uma entidade conter várias chaves guardadas em vários motores SNMP. A localização destas chaves é no motor autoritário, e o principal , tem em sua responsabilidade uma única chave, com isto, evita-se que existam várias cópias de uma chave em toda a rede.

Considerando estes propósitos, faz sentido considerar o receptor *Command Generator* juntamente com os "*inform* PDU´s" como motores autoritários, e, também a responsabilidade de calcularem o tempo de vida das mensagens. Se um "*trap*" de resposta é atrasado ou reenviado, podem ocorrer pequenas falhas, contudo, o *Command Generator* e algumas extensões do *Inform* PDU acabam por fazer um gerenciamento operacional da rede. Para que não surjam efeitos não desejados, é fundamental garantir que as PDU´s não sejam atrasadas ou reenviadas, podendo causar efeitos indesejados.

8.3.3 Parâmetros Utilizados nas Mensagens USM

Quando uma mensagem está pronta para ser enviada e após passar pelo Processador de Mensagens, é enviada para o USM (*User Security Model*) onde irão ser

introduzidos os parâmetros de segurança. No caso de se tratar de uma mensagem que é recebida ou uma mensagem que é enviada pelo processador de mensagens ao USM, são verificados os seguintes parâmetros:

- ***MsgAuthoritativeEngineID*** – cabe a este parâmetro (*AuthoritativeEngineID*) a responsabilidade de trocar as mensagens num motor SNMP autoritário. Com a análise do valor deste parâmetro é possível saber qual a origem de um "*trap*" (resposta ou relatório) e é também possível verificar o destino de um comando *get, getnext, getbulk*;

- ***MsgAuthoritativeEngineBoots*** – este parâmetro indica o número de vezes que o motor SNMP se inicializou ou reinicializou desde a última configuração;

- ***MsgAuthoritativeEngineTime*** – representa o tempo (incremento em segundos) em que o parâmetro "*AuthoritativeEngineBoots*" do motor SNMP se reinicializou pela última vez, sendo de cada motor SNMP a responsabilidade pelo incremento do seu parâmetro "*AuthoritativeEngineTime*", mas implementa uma função semelhante, que traduz, no incremento de um valor sempre que comunica com um motor autoritário, remotamente;

- ***MsgUserName*** – este parâmetro representa o nome do utilizador (principal) com quem a mensagem está, para ser trocada;

- ***MsgAuthenticationParameters*** – o valor deste parâmetro é nulo (*NULL*) quando a mensagem não possui autenticação. No caso contrário, indica um valor não nulo, pelo que possui autenticação. Para a definição corrente do USM a autenticação do parâmetro é uma mensagem de autenticação codificada tipo HMAC;

- ***MsgPrivacyParameter*** – esse parâmetro assume valor nulo (*NULL*) quando a mensagem não possui autenticação. Quando tem um valor não nulo, indica que possui autenticação. Para a definição corrente do USM o valor deste código é baseado no algoritmo DES CBC.

Havendo necessidade de enviar mensagens codificadas, as mensagens com encriptação, serão efetuadas em primeiro lugar. Assim sendo, a PDU recebida é encriptada com os campos necessários e destinados aos parâmetros de privacidade (*msgPrivacyParameters*) preenchidos. Quando necessário e reforçando a codificação, é efetuada a autenticação da mensagem , para tal, toda a mensagem é em seguida, colocada no HMAC, onde irão ser colocados os parâmetros de autenticação.

A figura B.5 mostra o funcionamento do USM.

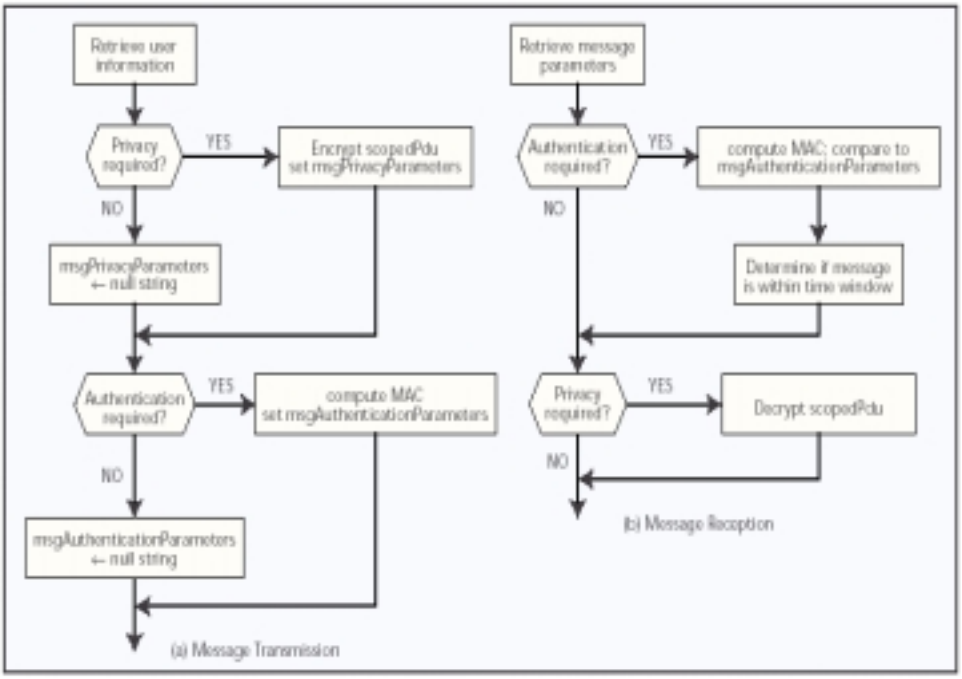

Figura 8.5 – Funcionamento do Processamento de Mensagem do USM

Tratando-se de recepção de mensagens, em primeiro lugar efetua-se a autenticação, quando necessário.

O USM compara o MAC recebido com o MAC que foi calculado. Se estes forem iguais, é assumido que a mensagem pode ser autenticada. Seguidamente o USM

verifica o tempo de vida da mensagem, caso o tempo de vida não coincidir, a mensagem não é autenticada e é eliminada.

Tratando-se de uma mensagem que foi encriptada, há necessidade de proceder a sua desencriptação, sendo esta devolvida em formato texto.

8.3.4 Mecanismo Tempo de Vida do USM

Para proteger as mensagens e as tornar confiáveis, o USM utiliza o mecanismo de tempo de vida, protegendo desta forma a mensagem contra atrasos e repetições. Um motor SNMP para poder funcionar como motor autoritário precisa manter dois objetos: o "*snmpEgineBoots*" e o "*snmpEngineTime*". Quando da instalação de um motor SNMP pela primeira vez, estes objetos deverão ser inicializados com ´0´. Como são complementares, quando o "*snmpEngineTime*" (é incrementado uma vez por segundo), ao atingir o valor máximo (2^{31} –1), o "*snmpEngineBoots*" é incrementado uma unidade. O mesmo se verifica quando o sistema for reinicializado. Posteriormente o "*snmpEngineTime*" é inicializado em ´0´ e recomeça a incrementar.

Usando um mecanismo de sincronismo, um motor não autoritário mantém uma estimativa dos valores dos tempos de comunicação que precisa para comunicar com os motores autoritários. Partindo deste pressuposto, os tempos estimados pelos motores não prioritários, são colocados nas mensagens emitidas, tornando assim possível ao motor autoritário que recebe a mensagem, verificar se esta ultrapassou ou não, o tempo de vida.

Um motor não autoritário guarda uma cópia de três variáveis por cada motor autoritário com quem comunica. O mecanismo de tempo de vida tem o seguinte esquema de funcionamento:

- *SnmpEngineBoots* – mantém o valor atualizado (mais recente) da variável "*snmpEngineBoots*" do motor autoritário remoto;

- *SnmpEngineTime* – variável utilizada para medir o tempo de um motor autoritário remoto. O valor desta variável é sincronizado remotamente por um motor autoritário. Entre os processos de sincronização este valor é logicamente incrementado uma vez por segundo, mantendo uma sincronização suave, com o motor autoritário remoto;

- *LastReceivedEngineTime* – trata-se do valor mais elevado da variável "*msgAuthoritativeEngineTime*", e é esse valor que é enviado pelo motor autoritário remoto ao motor autoritário utilizado localmente. Com esta variável é possível eliminar a existência, de mensagens repetidas, eliminando a possibilidade de um motor não autoritário incrementar a variável, "*snmpEngineTime*" podendo existir mensagens repetidas.

A função fundamental da existência destas três variáveis é a de informar o motor autoritário local, da existência dos outros motores autoritários remotos existentes na rede. Logicamente, estas variáveis são mantidas localmente utilizando uma fração reduzida de memória "*cache*", a sua indexação é conseguida através do valor único da variável "*snmpEngineID*" por cada motor autoritário remoto. Como já mencionado anteriormente, para que os motores não autoritários se sincronizem com outros motores autoritários, é necessário que o motor autoritário informe estes em cada mensagem trocada. Este objetivo é conseguido através do valor da variável do "*snmpEngineID*" em cada resposta, *trap* ou relatório, nos campos *msgAuthoritativeEngineID*, *msgAuthoritativeEngineTime* e *msgAuthoritativeEngineBoots*. Caso se trate de uma mensagem ser autêntica e com tempo de vida válido, então o motor não autoritário que recebe a mensagem , atualiza as suas variáveis locais em referência ao motor remoto atendendo as seguintes regras:

- Ocorre uma atualização, se pelo menos uma das seguintes condições for verdadeira:

(*msgAuthoritativeEngineBoots* > *snmpEngineBoots*) **or**
[(*msgAuthoritativeEngineBoots* = *snmpEngineBoots*) **and**
(*msgAuthoritativeEngineTime* > *lastReceiveEngineTime*)]

É possível verificar que a primeira condição indica que uma atualização ocorrerá se o valor *´boot´* do motor autoritário tiver sofrido um incremento desde a última atualização.

Na segunda condição conclui-se que, se o valor *´boot´* não tiver sofrido um incremento, e se o tempo de chegada do motor for maior que o último tempo recebido, então ocorrerá uma atualização. Se porventura as mensagens de chegada, chegarem fora de ordem, então o tempo de chegada do motor será obviamente, menor que o último tempo recebido do motor.
No caso de ser perdida uma atualização, então serão efetuadas as seguintes mudanças:

- Será copiado o valor da variável *msgAuthoritativeEngineBoots* para a variável *snmpEngineBoots*;

- Será copiado o valor da variável *msgAuthoritativeEngineTime* para a variável *snmpEngineTime*;

- Será copiado o valor da variável *msgAuthoritativeEngineTime* para a variável *latestReceivedEngineTime*.

É importante salientar que só se utiliza sincronização no caso de a mensagem utilizar autenticação que é implementada através do HMAC. Esta condição é fundamental porque só desta forma é possível garantir a confiabilidade das variáveis *msgAuthoritativeEngineID*, *msgAuthoritativeEngineTime* e *msgAuthoritativeEngineBoots*.

O protocolo SNMPv3 especifica também que uma mensagem tem de ser recebida dentro dos valores previstos numa janela temporal adequada, protegendo desta

forma a mensagem de atrasos e duplicação. Esta janela temporal deverá ser o mais pequena possível, considerando o tipo e precisão dos relógios envolvidos, os tempos de propagação da mensagem e a freqüência de sincronização dos relógios. Partindo destes pressupostos, se a janela temporal for inferior ao necessário, há o risco de mensagens válidas e autênticas serem consideradas não válidas e não autenticadas. Contrariamente, se o tempo razoável da janela para a recepção, for muito grande, as mensagens permanecem mais tempo na rede e desta forma, mais vulneráveis a ataques.

No caso de o tempo da mensagem recebida for superior ao da janela temporal, a mensagem será considerada não válida, e conseqüentemente emitido um erro indicativo (*notIntimeWindow*) ao módulo responsável pela emissão.

8.3.5 Localização da Chave

Nos serviços privados em SNMPv3, um dos requisitos é a necessidade de haver uma chave de autenticação secreta e uma chave privada secreta partilhadas, para existir comunicação entre um Principal em um motor não autoritário e um motor autoritário remoto. São estas chaves que permitem ao utilizador num motor não autoritário, aplicar autenticação e privacidade com motores não autorizados remotos que são gerenciados pelo utilizador. No RFC 2574 é descrito como devem ser gerenciadas, criadas e mantidas estas chaves.

Para poder simplificar gerenciamento de um grande número de chaves nos Principais, cada Principal só mantém uma única chave de autenticação e uma chave de encriptação. Estas chaves não são guardadas na MIB e não podem ser acessíveis via SNMP.

Quando um utilizador necessitar de uma chave privada de 16 *octetos* e de uma chave de autenticação de 16 ou 20 *octetos*, para que ele as possa utilizar seria necessário a conversão numa senha em texto perceptível e decorável e não numa senha baseada em *bits*, o que seria muito difícil e trabalhoso a sua utilização. É no RFC 2574 que se encontra definido o algoritmo que utiliza a senha do utilizador de 16 ou 20

octetos. O USM não coloca nenhuma restrição no tipo de senha, mas as políticas de gerenciamento locais especificam que os utilizadores devem escolher senhas dificilmente legíveis, sem nexo, para que seja difícil ser adivinhável.
A especificação de geração das senhas, dever ser feita segundo os seguintes pontos:

- Ao ser introduzida uma senha pelo utilizador, é gerado pelo software uma "*string*" do tamanho de 2²º *octetos*, repetindo a senha até os *octetos* ficarem preenchidos, e efetuando a truncagem se necessário, gerando uma "*string*" denominada por "*digest0*";

- Se for utilizada uma chave com 16 *octetos*, utiliza-se a função de "*hash*" MD5 para criar a "*string*" "*digest1*" a partir da "*string*" "*digest0*". Se for utilizada uma chave de 20 *octetos* utilizamos a função de "*hash*" SHA-1, sendo obtido desse modo a chave do utilizador.

Esta técnica tem várias vantagens, porque cria um atraso considerável em cada uma das tentativas de descobrir a senha, porque em cada tentativa o atacante tem que tentar muitas combinações, e, em cada combinação é preciso gerar a chave a partir da senha, e só depois verificar se a chave funciona com a encriptação utilizada. No caso de um atacante interceptar uma mensagem autenticada, pode tentar gerar o valor do HMAC com chaves diferentes escolhidas aleatoriamente, se o valor coincidir, pode ter acesso à mensagem e descobrir a senha. Mas com estes dois passos descritos este processo de tentativas de descoberta da senha levará muito mais tempo.

Uma outra vantagem desta técnica é que são separadas as chaves dos utilizadores de qualquer sistema de gerenciamento de rede (NMS – *Network Management System*). Sendo assim, nenhum NMS necessita possuir uma chave de utilizador guardada no sistema de gerenciamento, ou seja, quando a chave for necessária esta pode ser gerada a partir da senha do utilizador.

Porém, se a chave for guardada em vez de ser gerada a partir da senha, será necessário que exista um registro central onde possam ser armazenadas as chaves

secretas. Mas, como é óbvio, esta solução sobrecarrega o sistema e pode criar problemas quando o registro central onde são guardadas as chaves não está acessível.

Se por outro lado, existirem registros centrais duplicados, isso vai colocar as questões de segurança em risco, pois existiriam mais pontos passíveis de serem atacados.

- No caso dos registros centrais ou dos duplicados serem utilizados, estes deverão estar localizados em locais seguros, porque deste modo é possível reduzir o número de correções dos erros.

Mas, poderá ser utilizada uma única senha para autenticação e encriptação. Contudo, é logicamente mais seguro a utilização de uma senha para cada caso.

No RFC 2574 está definida uma chave localizada como sendo uma chave secreta partilhada pelo utilizador e pelo motor SNMP autoritário. O objetivo é que o utilizador não tenha que memorizar mais que uma (ou duas no caso de autenticação e privacidade) senha. Os dados secretos partilhados entre um determinado utilizador e o motor SNMP são diferentes, pois o processo de descodificação de uma determinada chave é convertida em múltiplas chaves (de valor único) para cada um dos respectivos motores SNMP e este processo é designado como localização da chave.

Para gerenciamento das chaves terão de ser definidos os seguintes objetivos:

- Numa rede distribuída cada agente SNMP tem uma chave única para cada utilizador autorizado a ser gerenciado na rede. No caso de existirem múltiplos agentes autorizados a serem gerenciados na rede, deve existir para cada um, uma única chave de autenticação e uma única chave de encriptação, o que possibilita que, ser por acaso, ocorrerem problemas com um determinado agente, os outros agentes não deixem de poder gerenciar a rede;

- Cada utilizador tem uma chave diferente para cada agente SNMP, possibilitando que, no caso de ocorrerem problemas com um determinado utilizador, somente as chaves designadas para este utilizador fiquem inativas, continuando válidas as chaves dos demais utilizadores;

- Deve ser possível o gerenciamento da rede a partir de qualquer ponto da rede, levando-se em conta a disponibilidade do sistema de gerenciamento da rede pré configurado (NMS). Isso permite que um agente autorizado desenvolva funções de gerenciamento a partir de qualquer estação de gerenciamento. Esta capacidade é disponibilizada pelo mecanismo de geração de senhas descrito anteriormente.

Deve-se ainda, levar em conta as seguintes situações:

- Um utilizador terá que relembrar ou gerenciar um vasto número de chaves. Chaves essas que aumentam com o número de agentes que são inseridos na rede.

- Um "*hacker*" que descodifique uma chave relativa a um determinado agente ou utilizador, poderá utilizar esta chave identificando-se como um agente perante os utilizadores, ou utilizador perante os agentes.

Uma forma de alcançar os objetivos e considerações anteriores, é mapear uma única chave como sendo uma função de um único sentido (não reversível). A chave é mapeada em diferentes chaves localizadas, dependendo dos motores autenticados. O processo de mapeamento e obtenção da chave localizada do utilizador consiste no seguinte:

Na criação da "*string*" "*digest2*" através da concatenação da "*string*" "*digest1*" com o valor da variável "*snmpEngineID*", podendo ser esta "*string*" (*digest2*) um valor de entrada nas funções de "*hash*" MD5 ou SHA-1.

Se pretende-se uma chave de 16 *octetos*, pode-se utilizar como valor da função de "*hash*" MD5 a "*string*" "*digest2*". Se pretende-se uma chave de 20 *octetos*, pode-se utilizar como valor da função de "*hash*" SHA-1 a "*string*" "*digest2*".
A chave localizada resultante, pode ser configurada no sistema agente de um modo seguro.

Devido ao fato de tanto a função MD5 como a SHA-1 serem de sentido único (não reversíveis), impossibilitam que qualquer "*hacker*" consiga descodificar a chave de um determinado utilizador mesmo que este tente descobrir a chave localizada.
A figura B.6 mostra qual o processo de localização da chave.

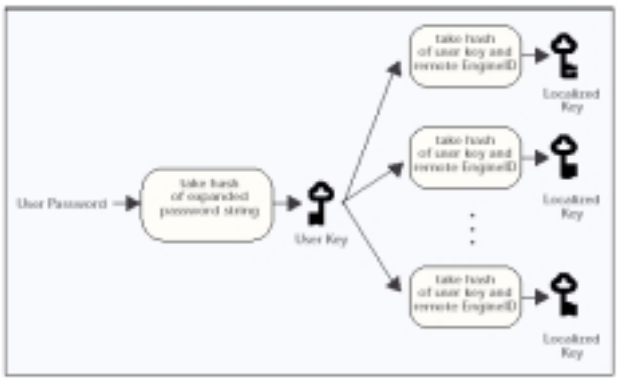

Figura 8.6 – Localização da Chave

8.4 View-Based Access Control

Uma função de segurança desempenhada ao nível de PDU é o controle de acesso. Um documento de controle de acessos define mecanismos para determinar quando deve ser permitido o acesso para um objeto gerenciado em uma MIB local por um "Principal" remoto. Neste conceito, o mecanismo de acessos múltiplos pode ser definido. Os documentos do SNMPv3 definem um modelo *View-Based Access Control* (VACM).
Tem-se então duas características importantes:

- Faz com que seja determinado se o acesso a objetos gerenciados na MIB local, por um “Principal” remoto, deve ou não ser permitido;

- Faz uso de uma MIB que, define a política de controle de acessos para um agente e também torna possível a configuração remota.

8.4.1 Elementos do Modelo VACM

Os elementos que constituem a estrutura do VACM são cinco: grupos, nível de segurança, contextos, visões da MIB e política de acessos. Segue descrição de cada um dos elementos.

8.4.1.1 Grupos

Para definir um grupo é utilizado um conjunto de zero ou mais tuplas <*securityModel*, *securityName*> pelos quais os objetos de gerenciamento SNMP podem ser acessados. A variável “*securityName*” refere-se a um “Principal”, e os direitos de acesso a todos os “Principais” num determinado grupo são idênticos. O “*groupName*” associado a cada grupo é único. O conceito de grupo é uma ferramenta útil para definir por categorias, relativamente aos direitos de acesso, os gerenciadores. Por exemplo, todos os gerenciadores do topo podem ter apenas um conjunto de direitos de acesso, enquanto que os gerenciadores intermediários podem ter vários conjuntos de direitos de acesso.

Qualquer combinação “*securityModel*-*securityName*” pode pertencer a pelo menos um grupo, isto é, para um agente, um dado “Principal” cujas comunicações estão protegidas por um determinado “*securityModel*”, pode apenas ser incluído num grupo.

8.4.1.2 Nível de Segurança

Em um grupo podem ser distinguidos os direitos de acesso, dependendo do nível de segurança da mensagem que contém o pedido. Por exemplo, um agente pode permitir o acesso apenas para leitura, atendendo a um pedido de comunicação não autenticado, mas pode pedir autenticação para um acesso de escrita. Para alguns objetos específicos, o agente pode requerer que o pedido e a respectiva resposta sejam comunicados usando o serviço de privacidade.

8.4.1.3 Contextos

O contexto de uma MIB é um subconjunto atribuído nas instâncias do objeto na MIB local. Os contextos são uma forma fácil de agregar objetos em coleções com diferentes políticas de acesso.

O contexto é um conceito relacionado com o controle de acessos. Quando uma estação de gerenciamento interage com um agente para aceder como o gerenciador à informação de gerenciamento, então a intereção faz-se entre a estação de gerenciamento “Principal” e motor SNMP agente, e, os privilégios de controle de acesso estão numa visão MIB que recorre a esse “Principal” e ao seu contexto. Os contextos têm as seguintes características chave:

- Em uma entidade SNMP, sendo identificada apenas por um “*contextEngineID*”, pode manter mais do que um contexto;

- Um objeto ou uma instância podem aparecer em mais do que um contexto;

- Se existirem contextos múltiplos, para identificar a instância de um objeto individual, o seu “*contextName*” e “*contextEngineID*” devem ser identificados com o seu tipo de objeto e instância.

8.4.1.4 Visões da MIB

Caso pretende-se restringir o acesso de um dado grupo a um subconjunto de objetos gerenciados em um agente, e para que se possa alcançar este objetivo, o acesso a um contexto é feito por intermédio de uma visão da MIB, a qual define um conjunto específico de objetos gerenciados. Então, para definir visões da MIB o VACM faz uso de uma técnica potente e flexível, baseado nos conceitos das visões de sub-árvores (*view subtrees*) e visões de famílias (*view families*). A visão MIB é definida como sendo uma coleção, ou família, de árvores com as suas ramificações, em que cada ramificação é incluída ou excluída da zona de visualização.

Os objetos gerenciados numa base de dados local são organizados de forma hierárquica, ou em árvore, baseada nos identificadores dos objetos. Esta base de dados local inclui um subconjunto de todos os tipos de objetos, definida de acordo com o *Internet-Standard Structure of Management Information* (SMI), e, inclui instâncias de objetos cujos identificadores estão de acordo com as convenções SMI.

O SNMPv3 introduz o conceito de sub-árvore. Uma sub-árvore não é mais do que um nodo na hierarquia de uma MIB, juntamente com todos os seus elementos subordinados. De um modo mais formal, uma sub-árvore pode ser definida como sendo o conjunto de todos os objetos e instâncias de objetos que têm o prefixo comum ASN.1 *OBJECT IDENTIFIER* para os seus nomes. O maior prefixo comum de todas as instâncias da sub-árvore é o identificador de objeto da raiz da sub-árvore.

A cada entrada na tabela de acesso VACM (*vacmAccessTable*) estão associadas as visões MIB da árvore, uma por cada, de leitura, escrita e notificação de acesso. Cada visão MIB consiste em um conjunto de sub-árvores. Cada sub-árvore na visão da MIB é especificada como sendo incluída ou excluída, isto é, a visão da MIB tanto pode incluir como excluir todas as instâncias de objetos contidas na sub-árvore. Além disso, a máscara da visão é definida com o objetivo de reduzir a quantidade de informação de configuração necessária, quando é requerido um controle de acesso “detalhado” (por exemplo, controle de acesso ao nível da instância de objeto).

8.4.1.5 Política de Acesso

O VACM permite que um motor SNMP seja configurado com um conjunto de direitos de acesso específico. Os fatores importantes que determinam o acesso são:

- Se for o “Principal” que solicita o acesso, o VACM torna possível um agente ceder diferentes tipos de privilégios de acesso a diferentes utilizadores. Por exemplo, um sistema de gerenciamento, responsável pela configuração da rede, ter autorização para alterar itens na MIB local, enquanto que um gerenciador de um nível médio com responsabilidades de monitoração, pode apenas ter autorização de acesso para leitura e estar limitado a ter acesso a um subconjunto da MIB local. Como já mencionado, “Principais” são atribuídos a grupos e política de acesso é especificada em relação aos grupos;

- Ao nível de segurança, normalmente numa mensagem SNMP é feito um pedido de autenticação para as mensagens que contenham um pedido de escrita;

- Qual o nível de segurança usado para o processamento da mensagem do pedido. No caso de vários modelos de segurança estarem implementados num agente, normalmente estará configurado para fornecer diferentes níveis de acesso para pedidos feitos por mensagens processadas por diferentes modelos de segurança. Se por exemplo, certos itens só podem ser acessíveis se a mensagem de pedido chegar através do USM, mas não são acessíveis se o modelo de segurança é o SNMPv1;

- O contexto da MIB para o pedido;

- A instância específica do objeto para a qual o acesso é requisitado. A informação retida pelos objetos não é feita de igual modo, pois uns são mais sensíveis a determinados tipos de informação do que

outros, isso implica que, a política de acesso depende da instância específica do objeto pedido;

- O tipo de acesso pedido (leitura, escrita, notificação). Quando se têm diferentes tipos de operação como são o caso de leitura, escrita e notificação. Terão de existir, políticas diferentes de controle de acesso para cada uma.

8.5 Processamento de Controle de Acessos

Uma aplicação SNMP invoca o VACM através da primitiva "*isAccessAllowed*", com os parâmetros de entrada "*securityModel*", "*securityName*", "*securityLevel*", "*viewType*", "*contextName*" e "*variableName*".

É preciso existir todos estes parâmetros para se poder tomar uma decisão sobre o controle de acesso. Por outro lado, o subsistema de Controle de Acessos é definido de forma a fornecer uma ferramenta flexível para configurar o controle de acessos no agente, separando os componentes de decisão do controle de acessos em seis variáveis.

Na figura B.7, apresenta-se uma forma prática de visualização das variáveis de entrada e também mostra como os vários quadros na MIB VACM se relacionam com as decisões do controle de acesso.

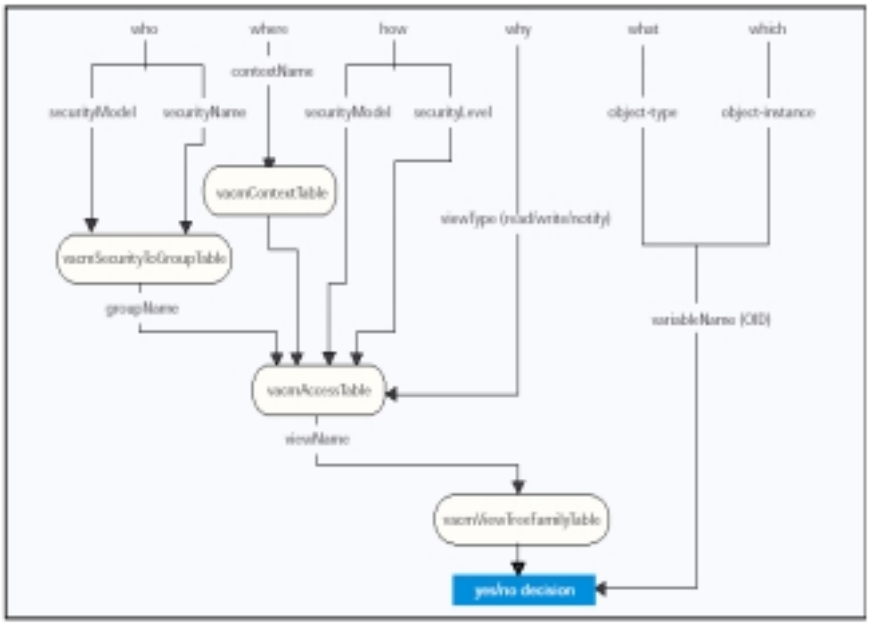

Figura 8.7 – Estrutura Lógica da VACM

- ***Who*** (quem) – é a combinação do Modelo de Segurança com o "*securityName*" que define o "quem" desta operação; identifica um dado "Principal" cujas comunicações são protegidas por um dado "*securityModel*". Esta combinação pertence a pelo menos um grupo neste motor SNMP. O "*vacmSecurityToGroupTable*" fornece o "*groupName*", tendo sido dado o "*securityModel*" e o "*securityName*";

- ***Where*** (onde) – O "*contextName*" explicita onde se deve encontrar o objeto de gerenciamento desejado. O "*vacmContextTable*" contém uma lista dos "*contextNames*" reconhecidos;

- ***How*** (como) – A combinação do "*securityModel*" com o "*securityLevel*" define "como" o pedido recebido ou "*inform* PDU" foi protegido. A combinação de "quem", "onde" e "como" identifica zero ou uma entradas na "*vacmAccessTable*";

- ***Why*** (porquê) – O "*viewType*" especifica o "porquê" do pedido de acesso: para uma operação de leitura, escrita ou notificação. A entrada selecionada na "*vacmAccessTable*" contém uma MIB

"*viewName*" para cada um dos três tipos de operação, e, o "*viewType*" é utilizado para selecionar um "*viewName*" específico. Este "*viewName*" seleciona a visão da MIB apropriada do "*vacmViewTreeFamilyTable*";

- ***What*** (o quê) – O "*variableName*" é um identificador do objeto, cujos prefixo e sufixo identificam respectivamente, um tipo específico e uma instância específica do objeto. O tipo do objeto indica qual o tipo de informação de gerenciamento solicitado;

- ***Which*** (qual) – A instância do objeto indica "qual" é o item específico de informação solicitada.

Finalmente, o "*variableName*" é comparado com a visão MIB devolvida. Se o "*variableName*" condiz com o elemento incluído na visão MIB o acesso é autorizado.

8.5.1 Motivação

Segue abaixo, uma breve análise dos conceitos que fizeram o VACM resultar numa definição mais complexa de controle de acesso. Tais conceitos, servem para esclarecer as relações envolvidas no acesso a informação de gerenciamento e para minimizar o armazenamento e requisitos de processamento no agente. Para entender essas motivações, é preciso ter atenção no seguinte: no SNMPv1 e SNMPv2, os conceitos são usados para representar as seguintes informações relativas à segurança:

- A identificação da entidade que faz o pedido (estação de gerenciamento);

- A identificação da entidade que faz a execução (agente que age para si próprio ou para qualquer uma entidade que necessita de um agente ("*proxy*"));

- A identificação do local a partir do qual se pode obter informação da localização da informação de gerenciamento a ser acessada (agente, ou dispositivo que necessita de um agente “*proxy*”);

- Autenticação da informação;

- Informação sobre o controle de acesso (autorização para executar as operações pretendidas);

- Informação da visão MIB.

Ao juntar todos estes conceitos numa única variável, a conseqüência é a perda da funcionalidade e flexibilidade. O VACM fornece este mesmo conjunto de informação relativa à segurança ao utilizar variáveis distintas para cada item. Esta é portanto, uma melhoria significativa em relação ao SNMPv1. Isto vai separar vários conceitos de forma que esses valores possam ser atribuídos a cada um, separadamente.

Portanto, pode-se concluir que o SNMPv3 é significativamente mais complexo que SNMPv1 e SNMPv2 e o propósito primário para o seu desenvolvimento foi: corrigir a falha mais óbvia das versões anteriores, a falta de segurança. Apesar de muitas mudanças incorporadas em SNMPv3, uma mensagem SNMPv3 ainda carrega uma PDU SNMPv2 dentro dela, porém as mensagens de SNMPv3 invocam as mesmas operações como mensagens de SNMPv2, só que em um modo altamente seguro. As novas capacidades de segurança podem ser categorizadas em duas áreas: características criptográficas e características de controle de acesso.

APÊNDICE C

9. ESPECIFICAÇÃO DA MIB PADRÃO (MIB 1213)

9.1 O que é a MIB 1213

Uma MIB (MILLER, 1999) é uma estrutura que contém as variáveis necessárias para monitorar, gerenciar ou administrar os componentes em redes Internet.

O RFC 1158 propôs a MIB-II, para uso com o protocolo TCP/IP, sendo aceita e formalizada como padrão no RFC 1213. A MIB-II expandiu a base de informações definidas na MIB-I.

A MIB 1213 é uma Base de Informações padrão que contém um conjunto de objetos bem definidos, conhecidos e aceitos pelos grupos padrão da Internet. Duas versões da MIB padrão estão disponíveis hoje em dia, e são chamadas MIB-I (RFC 1156/1066) e MIB-II (RFC 1213/1158).

Abaixo da subárvore MIB II estão os objetos usados para obter informações específicas dos dispositivos da rede. A **MIB II** fornece informações sobre o equipamento gerenciado como por exemplo, o estado da *interface*, informações sobre os protocolos de rede, número de pacotes com erros, número de pacotes enviados e

recebidos etc. Esses objetos são divididos em 11 grupos Para maiores informações sobre a MIB II ver Apêndice A.

Neste trabalho, optou-se pela utilização da MIB-II, mais especificamente a MIB 1213 devido ao fato desta MIB ter sido projetada para monitorar a mídia de rede, em vez de um dispositivo especifico. Portanto, ela é útil no monitoramento do tráfego de rede como um todo.

Para implementar um agente da MIB 1213, em um dispositivo, a interface de rede no dispositivo deve ser capaz de operar no modo promíscuo em que poderá aceitar pacotes não endereçados especificamente para ela. No escopo deste trabalho, não foi necessário configurar o dispositivo de rede para o modo promíscuo, porque os testes de simulação foram realizados na interface *loopback* do dispositivo.

Dentre os grupos de objetos da MIB 1213 que foram utilizados pelas simulações destacam-se o grupo *System* e o grupo *Interfaces*.

Nas seções a seguir, apresenta-se uma descrição de todos os objetos pertencentes aos grupos da MIB 1213, classificados por área funcional.

9.1.1 Grupo *System*

O grupo *System* contém informações sobre o sistema no qual se encontra a entidade gerenciada. Muitos destes objetos são usados no gerenciamento de configuração e gerenciamento de falhas.

A tabela C.1 apresenta os objetos do grupo *System* para Gerenciamento de Configuração.

Objeto	Informação usada no gerenciamento de configuração
SysDescr	descrição do sistema
SysLocation	localização física do sistema
SysContact	pessoa responsável pelo sistema
SysName	nome do sistema

Tabela 9.1 – Objetos do Grupo System para Gerenciamento de Configuração

O *SysDescr* informa a descrição do sistema. Este dado pode ser útil tanto para gerenciar a configuração do dispositivo como para diagnosticar falhas.

Os objetos *sysLocation*, *sysContact* e *sysName* são úteis quando há necessidade de contactar com alguém para um acesso físico a um dispositivo remoto.

Na tabela C.2 são apresentados os objetos do grupo *System* para o Gerenciamento de Falhas.

Objeto	Informação usada no gerenciamento de falhas
SysObjectID	fabricante do sistema
SysServices	qual camada de protocolo o sistema serve
SysUpTime	quanto tempo o sistema está operacional

Tabela 9.2 – Objetos do Grupo System para Gerenciamento de Falhas

O *SysServices* informa quais os níveis do modelo de referência da ISO, o dispositivo serve. Ele retorna a soma dos números de cada camada, usando, para cada camada, a fórmula 2(L-1), onde L é o número da camada. Esta informação é útil para rastrear problemas quando a funcionalidade do dispositivo é desconhecida.

9.1.2 Grupo *Interfaces*

O Grupo *Interfaces* oferece dados sobre cada interface de um dispositivo gerenciável da rede. Essas informações são úteis para o gerenciamento de falhas, de configuração, de performance, e de contabilização.

O objeto *ifTable* contém informações sobre todas as interfaces de uma entidade.

Na tabela C.3 são apresentados os objetos para o Gerenciamento de Falhas.

Objeto	Informação usada no gerenciamento de falhas
ifAdminStatus	indica se a interface esta administrativamente up/down/test
ifOperStatus	indica o status operacional da interface (up/down/test)
ifLastChange	indica quando a interface mudou seu estado operacional

Tabela 9.3 – Objetos do Grupo Interfaces para o Gerenciamento de Falhas

A combinação dos objetos *ifAdminStatus* e *ifOperStatus* determina o status da interface. A tabela C.4 abaixo, apresenta as possíveis combinações.

IfAdminStatus / IfOperStatus	Up(1)	Down(2)	Testing(3)
Up(1)	Operacional	N/A	N/A
Down(2)	Falha	Down	N/A
Testing(3)	N/A	N/A	em teste

N/A - não aplicável.

Tabela 9.4 – Combinação de Objetos do Grupo Interfaces para determinar o Status da Interface

Na tabela C.5 a seguir, são apresentados os objetos para o Gerenciamento de Configuração.

Objeto	Informação usada no gerenciamento de Configuração
ifDescr	nome da interface
ifType	tipo de interface
ifMtu	tamanho máximo do datagrama suportado pela interface
ifSpeed	largura de banda da interface
ifAdminStatus	indica se a interface está administrativamente up/down/test

Tabela 9.5 – Objetos do Grupo Interfaces para o Gerenciamento de Configuração

O objeto *ifSpeed* é um medidor da velocidade da interface em *bits* por segundo. Ele é útil quando se deseja saber a velocidade atual de uma interface que aloca banda passante de acordo com a demanda de tráfego.

O objeto *ifAdminStatus* permite que, através do comando SNMP *Set-Request*, se configure remotamente a interface para *on/off*.

A tabela C.6 apresenta os objetos para o Gerenciamento de Performance.

Objeto	**Informação usada no gerenciamento de Performance**
ifInDiscards	taxa de descartes de entrada
ifOutDiscards	taxa de descartes de saída
ifInErrors	taxa de erros de entrada
ifOutErros	taxa de erros de saída
ifInOctets	taxa de bytes recebidos
ifOutOctets	taxa de bytes enviados
ifInUcastPkts	taxa de pacotes unicast recebidos
ifOutUcastPkts	taxa de pacotes unicast enviados
ifInNUcastPkts	taxa de pacotes no-unicast recebidos
ifOutNucastPkts	taxa de pacotes no-unicast enviados
ifInUnknownProtos	taxa de pacotes de protocolos desconhecidos recebidos
ifOutQLen	total de pacotes na fila de saída

Tabela 9.6 – Objetos do Grupo Interfaces para o Gerenciamento de Performance

Com os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts*, *ifInErrors*, *ifOutErros* , pode-se calcular as porcentagens de erro de entrada/saída.

- porcentagem de erro de entrada = *ifInErrors*/(*ifInUcastPkts* + *ifInNUcastPkts*)
- porcentagem de erro de saída = *ifOutErrors*/(*ifOutUcastPkts* + *ifOutNUcastPkts*)

Da mesma forma, com os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts*, *ifInDiscards*, *ifOutDiscards* , pode-se calcular as porcentagens de descartes de entrada/saída.

O objeto *ifInUnknownProtos* informa o número de descartes realizados devido ao recebimento de pacotes de protocolo desconhecido. Portanto, não há a detecção de nenhum problema, caso o valor dos objetos *ifInUnknownProtos* e *ifInDiscards* estiverem crescendo proporcionalmente.

Com os objetos *ifInOctets* e *ifOutOctets* pode-se calcular a taxa de utilização de uma interface. Para isso, primeiro calcula-se o total de *bytes* recebidos e enviados em um intervalo de tempo entre x e y:

- total de *bytes* = (*ifInOctets*_y - *ifInOctets*_x)+(*ifOutOctets*_y - *ifOutOctets*_x)

Depois calcula-se o total de *bytes* e *bits* por segundo:

- total de *bytes* por segundo = total de *bytes*/(y - x)

- total de *bits* por segundo = total de *bytes* por segundo * 8

E, finalmente, a taxa de utilização:

- taxa de utilização = (total de *bits* por segundo)/*ifSpeed*

Detalhes das variáveis de MIB II usados na fórmula acima:

.1.3.6.1.2.1.2.2.1.10
ifInOctets OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Counter
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "The total number of octets received on the interface, including framing characters."
::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 10 }

.1.3.6.1.2.1.2.2.1.16
ifOutOctets OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Counter
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "The total number of octets transmitted out of the interface, including framing characters."
::= { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 16 }

.1.3.6.1.2.1.2.2.1.5
ifSpeed OBJECT-TYPE
-- FROM RFC1213-MIB, IF-MIB
SYNTAX Gauge
MAX-ACCESS read-only
STATUS Mandatory
DESCRIPTION "An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth."
::= { ISO(1) org(3) DOD(6) Internet(1) mgmt(2) mib-2(1) interfaces(2) ifTable(2) ifEntry(1) 5 }

O objeto *ifOutQLen* indica se o dispositivo está tendo problemas em enviar dados para fora. Seu valor aumenta de acordo com o aumento do número de pacotes esperando para deixar a interface.

Os objetos *ifOutOctets* e *ifOutDiscards* juntos podem sinalizar um congestionamento na rede. Isto ocorre se, caso houver um aumento no valor do *ifOutDiscards* devido ao descarte de muitos pacotes que tentam deixar a interface, e uma diminuição do número total de *bytes* de saída, indicado pelo objeto *ifOutOctets*.

A tabela C.7 apresenta os objetos utilizados para o Gerenciamento de Contabilização.

Objeto	Informação utilizada no gerenciamento de contabilização
IfInOctets	taxa de bytes recebidos
IfOutOctets	taxa de bytes enviados
IfInUcastPkts	taxa de pacotes unicast recebidos
IfOutUcastPkts	taxa de pacotes unicast enviados
IfInNUcastPkts	taxa de pacotes no-unicast recebidos
IfOutNucastPkts	taxa de pacotes no-unicast enviados

Tabela 9.7 – Objetos do Grupo Interfaces para o Gerenciamento de Contabilização

Com os objetos *ifInOctets* e *ifOutOctets*, uma aplicação de gerenciamento de contabilização pode determinar o número de *bytes* enviados e recebidos em uma interface. Se a unidade de contabilização utilizada for pacotes ao invés de *bytes*, são utilizados os objetos *ifInUcastPkts*, *ifOutUcastPkts*, *ifInNUcastPkts*, *ifOutNucastPkts* para calcular o número de pacotes recebidos e enviados.

No contexto deste trabalho, para a análise da sobrecarga do protocolo SNMPv3, foram utilizados os objetos *ifInOctets* e *ifOutOctets* para a obtenção do número de *bytes* recebidos e o número de *bytes* enviados, respectivamente, quando da execução das operações SNMP, conforme descrito no estudo de caso que é apresentado no capítulo V.

9.1.3 O Grupo *Address Translation*

O grupo *Address Translation* não constitui mais um grupo separado. Seus objetos foram incorporados aos grupos de protocolos. Portanto o uso de informações de

conversão de endereços no gerenciamento de redes serão analisados em cada um desses grupos.

9.1.4 O Grupo IP

O IP é um protocolo de rede que utiliza um modo de serviço sem conexão para entregar datagramas. O grupo IP provê informações sobre o protocolo IP na entidade. Estas informações são subdivididas em quatro grupos:

- objetos que informam erros e tipos dos pacotes IP vistos;
- tabela de informação sobre os endereços IP das entidades;
- tabela de roteamento IP da entidade;
- mapeamento de endereços IP para outros protocolos (substituindo o grupo *Address Translation*).

Os objetos do grupo IP podem ser aplicados ao gerenciamento de falhas, de configuração, de performance, e de contabilização.

A tabela C.8 a seguir apresenta os objetos para Gerenciamento de Falhas.

Objeto	Informação usada para gerenciamento de falhas
IpRouteTable	tabela de roteamento IP
IpNetToMediaTable	tabela de conversão de endereços IP

Tabela 9.8 – Objetos do Grupo IP para o Gerenciamento de Falhas

O ipRouteTable é uma tabela que possui as seguintes colunas:

IpRouteDest	endereço IP do destino
IpRouteIfIndex	número da interface
IpRouteMetric1	métrica de roteamento #1
IpRouteMetric2	métrica de roteamento #2
IpRouteMetric3	métrica de roteamento #3
IpRouteMetric4	métrica de roteamento #4
IpRouteMetric5	métrica de roteamento #5
IpRouteNextHop	próxima escala (endereço IP do roteador, usado para roteamento indireto)
IpRouteType	tipo(direto, indireto, válido, inválido)
IpRouteProto	mecanismo usado para determinar a rota
IpRouteAge	idade da rota em segundos
IpRouteMask	máscara da subrede para roteamento
IpRouteInfo	ponteiro MIB para um protocolo de roteamento específico

Tabela 9.9 – Colunas da tabela ipRouteTable do Grupo IP

Todos os objetos contidos no *objeto ipRouteTable* podem ser usados no gerenciamento de falhas. Por exemplo, eles podem ser usados para rastrear problemas de roteamento e de dispositivos que sinalizam informações de roteamento incorreto. Esses objetos permitem que a aplicação de gerenciamento de falhas produza a tabela de roteamento IP para um dispositivo e descubra rotas através da rede. Além disso os objetos *ipRouteType* e *IpRouteProto* informam como a informação de roteamento foi aprendida.

Assim como o *ipRouteTable*, o objeto *IpNetToMediaTable* é uma tabela com as seguintes entradas:

IpNetToMediaIfIndex	número da interface
IpNetToMediaPhysAddress	endereço do meio do mapeamento
IpNetToMediaNetAddress	endereço IP do mapeamento
IpNetToMediaType	como o mapeamento foi determinado (outro, inválido, dinâmico, estático)

Tabela 9.10 – Entradas da tabela IpNetToMediaTable do Grupo IP

Os objetos contidos no objeto *IpNetToMediaTable* informam o mapeamento de endereços IP para endereços em outros protocolos.

A tabela C.11 abaixo apresenta os objetos para Gerenciamento de Configuração.

Objeto	Informação usada para gerenciamento de configuração
IpFowarding	se o dispositivo está configurado para enviar datagramas IP
IpAddrTable	endereços IP do dispositivo
IpRouteTable	tabela de roteamento IP

Tabela 9.11 – Objetos do Grupo IP para o Gerenciamento de Configuração

O objeto IpAddrTable é composto dos seguintes objetos:

IpAdEntAddr	o endereço IP desta entrada
IpAdEntIfIndex	número da interface
IpAdEntNetMask	máscara de sub-rede para endereços IP
IpAdEntBcastAddr	o bit menos significativo do endereço IP de broadcast

Tabela 9.12 – Objetos Derivados do objeto IpAddrTable do Grupo IP

Estes dados são definidos na MIB somente para leitura (*read-only*),portanto a aplicação de gerenciamento de configuração poderá apenas consultá-los e não alterá-los.

Já o objeto *ipRouteTable* possui vários objetos definidos com permissão de leitura e escrita (*read-write*). Dessa forma, uma aplicação de gerenciamento de configuração pode entrar com novas rotas através do objeto *ipRouteDest* e mudar o tipo de uma rota com o objeto *ipRouteType*. Além disso é possível configurar-se as métricas de roteamento através dos objetos *ipRouteMetric1*, *ipRouteMetric2*, *ipRouteMetric3*, *ipRouteMetric4* e *ipRouteMetric5*.

Na tabela C.13 abaixo, são apresentados os objetos para Gerenciamento de Performance.

Objeto	Informação usada para gerenciamento de performance
IpInReceives	taxa de datagramas recebidos
IpInHdrErrors	taxa de erros de cabeçalho de entrada
IpInAddrErrors	taxa de erros de endereço de entrada
IpForwDatagrams	taxa de datagramas repassados
IpInUnknownProtos	taxa de datagramas de entrada para um protocolo desconhecido
IpInDiscards	taxa de datagramas de entrada descartados
IpInDelivers	taxa de datagramas de entrada entregues com sucesso
IpOutRequests	taxa de datagramas de saída (não inclui os datagramas repassados)
IpOutDiscards	taxa de datagramas de saída descartados
IpOutNoRoutes	taxa de descartes ocorridos por falta de informação de roteamento
IpRoutingDiscards	taxa de entradas de roteamento descartadas
IpReasmReqds	taxa de datagramas recebidos necessitando de remontagem
IpReasmOKs	taxa de datagramas com sucesso na remontagem
IpReasmFails	taxa de datagramas com falhas na remontagem
IpFragOKs	taxa de datagramas com sucesso na fragmentação
IpFragFails	taxa de datagramas com insucesso na fragmentação
IpFragCreates	taxa de fragmentos gerados

Tabela 9.13 – Objetos para Gerenciamento de Performance do Grupo IP

A seguir serão apresentadas algumas das formas que estes objetos podem ser utilizados para auxiliar no gerenciamento de performance.

Os objetos *ipInReceives* e *ipOutRequest* juntamente com alguns objetos do grupo *Interface* permitem o cálculo da taxa de tráfego IP de entrada e saída de uma entidade.

- porcentagem de tráfego IP de entrada:

  (ifInUcastPkts + ifInNUcastPkts)/ipInReceives

- porcentagem de tráfego IP de saída:

  (ifOutUcastPkkts + ifOutNUcastPkts)/ipOutRequest

- porcentagem de erros de entrada:

(ipInDiscards+ipInHdrErrors+ipInAddrErrors)/ipInReceives

- porcentagem de erros de saída:

(ipOutDiscards+ipOutHdrErrors+ipOutAddrErrors)/ipOutRequests

O aumento do valor do objeto *ipRoutingDiscards* pode significar que um dispositivo está descartando entradas válidas devido à falta de recursos.

Um freqüente aumento no valor do objeto *ipInUnknownProtos* pode causar problemas de performance, pois os recursos estarão sendo desperdiçados em checagens de erros e determinação do destino, sendo que por fim o datagrama será descartado por ser destinado a um protocolo da camada superior desconhecido.

É possível calcular a taxa de envio e de recepção de datagramas IP por um dispositivo em um intervalo de tempo entre x e y em segundos.

taxa de forwarding = (ipForwDatagrams_y - ipForwDatagrams_x)/(y - x)

taxa de entrada = (ipInReceives_y - ipInReceives_x)/(y - x)

Tendo estas duas taxas pode-se determinar se o sistema está repassando os datagramas IP rápido o bastante para satisfazer os requerimentos da rede. Se os pacotes recebidos estão sendo repassados a outros sistemas, a taxa de entrada e taxa de forwarding devem ser iguais. No entanto para que o cálculo seja mais exato deve-se subtrair da taxa de entrada a taxa de erros e pacotes IP destinados ao próprio sistema.

A tabela C.14 apresenta os objetos para Gerenciamento de Contabilização .

Objeto	Informação usada para gerenciamento de contabilização
IpOutRequests	número de pacotes IP enviados
IpInReceives	número de pacotes IP recebidos
IpInDelivers	número de pacotes IP de entrada entregues com sucesso

Tabela 9.14 – Objetos para Gerenciamento de Contablização do Grupo IP

O objeto *ipInDelivers* informa o número de pacotes IP entregues com sucesso a protocolos de camada superior e aplicações.

9.1.5 O Grupo ICMP

O ICMP é um protocolo que carrega mensagens de erro e controle para dispositivos IP. O grupo ICMP contém objetos que fornecem informações sobre o protocolo ICMP na entidade em questão. Todos os seus objetos são aplicados ao gerenciamento de performance.

Na tabela C.15 são apresentados os objetos do grupo ICMP.

Objeto	Informação usada para gerenciamento de performance
IcmpInMsgs	taxa de recebimento de mensagens
IcmpInErrors	taxa de erros de entrada
IcmpInDestUnreachs	taxa de mensagens de Destino não Alcançado recebidas
IcmpInTimeExcds	taxa de mensagens de Tempo Excedido recebidas
IcmpInParmProbs	taxa de mensagens de Problemas com Parâmetros recebidas
IcmpInSrcQuenchs	taxa de mensagens de Fonte Apagada recebidas
IcmpInRedirects	taxa de mensagens de Redirecionamento recebidas
IcmpInEchos	taxa de mensagens de Ecos (requisição) recebidas
IcmpInEchoReps	taxa de mensagens de Respostas a Ecos recebidas
IcmpInTimestamps	taxa de mensagens de requisição de Timestamp requisição recebidas
IcmpInTimestampReps	taxa de mensagens de respostas ao pedido de Timestamp recebidas
IcmpInAddrMasks	taxa de mensagens de requisição de máscaras de endereços recebidas
IcmpInAddrMaskReps	taxa de mensagens de resposta a mascaras de endereços recebidas
IcmpOutMsgs	taxa de saída de mensagens
IcmpOutErrors	taxa de erros de saia
IcmpOutDestUnreachs	taxa de mensagens de Destino não Alcançado enviadas
IcmpOutTimeExcds	taxa de mensagens de Tempo Excedido enviadas
IcmpOutParmProbs	taxa de mensagens de Problema com Parâmetros enviadas
IcmpOutSrcQuenchs	taxa de mensagens de Origem Apagada enviadas
IcmpOutRedirects	taxa de mensagens de Redirecionamento enviadas
IcmpOutEchos	taxa de mensagens de Eco (requisição) enviadas
IcmpOutEchoReps	taxa de mensagens Respostas de Eco enviadas
IcmpOutTimestamps	taxa de mensagens de requisição de Timestamp requisição enviadas
IcmpOutTimestampReps	taxa de mensagens de respostas ao pedido de Timestamp enviadas
IcmpOutAddrMasks	taxa de mensagens de Requisição de Mascara de Endereços enviadas
IcmpOutAddrMaskReps	taxa de mensagens de Resposta de Mascara de Endereços enviadas

Tabela 9.15 – Objetos do Grupo ICMP

9.1.6 O Grupo TCP

O TCP é um protocolo de transporte que provê conexões confiáveis entre aplicações. Muitas implementações do TCP incluem recursos adicionais para lidar com controle de fluxo, congestionamento da rede, e a retransmissão de segmentos perdidos.

O grupo TCP pode ajudar no gerenciamento de configuração , performance, de contabilização, e de segurança.
Este grupo é subdividido em dois grupos:

- objetos gerais sobre o TCP no sistema ;

- uma tabela de valores para cada conexão TCP corrente, a qual é alterada a cada começo e fim de uma conexão TCP.

Na tabela C.16 são apresentados os objetos para Gerenciamento de Configuração.

Objeto	**Informação usada para gerenciamento de configuração**
TcpRtoAlgorithm	algoritmo utilizado para determinar o "time out" de retransmissão de octetos TCP não confirmados
TcpRtoMin	valor mínimo permitido para o "time-out" de retransmissao TCP, em milissegundos
TcpRtoMax	valor máximo permitido para o "time-out" de retransmissao TCP, em milissegundos
TcpMaxConn	limite de conexões que podem ser abertas pela entidade de transporte do dispositivo
tcpCurrEstab	número de conexões de transporte corretamente abertas

Tabela 9.16 – Objetos para o Gerenciamento de Configuração do Grupo TCP

O objeto *tcpRtoAlgorithm* permite a configuração do algoritmo de retransmissão de *octetos* TCP não confirmados. A configuração inadequada pode resultar em congestionamento na rede ou distribuição injusta de banda passante. Consultando-se freqüentemente os objetos *tcpRtoMin*, *tcpRtoMax* e *tcpRtoAlgorithm* pode-se verificar se a configuração está adequada ao ambiente de seu sistema de rede.

O objeto *tcpMaxConn* ajuda a configurar a rede para suportar o número de conexões TCP remotas necessárias. Este número pode ser calculado observando-se o objeto *tcpCurrEstab* que informa o número de conexões TCP estabelecidas no momento.
Na tabela C.17 são apresentados os objetos para Gerenciamento de Performance.

Objeto	Informação usada para gerenciamento de performance
tcpAttempt Fails	número de tentativas de conexão falhadas
tcp EstsabResets	número de reinicializações de conexões estabelecidas
tcpRetransSegs	número de segmentos retransmitidos
tcpInErrs	número de pacotes recebidos com erro
tcpOutRsts	número de vezes que a entidade tentou reinicializar uma conexão
tcpInSegs	taxa de segmentos TCP recebidos
tcpOutSegs	taxa de segmentos TCP enviados

Tabela 9.17 – Objetos para o Gerenciamento de Performance do Grupo TCP

O objeto *tcpRetransSegs* informa o número de segmentos TCP que o sistema está retransmitindo, esta informação pode indicar se uma entidade está tendo que fazer várias retransmissões para garantir a confiabilidade.

O objeto *tcpInErrs* indica o número de segmentos recebidos com erro. O aumento deste objeto pode ser causado pelo encapsulamento incorreto dos segmentos pelo sistema de origem, alguma rede repassando os segmentos com erro, ou outras razões.

Na tabela C.18 são apresentados os objetos para Gerenciamento de Contabilização.

Objeto	Informação usada para gerenciamento de contabilização
TcpActiveOpens	número de vezes que o sistema abriu uma conexão
TcpPassiveOpens	número de vezes que o sistema recebeu um pedido de abertura de conexão
TcpInSegs	numero total de segmentos TCP recebidos
TcpOutSegs	número total de segmentos TCP emitidos
TcpConnTable	tabela das conexões TCP correntes

Tabela 9.18 – Objetos para o Gerenciamento de Contabilização do Grupo TCP

O objeto *tcpConnTable* é uma tabela com as atuais conexões TCP, e contém os campos conforme mostrado na tabela C.19:

tcpConnState	estado da conexão
tcpConnLocalAddress	endereço TCP local
tcpConnLocalPort	endereço IP local
tcpConnRemAddress	endereço TCP remoto
tcpConnRemPort	endereço IP remoto

Tabela 9.19 – Campos do objeto tcpConnTable do Grupo TCP

O campo *tcpConRemAddress* determina o endereço do sistema remoto que está conectado à entidade. A consulta freqüente a este campo permite o conhecimento de quais sistemas usam os recursos da rede e durante quanto tempo.

Muitas aplicações TCP usam portas bem definidas, tornando possível determinar-se quais aplicações estão fazendo ou recebendo conexões TCP.

As informações da tabela *tcpConnTable* também podem ser usados para gerenciamento de segurança, pois permitem o conhecimento dos sistemas que acessam recursos via TCP. O tempo de *polling* influenciará grandemente na eficiência do gerenciamento, pois um intruso pode levar apenas alguns segundos para pegar as informações que deseja e fechar a conexão. Se nenhum *poll* for feito neste intervalo, o intruso não será detectado.

9.1.7 O Grupo UDP

O UDP é um protocolo de transporte que, ao contrário do TCP, não garante segurança e nem estabelece conexões, ao invés disso ele usa um fluxo de datagramas para transportar as informações. O grupo UDP possui um número limitado de objetos. O grupo UDP pode ajudar no gerenciamento de performance , de contabilização, e de configuração.

Este grupo é subdividido em dois grupos:

- objetos gerais sobre o UDP nesta entidade;

- entradas sobre as aplicações UDP, que estão recebendo datagramas correntemente, na entidade em questão.

A tabela C.20 a seguir apresenta os objetos para Gerenciamento de Performance.

Objeto	Informação usada para gerenciamento de performance
UdpInDatagrams	taxa de datagramas recebidos
UdpOutDatagrams	taxa de datagramas enviados
UdpNoPorts	taxa de datagramas que não foram enviados para uma porta válida
UdpInErrors	taxa de datagramas UDP recebidos com erro

Tabela 9.20 – Objetos para o Gerenciamento de Performance do Grupo UDP

Na tabela C.21 são apresentados os objetos para Gerenciamento de Contabilização.

Objeto	Informação usada para gerenciamento de contabilização
UdpInDatagrams	número total de datagramas UDP recebidos
UdpOutDatagrams	número total de datagramas UDP enviados
UdpTable	portas UDP recebendo datagramas correntemente

Tabela 9.21 – Objetos para o Gerenciamento de Performance do Grupo UDP

O objeto *udpTable* é composto pelos campos conforme tabela C.22:

UdpLocalAddress	endereço IP local
UdpLocalPort	porta UDP local

Tabela 9.22 – Campos do objeto udpTable do Grupo UDP

Como o UDP não é um protocolo baseado em conexão, as entradas da tabela acima são válidas somente para o período em que a aplicação escuta uma porta.

O Gerenciamento de Configuração é realizado checando-se o objeto *udpTable*, com isso, pode-se determinar se as aplicações da entidade estão configuradas corretamente. Por exemplo, se é conhecido que uma entidade tem uma aplicação que oferece impressão remota em uma determinada porta, esta configuração pode ser facilmente verificada usando o objeto *udpTable*.

O objeto *udpTable* também pode ser usado para gerenciamento de segurança. Pode-se consultar este objeto para assegurar que uma entidade não executou uma determinada aplicação. Por exemplo, se determinarmos que uma determinada aplicação escuta requisições por uma porta UDP específica. A ferramenta de gerenciamento pode consultar o objeto *udpTable* de todos os sistemas para verificar se esta porta UDP local está sendo escutada.

9.1.8 O Grupo EGP

O EGP é um protocolo que informa a um dispositivo de rede IP como alcançar outras redes IP. Ele não informa a rota completa para a outra rede, mas ele permite que um dispositivo saiba em que direção que a rede existe. Redes IP podem ser agrupadas em áreas lógicas chamadas sistemas autônomos. Um sistema autônomo geralmente é uma rede e suas sub-redes associadas ou uma coleção de redes e sub-redes sob uma mesma administração. Dois dispositivos de rede em dois sistemas autônomos distintos podem compartilhar informações de alcançabilidade via EGP.

Os dispositivos de rede que comunicam com o EGP entre sistemas autônomos são chamados vizinhos EGP. Cada processo EGP tem uma relação um-para-um com cada vizinho. Cada vizinho EGP conversa um protocolo *hello* que periodicamente informa outros vizinhos que ele ainda está ativo. Quando o sistema consulta a informação de alcançabilidade do vizinho, ele está fazendo um EGP *poll*.
Os objetos do grupo EGP são subdivididos em dois grupos:

- informações sobre o EGP nesta entidade;
- uma tabela de entradas contendo informações sobre cada vizinho EGP.

Os objetos do grupo EGP podem ser aplicados ao gerenciamento de falhas, de configuração, de performance, e de contabilização.
A tabela C.23 apresenta os objetos para Gerenciamento de Falhas.

Objeto	Informação usada para gerenciamento de falhas
EgpNeighState	estado de cada vizinho EGP
EgpNeighStateUps	número de vezes que um vizinho EGP entrou no estado UP
EgpNeighStateDows	número de vezes que um vizinho EGP entrou no estado DOWN

Tabela 9.23 – Objetos para o Gerenciamento de Falhas do Grupo EGP

Os objetos listados acima estão contidos na tabela de vizinhos EGP (objeto *egpNeighTable*).

O estado de um vizinho EGP pode prover informações de como a informação de roteamento é injetada no sistema autônomo. A aplicação de gerenciamento de falhas pode usar o objeto *egpNeighState* para conhecer o estado atual de um vizinho EGP. Se o vizinho EGP está no estado *up*, então ele deverá estar enviando informações sobre alcançabilidade de redes ao processo EGP local.

Sabendo-se quando um vizinho entra no estado *up* pode-se sinalizar sobre novas informações de roteamento que devem entrar no sistema autônomo. Saber quando o vizinho pára a comunicação, e entra no estado *down*, pode ser útil na resolução de problemas de roteamento.

Na tabela C.24 são apresentados os objetos para Gerenciamento de Configuração.

Objeto	Informação usada para gerenciamento de configuração
EgpNeighState	estado de cada vizinho EGP
EgpNeighAddr	endereço IP do vizinho EGP
EgpNeighAs	sistema autônomo do vizinho EGP
egpNeighIntervalHello	intervalo entre retransmissões de comandos *Hello*
egpNeighIntervalPoll	intervalo entre retransmissões de comandos *Poll*
egpNighMode	modo de *polling* desta entidade EGP
egpNeighEventTrigger	permite iniciar ou finalizar uma comunicação
egpAs	sistema autônomo local

Tabela 9.24 – Objetos para o Gerenciamento de Falhas do Grupo EGP

O objeto *egpAs* informa o número do sistema autônomo da entidade EGP local. Os demais objetos estão contidos no objeto *egpNeighTable* e dizem respeito à configuração de um vizinho específico.

O objeto *egpNeighEventTrigger* pode ser usado para iniciar e encerrar uma comunicação com um vizinho EGP (já existente). Este objeto permite o controle do processo EGP do sistema. Este é o único objeto do grupo EGP que pode ser configurado pelo engenheiro de rede através de um comando SNMP *Set-Request*.

A tabela C.25 apresenta os objetos para Gerenciamento de Performance.

Objeto	**Informação usada para gerenciamento de performance**
EgpInMsgs	taxa de mensagens recebidas
EgpInErrors	taxa de mensagens recebidas com erro
EgpOutMsgs	taxa de mensagens enviadas
EgpOutErrors	taxa de mensagens não enviadas devido a ocorrência de erros
EgpNeighInMsgs	taxa de mensagens recebidas deste vizinho EGP
EgpNeighInErrs	taxa de mensagens recebidas com erro deste vizinho EGP
EgpNeighOutMsgs	taxa de mensagens enviadas a este vizinho EGP
EgpNeighOutErrs	taxa de mensagens não enviadas ao vizinho EGP devido erros
EgpNeighInErrMsgs	taxa de mensagens de erro recebidas deste vizinho EGP
EgpNeighOutErrMsgs	taxa de mensagens de erro enviadas a este vizinho EGP

Tabela 9.25 – Objetos para o Gerenciamento de Performance do Grupo EGP

Os objetos *egpInMessages* e *egpOutMessages* permitem calcular a taxa de mensagens EGP que entram e saem da entidade. Geralmente esta taxa será insignificante, mas em alguns momentos de instabilidade da rede entre os vizinhos EGP, essa taxa pode aumentar e influenciar na performance da entidade.

O aumento do valor dos objetos *egpInErrors* e *egpOutErrors* geralmente coincide com o aumento do número de mensagens recebidas e enviadas pela entidade. Se uma mensagem é recebida com erro e uma resposta válida não é enviada, o vizinho EGP originador deverá retransmitir a mensagem. Quando a entidade não pode enviar mensagens EGP válidas devido a limitações de recursos, o valor do objeto *egpOutErros* irá aumentar. Conseqüentemente, quando a taxa de *egpInMessages* aproxima-se da taxa de *egpOutErrors*, a entidade provavelmente estará tendo dificuldades em construir e enviar mensagens EGP.

Da mesma forma, usando os objetos *egpNeighInMsgs*, *egpNeighInErrs*, *egpNeighOutMsgs* e *egpNeighOutErrs*, pode-se calcular a taxa de entrada e saída de mensagens e erros de cada vizinho.

9.1.9 O Grupo CMOT

O Grupo CMOT existe somente por razões históricas. O CMOT é um protocolo que ajuda na transição do SNMP para o CMIS/CMIP. No entanto, embora a definição para o CMOT exista, nenhum trabalho significativo tem sido feito em cima deste protocolo desde algum tempo, e logo não há nenhum objeto neste grupo.

9.1.10 O Grupo Transmission

O Grupo *Transmission* provê informações sobre o meio específico que forma a base das interfaces no sistema. Quando os padrões Internet para gerenciar vários tipos de meios forem definidos, este grupo será o prefixo para estas informações.

9.1.11 O Grupo SNMP

Os objetos do grupo SNMP podem ser aplicados em todas as cinco áreas de gerenciamento. Aplicações de gerenciamento de falhas observando os problemas SNMP podem achar útil conhecer o número de erros SNMP e sua freqüência, enquanto aplicações de gerenciamento performance podem calcular a taxa de pacotes SNMP entrando e deixando a entidade. Já aplicações de gerenciamento de contabilização podem usar os objetos SNMP para encontrar o número de pacotes SNMP enviados ou recebidos pela entidade. ,E por fim, alguns objetos do grupo SNMP podem ajudar no gerenciamento de configuração e segurança .

Na tabela C.26 são apresentados os objetos para Gerenciamento de Falhas.

Objeto	Informação usada para gerenciamento de falhas
SnmpInASNParseErrs	total de mensagens recebidas com erros ASN
SnmpInTooBigs	total de mensagens recebidas com erro "too big"
SnmpInNoSuchNames	total de mensagens recebidas com erro "noSuchName"
SnmpInBadValues	total de mensagens recebidas com erro "badValue"
SnmpInReadOnlys	total de mensagens recebidas com erro "readOnly"
SnmpInGenErrs	total de mensagens recebidas com erro "genErr"
SnmpOutTooBigs	total de mensagens enviadas com erro "too big"
SnmpOutNoSuchNames	total de mensagens enviadas com erro "noSuchName"
SnmpOutBadValues	total de mensagens enviadas com erro "badValue"
SnmpOutGenErrs	total de mensagens enviadas com erro "genErr"

Tabela 9.26 – Objetos para o Gerenciamento de Falhas do Grupo SNMP

Os objetos listados informam erros referentes a mensagens SNMP. Esses erros não indicam erros na rede em si, mas pode informar que a entidade não está manipulando os pacotes SNMP apropriadamente. O número e tipos de erros também podem indicar que a entidade está recebendo pacotes SNMP com erros dos dispositivos da rede. A solução para esses erros geralmente está na configuração do gerente ou agente SNMP. Se a reconfiguração não diminuir o número de erros, o problema provavelmente residirá na implementação do gerente ou agente SNMP.

A tabela C.27 apresenta os objetos para Gerenciamento de Performance.

Objeto	Informação usada para gerenciamento de performance
SnmpInPkts	taxa de pacotes SNMP recebidos
SnmpOutPkts	taxa de pacotes SNMP enviados
SnmpInTotalReqVars	taxa de Get/Get-Next-Requests recebidas
SnmpInTotalSetVars	taxa de Set-Requests recebidas
SnmpInGetRequests	taxa de Get-Requests recebidas
SnmpInGetNexts	taxa de Get-Next-Requests recebidas
SnmpInSetRequests	taxa de Set-Requests recebidas
SnmpInGetResponses	taxa de Get-Responses recebidas
SnmpInTraps	taxa de Traps recebidas
SnmpOutGetRequests	taxa de Get-Requests enviadas
SnmpOutGetNexts	taxa de Get-Next-Requests enviadas
SnmpOutSetRequests	taxa de Set-Requests enviadas
SnmpOutGetResponses	taxa de Get-Responses enviadas
SnmpOutTraps	taxa de Traps enviadas

Tabela 9.27 – Objetos para o Gerenciamento de Performance do Grupo SNMP

Como qualquer outra atividade da entidade, o SNMP pode afetar a performance do sistema. Caso deseja-se conhecer a porcentagem de recursos que uma entidade está usando para manipular o SNMP, pode-se calcular a taxa de pacotes SNMP recebidos ou enviados, usando os objetos snmpInPkts e snmpOutPkts.

Os demais objetos listados na tabela acima permite que se conheça os tipos de pacotes SNMP que a entidade está manipulando.

Na tabela C.28 são apresentados os objetos para Gerenciamento de Contabilização.

Objeto	Informação usada para gerenciamento de contabilização
SnmpInPkts	taxa de pacotes SNMP recebidos
SnmpOutPkts	taxa de pacotes SNMP enviados
SnmpInTraps	taxa de traps recebidas
SnmpOutTraps	taxa de traps enviadas

Tabela 9.28 – Objetos para o Gerenciamento de Contabilização do Grupo SNMP

A tabela C.29 apresenta os objetos para Gerenciamento de Segurança.

Objeto	Informação para gerenciamento de segurança
SnmpInBadCommunityNames	total de pacotes com uma community string incorreta
SnmpInBadCommunityUses	total de pacotes com community string que não permite a operação requisitada

Tabela 9.29 – Objetos para o Gerenciamento de Segurança do Grupo SNMP

O objeto *snmpInBadCommunityNames* conta o número de vezes que um usuário ou aplicação, na tentativa de comunicar-se com o SNMP de uma entidade, não informou a *community string* correta.

Na tabela C.30 é apresentado o único objeto para Gerenciamento de Configuração.

Objeto	Informação usada para gerenciamento de configuração
SnmpEnableAuthenTraps	indica se o agente SNMP pode enviar traps

Tabela 9.30 – Objetos para o Gerenciamento de Segurança do Grupo SNMP

Portanto, apresentou-se o conceito da MIB padrão (MIB 1213) que é uma Base de Informações padrão que contém um conjunto de objetos bem definidos, conhecidos e aceitos pelos grupos padrão da Internet; bem como, a descrição de todos os seus grupos com seus respectivos objetos, que servem para a coleta de dados dos dispositivos gerenciados. Também foram apresentadas as fórmulas para calcular a taxa de utilização de uma interface utilizando-se os objetos *ifInOctets* e *ifOutOctets*.