

**UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**Desenvolvimento e Adaptação de Aplicações Visando a
Gerência com SNMP: Análise de Requisitos e Estudo
de Caso para um Sistema de Impressão**

Dálmer Barbosa de Azevedo Jr.

**Campina Grande - PB
Outubro de 1998**

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Dálmer Barbosa de Azevedo Jr.

**DESENVOLVIMENTO E ADAPTAÇÃO DE APLICAÇÕES VISANDO A
GERÊNCIA COM SNMP: Análise de Requisitos e Estudo de Caso para
um Sistema de Impressão**

Dissertação apresentada ao curso de MESTRADO
EM INFORMÁTICA da Universidade Federal da
Paraíba, em cumprimento às normas para obtenção
do Grau de Mestre.

Jacques Philippe Sauvé
(*Orientador*)



A994d Azevedo Júnior, Dálmer Barbosa de.
Desenvolvimento e adaptação de aplicações visando a gerência com SNMP : análise de requisitos e estudo de caso para um sistema de impressão / Dálmer Barbosa de Azevedo Júnior. - Campina Grande, 1998.
133 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Humanidades, 1998.
"Orientação : Prof. Dr. Jacques Philippe Sauvé".
Referências.

1. Redes de Computadores. 2. Gerência de Redes. 3. Gerência - SNMP. 4. Dissertação - Informática. I. Sauvé, Jacques Philippe. II. Universidade Federal da Paraíba - Campina Grande (PB). III. Título

CDU 004.7(043)

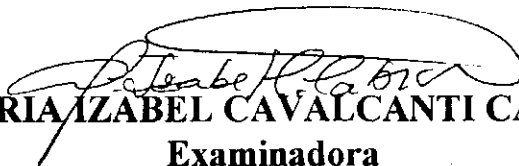
**DESENVOLVIMENTO E ADAPTAÇÃO DE APLICAÇÕES
VISANDO A GERÊNCIA COM SNMP: ANÁLISE DE REQUISITOS
E ESTUDO DE CASO PARA UM SISTEMA DE IMPRESSÃO**

DÁLMER BARBOSA DE AZEVEDO JÚNIOR

DISSERTAÇÃO APROVADA EM 26.10.98



PROF. JACQUES PHILIPPE SAUVÉ, Ph.D
Presidente



PROF^ª MARIA IZABEL CAVALCANTI CABRAL, D.Sc
Examinadora



PROF. JOSÉ NEUMAN DE SOUSA, Dr.
Examinador

CAMPINA GRANDE - PB

A todos aqueles que me ajudaram a terminar esta dissertação. Principalmente meus pais, que se sacrificaram durante a vida e me deram a educação necessária para estar aqui e minha esposa que com amor conseguiu mudar o meu perfil desorganizado e me orientou para finalizar este trabalho.

Agradecimentos

Ao meu orientador, Dr. Jacques Philippe Sauvé, pela orientação, pela experiência transmitida, pelas lições de vida dadas nos diversos encontros que tivemos durante a minha passagem pela Infocon e acima de tudo, por ter sido paciente e ter acreditado que poderíamos fazer, juntos, um bom trabalho.

Meus agradecimentos ao Dr. José Antônio Beltrão Moura, pelos conselhos e orientação dada durante essa árdua jornada. Agradeço também a Fernanda Azevedo, minha esposa, pelos conhecimentos transmitidos, pela experiência, e acima de tudo por todas as horas que ela esteve, literalmente, lado a lado comigo.

Ao meu amigo Marcos Sebastian por ter dividido o tempo comigo e me ajudado a terminar esta dissertação. Agradeço também a todo aquele que direta ou indiretamente esteve comigo nos momentos de prazer ou tristeza: Adriano Sérgio, Adriano Lemos, José Tenório Neto, Alessandro Jatobá, João Carvalho e família, Joni Girard, Alex Lucena, enfim, todos os amigos que fiz no Brasil e nos EUA.

Resumo

A importância de gerência de redes nos dias atuais é incontestável. Redes de computadores estão crescendo em tamanho e complexidade e se torna necessário o desenvolvimento de ferramentas para auxiliar profissionais na tarefa de gerenciar redes de computadores. Esse trabalho apresenta um estudo contendo sugestões práticas e requisitos para o desenvolvimento e/ou adaptação de aplicações que se destinam a ser gerenciadas pelo protocolo SNMP. Foram abordados aspectos como arquitetura para o gerenciamento de redes, protocolos, base de informação e tecnologias emergentes na gerência de redes e aplicações. Com base neste estudo, especificamos a adaptação de um produto no mercado, SpoolView (sistema de impressão distribuído) para ser gerenciado pelo protocolo SNMP, fornecendo assim um subsídio importante ao desenvolvedor de aplicações para a gerência.

Abstract

There is no doubt about the importance of network management. Computer networks are growing in complexity and size. The complexity of such systems dictates the use of automated network tools to help the tasks of network management. This work presents practical advice, considerations and requirements for development and/or modification of software products to SNMP. It includes subjects such as network management, application management, management information base and emerging management technologies. Based on these considerations, we do a practical study, modifying a software product on the market, SpoolView (Network Print Manager), to SNMP. The importance of such a study is to allow the software developer apply all the concepts presented in the dissertation in the application management area.

Lista de Figuras

<i>Figura 2.1. Rede de Computadores</i>	7
<i>Figura 2.2. Quebras de Segurança</i>	16
<i>Figura 2.3. Modelo Agente/Estação de Gerência (Cliente/Servidor)</i>	21
<i>Figura 2.4. A Arquitetura de um Agente Procurador.</i>	23
<i>Figura 2.5. Representação da MIB em Árvore</i>	28
<i>Figura 2.6. Representação de uma Tabela como Árvore de OIDs</i>	31
<i>Figura 3.1. Classificação dos dados Segundo Nível de Informação Fornecida</i>	51
<i>Figura 3.2. Propsta AWG para gerência de Aplicações</i>	52
<i>Figura 3.3. Relacionamento entre MIBs para Gerência de Aplicações</i>	53
<i>Figura 3.4. Relacionamento das Tabelas no Grupo de Canais</i>	67
<i>Figura 4.1. Ciclo de Vida e Estados dos Pedidos</i>	75
<i>Figura 4.2. Configuração Cliente/Impressora – Agente na Impressora</i>	81
<i>Figura 4.3. Configuração Cliente/Servidor/Impressora – Agente no Servidor</i>	82
<i>Figura 4.4. Configuração Cliente/Servidor/Impressora – Cliente Monitora Agente na Impressora e o Servidor.</i>	83
<i>Figura 5.1. A Arquitetura do SpoolView</i>	96
<i>Figura 5.2. Arquitetura AgentX com um Sub-Agente</i>	113

Lista de Tabelas

<i>Tabela 2.1. Indicadores de Desempenho na Rede</i>	11
<i>Tabela 2.2. Identificador da Instância do Objeto sysDescr</i>	30
<i>Tabela 2.3. Localização de uma Instância na Tabela.</i>	31
<i>Tabela 3.1. SysApplInstallPkgTable – Informações sobre os Pacotes de Software no Sistema</i>	59
<i>Tabela 3.2. SysApplInstallElmtTable - Exemplo de uma linha Linha na tabela Tabela de elementos Elementos da Aplicação</i>	59
<i>Tabela 3.3 Tabela de Estados dos Elementos da Aplicação</i>	68
<i>Tabela 3.4 Tabela de Controle de Elementos da Aplicação</i>	68
<i>Tabela 4.1. Mapeamento de um Job LPD para jmJobSubmissionID</i>	87
<i>Tabela 4.2. Resumo dos Atributos da Job Monitoring MIB</i>	92
<i>Tabela 5.1. Comandos do SpoolView Implementados no Agente</i>	99
<i>Tabela 5.2. Índice JmJobSubmissionID para o SpoolView</i>	103
<i>Tabela 5.3. Mapeamento da General Table para o SpoolView</i>	104
<i>Tabela 5.4. Mapeamento da Job Table para o SpoolView</i>	106
<i>Tabela 5.5. Attribute Table mapeada para o SpoolView</i>	107
<i>Tabela 5.6. Controle dos pedido de impressão</i>	107
<i>Tabela 5.7. Controle de Impressoras no SpoolView</i>	108
<i>Tabela 5.8. Tabela de Alertas da SpoolView MIB</i>	110
<i>Tabela 5.9. Tabela de Limiares da SpoolView MIB</i>	111

Sumário

1	INTRODUÇÃO.....	1
1.1	MOTIVAÇÃO	2
1.2	OBJETIVOS DO TRABALHO.....	3
1.3	CONTRIBUIÇÕES.....	3
1.4	ORGANIZAÇÃO DO RESTANTE DA DISSERTAÇÃO	4
2	GERÊNCIA DE REDES E SISTEMAS	5
2.1	O QUE É GERENCIAR REDES	6
2.1.1	<i>Áreas Funcionais da Gerência de Redes.....</i>	<i>7</i>
2.2	GERENCIAR É MONITORAR E CONTROLAR.....	9
2.2.1	<i>Monitoração</i>	<i>9</i>
2.2.2	<i>Controle da Rede</i>	<i>14</i>
2.3	METAS PARA A GERÊNCIA DE REDES	17
2.4	ARQUITETURA PARA GERÊNCIA DE REDES E SISTEMAS.....	18
2.5	SISTEMAS ABERTOS	19
2.6	O PROTOCOLO SNMP.....	20
2.6.1	<i>O Modelo.....</i>	<i>20</i>
2.6.2	<i>Métodos de Coleta de Informações.....</i>	<i>23</i>
2.6.3	<i>Representação dos Dados</i>	<i>25</i>
2.6.4	<i>O protocolo SNMP.....</i>	<i>32</i>
2.7	SUMÁRIO.....	35
3	ANÁLISE DE REQUISITOS PARA A GERÊNCIA DE APLICAÇÕES	37
3.1	O QUE SIGNIFICA GERENCIAR APLICAÇÕES.....	38
3.2	REQUISITOS PARA A GERÊNCIA DE APLICAÇÕES.....	39
3.2.1	<i>Requisitos para a Gerência de Desempenho</i>	<i>39</i>
3.2.2	<i>Requisitos para a Gerência de Faltas.....</i>	<i>42</i>
3.2.3	<i>Requisitos para a Gerência de Contabilidade.....</i>	<i>44</i>
3.2.4	<i>Requisitos para a Gerência de Configuração.....</i>	<i>45</i>
3.2.5	<i>Requisitos para a Gerência de Segurança.....</i>	<i>48</i>
3.2.6	<i>Requisitos para a Aplicação a ser Gerenciada.....</i>	<i>48</i>
3.3	ESFORÇOS PARA A PADRONIZAÇÃO.....	51

3.3.1	<i>Arquitetura</i>	51
3.3.2	<i>Host Resource MIB</i>	53
3.3.3	<i>System Application MIB (SysApplMIB)</i>	56
3.3.4	<i>Application MIB (ApplMIB)</i>	63
3.4	SUMÁRIO	69
4	GERÊNCIA DE UM SISTEMA DE IMPRESSÃO	70
4.1	O QUE É UM SISTEMA DE IMPRESSÃO	71
4.2	REQUISITOS PARA A GERÊNCIA DE UM SISTEMA DE IMPRESSÃO	72
4.2.1	<i>Independência do Protocolo de Impressão</i>	73
4.2.2	<i>Requisitos para o Controle do Sistema de Impressão</i>	73
4.2.3	<i>Requisitos para a Monitoração</i>	74
4.2.4	<i>Requisitos para a Contabilidade</i>	78
4.2.5	<i>Requisitos para o Usuário Final</i>	78
4.2.6	<i>Requisitos para o Administrador do Sistema de Impressão</i>	79
4.2.7	<i>Notificações do Agente</i>	79
4.3	CONFIGURAÇÃO PARA A GERÊNCIA DE SERVIÇO DE IMPRESSÃO	80
4.3.1	<i>Configuração Cliente/Impressora - Agente Localizado na Impressora</i>	81
4.3.2	<i>Configuração Cliente/Servidor/Impressora - Agente Localizado no Servidor</i>	81
4.3.3	<i>Configuração Cliente/Servidor/Impressora - Agente na Impressora - Cenário 2</i>	83
4.4	JOB MONITORING MIB	83
4.4.1	<i>Objetos da Job Monitoring MIB</i>	85
4.4.2	<i>General Table (Tabela de Impressoras)</i>	85
4.4.3	<i>Job ID Table</i>	86
4.4.4	<i>Job Table (Tabela de Pedidos)</i>	87
4.4.5	<i>Attribute Table (Tabela de Atributos)</i>	88
4.5	SUMÁRIO	92
5	GERÊNCIA DE UM SERVIÇO DE IMPRESSÃO VIA SNMP	94
5.1	SPOOLVIEW	95
5.1.1	<i>Arquitetura do SpoolView</i>	95
5.1.2	<i>Objetos Manipulados pelo SpoolView e Seus Atributos</i>	97
5.2	REQUISITOS ESPECÍFICOS PARA A GERÊNCIA DO SPOOLVIEW	98
5.2.1	<i>Controle do SpoolView</i>	98
5.2.2	<i>Traps (Notificações do Agente)</i>	100
5.3	BASE DE DADOS PARA A GERÊNCIA DO SPOOLVIEW	101
5.3.1	<i>Relacionamento com as MIBs da IETF</i>	102
5.3.2	<i>O Grupo de Controle</i>	107
5.3.3	<i>O Grupo de Alertas</i>	109
5.3.4	<i>O Grupo de Limiares</i>	110
5.4	CONSIDERAÇÕES NA IMPLEMENTAÇÃO	111
5.4.1	<i>Arquitetura para a Gerência SpoolView</i>	111

5.5 RESUMO	115
6 CONCLUSÃO.....	116
6.1 AVALIAÇÃO DOS OBJETIVOS DO TRABALHO	117
6.2 CONCLUSÕES FINAIS E TRABALHOS FUTUROS	120
A. A MIB SPOOLVIEW.....	121

1 Introdução

Nos últimos anos, as redes de computadores têm tido especial destaque, quer pela proliferação das redes locais (LAN¹), interligando computadores pessoais, estações de trabalhos, computadores de grande porte (*mainframes*), quer pelo aumento do acesso a redes de longa distância (WAN²) através da Internet, por exemplo, facilitando o processo de comunicação. Elas têm crescido em tamanho, complexidade e heterogeneidade exigindo um grande investimento na área de gerência de redes.

Com o intuito de controlar custos e padronizar ferramentas que possam ser usadas em diversos tipos de produtos (independentes de fabricante), incluindo sistemas de software, ações para a padronização vêm sendo tomadas através do uso de protocolos abertos de gerência e da definição de bases de dados que modelam componentes da rede. Porém, a transição para um ambiente gerenciável não é um processo barato, pois envolve a aquisição de produtos de software para o gerenciamento e um processo de adaptação da base de software instalada na empresa para o protocolo aberto desejado.

Este processo de desenvolvimento e adaptação de aplicações para um protocolo aberto exige um estudo das diversas alternativas existentes no mercado para a gerência de rede. Essas alternativas, em sua maioria, apresentam bases de dados para a gerência das aplicações e das funcionalidades oferecidas por estas.

¹ LAN – *Local Area Network*

² WAN - *Wide Area Network*

1.1 Motivação

Podemos definir gerência de rede, nesse trabalho, como o uso de ferramentas (protocolos, *software*, modelos) que permitam monitorar e/ou configurar recursos (*hardware* e *software*) no ambiente de rede. Estes recursos, em sua maioria, são recursos de *hardware* utilizados para interconectar máquinas na rede. É responsabilidade do gerente de rede monitorar e controlar esses recursos de forma a manter a rede sob certos patamares de desempenho e qualidade.

Porém, imagine uma rede de computadores sem aplicações. As aplicações são os elementos que fornecem serviços aos usuários finais, que fornecem interfaces, que transmitem e recebem as informações. Não existe "rede" sem as aplicações. Logo, gerenciar aplicações é tão importante quanto gerenciar o *hardware* que interconecta as máquinas.

O foco em gerência de aplicações vem aumentando e várias propostas estão surgindo como soluções para o gerenciamento de aplicações. Porém, essas propostas ainda se encontram dispersas na bibliografia. Não existe um trabalho que reúna e discuta de forma clara, os requisitos necessários para tornar uma aplicação gerenciável. Em adição a este problema, as propostas que vêm surgindo são insuficientes para gerenciar qualquer tipo de aplicação. Isto faz com que o programador tenha que definir base de dados adicionais para gerenciar uma aplicação específica. Daí, a necessidade de se discutir e resolver os problemas mencionados.

Devido ao aumento da complexidade das redes e à necessidade de gerenciamento, os protocolos abertos mostram-se como ferramentas indispensáveis para o gerenciamento de rede. Na realidade, eles já se apresentam fortes no gerenciamento de recursos de hardware, principalmente pelo uso do SNMP (*Simple Network Management Protocol*), que se tornou um padrão de fato.

A partir da pesquisa bibliográfica, constatou-se que, atualmente, as informações de caráter prático apresentam-se de forma "mística", não dando ao programador um guia objetivo e lógico para o desenvolvimento de soluções para a gerência de aplicações. Várias soluções vêm sendo propostas pela comunidade científica e estas ainda se apresentam de forma dispersa.

1.2 Objetivos do Trabalho

Esse trabalho se propõe a realizar um estudo na área de gerenciamento de redes, especificamente a gerência de aplicações, dando um enfoque especial no desenvolvimento de soluções para a gerência de aplicações baseados em um protocolo aberto, o SNMP. O seu principal objetivo é apresentar informações que auxiliem e situem o programador nas arquiteturas e modelos para adaptar uma aplicação e torná-la gerenciável. Os objetivos específicos são:

- Apresentar de forma clara o que significa gerenciar aplicações;
- Apresentar as soluções proposta pela IETF (*Internet Engineering Task Force*) para a gerência de aplicações;
- Apresentar os requisitos que nortearam o desenvolvimento das soluções para a gerência de aplicações;
- Mostrar que outras bases de dados de gerência são necessárias para a gerência de aplicações;
- Apresentar os aspectos práticos de como tornar uma aplicação gerenciável;
- Avaliar o estudo apresentado para a adaptação e desenvolvimento de uma aplicação em SNMP, a partir de um estudo de caso.

1.3 Contribuições

Esse trabalho contribui para as áreas de Redes de Computadores e Engenharia de Software com um texto didático e sintético sobre a gerência de aplicações. Especificamente este trabalho contribuiu:

1. Apresentando, de forma clara, em um único texto, as diversas propostas da comunidade científica para a gerência de aplicações, mostrando que estas não são suficientes para a gerência de aplicações;
2. Reunindo requisitos e considerações a serem usadas pelo programador quando desenvolvendo ou adaptando uma aplicação que será gerenciável;
3. Apresentando um estudo de caso que esclarece todos os pontos discutidos na dissertação e que serve de modelo para o desenvolvimento de soluções de gerência para outros tipos de aplicações.

1.4 Organização do Restante da Dissertação

Além deste capítulo, a dissertação possui outros quatro, cuja estruturação é apresentada abaixo, de forma sintética.

O Capítulo 2, um apanhado bibliográfico, apresenta os principais conceitos envolvidos na gerência de redes discutindo as áreas nas quais ela pode ser aplicada. Apresentamos as principais metas buscadas pela gerência de redes e como estas metas podem ser alcançadas. O capítulo é finalizado apresentando um estudo sobre o protocolo SNMP.

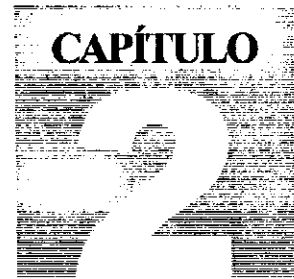
Discutimos gerência de aplicações no Capítulo 3 definindo o que significa gerenciar aplicações. Como contribuição do trabalho, identificamos os principais requisitos para a gerência de aplicações e organizamos estes requisitos baseados nas áreas funcionais de gerência. Apresentamos uma discussão sobre as propostas da IETF para gerência de aplicações e como as bases de dados definidas nestas propostas satisfazem os requisitos apresentados. Por fim, baseados nestas propostas definimos uma arquitetura na qual classificamos as bases de dados de acordo com o nível de detalhes das informações, estendendo assim a proposta da IETF.

No Capítulo 4 aplicamos as idéias apresentadas até o momento num estudo de caso concreto. Pretendemos assim identificar claramente até onde as propostas da IETF podem ser utilizadas para a gerência de aplicações, delineando claramente o que ainda deve ser incluído em soluções não padronizadas. Escolhemos um Serviço de Impressão em rede como estudo de caso. Neste capítulo levantamos os requisitos para a gerência de um sistema de impressão qualquer e apresentamos os esforços da IETF para a padronização de uma MIB para a gerência de sistemas de impressão.

O Capítulo 5 apresenta um estudo de caso envolvendo a adaptação de um sistema de impressão distribuído para ser gerenciado pelo protocolo SNMP. São descritos detalhes do projeto, considerações para a extensão de agentes SNMP e alternativas para a comunicação entre agente e aplicação sendo gerenciada.

O Capítulo 6 se refere às conclusões e contribuições obtidas com o trabalho. Para trabalhos futuros, são apresentadas algumas sugestões envolvendo aspectos que não foram cobertos por este trabalho.

Por fim, não menos importante, formalizamos a MIB SpoolView proposta na dissertação. A listagem completa, em ASN.1, é apresentada no Apêndice A.



2 Gerência de Redes e Sistemas

Os conceitos que serão apresentados neste capítulo têm a intenção de tornar mais cômodo para o leitor o acompanhamento do estudo que será apresentado no restante da dissertação. O objetivo do capítulo não é esgotar o assunto sobre gerência, mas fornecer, através de um texto fácil e compreensível, uma visão geral do processo de gerenciamento de rede e sistemas, bem como disponibilizar referências para o assunto. Ao final do capítulo, teremos respondido a questões como: O que é gerenciar redes, por que gerenciar redes, como gerenciar redes e, por fim, o que queremos alcançar com a gerência de redes.

Com o intuito de fornecer respostas às questões acima, o capítulo foi organizado em torno de dois temas: O que é gerência de redes e sistemas e como a gerência de redes é implementada. Em um primeiro momento, apresentamos a infra-estrutura necessária para o desenvolvimento de aplicações de gerência, e classificamos gerência de redes segundo suas áreas funcionais. A partir do entendimento das áreas funcionais, buscamos agrupá-las em duas categorias de gerenciamento: monitoração e controle. Definimos, então, os elementos envolvidos no processo de gerência, associando-os às arquiteturas e modelos para o gerenciamento de rede. Em um segundo instante, discutimos sistemas abertos, e apresentamos os principais protocolos utilizados para a implementação de aplicações de monitoração e controle do ambiente de rede. Damos um enfoque especial ao protocolo SNMP (*Simple Network Management Protocol*), que vem se apresentando como um padrão para o desenvolvimento de aplicações de gerência.

2.1 O que é Gerenciar Redes

“A primeira coisa que Alex ouviu ao chegar pela manhã ao trabalho, foi pelo menos uma centena de reclamações e uma pilha de notas em seu terminal, sua secretária eletrônica já não mais suportava nenhuma mensagem, além do que, uma multidão já se formava ao redor de seu escritório, e todos reclamavam , ‘a rede está fora’.

Este é outro dia na vida de um engenheiro de rede. Alex, ao passear pelos corredores, sentia a tensão no ambiente, sabia que não se tinha acesso a Brasília, São Paulo, muito menos Pembroke Pines na Florida. Maior parte da rede que interligava a corporação parecia ter desaparecido em um buraco negro. No suporte só se ouvia, ‘você pode ligar mais tarde, o computador está fora’; na outra sala, no departamento de vendas, o pessoal estava em um acirrado jogo de cartas enquanto esperavam que alguma venda aparecesse no computador; as únicas pessoas conscientes eram Alex e toda a diretoria ao seu redor.

Alex sabia como ia ser o seu dia: localizar o problema, isolar o problema, consertar o problema e gerar uma série de relatórios gerenciais mostrando o que aconteceu e porque isso não acontecerá novamente. Com um pouco de sorte Alex vai ter um tempo para tomar uma xícara de café. Antes de pegar as suas ferramentas, Alex colocou uma outra nota em seu monitor. "Investigar sistema de gerência de redes - com urgência!".

O cenário acima, relatado por Allan Leinwand [ALLA 93] é familiar a várias pessoas que lidam com rede. A importância das redes de computadores está na capacidade de fornecer acesso às informações de forma excepcionalmente rápida e eficiente. Hoje, dependemos tanto das redes de computadores que a inoperância em uma rede pode significar a desestruturação em uma empresa, resultando em insatisfação dos usuários ou clientes, demora na recepção de dados críticos para a empresa ou possível perda de lucros. Logo, manter a rede de computadores funcionando é uma fator crítico. Ai é que entra a gerência de redes.

Podemos definir uma rede de computadores como uma coleção de dispositivos e circuitos que pode prover formas para a transferência de dados entre computadores [ALLA 93]. Uma rede de computadores é composta basicamente por três tipos componentes como ilustrado na Figura 2.1 [SAUV 98].

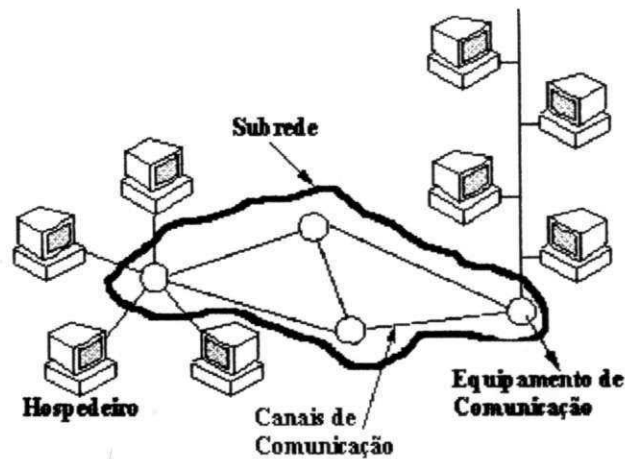


Figura 2.1. Rede de Computadores

- Computadores hospedeiros. São dispositivos (terminais, estações de trabalho, PCs, etc.) que estão nas pontas das redes. Estas máquinas são usuárias da infra-estrutura de comunicação e são responsáveis pela execução das aplicações que oferecem serviços aos usuários.
- Equipamentos de comunicação. São dispositivos utilizados para interligar diferentes redes de forma que um nó de uma rede³ possa se comunicar com outro nó de rede. Estas máquinas provêem serviços de comunicação, mas não executam aplicações finais.
- Canais de comunicação. Os computadores hospedeiros estão conectados na rede através de cabos (canais de comunicação). Estes canais provêem um meio para a transmissão eletrônica de sinais.

Como todo sistema, uma rede de computadores tem que ser gerenciada para prover serviços com qualidade. A qualidade dos serviços vem da constante monitoração e controle da rede. Através disso, é possível detectar faltas eminentes, problemas de desempenho, mal funcionamento de sistemas, etc. e com isso, tomar as providências necessárias para que tais problemas não causem impacto nos serviços prestados pela rede.

2.1.1 Áreas Funcionais da Gerência de Redes

Para alcançar qualidade nos serviços oferecidos pela rede, a ISO (*International Organization for Standardization*) dividiu gerência de redes em áreas funcionais: gerência de faltas, gerência de contabilidade, gerência de configuração, gerência de desempenho e gerência de segurança [ISO 84].

³ Nó de rede. Uma estação de trabalho, servidor de arquivo, servidor de impressão, ou outro dispositivo qualquer que seja endereçável em uma rede.

- **Gerência de Falhas.** Falta é uma condição anormal do sistema que requer reparos ou ação por parte do gerente. Quando uma falta ocorre, é importante determinar exatamente onde ela ocorreu, isolar o resto da rede, reconfigurar ou modificar a rede, de modo a minimizar o impacto da operação de conserto do problema e reparar ou repor o componente que está com problema, retornando a rede ao seu estado inicial.
- **Gerência de Contabilidade.** Essa área permite mensurar o uso de recursos ou serviços da rede, e atribuir o seu uso a determinados grupos de pessoas, de forma que eles possam ser cobrados ou apenas monitorados.
- **Gerência de Configuração.** Redes de computadores utilizam equipamentos complexos (roteadores, por exemplo) que têm de ser configurados corretamente para que possam desempenhar as suas funções. Cabe a gerência de configuração oferecer o ferramental para facilitar a configuração de tais equipamentos.
- **Gerência de Desempenho.** Cabe à gerência de desempenho monitorar e controlar componentes de rede. Monitorar é a função de manter estados das atividades do ambiente. Já a função de controle permite gerenciar o desempenho através de ajustes sobre determinados componentes da rede.
- **Gerência de Segurança.** Essa área diz respeito à geração, distribuição e armazenamento de chaves de acesso aos computadores hospedeiros. Também está relacionada com a monitoração e o controle de acesso às informações contidas nos nós da rede (computadores da rede, equipamentos de comunicação, etc.) por pessoas não autorizadas. O intuito é proteger as informações contidas nestes nós.

Além das áreas definidas pela ISO, duas outras áreas podem ser inseridas dentro das áreas funcionais de gerência de redes: gerência de sistemas e gerência de custos [STEV 94].

- Gerência de sistemas é a gerência e administração dos serviços providos na rede. Muitas implementações deixam de fora esta parte crucial da gerência, que vem mostrando ser uma das áreas em que a gerência de redes mais evoluiu recentemente.
- Gerência de custos é uma área onde estão envolvidos fatores como confiabilidade, operabilidade e manutenibilidade dos objetos que estão sendo gerenciados. Esta função permite determinar o *upgrade* de equipamentos, remoção de serviços não utilizados e ajuste das funcionalidades dos servidores em relação aos serviços providos. Esta área é associada diretamente ao uso, de forma mais adequada e equilibrada, dos recursos financeiros alocados para a gerência de redes.

Como se pode ver, a gerência de redes envolve vários fatores, e estes fatores têm que ser tratados para se ter qualidade nos serviços prestados pela rede.

2.2 Gerenciar é Monitorar e Controlar

A seção anterior apresentou a divisão da gerência de redes, pela ISO, em áreas funcionais. Vamos nesse momento agrupar as áreas funcionais de gerência em duas categorias: monitoração e controle. O entendimento destas categorias de gerenciamento é primordial para o entendimento do que é gerenciar redes e sistemas.

No restante da dissertação utilizaremos o termo gerência de redes para nos referirmos a gerência de todos os recursos (aplicações, máquinas hospedeiras, equipamentos de comunicação, etc.) em um sistema aberto.

2.2.1 Monitoração

Os recursos em uma rede de computadores têm que ser monitorados. Qualquer comportamento anormal de um determinado recurso que prejudique o desempenho da rede, ou dos serviços prestados por essa rede tem que ser detectado e reportado. A função de monitoração representa o guardião da rede, e consiste basicamente da observação e análise de determinados parâmetros de rede, como por exemplo, desempenho e funcionamento de recursos gerenciados. Chiu e Sudama [CHIU 92] sugerem que a monitoração de rede consiste de três áreas:

1. Acesso à informação monitorada. Consiste em definir como é a informação e a forma como ela vai ser obtida dos recursos gerenciados.
2. Projeto dos mecanismos de monitoração. Consiste no estudo da melhor forma de se obter as informações dos recursos.
3. Aplicação das informações monitoradas. A principal questão é definir como as informações serão utilizadas nas diversas áreas funcionais.

Antes de descrevermos a monitoração segundo as áreas funcionais de gerência, é importante pararmos e termos idéia da informação a ser gerenciada. Estas informações são extraídas dos nós gerenciados no processo de monitoração da rede, e, podem ser classificadas da seguinte forma [STAL 93]:

- **Informações Estáticas** - Informações que caracterizam a configuração corrente e os elementos dessa configuração, tais como número e identificação de portas em um roteador. Essas informações não mudam com muita frequência.
- **Informações Dinâmicas** - Informações relacionadas a eventos na rede. A mudança de um estado em um protocolo, por exemplo.

- **Informações Estatísticas** - Informações que podem ser derivadas da dinâmica, tais como o número de conexões por dia em um determinado servidor FTP⁴ (*File Transfer Protocol*).

A monitoração de rede engloba todas as áreas funcionais da gerência definidas anteriormente, mas focaremos em três áreas consideradas de maior importância para a monitoração: monitoração de desempenho, monitoração de faltas e monitoração de contabilidade.

2.2.1.1 Monitoração de Desempenho

A monitoração de desempenho é uma das funções mais importantes no gerenciamento de redes. É através das informações coletadas dos nós gerenciados e da análise destas informações que se pode determinar se a rede, e os serviços que esta rede presta, estão em um nível aceitável de qualidade. O processo de monitoração consiste de coleta de dados, análise dos dados coletados e acionamento de outras funções de gerenciamento, tais como controle de configuração para ajuste do problema, ou gargalo detectado. Porém, como saber se uma determina aplicação ou parte da rede tem problemas de desempenho? O que podemos fazer é selecionar alguns indicadores para mensurar o desempenho; a dificuldade está em selecionar os indicadores apropriados.

Existem diversos indicadores para mensurar o desempenho; porém, podemos quebrar esses indicadores em duas categorias: medidas orientadas a serviços e medidas orientadas a eficiência. A Tabela 2.1, proposta por Terplam [TERP 92] dá uma breve descrição dos principais indicadores agrupados nas duas categorias citadas.

Medidas orientadas a serviços visam manter os serviços prestados pela rede a um certo nível de qualidade com o objetivo de manter os usuários sempre satisfeitos. O custo de atingir os requisitos de qualidade tem que ser mantido baixo, daí a necessidade dos indicadores orientados a eficiência.

⁴ Protocolo utilizado para a transferência de arquivos entre computadores na rede.

Orientadas a Serviços	
Disponibilidade	Porcentagem de tempo que um sistema de rede, um componente ou uma aplicação está disponível ao usuário.
Tempo de Resposta	Quanto tempo leva para uma resposta aparecer no terminal de usuário depois de uma ação que espera por uma resposta.
Precisão	Percentual de tempo em que não ocorrem erros em uma transação de rede.
Orientadas a Eficiência	
Vazão	Frequência na qual um determinado evento (por exemplo, transações, transferência de arquivos) ocorre em uma aplicação.
Utilização	O percentual da capacidade teórica do recurso (exemplo: multiplexador, linha de transmissão) que está sendo usado.

Tabela 2.1. Indicadores de Desempenho na Rede

2.2.1.2 Monitoração de Falhas

O principal objetivo da monitoração de falhas é determinar as falhas tão rápido quanto possível e identificar a sua causa de forma a minimizar o trabalho de reparo. Fica claro que gerenciar falhas envolve três passos: detectar, isolar e corrigir (se possível). Várias são as dificuldades encontradas para realizar estas três tarefas. Vejamos alguns problemas relacionados à observação e detecção de falhas:

- **Faltas não observadas.** Certos problemas são difíceis de serem observados localmente. Por exemplo, existência de *deadlock* entre processos cooperativos distribuídos.
- **Faltas parcialmente observadas.** Uma falta em um nó pode ser observada, mas a observação pode ser insuficiente para determinar precisamente o que

está acontecendo. Por exemplo, um nó pode estar com problemas de desempenho devido a algum protocolo de baixo nível.

- **Faltas em observação.** Mesmo quando observações detalhadas são possíveis, elas podem ser incertas e mesmo inconsistentes, quando associadas com a observação. Por exemplo, a falta de resposta de um determinado dispositivo remoto pode significar que o dispositivo está com mal funcionamento ou que a rede está lenta, o que está causando um atraso na resposta.

Uma vez que a falta é detectada, ela tem que ser isolada para que se possa consertar o problema. Neste momento também temos algumas dificuldades, estas, são citadas abaixo[STAL 93]:

- **Múltiplas causas.** Quando múltiplas tecnologias estão envolvidas, os pontos potenciais de faltas e os tipos de faltas também crescem. A correlação entre faltas é necessária para interpretar as causas corretamente e apresentá-la ao gerente de rede.
- **Interferência entre diagnose e procedimentos locais de recomposição.** Procedimentos locais para conserto de faltas podem destruir evidências da natureza da falta, removendo a possibilidade de uma diagnose.
- **Ausência de ferramentas automáticas.** Testar e isolar faltas, sem ferramentas adequadas, é difícil e pode custar muito para o administrador.

O principal benefício trazido pela monitoração de faltas é o aumento da confiabilidade da rede, dando ao engenheiro ferramentas para a detecção da falta tão logo ela ocorra. Novamente a satisfação do usuário é a meta que se quer atingir com a monitoração. Os usuários esperam que a rede esteja permanentemente disponível; apesar de sabermos que é praticamente impossível uma rede funcionar sem problemas, é função do engenheiro responsável resolver tais problemas de forma que não afete a produtividade do usuário.

A monitoração de faltas oferece uma variedade de ferramentas que provêm informações sobre como está a rede. Idealmente estas ferramentas podem determinar exatamente onde a falta está ocorrendo, relatar tal problema, e, até mesmo consertá-lo. Muitos métodos para automatizar a identificação de faltas têm sido utilizados. Dentre eles, o mais comum chama-se correlação de eventos, ou de alarmes⁵.

Correlação de eventos consiste da interpretação de múltiplos eventos gerados na rede tratando esse conjunto de eventos como uma unidade, com um novo significado.

⁵ Um alarme consiste de uma notificação sobre a ocorrência de um evento específico, que pode ou não representar um erro [MEIR 97].

As técnicas de correlação de eventos procuram identificar padrões ou tendências nos alarmes gerados de forma que múltiplas manifestações de uma única falta possam ser tratadas como uma unidade [FREI 98].

As anormalidades que ocorrem durante as operações da rede provocam a emissão automática de alarmes, as quais são recebidas na estação de gerência. Cabe ao sistema que implementa correlação de eventos identificar e diagnosticar a falta ocorrida⁶. Uma das razões de se utilizar técnicas de correlação de eventos é diminuir o número de informações redundantes apresentadas ao engenheiro de rede.

2.2.1.3 Monitoração de Contabilidade

Monitoração de contabilidade evolve mensurar a utilização dos recursos de rede pelos usuários com o objetivo de se estabelecer medidas de consumo, cotas e cobrança dos serviços. O processo de monitoração de contabilidade consiste de três passos:

1. Obter informações da utilização dos recursos;
2. Ajustar cotas utilizando medidas para isso, e;
3. Cobrar o usuário pelo uso dos recursos da rede.

Em suma, monitoração de contabilidade é o processo de coletar estatísticas sobre a rede. Estas informações são úteis, pois ajudam o engenheiro de rede tomar decisões com relação a alocação de recursos de rede. Tais informações estatísticas também podem ser usadas para a gerência de recursos do sistema, tais como, espaço em disco, uso de CPU, uso de impressoras, etc.

O principal benefício da monitoração de contabilidade é a capacidade de se relatar informações de contabilidade (uso de recursos) usados por um indivíduo ou grupo de usuários, e então, cobrar por tais recursos. Cobrar aos usuários obviamente é essencial para se recuperar os investimentos feitos na rede, porém mais que isso, saber quem utiliza o quê na rede é importantíssimo, pois permite utilizar tais informações no processo de planejamento de capacidade da rede, política de segurança, *design* da rede, entre outras áreas. Por fim, outro benefício, é que examinando métricas e cotas pode-se assegurar que o usuário tem recursos suficientes para cumprir as suas tarefas.

⁶ Diagnóstico de falta é uma etapa no processo de gerência de faltas que consiste em descobrir qual a causa original para os sintomas (representados pelos alarmes) recebidos [MEIR 97].

2.2.2 Controle da Rede

Parte da gerência de redes que está ligada à modificação de parâmetros sobre os nós gerenciados. Essas mudanças têm como efeito a execução de ações predefinidas. As duas áreas de gerenciamento que apresentam maior importância a categoria de controle são: controle de configuração e controle de segurança, as quais são discutidas adiante.

2.2.2.1 Controle de Configuração

Controle de configuração diz respeito à inicialização, manutenção, e desligamento de componentes individuais e sub-sistemas lógicos dentro de uma configuração de máquinas e recursos [STAL 93]. Mais especificamente, é o processo de se obter informações e usar estas informações para gerenciar a configuração dos dispositivos de rede. O processo consiste de:

- Obter informação sobre a configuração atual da rede;
- Usar as informações coletadas no processo de monitoração para modificar a configuração da rede, e;
- Armazenar as informações, manter e atualizar informações de inventário, e produzir relatórios baseados nas informações coletadas.

O principal benefício da gerência de configuração é que ele permite ao gerente de rede ter um maior controle sobre o comportamento dos dispositivos de rede, este controle é alcançado devido ao acesso rápido às informações de configuração dos dispositivos distribuídos. Outra possibilidade é o uso da comparação da configuração atual da rede com configurações previamente armazenadas no sistema, facilitando a criação de novas configurações de rede quando necessário.

É fácil ver como a gerência de configuração pode ser utilizada em uma situação real. Considere uma empresa que possui uma rede geograficamente distribuída e que um dos serviços primordiais nessa empresa seja impressão, e que o número de pedidos de impressão nessa empresa chegue à casa dos milhares por dia. Alex, o nosso engenheiro é responsável pela gerência da rede de impressão. Ele nota, através da monitoração das impressoras que existe algum problema no "Macro", uma das suas maiores lojas. Uma das impressoras não está dando vazão ao número de pedidos. Através da monitoração (desempenho e faltas) ele pôde identificar exatamente qual é a impressora em questão. Estudando a configuração de sua rede ele verifica qual a impressora mais próxima e que pode ser usada para escoar tais pedidos. Finalmente,

com a gerência de configuração, ele modifica alguns parâmetros em seu software de impressão que dispara um comando para a criação de um filtro e desvio de parte dos pedidos de impressão para a impressora menos sobrecarregada.

O exemplo acima, mostra claramente o trabalho em conjunto das duas categorias de gerenciamento, inicialmente monitoração e em um segundo momento configuração. Fica claro então que a modificação de certos atributos que representam um determinado nó gerenciado é responsável por uma determinada ação a ser tomada na rede. Desta forma, o gerente de rede, fazendo uso dos recursos de monitoração, pode determinar o que deve ser modificado em um nó de rede para que uma determinada ação seja tomada.

2.2.2.2 Controle de segurança

O controle da segurança das informações dentro de uma organização vem se tornando cada vez mais importante. Com a introdução dos computadores, redes e sistemas distribuídos, a necessidade de ferramentas automáticas para a proteção de arquivos e outras informações tornou-se evidente. A segurança de rede, surge como o meio de proteger as informações encontradas nos dispositivos da rede (equipamentos de comunicação, máquinas hospedeiras) bem como as informações que trafegam nas redes de computadores.

O controle da segurança é a parte do gerenciamento responsável pela segurança de computador, segurança de rede e de recursos sendo gerenciados na rede.

A quebra de segurança em um sistema de computador é melhor ilustrada quando imaginamos um cenário de envio e recebimento de informações. Podemos então mostrar as diversas categorias de quebra de segurança (ver Figura 2.2 [STAL 93]):

- **Interrupção.** A informação não consegue chegar ao seu destino, devido à interrupção do fluxo normal de informações.
- **Interceptação.** Alguém não autorizado ganha acesso ao sistema e consegue “ouvir” a comunicação.
- **Modificação.** Alguém não autorizado não apenas ganha acesso, mas também tem a capacidade de modificar a informação recebida.
- **Mascaramento.** Alguém não autorizado simula ser parte do sistema, gerando a sua própria informação para o destino.

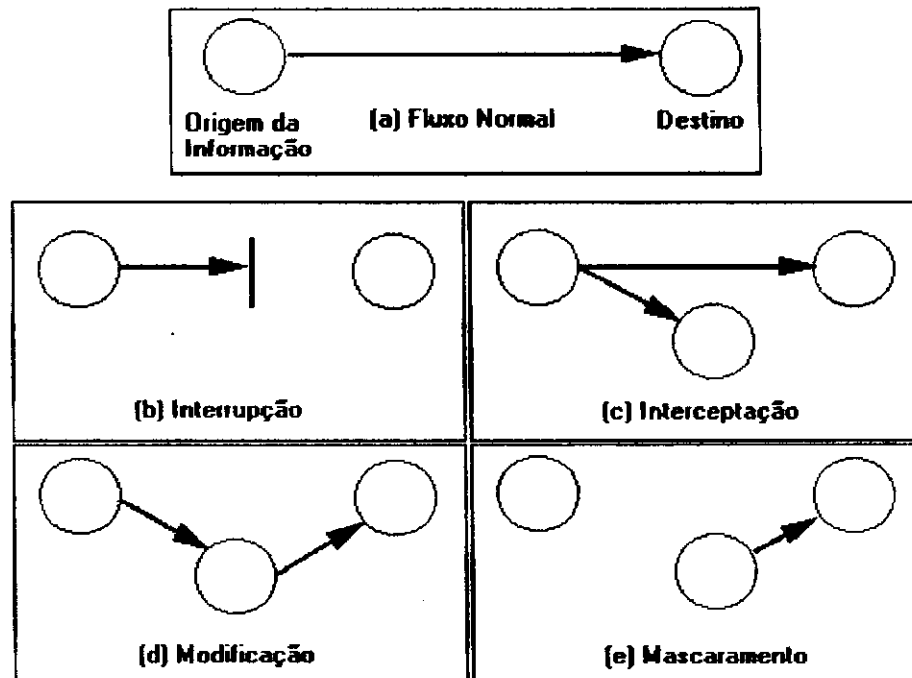


Figura 2.2. Quebras de Segurança

O esquema de segurança deve suprir funcionalidades de forma a tentar prevenir os ataques acima mencionados. Estas funcionalidades incluem a manutenção da segurança da informação, o controle de acesso a recursos e o controle do processo de criptografia.

A idéia de manutenção da segurança é manter uma monitoração constante sobre as informações e nós gerenciados, mantendo um *log* de todos os acessos ou tentativas de acesso, com o objetivo de detectar e se recuperar de possíveis ataques.

O controle de acesso diz respeito à identificação das entidades ("Quem é você?") autenticação dos mesmos ("Será que você é quem diz ser?") e autorização para utilização dos serviços ("O que você pode fazer?"). Essa função deve ser projetada para proteger uma série de recursos no sistema, como por exemplo: códigos de segurança, diretórios, tabelas de roteamento, arquivo de senhas, etc.

Outro fator que deve ser contemplado por um esquema de gerenciamento de segurança é a criptografia ("Se alguém capturar essa informação, não poderá decifrá-la") de informações trocadas entre as entidades de rede, quando necessária. Esta função envolve algoritmos de criptografia e a distribuição de chaves para os mesmos.

Enfim, na área de controle de segurança a responsabilidade do sistema de gerenciamento é coordenar e controlar os mecanismos de segurança. Esses mecanismos

de segurança têm como função proteger o usuário e os recursos do sistema, incluindo o próprio sistema de gerenciamento.

2.3 Metas para a Gerência de Redes

As metas que a gerência de redes pretende alcançar podem ser resumidas nos tópicos abaixo:

Maior disponibilidade dos recursos da rede. O gerenciamento de rede visa garantir uma maior disponibilidade dos dispositivos sendo gerenciados, através do uso constante de monitoração (faltas, desempenho) dos dispositivos e ajustes através da função de controle.

Redução de custos nas operações de rede. Ultimamente, a redução de custos é o principal motivo por trás da gerência de redes. Como a tecnologia muda rapidamente, é interessante ter um sistema de gerência que possa ser utilizado para gerenciar redes heterogêneas.

Redução dos gargalos na rede. A redução de gargalos na rede pode ser feita através da monitoração e ajuste de componentes na rede, bem como na escolha do modelo de gerência (centralizado ou distribuído) que melhor se adequa a rede.

Aumento da flexibilidade de operação e integração. Tecnologias de rede estão mudando constantemente. Com a adoção de padronizações na gerência de redes é possível absorver tais tecnologias com um custo mínimo. Por exemplo, a manipulação de impressoras em rede é muito comum, porém gostaríamos de integrar periféricos externos como copiadoras e máquinas de fax ao nosso esquema de gerência.

Aumento da eficiência. Em alguns casos, as metas na gerência de redes sobrepõem-se. Fica claro que aumentamos a eficiência geral da rede quando metas como redução do custo operacional, aumento da disponibilidade dos componentes de rede, entre outras são atingidas.

Facilidade de uso. A interface final para o gerente de rede é crítica para o sucesso de uma plataforma de gerenciamento. A plataforma de gerência deve oferecer uma interface onde a curva de aprendizado não seja tão acentuada.

Segurança. Algumas funções de gerenciamento precisam de características de segurança. Podemos fornecer segurança em dois níveis: a nível de computador e a nível de rede. Segurança a nível de computador consiste da proteção das informações nos

sistemas de software, enquanto segurança a nível de rede provê esquemas de segurança para as informações que trafegam na rede e para os nós sendo gerenciados.

2.4 Arquitetura para Gerência de Redes e Sistemas

A seção 2.2 apresentou uma descrição do que é a gerência de redes. Continuamos, aqui, descrevendo como podemos implementar a gerência de redes de forma a atingir as metas colocadas na seção 2.3. O primeiro ponto a se observar é que a gerência de redes trata de objetos (roteadores, canais de comunicação, etc.) distribuídos em uma rede de comunicação. Portanto deve haver algum tipo de troca de informação entre as partes envolvidas e esta informação de gerência deverá trafegar pela rede [SAUV 98].

Um sistema de gerenciamento é uma coleção de ferramentas para a monitoração e controle da rede com as seguintes características:

- Uma única interface de operações, com um poderoso e amigável conjunto de comandos para realizar a maioria ou todas as tarefas de gerência.
- Quantidade mínima de equipamentos distintos, isto é, o *hardware* e *software* necessário para a gerência de redes devem ser incorporados dentro do equipamento do usuário.

Um sistema de gerência é projetado para ver a rede como uma arquitetura integrada, com endereços e identificadores associados a cada elemento de rede, estes possuindo seus atributos específicos. Cabe a estes elementos sendo gerenciados, alimentar constantemente a estação de gerência com informações.

Fica claro que os nós gerenciados (computadores hospedeiros, equipamentos de comunicação, etc.) sabem da existência de problemas. Por exemplo, um sistema de impressão "sabe" que uma impressora não está funcionando. Porém, quem trata destes problemas? Dentro dos nós gerenciados, existe um software ao qual chamamos de agente. Este software é responsável pelas informações contidas nos nós gerenciados.

A segunda parte no processo de gerência é a estação de gerência de redes (*Network Management Station* ou NMS). NMS é um equipamento que executa um programa chamado gerente. O gerente normalmente possui uma interface gráfica única utilizada para acessar os agentes. Entre as funções do gerente destacamos: comunicação com os agentes, coletar e apresentar os dados dos nós gerenciados, bem como armazenar informações e gerar gráficos e relatórios.

Temos então duas peças em nossa arquitetura, agentes e gerentes, mas o que se passa entre eles? Na realidade o tempo todo gerentes e agentes conversam sobre a gerência dos nós, essa conversa utiliza um protocolo de gerência (o protocolo mais popularmente utilizado é o SNMP).

Os agentes mantêm variáveis sobre os nós gerenciados; essas variáveis formam uma estrutura de dados que modela o nó sendo gerenciado. Por exemplo, um agente que gerência um computador hospedeiro deve manter variáveis sobre dispositivos na máquina, sistemas de arquivos, aplicações instaladas na máquina, etc. essas variáveis modelam recursos associados a um hospedeiro. Chamamos a esse conjunto de variáveis MIB (*Management Information Base*) ou base de informação de gerência.

A idéia, então, é simples. Temos em uma arquitetura típica um gerente (rodando na NMS) que se comunica com vários agentes (que representam os nós gerenciados). Este gerente solicita aos agentes determinados valores de variáveis (número de pedidos em uma fila de impressão, taxa de transmissão, etc.). A aplicação de gerência pode manipular essas informações e utilizá-la nas diversas áreas de gerência.

Existem outros modelos, ainda não muito populares, de implantar a gerência de forma distribuída para atingir escalabilidade [STAL 93, CARR 94, DOUG 93, NEUM 95, PRAS 95]. Não entraremos em detalhes sobre gerência distribuída de rede, pois foge ao escopo do trabalho.

2.5 Sistemas Abertos

A integração de sistemas heterogêneos tem se tornado cada vez mais complexa, quer seja pela interoperabilidade entre diversos domínios organizacionais, cada um com suas necessidades e políticas de gerenciamento próprias, quer seja pela heterogeneidade dos componentes de diferentes fornecedores, dentro de um mesmo ambiente. Como consequência, temos o aumento do custo na operação e manutenção do sistema.

Uma proposta de solução para o problema da heterogeneidade é a adoção de plataformas de gerenciamento. Estas plataformas definem um conjunto de APIs (*Application Programming Interface*) que abstraem do usuário múltiplos protocolos proprietários ou padronizados, através da implementação de serviços comuns de gerência. Porém, existe uma complexidade muito grande no desenvolvimento e integração das diversas APIs.

Outra alternativa é a adoção de sistemas abertos (arquiteturas, funções, estrutura de informações, protocolo de gerenciamento, etc.) por parte dos fabricantes.

Hoje em dia, a maioria dos produtos de gerência disponíveis no mercado são baseados em dois padrões: SNMP (*Simple Network Management protocol*), tecnologia Internet, ou CMIP (*Common Management Information Protocol*), da arquitetura OSI. O protocolo CMIP fornece um modelo mais poderoso (abrangente) de gerenciamento que o SNMP, porém, a sua eficácia vem se tornando duvidosa, devido à complexidade do protocolo.

O protocolo SNMP obteve popularidade devido a sua simplicidade. Por ser um protocolo mais simples, ele não exige tantos recursos do ambiente; entretanto, ele não se apresenta como o protocolo ideal para a gerência de redes, devido às deficiências para representar objetos complexos e por não possuir um esquema de segurança eficiente.

Devido à simplicidade do protocolo, por ser o mais utilizado, e à disponibilidade de ferramentas, escolhemos o SNMP como protocolo para o desenvolvimento da nossa solução de gerência.

2.6 O Protocolo SNMP

O Protocolo SNMP foi introduzido como padrão Internet em 1988 e, desde então, tem se tornado a base para a gerência de dispositivos heterogêneos em redes locais ou WANs (*Wide Area Network*). Ele foi criado para suprir às necessidades de gerência que vinham aumentando com o crescimento da Internet [STEV 94].

Antes de entrarmos no estudo mais detalhado do protocolo SNMP, introduziremos os principais conceitos, formando basicamente o modelo de gerenciamento e representação dos dados propostos pelo conjunto que forma o protocolo de gerência.

2.6.1 O Modelo

O modelo utilizado pelo SNMP é definido em [RFC1157] como sendo o modelo agente/estação de gerência. Um agente em SNMP é um software capaz de responder a pedidos de uma estação de gerência sobre informações definidas em uma base de dados de gerência (uma MIB). Temos então, um agente para cada nó gerenciado que fornece informações para a estação de gerência. Para que esse modelo,

essencialmente cliente/servidor, funciona, agentes e estação de gerência têm que falar a mesma língua, no nosso caso SNMP. A Figura 2.3 mostra o relacionamento entre agentes SNMP e estação de gerência.

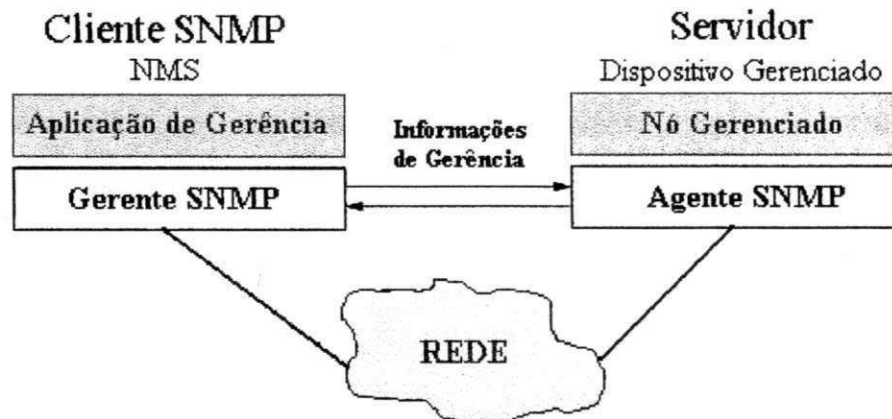


Figura 2.3. Modelo Agente/Estação de Gerência (Cliente/Servidor)

Com base no modelo mostrado acima, vamos descrever as entidades que fazem parte de todo o esquema de gerência em SNMP. Estas entidades são: nós gerenciados, a estação de gerência, o protocolo e as informações de gerência [STEV 94].

Nós Gerenciados

Um nó gerenciado refere-se a um dispositivo de algum tipo, podendo ser:

- Algum tipo de *hardware* como estações de trabalho, servidores de terminais, ou impressoras;
- Um sistema de roteamento (roteadores);
- Um dispositivo de interconexão de rede como pontes, *hubs* ou multiplexadores;
- E mais especificamente, em nosso caso, sistemas de software.

Um agente SNMP é um *software* que normalmente reside no sistema gerenciado, como um servidor de arquivo ou um *hub*. Esta entidade é responsável por receber pedidos de gerenciamento e interpretá-los, executando-os sobre o sistema gerenciado⁷ e retornando com a resposta apropriada. Em adição, o agente pode reportar ao gerente situações que foram pré-definidas como não sendo situações normais do sistema.

⁷ Sistema gerenciado corresponde a um conjunto de nós gerenciados

Estações de Gerência

Numa visão simplista, uma estação de gerência é um computador que contém aplicações que realizam tarefas de gerenciamento, mantendo uma comunicação constante com os seus agentes. Normalmente, a estação provê uma interface para o gerente de rede, centralizando assim todas as atividades de gerência.

Poderíamos definir melhor uma estação de gerência como sendo um sistema com uma ou mais aplicações que estão interagindo com os agentes através de um protocolo de gerência. O protocolo provê o mecanismo para o gerenciamento, enquanto que as aplicações determinam a política usada para esse gerenciamento [ROSE 91].

O Protocolo de Gerência de Rede

O protocolo de gerência, SNMP, é utilizado para transferir informações entre as estações de gerência e os diversos agentes distribuídos pela rede. O SNMP utiliza o paradigma de depuração remota (*Remote Debugging*), no qual a estação de gerência pode inspecionar ou alterar variáveis sobre uma base de dados do sistema gerenciado.

Em adição à leitura e escrita, segundo o paradigma de depuração remota, fazem-se necessárias mais duas operações:

- Uma operação para percorrer a base de dados, chamada *Traversal*, que permite a estação de gerência determinar quais são as variáveis que compõem o nó gerenciado através de uma varredura sobre a estrutura que representa os nós gerenciados;
- E uma operação de *Trap*, que permite um nó gerenciado reportar um evento extraordinário.

Gerência por Procuração

O uso de SNMP requer que ambos, agentes e estação de gerência, utilizem o conjunto de protocolos Internet TCP/IP. Este fator limita os dispositivos a serem gerenciados, pois todos eles precisam conhecer TCP/IP para serem gerenciados utilizando o SNMP. Isto não tem sido problema, visto que TCP/IP se tornou um padrão de fato nas grandes redes.

Para suportar dispositivos que não implementam SNMP, o conceito de *proxy*⁸ foi empregado. Neste esquema (ver Figura 2.4 [NEUM 95]), o agente comporta-se como

⁸ Proxy (procurador) - Peça de software que representa um objeto gerenciado que não entende SNMP. Esse software normalmente entende SNMP e o protocolo proprietário do nó gerenciado.

um tradutor, mediando a conversa entre a estação de gerência e os dispositivos que não implementam o SNMP.

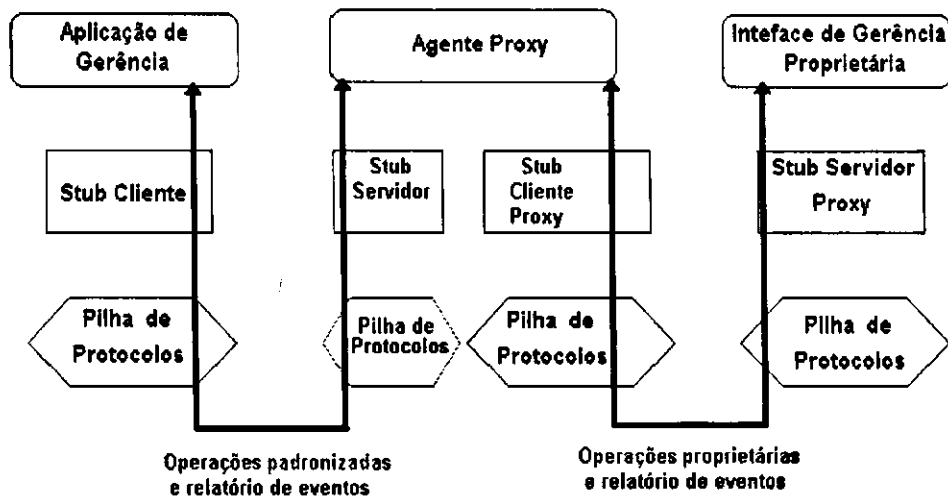


Figura 2.4. A Arquitetura de um Agente Procurador.

Em suma, na arquitetura SNMP temos uma coleção de estações de gerência e diversos nós de rede. As estações executam as funções de gerência, monitorando e controlando os diversos nós de rede. Entre estas duas peças importante, está o protocolo SNMP, que é usado para trocar informações de gerência entre as estações e os agentes.

Não ficou claro porém, onde estão as informações sobre os nós gerenciados, qual o formato para tal estrutura de dados, e como estas informações podem ser coletadas. Nas próximas seções apresentamos dois métodos de acesso às informações e a seguir discutiremos outra peça chave no modelo SNMP, definida como a base de dados de informação de gerência ou simplesmente MIB.

2.6.2 Métodos de Coleta de Informações

Basicamente as estações de gerência coletam informações dos agentes utilizando *polling* que é uma interação do tipo pedido/resposta entre o gerente e o agente SNMP, ou *trap*, que é uma comunicação iniciada pelo agente, nesse caso o gerente assume uma posição de ouvinte, e recebe um relatório do agente informando ao gerente o seu estado corrente.

Várias são as discussões geradas em torno de qual o método que deve ser usado, não entrando no mérito da discussão lembramos um ponto importante a ser comentado,

no caso de *polling* qual o tempo ótimo para se questionar periodicamente o agente sobre o seu *status*?

2.6.2.1 Tempos de Polling

Devido à limitação do número de *traps* definidas pelo protocolo SNMP, a maioria das informações adquiridas pelo gerente é através de *polling*. Devido ao *polling* ser feito apenas no momento de inicialização ou em resposta a uma *trap*, a estação de gerência pode ter uma visão desatualizada da rede.

Segue que, precisamos definir uma política para a frequência com que a estação de gerência deve fazer *polling*. Esta por sua vez, está relacionada com o tamanho da rede e o número de agentes que podem ser efetivamente gerenciados pela estação de gerência. É difícil estabelecer a performance da estação, pois esta depende de diversos fatores como velocidade de processamento, a velocidade de transferência de vários segmentos da rede, o nível de congestionamento na rede, entre outros. Entretanto, podemos prover uma fórmula que dará uma idéia do valor escalar para esse tempo.

Para simplificar o problema, vamos supor que uma estação possa apenas lidar com um agente por vez. Isto é, quando um gerente requisita algo ao agente, ele não faz outra coisa até ter terminado com o agente. Nós podemos determinar o número máximo de agentes que a estação pode gerenciar, considerando a situação em que a estação de gerência está engajada *full-time* no *polling* [STAL 93].

$N \leq T/\Delta$, onde N é o número de agentes na estrutura, T é o intervalo de *polling* desejado, que é o tempo levado entre os *pollings* para um mesmo agente, e Δ é o tempo médio para uma simples requisição. A quantidade Δ depende de diversos fatores:

- Tempo que a estação de gerência leva para gerar um pedido;
- *Delay* do gerente ao agente;
- Tempo de processamento do agente para interpretar a mensagem;
- Tempo de processamento do agente para gerar a resposta;
- *Delay* do agente ao gerente;
- Tempo de processamento do gerente para receber e interpretar a resposta;
- Número de pedidos/respostas trocadas para se obter toda a informação desejada do agente.

2.6.2.2 Traps

Se uma estação é responsável por vários agentes, e cada agente mantém diversos recursos, torna-se impraticável para a estação colher informações dos agentes para todo objeto sendo gerenciado. Para contornar esse problema, temos uma técnica conhecida como *trap-direct polling* ou *event report*.

A estratégia é a seguinte: quando inicializado, e talvez em intervalos não tão frequentes, a estação deve coletar as informações de seus agentes, a partir de agora, ela ficará aguardando notificações dos agentes sobre qualquer evento não usual. Cabe ao agente reportar tais eventos às estações de gerência quando os mesmos acontecem.

Trap-directed polling pode representar uma economia muito grande dos recursos de rede e tempo de processamento dos agente. Temos que considerar que a rede não foi feita para carregar informações de gerência que as estações não necessitam, e agentes não foram desenvolvidos para responder aos pedidos desnecessários.

2.6.3 Representação dos Dados

A base de informação de gerência (MIB) é um modelo de dados que define precisamente as informações que estão acessíveis utilizando um protocolo de gerência. Usando uma estrutura hierárquica, a MIB modela, através de variáveis, cada dispositivo sendo gerenciado.

A [RFC 1155⁹] descreve a sintaxe e o tipo de informação disponíveis na MIB para a gerência de redes TCP/IP, ou seja, regras simples para a criação de diferentes tipos de informações dentro da base de dados de gerência.

Necessitamos, porém, de um novo formalismo para representar essas estruturas complexas trocadas na camada de aplicação do modelo OSI [COMM 91]. Este novo formalismo é denominado Sintaxe Abstrata, que é usado para definir dados sem considerar a plataforma com que se está trabalhando (independência de sistema) [ROSE91].

No SNMP utilizaremos um subconjunto de uma linguagem OSI chamada *Abstract Syntax Notation 1 (ASN.1)* com dois propósitos:

- Definir o formato das informações trocadas pelo protocolo de gerência; e,

⁹ RFC1155 - Structure and Identification of Management Information for TCP/IP Networks (SMI). Define regras para a criação de diferentes tipos de dados dentro da MIB.

- Definir as estruturas de dados que representam os nós gerenciados.

Portanto, a Sintaxe Abstrata é utilizada para descrever as estruturas de dados trocadas pelo protocolo, junto com a informação gerenciada, que é representada através dessas estruturas de dados. Não entraremos em detalhes sobre a linguagem ASN.1, porém temos a necessidade de discutir alguns conceitos que serão usados mais adiante.

Identificador de Objeto (*Object Identifier*)

Em ASN.1 as informações estão distribuídas segundo um modelo baseado em árvore. Cada peça de informação nessa árvore é um objeto¹⁰ que contém as seguintes informações:

1. Um identificador de objeto (OID).
2. Um texto curto que define esse objeto.

Os identificadores de objetos (OIDs) são uma seqüência de inteiros, não negativos, separados por um ponto e organizados hierarquicamente em árvore [DAVI 93].

Para facilitar a utilização destes identificadores, associamos nomes aos componentes de um OID. OIDs são escritos em um dos seguintes formatos:

```
Sintaxe:
  "{ " { { <nome> ["("<número>")" ] } | <número> ]... " }"
Ou
<número> [ "." <número> ] ...
```

onde:

- Os valores literais são especificados entre aspas, por exemplo "ready-only";
- Três pontos são utilizados para indicar que o item que o precede pode se repetir uma ou mais vezes;
- Colchetes são utilizados para delimitar itens opcionais;
- Chaves são utilizadas para agrupar itens;
- Barra vertical indica uma escolha entre itens,
- <nome> é um componente alfanumérico;
- <número> é um inteiro não negativo.

¹⁰ Em nosso contexto um objeto SNMP é uma estrutura de dados identificado unicamente por um OID e que representa um nó gerenciado.

Exemplo: {iso org(3) dod(6) internet(1) } ou 1.3.6.1. Em ASN.1 a forma de se declarar este identificador seria : internet OBJECT IDENTIFIER ::= {iso org(3) dod(6) 1 }.

Como podemos ver na Figura 2.5, o objeto Internet possui o identificador de objeto 1.3.6.1. Este valor serve como prefixo para os objetos que estão mais abaixo na hierarquia. Ex.: mgmt é identificado como 1.3.6.1.2 ou iso org(3) dod(6) internet(1) mgmt(2).

A MIB-I, primeira versão da MIB para uso com o protocolo TCP/IP, explica e define as informações necessárias para monitorar e controlar redes baseadas em TCP/IP [RFC1156]. A segunda versão veio com a MIB-II formalizada através da [RFC1213], estende as informações definidas na MIB-I. Utilizaremos o termo MIB quando se referindo a uma base de dados qualquer.

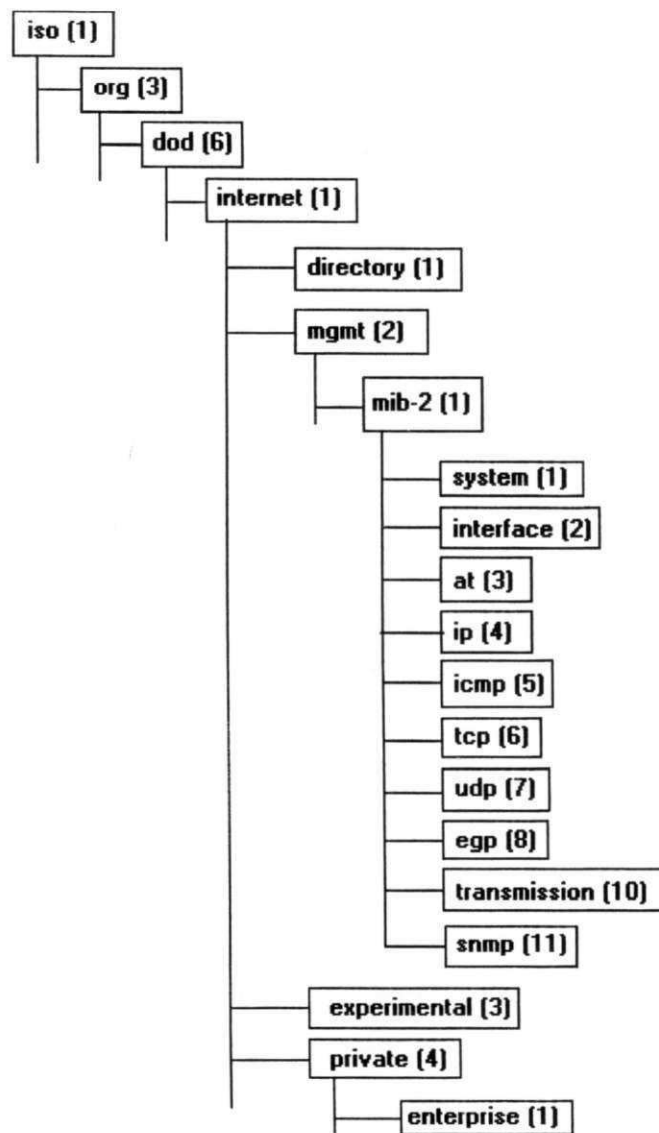


Figura 2.5. Representação da MIB em Árvore

A árvore da MIB designa uma estrutura hierárquica, nas quais podemos destacar os objetos abaixo do identificador Internet:

1. **directory:** Esta sub-árvore é reservada para uso futuro com o sistema de diretório da OSI (X.500).
2. **mgmt:** Esta sub-árvore é reservada para objetos definidos nos documentos aprovados pela IAB¹¹ (*Internet Activities Board*).
3. **experimental:** Esta sub-árvore é usada para identificar objetos usados em experimentos na Internet.

¹¹ IAB - Entidade que aprova padronizações na Internet.

2.6.3.1 Management Information Base (MIB)

O alicerce para um sistema de gerência baseado em TCP/IP é um banco de dados contendo informações sobre os nós a serem gerenciados. Chamamos este banco de dados de MIB (*Management Information Base*). Cada nó a ser gerenciado é modelado como um objeto e armazenado na MIB. A idéia é que, em cada nó gerenciado, tenhamos uma MIB que será uma abstração deste nó. Por exemplo, as informações contidas na MIB de um roteador modela os estados do roteador naquele momento. Uma estação de gerência poderá monitorar estes nós lendo os valores na MIB, e controlá-los pela modificação destes valores.

Uma base de dados é definida segundo um conjunto de regras chamado SMI¹² (*Structure of Management Information*) [RFC1155]. O SMI define como as informações de gerência são agrupadas e nomeadas; quais operações e tipos de dados são permitidos; e qual a sintaxe para a definição das MIBs. Poderíamos compará-lo a um esquema de banco de dados, que define o modelo dos objetos gerenciados e as operações que podem ser aplicadas sobre os objetos; bem como os tipos de dados que são permitidos para estes objetos [DAVI 93].

Um objeto da MIB é uma abstração com sintaxe e semântica. A sintaxe define como uma instância do objeto apresenta-se na rede. A semântica define o que o objeto representa naquele momento [WW06 95]. Fazendo um paralelo com a linguagem C, uma estrutura que represente um nó gerenciado em ASN.1 nada mais é do que uma *struct* composta por diversos tipos básicos. A vantagem do uso do ASN.1 é a não dependência de plataforma.

2.6.3.2 Considerações sobre Instâncias

Foi visto que todo objeto na MIB tem um identificador único, que é definido pela posição do objeto na árvore da MIB. Entretanto, quando um acesso é feito a MIB via SNMP, é uma instância específica de um objeto que se quer e não o seu tipo [STAL 93]. Onde são armazenadas e como são recuperadas as instâncias dos objetos definidos na MIB?

Antes de entrarmos em detalhes sobre a identificação de instâncias em uma MIB se faz necessário definir alguns conceitos. Alguns objetos na MIB (Ex.: descrição do

¹² O SMI é um subconjunto do ASN.1 munido de algumas macros e tipos pré-definidos

sistema) possuem uma única instância, enquanto outros (Ex.: conexões de rede) possuem múltiplas instâncias. Objetos relacionados que possuem o mesmo tipo de instância são organizados na MIB em tabelas conceituais. A identidade de um objeto associado com a sua instância é chamado variável SNMP.

A identificação de uma instância para uma variável SNMP é feita pela concatenação de um zero ao fim de seu OID. Por exemplo a variável SNMP "sysDescr"¹³ que não está em uma tabela é (ver Tabela 2.2):

Iso	Org	dod	internet	Mgmt	mib	System	Sysdescr	Instância
1	3	6	1	2	1	1	1	0

Tabela 2.2. Identificador da Instância do Objeto sysDescr

Ou 1.3.6.1.2.1.1.1.0 ou sysDescr.0.

Para objetos em tabelas os OIDs não são suficientes para identificar as instâncias: existe uma instância de cada objeto para cada linha na tabela. O SMI coloca que as instâncias de objetos não são definidas na MIB. A referência a uma instância de objeto é feita por um protocolo específico, que deve ser definido pelo desenvolvedor.

Porém, SNMP define algumas técnicas para identificar uma instância específica. A primeira técnica é acesso serial baseado em ordem lexicográfica. A outra, é acesso randômico. Por questões de simplicidade e para facilitar o entendimento, veremos primeiro o acesso randômico.

A Figura 2.6 representa uma tabela de roteamento. O objeto TabelaRoteamento identifica a tabela como uma todo, o objeto LinhaTabela identifica uma linha da tabela que por sua vez é formado por três objetos SNMP:

- IndiceRot - Inteiro, não negativo, que identifica unicamente uma linha da tabela;
- EndLocal - Endereço IP local;
- EndRemoto - Endereço IP de destino.

Cada objeto nessa descrição possui o seu próprio OID. Se a estrutura representa uma tabela, como identificamos múltiplas instância das linhas da tabela?

¹³ Variável na MIB-I que contém a descrição do sistema.

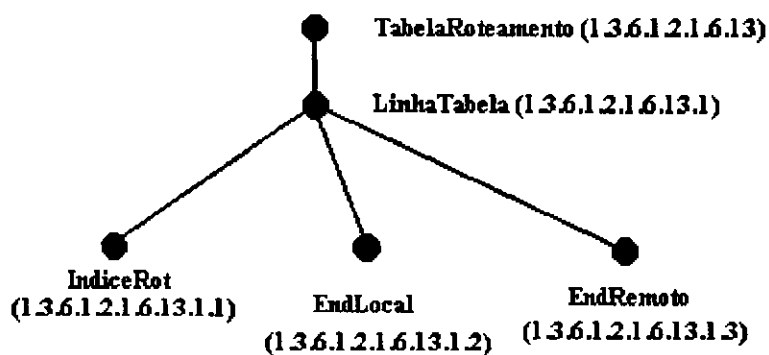


Figura 2.6. Representação de uma Tabela como Árvore de OIDs

O SMI define a cláusula INDEX que identifica quais os objetos na tabela cujos valores serão utilizados para distinguir uma linha de outra. Logo, a combinação de um OID para a localização da coluna na tabela concatenado com um conjunto de valores dos objetos INDEX podem ser usados para recuperar um objeto em uma tabela [STAL 93].

É fácil vermos como localizar uma instância através de um exemplo. Considere a mesma tabela de roteamento. Suponha que precisamos saber que endereço remoto (rota) esta associada a entrada de indice 2. O OID para EndRemoto é 1.3.6.1.2.1.6.13.1.3, o valor do índice de interesse, no nosso caso, é 2. Logo, o identificador de instância do endereço remoto correspondente a linha que contém o valor do objeto IndiceRot igual a 2 é 1.3.6.1.2.1.6.13.1.3.2. Nós apenas concatenamos o valor do objeto IndiceRot ao identificador do objeto EndRemoto (Ver Tabela 2.3).

TabelaRoteamento (1.3.6.1.2.1.6.13)			
IndiceRot (1.3.6.1.2.1.6.13.1.1)	EndLocal (1.3.6.1.2.1.6.13.1.2)	EndRemoto (1.3.6.1.2.1.6.13.1.3)	LinhaTabela (1.3.6.1.2.1.6.13.1)
1	202.10.4.1	150.165.2.1	LinhaTabela (1.3.6.1.2.1.6.13.1)
2	202.10.6.10	150.165.10.1	LinhaTabela (1.3.6.1.2.1.6.13.1)

Tabela 2.3. Localização de uma Instância na Tabela.

De forma mais simples, a identificação de uma instância na tabela é feita através da concatenação do identificador da coluna que você quer recuperar com os valores dos objetos índices para a linha que se está buscando.

2.6.3.3 Percorrimento Sequencial e Ordem Lexicográfica

Um OID é uma sequência de inteiros que reflete uma hierarquia ou uma estrutura de árvore dos objetos de uma MIB. Dada uma MIB, o OID de um objeto em particular pode ser derivado traçando-se um caminho da raiz até o objeto.

Devido os objetos se apresentarem como uma sequência de inteiros, eles podem ser ordenados lexicograficamente. A ordem lexicográfica é conseguida organizando-se a árvore de forma que os galhos de cada nó seja arranjado em ordem crescente da esquerda para direita. Com esta convenção podemos definir o percorrimto de uma árvore da MIB de forma recursiva como:

1. Visite a raiz.
2. Leia as sub-árvores da esquerda para a direita.

A principal razão para o uso de ordem lexicográfica na MIB é porque uma estação de gerência pode não saber como os objetos estão dispostos e que tipo de acesso ela possui. A estação de gerência precisa de algum mecanismo para acessar ou procurar por elementos na MIB sem a necessidade de conhecer o seu OID. Com o uso da ordem lexicográfica, a estação de gerência pode percorrer toda a estrutura da MIB, basta para isso, fornecer um identificador de instância para o objeto e solicitar pelo objeto que seja o próximo na ordem. Veremos na próxima seção que o protocolo SNMP oferece um comando específico utilizado para percorrer uma árvore.

2.6.4 O protocolo SNMP

O protocolo SNMP define o mecanismo de troca de mensagens entre os gerentes e os agentes da rede. O SNMP provê um conjunto de operações simples que podem ser utilizadas sobre os agentes com o intuito de manipular a(s) MIB(s).

A forma como o protocolo trabalha é bastante simples: ele troca informações através de mensagens a qual chamaremos de mensagem SNMP. Uma mensagem SNMP além de outros elementos, carrega consigo uma PDU (*protocol data unit*). Uma PDU identifica que tipo de serviço e também parâmetros utilizados nestes comandos.

Em SNMP, as operações são aquelas que permitem a alteração ou inspeção de variáveis na MIB. Especificamente quatro operações de propósito geral são permitidas:

1. Get: Uma estação de gerência solicita o valor de um objeto a um agente.

2. Set: Uma estação de gerência solicita a modificação do valor de um objeto a um agente.
3. Trap: Operação que permite ao agente, em situações pré-definidas, enviar uma mensagem à estação de gerência.
4. GetNext. Uma estação solicita o próximo objeto na MIB baseado na ordem lexicográfica.

Segundo [STAL 93], as restrições nos comandos SNMP, simplificam enormemente a implementação de aplicações; por outro lado, fraquezas como a segurança limitam o escopo no desenvolvimento de aplicações.

Antes de discutirmos o protocolo, é importante vermos qual a política de autenticação e autorização propostas pelo SNMP.

2.6.4.1 Segurança no SNMP

Existe a necessidade de agentes protegerem suas MIBs de acessos não desejados. O SNMP possui uma capacidade limitada e primitiva para tratar segurança [RFC1157]. Ele introduz o conceito de comunidade.

A comunidade define um relacionamento entre um agente SNMP e uma ou mais estações de gerência; enfim, define características de acesso, autenticação e *proxy* [STAL 93]. A cada comunidade é dada um nome (nome da comunidade) que é único dentro deste agente. A cada estação dentro desta comunidade é fornecido este nome que deve ser apresentado sempre que qualquer operação seja feita.

A gerência de redes pode ser vista como uma aplicação distribuída, envolvendo como elementos chaves, estações de gerência e agentes, ligados pelo SNMP. Devido à estrutura distribuída, devemos nos preocupar com a política de liberação de informações da MIB. Esta política possui três aspectos:

1. Serviço de autenticação. Autenticar é ter a certeza de que você é quem diz realmente ser. O SNMP, como definido em [RFC1157], possui um esquema de autenticação trivial, onde toda mensagem de uma estação de gerência para um agente possui o nome da comunidade, este funciona como uma senha. A mensagem é assumida ser autêntica caso o nome da comunidade, armazenado na mensagem coincida com o dos agentes.

2. Política de acesso: Pela definição de uma comunidade, o agente limita o acesso a sua MIB para um determinado conjunto de estações de gerência. Com o uso de diversas comunidades o agente pode prover diferentes formas de acesso a MIB às diversas estações. Este controle de acesso tem dois aspectos importantes: vista da MIB (define quais objetos são visíveis a uma determinada estação de gerência) e modo de acesso que limita o tipo de acesso fornecido a estação de gerência baseado na comunidade que ele possui (um modo de acesso é definido para cada comunidade, podendo ser READ-ONLY ou READ-WRITE).
3. Serviço de *Proxy*: Um agente pode agir como um *proxy* para outras estações de gerência.

A segurança no SNMP nada mais é de que um esquema que combina identidade e senha no nome da comunidade. Quem possui o nome da comunidade estará ao mesmo tempo identificado, autenticado e autorizado. É fácil de se perceber o quão fraco é este esquema.

2.6.4.2 Comandos SNMP

O SNMP é um protocolo de aplicação pelo qual as variáveis da MIB de um agente podem ser inspecionadas ou alteradas [RFC1157]. A comunicação entre as entidades do protocolo é caracterizada pela troca de mensagens representadas por uma mensagem SNMP). Uma mensagem SNMP consiste de um identificador de versão, o nome da comunidade SNMP, e uma Unidade de Dados de Protocolo (PDU).

Um comando do SNMP é a forma de interação entre uma entidade de gerência e seus respectivos agentes. Eles se traduzem em mensagens que são transmitidas e recebidas no processo de comunicação.

SNMP Get. O comando SNMP Get inicia o processo de leitura de informação de um agente remoto, utilizando uma PDU GetRequest. O resultado da operação, é uma resposta do agente com o valor de uma determinada variável da MIB [ROSE 91, STAL 93].

SNMP GetNext. A operação GetNext é utilizada para recuperar elementos da MIB sem a necessidade de se ter um OID do objeto. Esta operação é utilizada para fazer uma busca através de uma sub-árvore da MIB, com a utilização da ordem

lexicográfica. Para cada variável contida na PDU, a instância lexicograficamente [ROSE 91, STAL 93] posterior é retornada com o seu respectivo valor.

SNMP Set. O comando *SNMP set* inicia o processo de escrita em uma variável na MIB. A função do *set* é alterar o valor de uma variável SNMP através de um agente em uma determinada MIB. Uma característica importante do comando *set* é a unicidade. O comando *set* permite que se possa alterar os valores de mais de uma variável em um único comando. Caso o *set* falhe para uma das variáveis, as outras não serão modificadas (unicidade do comando, ou seja, “ou todas ou nenhuma”).

Traps. As *traps* são notificações feitas pelos agentes para às estações de gerência em situações extraordinárias. Elas devem ser definidas no momento da construção da MIB, notificando quais são as *traps* permitidas para aquele agente. A característica mais importante na *trap* é a capacidade de se adicionar novos tipos de *traps* em adição às definidas pelo protocolo.

2.7 Sumário

"Alex gastou a manhã escrevendo os últimos relatórios sobre o incidente que praticamente parou a rede de sua empresa. No processo de limpar o seu monitor, que para variar estava cheio de notinhas amarelas, Alex encontrou uma nota que dizia – ‘Investigando sistema de gerência de redes -- com urgência!’.

Vendo que a rede parecia relativamente estável e que não existia nenhuma tarefa para cumprir no momento, resolveu fazer exatamente o que dizia a sua nota. Pegou algumas revistas sobre gerência de redes na biblioteca, sentou e leu.

Horas depois, Alex descobriu que gerência de redes era muito mais que apenas fazer as coisas vermelhas no seu monitor virarem verde."

Uma vez que uma rede é instalada, ela tem que ser gerenciada para maximizar o seu potencial. Esta gerência é feita através de um processo ao qual chamamos gerência de redes, que consiste de cinco áreas funcionais: gerência de faltas, gerência de desempenho, gerência de contabilidade, gerência de configuração e gerência de segurança. Para auxiliar no processo de gerência de redes, os engenheiros usam o que chamamos sistema de gerência de redes que é um conjunto de software distribuído e organizado através de uma arquitetura bem definida. Esse sistema se utiliza de um

protocolo de comunicação. Atualmente SNMP vem sendo o mais utilizado para a gerência de redes.

3 Análise de Requisitos para a Gerência de Aplicações

Nas bibliografias disponíveis sobre gerência, é comum abordar o assunto sob uma perspectiva da infra-estrutura do que é gerenciar redes. Por exemplo, fala-se em gerência de faltas sob a visão de que é necessário detectar faltas físicas em dispositivos para o bom andamento da rede. Esquece-se, porém, de dar uma maior importância às aplicações que estão rodando nos computadores hospedeiros e que fornecem serviços aos usuários (e-mail, acesso ao banco de dados corporativo, videoconferência, etc.). A qualidade e disponibilidade dos serviços estão intimamente ligadas à produtividade e satisfação do usuário. A satisfação do usuário final é a meta que tentamos atingir com a gerência de redes, não apenas performance, tolerância a falhas, segurança ou outra característica intrínseca da rede. O relacionamento usuário/aplicação (fornecedora de serviços) justifica um estudo mais apurado das técnicas e métodos utilizados para o gerenciamento das aplicações.

Nesse capítulo, discutiremos a gerência de aplicações de forma geral e sua importância. Baseado nos estudos, definimos os requisitos para uma solução de gerência de aplicações, dividindo o tema em requisitos para as áreas funcionais de gerência (desempenho, faltas, contabilidade, configuração e segurança) e requisitos para a aplicação gerenciada. As informações adquiridas a partir destes requisitos poderão ser utilizadas para a gerência de qualquer aplicação. Particularmente em nosso estudo, apresentaremos uma proposta para a gerência de um sistema de impressão; este assunto será abordado em capítulos subsequentes. Por fim, descreveremos o que está sendo

desenvolvido pela comunidade científica para a padronização da gerência de aplicações¹⁴.

3.1 O que Significa Gerenciar Aplicações

A configuração, detecção de faltas, monitoração de performance e controle de aplicações durante o seu ciclo de "vida" em um computador são de grande importância econômica, pois as aplicações são as entidades que fornecem os serviços que mantêm os usuários produtivos. O que queremos garantir com a gerência de redes é que os serviços estejam disponíveis e sejam oferecidos aos usuários dentro de um certo padrão de qualidade. Porém, a gerência não é aplicada diretamente aos serviços, mas sim às entidades ativas que fornecem tais serviços: as aplicações.

Nesse trabalho, definimos uma aplicação como sendo uma ou mais unidades de código executável ou arquivos relacionados a esses módulos executáveis, instalados em um computador, e que podem ser vistos, externamente, como um único objeto a ser gerenciado [RFC2287].

Não existe uma definição formal do que significa "gerenciar aplicações". Porém, podemos dizer que gerenciar aplicações significa assegurar que os serviços fornecidos por estas, em um determinado ambiente, estão "no ar", disponíveis ao usuário e executando de acordo com suas especificações de desempenho e qualidade. Essa meta é conseguida pela constante automação¹⁵ do processo de gerência e monitoração de aplicações, banco de dados, serviços Internet e recursos que afetam a operação e disponibilidade dos serviços oferecidos. Em outras palavras, gerenciar aplicações significa ter uma visão global de todos os recursos importantes e necessários para manter uma organização funcionando, bem como ser capaz de detectar e responder a faltas antes que estas afetem a produtividade dos usuários.

Na próxima seção, serão apresentados os requisitos básicos que devem ser considerados no projeto de soluções que se destinem à gerência de aplicações.

¹⁴ Note que gerência de aplicação difere de aplicação de gerência. Aplicação de gerência é um software que está executando na estação de gerência e é responsável pelo processamento dos dados colhidos dos agentes.

¹⁵ Busca de ferramentas que auxiliem o processo de gerência.

3.2 Requisitos para a Gerência de Aplicações

Ao projetar uma solução voltada para a gerência de aplicações, o projetista deve considerar um conjunto básico de requisitos que podem ter impacto direto na qualidade da implementação. Os requisitos estão divididos em três grupos: requisitos das áreas funcionais de gerência (estas áreas foram discutidas no capítulo 2), requisitos da aplicação sendo gerenciada e requisitos da base de informações a ser utilizada para gerenciar a aplicação. Iniciaremos então com as áreas funcionais (desempenho, falta, contabilidade, configuração e segurança).

3.2.1 Requisitos para a Gerência de Desempenho

No contexto desse trabalho, uma solução de gerência que contempla a gerência de desempenho provê a habilidade de relacionar aplicações com métricas para a medida de desempenho. As tarefas para a gerência de desempenho incluem:

- Monitoração do tempo de resposta;
- Monitoração do fluxo de informações (vazão);
- Precisão;
- Monitoração da utilização de recursos pela aplicação;
- Planejamento de capacidade;
- *Baselines* de desempenho;

3.2.1.1 Monitoração do Tempo de Resposta

O problema com o tempo de resposta representa uma frustração tanto para o usuário da aplicação, quanto para o administrador. Para o usuário, o tempo de resposta é um fator que afeta diretamente sua produtividade. É inadmissível para ele que os serviços fornecidos pelas aplicações não estejam funcionando eficientemente. Para administradores, a maior frustração é que eles, na maioria das vezes, não saberão que está ocorrendo algum problema, até que os telefones comecem a tocar.

A monitoração do tempo de resposta está relacionada diretamente com o desempenho da aplicação, ou seja, a que nível de qualidade as aplicações estão executando. Uma solução de gerência deve utilizar-se dessa monitoração para identificar o que está causando o excesso de atraso. Por exemplo, em uma aplicação tipicamente cliente/servidor, no caso de uma falta relacionada a tempo de resposta,

temos que identificar se o servidor está no ar, porém ocupado realizando outras tarefas, ou se ele não está executando. Com as causas identificadas, é possível utilizar outras áreas de gerência para resolver a falta. Para o usuário, certamente, só restarão duas perguntas: Está funcionando? Quanto tempo leva para consertar?

3.2.1.2 Monitoração do fluxo de Informações (Vazão)

Vazão é uma métrica para a medida de desempenho que procura relacionar uma unidade de trabalho (tarefa) na aplicação com o tempo para realizar esta tarefa. Como exemplo podemos citar:

- Número de transações de um dado tipo durante um certo período de tempo;
- Número de canais de comunicação abertos para uma dada aplicação durante um certo período de tempo;
- Número de bytes transmitidos por uma dada aplicação ou serviço durante um certo período de tempo.

É importante termos mecanismos na solução de gerência que permitam monitorar a vazão de informações. Estes mecanismos devem fornecer informações sobre canais de comunicação (conexões de rede, arquivos, etc.) abertos pelas aplicações, bem como, permitir computar os dados que trafegam nestes canais. Por exemplo, em um software de FTP seria interessante sabermos quantos bytes foram lidos ou escritos no canal de comunicação estabelecido entre o cliente e servidor. Dividindo tais valores pelo tempo de transmissão, temos a vazão de informações neste canal.

Manter um registro da monitoração da vazão durante um certo tempo é importante para a determinação de possíveis problemas de desempenho.

3.2.1.3 Precisão

Precisão na transmissão de dados é essencial para o funcionamento das aplicações. A precisão está relacionada com a taxa de erros de comunicação entre os elementos de uma aplicação (processos).

Uma solução de gerência deve fornecer informações sobre as taxas de erros na comunicação entre dois processos. Percebe-se então que a precisão está relacionada com a vazão da seguinte forma: tome como exemplo o software de FTP citado na seção anterior. Sobre o mesmo canal de comunicação aberto pela aplicação podemos

computar a taxa de erros de transmissão e recepção de dados entre as entidades cliente e servidora.

Monitorar a precisão na comunicação auxilia na detecção de gargalos causados por problemas físicos na rede ou outros fatores que influenciam a transmissão de dados.

3.2.1.4 Monitoração da Utilização de Recursos

A monitoração de recursos consiste em determinar quais recursos estão sendo utilizados para que uma determinada tarefa seja realizada. As informações coletadas neste processo de monitoração podem ser utilizadas em outras áreas, como no planejamento de capacidade, gerência de falta e de contabilidade.

Na gerência de aplicação queremos saber quais recursos estão sendo utilizados por uma determinada aplicação. Por exemplo, o tempo de CPU e a quantidade de memória consumida pela aplicação para realizar um serviço. Estas informações têm que estar acessíveis e organizadas de tal forma que possamos escolher a que nível desejamos a informação. Como exemplo: quanto tempo de CPU a aplicação como um todo consumiu? Ou então: quanta memória o serviço de transferência dessa aplicação consumiu?

3.2.1.5 Planejamento de Capacidade

O planejamento de capacidade tem como objetivo prover informações sobre a utilização dos recursos da rede. Com isso, podemos utilizar a capacidade da rede de forma mais eficiente. Analisar o impacto causado pela adição de novas aplicações, identificando a carga adicional imposta; avaliar quedas de desempenho; e identificar picos de utilização dos recursos da rede são questões de planejamento de capacidade.

O planejamento de capacidade está relacionado com o desempenho. Estuda-se a rede, o comportamento das aplicações, a distribuição das aplicações e dos recursos sendo utilizados para oferecer ao usuário final serviços com desempenho aceitável. A idéia é reduzir os custos enquanto se aproveitam os recursos disponíveis da melhor maneira possível.

Com o planejamento de capacidade descrito devemos pensar em como estas informações são disponibilizadas. Não existem variáveis específicas que informe a capacidade de um sistema. O planejamento é feito através da análise das diversas

informações obtidas nas diversas áreas de gerência. Estes dados são analisados estatisticamente para projetar futuros requisitos de capacidade.

Devido a isso, soluções para gerência de aplicações devem fornecer informações a serem armazenadas em banco de dados. Tais informações devem incluir: suporte a medidas de tráfego na rede gerado pela aplicação, utilização de recursos (arquivos em uso, estatísticas de E/S, uso de recursos da camada de aplicação, uso de CPU, uso de memória). Além do que, acesso às informações de configuração como pacotes instalados e elementos que compõem os pacotes de software.

3.2.1.6 Baselines de Desempenho

Baselines provêm informações sobre o desempenho geral da rede. Podemos ver *baselining* como um retrato instantâneo. Porém, o que este retrato revela são parâmetros de desempenho (vazão, tempo de resposta, etc.) da rede. Estes retratos podem ser armazenados, comparados, analisados e utilizados com o intuito de detectar problemas de desempenho na rede.

A idéia é determinar um *baseline* de desempenho ideal (considerado “ótimo” em termos de desempenho) e utilizar este *baseline* como uma limiar para o desempenho da rede. Qualquer outro *baseline* que esteja muito acima deste limiar, ou muito abaixo é considerado suspeito e uma análise mais detalhada da rede, neste momento, é necessária para determinar a causa desta diferença.

3.2.2 Requisitos para a Gerência de Faltas

A gerência de faltas envolve três etapas: detectar (através da gerência de performance e disponibilidade), localizar, e determinar como corrigir a falta. É importante que a solução de gerência seja capaz de identificar quais aplicações estavam executando no momento da falta, bem como que componentes ou infra-estrutura estavam relacionadas a tais aplicações. Isso fornece um retrato instantâneo do ambiente e reduz consideravelmente as variáveis, o que facilita a localização da falta e o isolamento e correção da mesma.

A gerência de faltas está intimamente relacionada com outras áreas de gerência. Por exemplo, para gerenciarmos o desempenho, temos que ter detectado algum problema que estava afetando o desempenho da rede ou vice-versa. Esse problema provavelmente foi detectado através da monitoração de faltas. Dentre as tarefas na

monitoração de falta para aplicações destacamos: monitoração de disponibilidade, monitoração dos estados das aplicações e detecção de erros pelos agentes.

3.2.2.1 Monitoração da Disponibilidade

A gerência de aplicações é uma atividade integrada, o que permite que tenhamos atividades iguais em diferentes áreas. Apesar de ser uma métrica de desempenho, a disponibilidade de uma aplicação pode ser utilizada também como uma forma de detecção de faltas.

A disponibilidade de uma aplicação pode ser descrita de forma simples: a aplicação e toda tecnologia da qual ela depende estão funcionando, ou não?

Uma frustração comum para os usuários e administradores é uma aplicação que não mais funciona (as vezes o tempo de resposta é tão lento que os usuários acreditam que a aplicação não está mais executando). Nestas situações, o mais importante é detectar o que está causando o problema e reportá-lo para o administrador. A esta função vamos chamar de gerência de disponibilidade.

A gerência de disponibilidade é o processo de monitorar a aplicação e seus componentes durante a execução e, automaticamente, tomar ações corretivas ou preventivas, se necessário [WW09 98].

Para a gerência de disponibilidade, precisamos de informações sobre os elementos que formam uma aplicação, o tipo de cada elemento, onde ele está localizado além de ter a capacidade de monitorar tais elementos em execução (monitoração de processos). Sabendo quais arquivos compõem uma aplicação e os seus comportamentos, podemos detectar possíveis faltas antes que elas ocorram, aumentando assim a disponibilidade da aplicação. Por exemplo, monitorando a configuração de uma aplicação verificamos que uma determinada biblioteca (um arquivo .DLL no Windows, por exemplo) estava faltando no diretório de instalação, impossibilitando a execução da mesma. A reinstalação da aplicação, neste caso, seria necessária.

3.2.2.2 Monitoração do Estado das Aplicações

Uma solução de gerência deve permitir ao operador ter informações instantâneas sobre as aplicações. Ele deve ter acesso ao estado de cada aplicação e os elementos que

compõem esta aplicação, de forma que se possa determinar como elas, as aplicações, estão executando. E com isso, determinar faltas.

Uma aplicação poderia assumir os estados: **executando**, a **executar** – aguardando por recursos para iniciar a execução, **aguardando** – a aplicação está aguardando algum evento para prosseguir a execução (este evento pode ser um resultado de outro processo), **terminando** – a aplicação está no processo de término de execução.

3.2.2.3 Detecção de Erros pelos Agentes

É claro que em um ambiente de rede os sistemas gerenciados, hospedeiros e equipamentos de comunicação sabem da existência de problemas quando estes ocorrem. Por exemplo, uma aplicação que controla impressoras "sabe" quando uma impressora não está funcionando. Estes nós gerenciados devem, através de algum mecanismo, reportar ao gerente de rede que determinado serviço não está sendo prestado corretamente porque existe algum problema a ser resolvido.

Como discutido na seção anterior, existem dois mecanismos de comunicação entre gerentes e agentes: *polling* e *traps*. *Trap* é a solução que mais se adequa ao problema acima. Os agentes estão próximos e são responsáveis pelos nós gerenciados. Eles sabem que um problema está acontecendo; logo, é sua responsabilidade avisar o ao gerente a existência de problemas.

Uma solução de gerência deve utilizar *traps*, quando conveniente, para notificar faltas nas aplicações. *Traps* podem ser utilizados em todas as áreas de gerência, porém, são comumente utilizadas na gerência de faltas.

3.2.3 Requisitos para a Gerência de Contabilidade

O conceito de contabilização de recursos também pode ser utilizado em aplicações. Esse tipo de monitoração permite saber quem, onde, quais e quantas aplicações estão sendo executadas no ambiente de rede, facilitando assim a capacidade de planejamento e contabilidade do uso dos recursos.

Através da monitoração das aplicações, buscamos informações que serão de utilidade na contabilidade do uso de recursos e no planejamento da capacidade da rede. Por exemplo, o número de transações em um determinado sistema por indivíduo ou grupo de indivíduos, estatísticas de entrada/saída da aplicação. Essas informações

podem ser usadas para cobrar por serviços oferecidos, como também servem para o planejamento da capacidade da rede. Pode ser que o sistema não esteja suportando o número de usuários, aplicações, e que os serviços fornecidos estejam perdendo qualidade. Neste caso, medidas para remediar tais problemas devem ser tomadas.

Informações para a contabilização se apresentam de formas variadas nas aplicações. Uma forma de extrair informações de contabilização é modelar a aplicação através de canais de comunicação abertos, podendo assim computar bytes transmitidos/recebidos ou até transações. Além do mais, a solução de gerência deve ser capaz de distinguir e contabilizar os diversos serviços prestados pela aplicação (como discutimos, uma aplicação pode fornecer diversos serviços).

Vejamos um exemplo: em um provedor de acesso, podemos modelar um servidor PPP pela conexão estabelecida com um cliente. A partir desta conexão podemos extrair: quantidade de bytes transmitidos, quantidade de bytes recebidos, tempo de conexão, identificação do usuário que está requisitando o serviço, etc.

Concluimos portanto que uma solução para a gerência de aplicações deve prover a capacidade de definir o que é uma unidade de trabalho para aplicação. Por exemplo, uma aplicação bancária pode ser contabilizada pelo número de transações. Uma aplicação de transferência de arquivos pode ser contabilizada por bytes ou Kbytes transferidos/recebidos. Uma aplicação que provê um serviço de conexão Internet utiliza o tempo como unidade de contabilização. A flexibilidade em definir tais unidades é um requisito desejável quer seja em gerência de contabilidade ou em desempenho.

Em adição, uma solução de gerência de aplicações deve ser estruturada de forma que se possa facilmente transferir informações de contabilidade para um sistema especializado em contabilização do uso de recursos, normalmente um banco de dados. Esta estrutura deve facilitar a associação de um usuário ou grupo de usuários a uma conta, e deve acumular informações relevantes sobre o uso de recursos.

3.2.4 Requisitos para a Gerência de Configuração

O controle de configuração está relacionado com a administração das aplicações em um ambiente de rede. O processo de administração envolve aspectos como: distribuição, instalação e remoção das aplicações, bem como, a coleta e o armazenamento de informações que possam auxiliar na atualização das aplicações e, conseqüentemente, nos serviços fornecidos no ambiente de rede.

Distribuição e instalação são processos para ajustar o ambiente de aplicações de forma que se possa assegurar a correta associação entre as aplicações e os usuários. Em outras palavras, estes processos se preocupam com a instalação, configuração, atualização e remoção das aplicações no ambiente de rede [WW09 98].

Normalmente, os componentes de uma aplicação são dependentes de hardware e software. Como exemplo de tais dependências temos: existência da versão correta do sistema operacional, disponibilidade de espaço em disco para instalar a aplicação, existência do tipo certo de processador, etc. Essas dependências devem ser observadas antes da instalação e uso da aplicação.

Uma solução de gerência de aplicação deve disponibilizar informações sobre a configuração das aplicações no sistema sendo gerenciado. As informações de configuração devem descrever os pacotes de software instalados na máquina. As informações devem incluir a localização, o fabricante, o número de série, a data de instalação entre outras informações que auxiliem a detecção de problemas de instalação. Em adição, a solução deve ser capaz de identificar cada elemento que compõe o pacote de software e o que ele representa no contexto. Por exemplo, uma planilha de cálculo não é composta de um único arquivo; ela possui DLLs, arquivos de configuração, etc. e essas peças têm que estar bem definidas em uma base de dados.

Com tais informações, é possível montar *baselines* de configuração¹⁶ e armazená-los em um banco de dados. *Baselines* são interessantes porque podemos utilizá-los como parâmetros para a reconfiguração do sistema. Ou seja, a solução de gerência pode ser capaz de restaurar a configuração do ambiente baseado em *baselines* previamente armazenados.

Outros tópicos relacionados ao controle da configuração de aplicações são: controle operacional e relacionamento entre pacotes de software e processos. Estes assuntos serão abordados nos tópicos seguintes.

3.2.4.1 Controle Operacional da Aplicação

¹⁶ É importante o leitor diferenciar *baselines* de desempenho e *baselines* de configuração. Em desempenho estamos interessados em definir limiares de desempenho. Em configuração estamos interessados em manter registros das diversas configurações da rede (pacotes instalados).

Até agora, discutimos requisitos da aplicação de gerência baseados em monitoração (monitorando faltas e então identificando problemas). O uso da gerência de configuração se faz necessário para que possamos, entre outras coisas, consertar faltas quando possível.

Em algumas situações é importante termos algum nível de controle sobre as aplicações. Esse controle inicialmente pode se resumir a:

- Parar elementos de uma aplicação;
- Suspender e restaurar a execução da aplicação;
- Pedir para que a aplicação seja reconfigurada.

Considere um ambiente de rede, onde temos um software para o controle de impressão. Em um determinado instante, a aplicação de gerência detectou que a impressora no departamento de recursos humano estava com problemas. Com a falta detectada, os seguintes passos tinham que ser seguidos: solicitar que a aplicação que controla a impressora parasse de imprimir (suspender o servidor de impressão), trocar o papel da impressora (em nossa situação o papel tinha "enganchado") e restaurar a execução do servidor para que os pedidos de impressão voltassem a escoar. O importante nesse exemplo é que tudo foi feito, com exceção de trocar o papel, a partir de uma interface de gerência, centralizada, utilizada para controlar diversas aplicações.

3.2.4.2 Relacionando Pacotes de Software e Processos

Apresentamos acima dois requisitos básicos para gerência de configuração: obtenção de informações sobre pacotes de software e controle da aplicação. É importante em uma solução de gerência procurarmos relacionar estes dois requisitos. Expliquemos melhor. Existem situações em que é preciso descobrir informações sobre o pacote de software ao qual um processo sendo monitorado pertence e vice-versa. A solução deve permitir uma fácil navegação entre informações estáticas de pacotes de software e processos em execução e vice-versa.

É importante termos em mente o conceito de aplicação (ver seção 3.1). Uma aplicação nem sempre é formada por apenas um elemento executável; por isso, a necessidade de se identificar todos os elementos que formam uma aplicação e qual o seu papel dentro do contexto da aplicação. Aplicações podem ser formadas por um ou mais elementos executáveis aos quais, em nosso contexto, chamamos processos.

3.2.6.1 Informações obtidas sem instrumentação

Podemos obter informações genéricas sobre a aplicação sem a necessidade de instrumentação. Isso pode ser feito através de comandos UNIX, por exemplo, que provêm informações sobre pacotes de software instalados, sobre processos executando, entre outros, ou através de uso de arquivos de log fornecidos pela aplicação. Em suma, podemos sugerir como forma de coletar dados sem instrumentação [CARR 94]:

1. Abertura de quadros de rede. Todos os quadros transitando na rede são analisados, e aqueles quadros específicos da aplicação contendo informações que serão armazenadas na MIB são abertos. Essa forma de coleta de dados requer conhecimento do formato dos quadros que trafegam na rede para a aplicação.
2. Conhecimento da arquitetura do software sendo gerenciado. Pode-se usar arquivos de configuração proprietários do software como fonte de informações para a MIB. Pode-se implementar um agente que possa ser usado como *proxy* para a aplicação. Nesse caso, a aplicação alimenta os seus próprios arquivos que são acessados por um agente. Estas informações são então armazenadas na MIB.
3. Uso de comando específicos do sistema operacional que forneça informações genéricas sobre as aplicações instaladas e executando no sistema. Essa é uma forma fácil de integração, mas apenas fornece dados genéricos e superficiais sobre o ambiente de aplicação e a aplicação em si.

3.2.6.2 Instrumentação da Aplicação

Para se obter informações mais específicas de uma aplicação, é necessário mudar o código fonte da aplicação sendo gerenciada. Nesse esquema o código usado para a coleta de dados é inserido dentro do fonte da aplicação. Podemos inserir instruções para fornecer informações que serão utilizadas nas MIBs genéricas, ou prover informações para uma MIB específica da aplicação. A desvantagem desse esquema é a necessidade de se ter acesso ao código fonte da aplicação sendo modificada [CARR 94].

3.2.6.3 Requisitos para a Base de Informações

Na seção anterior, tratamos alguns aspectos relacionados à instrumentação ou não das aplicações. O objetivo principal era ilustrar o acesso às informações pelos agentes. Nessa seção, continuamos com a idéia de modelagem da aplicação para torná-la gerenciável, porém, trataremos o assunto sob o aspecto da base de dados.

Uma das dificuldades na gerência de aplicação está em como mapear uma aplicação para um modelo de dados, e como definir tal modelo de dados de forma suficientemente genérica para representar qualquer aplicação.

A maioria das soluções de gerência apresentam o que podemos chamar de monitoração baseada em logs¹⁸, ou seja, a aplicação mantém uma lista das atividades em um (ou mais) arquivo, e este arquivo é a ponte de comunicação entre a aplicação e o agente. Esse modelo fornece informações importantes, porém, ainda muito genéricas.

Daí parte a necessidade de se instrumentar a aplicação de forma que ela forneça informações mais detalhadas e que nos permita tomar ações mais precisas. Podemos resumir isso usando a frase de Morris : "O problema de não ter instrumentação na aplicação é que você está vendo a informação bruta, sem detalhes; isso faz com que as ações para a recuperação de algum problema tendam a ser brutas, sem precisão"[SHIM 96].

Uma solução de gerência deve fornecer informações em diversos níveis, do mais genérico ao mais específico. Informações genéricas sem instrumentação da aplicação se justifica pelo fato de atender as aplicações que já estão desenvolvidas e quando não temos acesso ao código fonte. Por outro lado, a instrumentação nos dá um nível de informação muito mais refinado, mas requer acesso ao código fonte da aplicação. Percebe-se que a solução não é trivial devido à necessidade de atender aplicações já existentes.

Sendo assim podemos organizar as informações segundo dois níveis: informações genéricas da aplicação e informações específicas (ver Figura 3.1).

A nível genérico (ilustrado em cinza) a aplicação pode adquirir as informações da aplicação com ou sem instrumentação. A nível específico, as informações referem-se a atributos e comportamentos intrínsecos do software sendo gerenciado.

¹⁸ Normalmente, quando a aplicação não é instrumentada, a interface entre o agente e a aplicação é um arquivo de log gerado pela aplicação gerenciada.

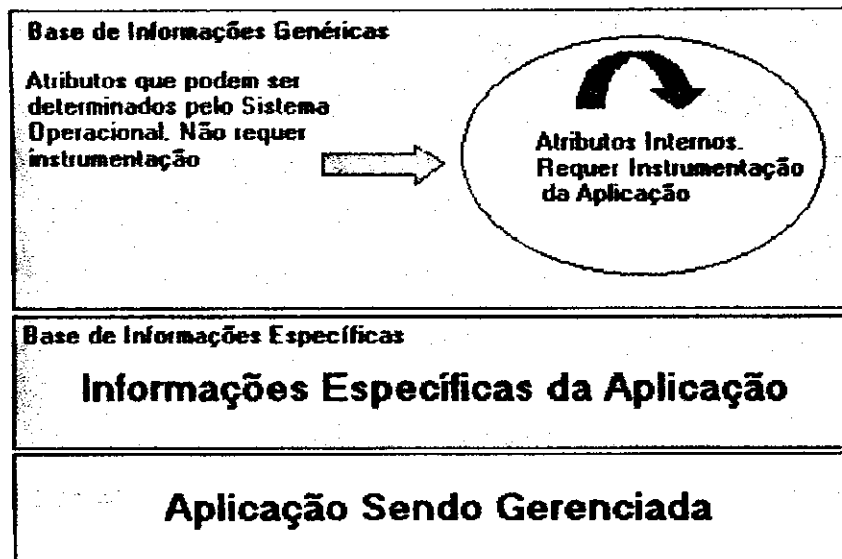


Figura 3.1. Classificação dos dados Segundo Nível de Informação Fornecida

3.3 Esforços para a Padronização

Nessa seção, discutiremos os esforços despendidos pela comunidade científica em busca da padronização de MIBs para a gerência de aplicações. Veremos o relacionamento do modelo proposto na seção anterior com os padrões que vêm surgindo e como tais padrões buscam atender aos requisitos apresentados no início do capítulo.

Felizmente, a comunidade científica tem feito grandes avanços em direção a uma padronização para MIBs de gerência de aplicações. Hoje, podemos citar como resultados desses esforços, Host Resource MIB e as MIBs propostas pelo *Application Working Group* (AWG) da IETF (*Internet Engineering Task Force*). Essas MIBs serão o foco do nosso estudo nas seções seguintes, pois, por estarem se tornando padrão, elas representam um conjunto de atributos que facilitam o desenvolvimento de aplicações de gerência interoperáveis.

3.3.1 Arquitetura

A arquitetura proposta pelo AWG é um modelo hierárquico onde cada MIB na hierarquia representa um nível de informação. A organização vai de informações mais genéricas (Host Resource MIB, SysApplMIB) até as informações mais específicas (ApplMIB). O relacionamento entre estas MIBs é feito através de índices comuns, o

que permite, por exemplo, a ApplMIB referenciar informações da MIB de sistema (sysApplMIB).

Podemos então representar a solução proposta pelo AWG utilizando a Figura 3.2.

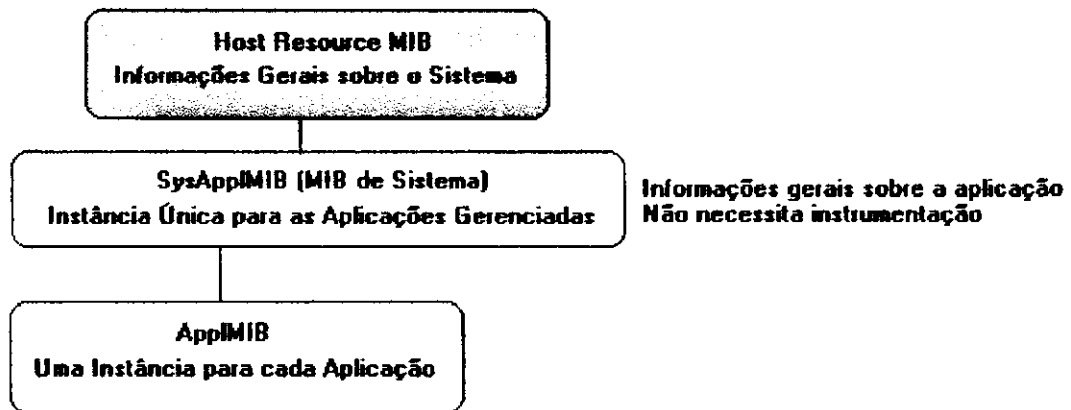


Figura 3.2. Proposta AWG para gestão de Aplicações

A Figura 3.2, acima, ilustra o relacionamento das diversas MIBs, propostas pela AWG, usadas para a gestão de aplicação. Se analisada de cima para baixo, cada caixa (MIB) representa uma especialização da MIB mais acima na hierarquia. ApplMIB apresenta informações genéricas sobre as aplicações que não necessitam instrumentação, SysApplMIB define informações genéricas adquiridas diretamente da aplicação pelo sistema operacional, Host Resource MIB apresenta informações gerais sobre o hospedeiro.

As MIBs AWG representam os primeiros passos para a gestão de aplicações. Porém, não são suficientes para gerenciar características específicas de uma aplicação. Propomos, então, uma extensão da arquitetura AWG através da adição de dois níveis de informações, ilustradas na Figura 3.3 como caixas verdes. Estas caixas representam um nível de informação mais específico sobre a aplicação gerenciada. Elas tendem a fornecer informações que auxiliarão os agentes a fornecerem serviços especializados e intrínsecos da aplicação gerenciada. Um exemplo ajudará a esclarecer a situação.

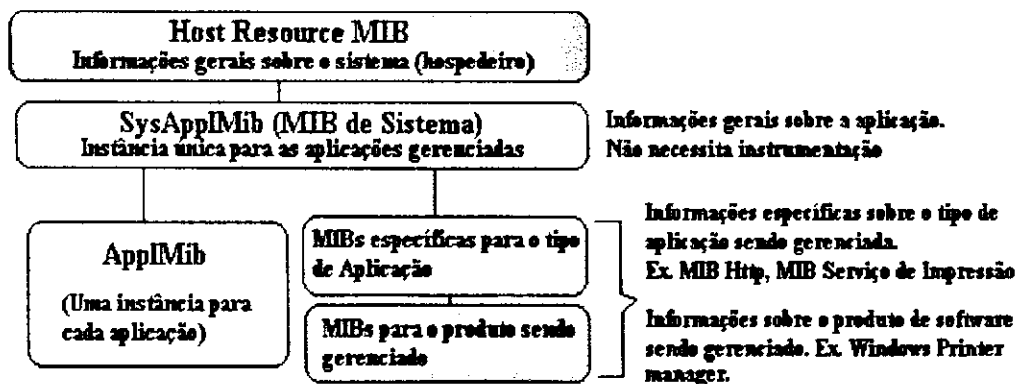


Figura 3.3. Relacionamento entre MIBs para Gerência de Aplicações

Software de impressão representa uma categoria de software que oferece um serviço de impressão. Uma MIB pode ser criada para modelar a categoria “serviço de impressão”. Esta MIB apresenta informações específicas para serviços de impressão, porém genéricas o suficiente para representar qualquer produto (Print Manager, Easy Spooler, SpoolView¹⁹, etc.), por outro lado, um software de impressão qualquer, digamos o *Printer Manager* do Windows, possui características específicas que não podem ser incluídas na MIB genérica de serviço de impressão. Nesse caso uma nova MIB tem que ser criada para suprir as necessidades de informações da MIB do serviço de impressão. A idéia apresentada não é nova, e é tradicionalmente aplicada ao modelar dispositivos de rede para a um ambiente de gerência de rede.

As MIBs específicas serão discutidas no capítulo 4 e 5. O importante para se ter em mente é que MIBs permitem definir níveis hierárquicos de informações de acordo com a necessidade da solução. Discutimos abaixo cada MIB da hierarquia que auxilia na gerência de aplicações. O destaque maior será dado às MIBs definidas pelo AWG.

3.3.2 Host Resource MIB

Essa MIB define um conjunto de objetos utilizados para a gerência de computadores hospedeiros. Em sua estrutura, a *Host Resource* MIB inclui 6 grupos, dos quais três são destinados à gerência do *hardware* e os outros três à gerência de *softwares* instalados no computador. Como o nosso foco é a gerência de aplicações, daremos maior importância aos grupos relacionados a *software*.

¹⁹ Estes são sistemas de impressão disponíveis no mercado.

3.3.2.1 Pacotes de Software

A organização das informações na *Host Resource* MIB, parte da premissa que toda máquina estará executando *software* e que este *software* estará dentro de uma das três categorias citadas abaixo.

- **Sistema Operacional (SO).** Conjunto de rotinas responsáveis pela coordenação dos recursos da máquina, e por prover uma interface de acesso uniforme aos diversos componentes de hardware.
- **Driver de Dispositivo.** Rotinas usadas pelo SO para acessar um tipo específico de hardware, tal como um disco ou um adaptador de rede; e,
- **Aplicação para o Usuário.** Um programa, ou uma coleção de programas que realiza alguma tarefa distinta para o usuário, tais como, um processador de texto ou uma planilha de cálculo.

A finalidade da *Host Resource* MIB é de prover informações sobre as aplicações instaladas na máquina, sobre os processos que estão rodando, bem como, sobre a performance destas aplicações.

A MIB *Host Resource* define três grupos relacionados à gerência de aplicações:

- O grupo `hrSWRun` identifica os processos que estão executando.
- O grupo `hrSWRunPerf` que identifica a utilização dos recursos das aplicações que estão em execução.
- O grupo `hrSWInstalled` que identifica as aplicações que estão instaladas na máquina.

Cada grupo apresentado está relacionado com uma área de gerência e satisfaz alguns dos requisitos apresentados no início do capítulo.

3.3.2.2 HrSWRun

Esse grupo contém uma entrada para cada peça de *software* que esteja executando ou na memória do computador hospedeiro. Estes *softwares* incluem: sistema operacional, controladores de dispositivos, e aplicações.

Este grupo permite o acompanhamento dos processos, que formam as aplicações. É possível através dessa MIB controlar o estado da aplicação (controle operacional da aplicação), modificando valores das variáveis na MIB. Um exemplo claro, é a variável `hrSWRunStatus`; ela representa o estado de uma peça de software que esteja executando. Esta variável pode assumir os estados:

- `running`. O processo está executando.
- `runnable`. O processo vai executar, porém, está aguardando recursos no sistema. Esses recursos podem ser memória, CPU, acesso a dispositivos de entrada e saída, etc.
- `notRunnable`. O processo está carregado, porém aguardando um evento. Este evento pode ser um retorno de outro processo, liberação de um dispositivo de saída, etc.
- `invalid`. O processo não está carregado.

Através dessa variável é possível controlar o estado do processo. Por exemplo, a mudança do valor dessa variável para o estado inválido (*invalid*) faz com que o software seja parado e descarregado da memória.

Fica claro que, através deste grupo, podemos identificar eventuais problemas com os processos que formam a aplicação (gerência de faltas), pela análise do campo `hrSWRunStatus`. Podemos identificar o tipo de processo (`hrSWRunType`), onde o arquivo se localiza no disco entre outras informações relacionadas aos processos executando no hospedeiro.

3.3.2.3 HrSWRunPerf

Este grupo define variáveis relacionadas com o desempenho das aplicações. Podemos através delas coletar informações tais como: quanto de CPU o processo está consumindo (variável `hrSWRunPerfCPU`), ou o quanta memória está alocada para esse sistema (variável `hrSWRunPerfMem`).

Relacionar estas variáveis com uma unidade de tempo permite coletar informações importantes que poderão ser usadas tanto na gerência de desempenho como no planejamento de capacidade. Por exemplo, podemos identificar que uma determinada aplicação necessita de mais memória para executar com qualidade aceitável.

3.3.2.4 HrSWInstalled

Esse grupo contém informações sobre cada software instalado na máquina local. Esta tabela é útil para identificar e catalogar os *softwares* em um hospedeiro e para diagnosticar incompatibilidades e problemas de versão entre as diversas partes do

software. Este grupo representa a base para o controle da configuração do ambiente de aplicações, principalmente no que diz respeito à instalação e à distribuição.

Essa MIB é um bom ponto de partida para o gerenciamento de sistemas, porém ela apresenta alguns problemas de arquitetura. A MIB *Host Resource* define uma tabela (*hrSWInstalledTable*) para os softwares que estão instalados no sistema. O formato dessa tabela não permite o gerenciamento de versões dos pacotes de software instalados, pois a mesma não possui uma coluna para identificar tais versões. Esse problema pode ser contornado através da combinação do nome do software com o número da versão. Essa solução é paliativa, mas necessita de intervenção humana para interpretar que porção do nome do software é a versão. Veremos que MIBs específicas para a gerência de aplicações solucionam este problema.

Um outro problema é que nem todo sistema operacional suporta um esquema unificado de instalação e distribuição de software, o que dificulta a descoberta dos softwares que estão instalados na máquina. Uma forma de contornar tal limitação é varrer o sistema em busca dos softwares instalados, mas essa tarefa consome muito tempo. Outra solução seria adotar o esquema *Windows* que descobre os softwares varrendo um banco de dados chamado *Registry* ou através da comparação dos arquivos encontrados, após uma busca no HD, com uma lista mestre mantida na máquina. Essa última estratégia pode ser prática quando falamos de computadores pessoais, mas não é escalável quando tratamos de sistemas multi-usuários com discos grandes.

A MIB *Host Resource* é um bom ponto de partida quando se tem em mente integrar rede e gerenciamento de sistemas. Entretanto, por ser uma MIB genérica, algumas funcionalidades de gerência são deixadas de lado, principalmente no que diz respeito a aplicações. Esse é um dos motivos que levou a IETF (*Internet Engineering Task Force*) a formar um grupo de pesquisa com o objetivo de definir objetos específicos para a gerência de aplicações. Esse grupo, conhecido como AWG, (*Application Working Group*) definiu algumas MIBs adicionais que serão discutidas adiante.

3.3.3 System Application MIB (SysAppMIB)

O grupo AWG foi criado com o objetivo de definir um conjunto de objetos para a monitoração e controle de aplicações distribuídas. Especificamente, esses objetos irão prover informações sobre a gerência de configuração (incluindo dependências e

associações entre aplicações), faltas (incluindo informações sobre o estado da aplicação) e desempenho (incluindo utilização de recursos) das aplicações distribuídas.

A arquitetura para gerência de aplicações apresentada pelo AWG pode ser vista como uma estrutura hierárquica, onde um conjunto de objetos apresenta informações genéricas sobre as aplicações e informações que podem ser determinadas pelo computador sem a necessidade de instrumentação da aplicação; essa MIB é chamada sysApplMIB. E, um conjunto de objetos que fornecem informações genéricas sobre a aplicação. Estas informações são obtidas através da instrumentação da aplicação, essa MIB é chamada ApplMIB.

É importante notar que essas MIBs são o passo inicial para a gerência de uma aplicação qualquer; porém, por serem genéricas, elas não fornecem um nível de informação detalhado, sendo portanto necessária a definição de outras MIBs para suprir a necessidade de informações intrínsecas da aplicação sendo gerenciada.

A *System Application* MIB provê informações sobre os pacotes de software instalados no sistema, bem como todos os arquivos relacionados que formam uma aplicação. A MIB também apresenta informações sobre as atividades das aplicações e recursos relacionados que estavam executando ou executaram em um determinado sistema.

Os objetos são organizados nos seguintes grupos:

- System Application Installed Group (Grupo de Aplicações Instaladas).
 - SysApplInstallPkgTable (Pacotes de software instalados).
 - SysApplInstallElmtTable (Elementos que compõem o pacote de software).
- System Application Run Group (Grupo de aplicações que estão executando/executaram).
 - SysApplRunTable (Aplicações executando).
 - SysApplPastRunTable (Aplicações que executaram).
 - SysApplElmtRunTable (Elementos das aplicações executando).
 - SysApplElmtPastRunTable (Elementos das aplicações que executaram).
- Escalares para restringir o tamanho das tabelas.
- System Application Map Group (Grupo para mapear processos em pacotes de software).
 - sysApplMapTable

A organização da MIB é simples. Primeiro ela oferece uma visão global das aplicações instaladas no sistema, e depois a quebra em várias tabelas, de forma a apresentar informações sobre cada elemento do pacote de software instalado (módulos executáveis e arquivos relacionados). Outros grupos tratam dos processos ou aplicações que estão executando no sistema e/ou executaram no passado.

As informações nesta MIB podem ser usadas para a gerência de configuração e de faltas. Ela fornece a base para a localização de *softwares* no sistema e a verificação da execução dos mesmos. Nessa primeira fase, características como o controle da aplicação não são contempladas. Apenas informações que podem ser obtidas a partir do sistema operacional, ou seja, sem instrumentação da aplicação, estão disponíveis nessa MIB.

Vamos descrever um cenário para ilustrar as diversas tabelas que formam essa MIB. Considere uma aplicação para a gerência de impressoras. O pacote de software é composto por vinte comandos de linha, um programa que é uma interface semi-gráfica para o usuário e dois servidores (um *despooler*²⁰ de impressão e um servidor para recepção de pedidos remotos). A principal função deste software, como a descrição sugere, é o controle de impressoras e pedidos de impressão em um ambiente de rede.

3.3.3.1 Grupo de Aplicações Instaladas (System Application Installed Group)

Esse grupo é formado por duas tabelas. Através destas tabelas, o administrador é capaz de identificar todos os pacotes de software instalados no sistema, bem como todos os arquivos que compõem esses pacotes. A primeira tabela, *sysApplInstallPkgTable*, lista os pacotes instalados na máquina. A segunda, *sysApplInstallElmtTable*, apresenta informações sobre arquivos não executáveis, ou elementos que coletivamente formam a aplicação [RFC2287].

Como essa tabela pode ser usada para a gerência de configuração? Considere a aplicação para a gerência de impressão. No momento da instalação, devemos armazenar informações sobre o pacote de software (versão, fabricante, etc.). Esse primeiro nível de informação é armazenado em *SysApplInstallPkgTable* e apresenta o pacote de software (ver Tabela. 3.1).

²⁰ Programa responsável pela impressão dos pedidos nas filas de impressão.

<i>OID</i>	<i>Descrição</i>	<i>Valor</i>
SysApplInstallPkgIndex	Inteiro usado para indexar a tabela.	1
SysApplInstallPkgManufacturer	Nome do fabricante do pacote de software	Light-Infocon
SysApplInstallPkgProductName	Nome associado pelo fabricante ao pacote de software.	SpoolView
SysApplInstallPkgVersion	Versão do pacote de software.	4.01
SysApplInstallPkgLocation { SysApplInstallPkgEntry 7 }	Local onde o pacote de software foi instalado. Diretório de instalação	/usr/spool/view

Tabela 3.1. SysApplInstallPkgTable – Informações sobre os Pacotes de Software no Sistema

Agora precisamos descrever cada elemento que compõe o pacote de software. Em nosso exemplo, vinte comandos de linha, uma interface do usuário e dois servidores compõem o gerente de impressão. Veja como ficaria uma linha na tabela SysApplInstallElmtTable (ver Tabela 3.2).

<i>OID</i>	<i>Descrição</i>	<i>Valor</i>
SysApplInstallElmtIndex	Inteiro, positivo, utilizado como índice da tabela.	1
SysApplInstallElmtName	O nome do elemento que compõe a aplicação	Sv
SysApplInstallElmtType	O tipo do elemento que compõe a aplicação. Este valor pode ser: unknown(1), nonexecutable(2), operatingSystem(3), deviceDrive(4), application(5).	Application(5)
SysApplInstallElmtPath	Diretório em que o elemento foi instalado.	/usr/spool/view/bin
SysApplInstallElmtRole	Mapa de bits que determina estado do elemento que compõem a aplicação. Este mapa pode assumir os valores: executable(0), exclusive(1), primary(2), required(3), dependent(4), Unknown(5).	Executable(0)

Tabela 3.2. SysApplInstallElmtTable - Exemplo de uma linha Linha na tabela Tabela de elementos Elementos da Aplicação

As informações fornecidas por estas tabelas são basicamente informações de configuração. Permite a localização do pacote de software instalado, identifica quais

elementos fazem parte do pacote de software e qual a função desempenhada por cada elemento na aplicação. As informações também podem ser usadas para o controle das versões das aplicações instaladas.

Para a gerência de configuração o campo `sysApplInstallElmRole` é um dos mais importantes. Este OID é do tipo BITS (mapa de bits) e pode assumir os valores:

- `executable` – uma aplicação pode ser composta por um ou mais arquivos executáveis;
- `exclusive` – apenas uma cópia deste elemento pode estar executando por instância de cada aplicação;
- `primary` – existe apenas um elemento do tipo *primary* na aplicação. Este é usado para iniciar (executar) a aplicação;
- `required` – todos os elementos com valores *required* têm que estar rodando para que a aplicação esteja executando em perfeitas condições.
- `dependent` – elementos dependentes podem não estar executando, a menos que os *requireds* o estejam.
- `Unknown` – valor *default* caso não tenha sido assinalado nenhum valor ao mapa de BITS.

Este OID é semelhante ao apresentado na Host Resource MIB. Ele permite identificar que tipo de elemento compõe a aplicação e em qual estado este elemento se encontra. Um elemento ser do tipo *primary* indica que podemos utilizá-lo para executar a aplicação. Para garantir que a aplicação esteja executando corretamente, todos os elementos *requireds* têm que estar executando. Ou seja, se existem elementos *requireds* que não estão executando, podemos dizer que identificamos um problema com a aplicação. Lembre-se que até então não temos informações sobre elementos que estejam executando, mas veremos adiante que tais informações poderão ser adquiridas em outros grupos da `sysApplMIB` e então relacionados com esta tabela. Essa é a forma de identificar quais processos fazem parte do pacote de software.

Um pré-requisito para essa MIB é que as informações sobre as aplicações instaladas possam ser descobertas sem instrumentação adicional na aplicação. Isso nos leva a requerer certas convenções como o uso de um software de instalação centralizado (pkadd no UNIX) ou de registro (esquema semelhante ao *Registry* do Windows). Esse software/comando de instalação deve prover e disponibilizar as informações necessárias sobre os pacotes de software instalados, de forma que o agente possa coletar tais informações e alimentar a MIB.

3.3.3.2 Grupo de Aplicações em Execução/Executadas (System Application Run Group)

Esse grupo apresenta informações sobre as aplicações que estão executando ou já executaram na máquina. O foco desse grupo é a monitoração. Podemos utilizar as informações desse grupo, adicionando alguma semântica, para detectar anormalidades no software instalado.

O grupo é formado por quatro tabelas. Estas representam as aplicações que estão executando no ambiente (`SysApplRunTable`) além das aplicações que já executaram no ambiente (`SysApplPastRunTable`). Para cada aplicação temos os elementos que compõem esta aplicação e que estão executando (`SysApplElmRunTable`) e os elementos das aplicações que executaram (`SysApplElmPastRunTable`).

Vejam como estas tabelas são usadas em um exemplo. Considere a gerência de impressão através de nosso pacote de software. Cada vez que uma aplicação do pacote é executada, uma linha na tabela `SysApplRunTable` é criada. Essa linha representa uma instância da aplicação executando. Podemos destacar como campos desta tabela:

- `SysApplRunStarted`. Data e hora em que a aplicação foi iniciada.
- `SysApplRunCurrentState`. Estado corrente da instância da aplicação. Este estado pode ser: `running`, `runnable` a executar mas aguardando por recurso como CPU por exemplo, `waiting` aguardando por um evento, `exiting` ou `other`.

Uma linha da tabela só será removida quando a aplicação terminar a execução. Nesse momento, uma entrada na tabela `sysApplPastRunTable` é criada.

Como deu para perceber, `sysApplPastRunTable` mantém informações das instâncias que executaram no passado, nesse ambiente. Devido ao crescimento desta tabela, dois escalares são definidos com o intuito de limitar o número de linhas permitidas e a idade da tabela; são eles respectivamente, `sysApplPastRunMaxRows` e `sysApplPastRunTblTimeLimit`.

Uma aplicação normalmente não é formada apenas por um módulo executável; outros elementos (outros módulos executáveis, bibliotecas dinâmicas, etc.) executam junto com a aplicação. Estes elementos são representados na tabela

`SysAppElmRunTable` que contém uma linha para cada processo executando na máquina. Destacamos os seguintes campos desta tabela:

- `SysAppElmtRunInstallPkg`. Forma parte do índice da tabela. Identifica a qual pacote de software este processo pertence. Se não podemos associar o elemento a um pacote de software, este OID terá o valor zero. Caso possamos determinar o pacote de software, este índice deve ter o mesmo valor de `sysAppInstallPkgIndex`. Isto é importante pois podemos utilizar este índice para obter informações mais detalhadas sobre o software ao qual este processo pertence.
- `SysAppElmtRunParameters`. Os parâmetros utilizados para a inicialização do processo.
- `SysAppElmtRunCPU`. Total de tempo de CPU consumido pelo processo.
- `SysAppElmtRunMemory`. O total de memória alocada a este processo.
- `SysAppElmtRunUser`. Nome do dono do processo.

A entrada nesta tabela é removida quando o processo termina a execução. Essas linhas são movidas para a tabela `SysAppElmPastRunTable` caso o processo seja associado a uma aplicação devidamente identificada na MIB. Vamos exemplificar como podemos usar informações desta MIB para a detecção de erros e controle da aplicação.

Em um dado momento, o operador percebe na estação de gerência que uma impressora está sem funcionar²¹ (O agente foi programado para emitir um *trap* caso notasse que elementos *requires* da aplicação não estivessem executando). Na mesma interface, o operador verificou no grupo de aplicações em execução que o nosso servidor de impressão não estava executando (detectamos uma falta). A ação tomada foi reativar o servidor de impressão (controle da aplicação). A reinicialização do servidor utiliza alguns campos da `SysAppMIB`. O procedimento envolve: descobrir qual é o elemento primário da aplicação na tabela de elementos que compõe o pacote de

²¹ Estações de gerência emite sinais visuais e/ou sonoros para alertar problemas na rede. Algumas estações podem automaticamente ligar para o biper do operador.

software, no nosso caso o servidor, e quais parâmetros deveremos utilizar para executá-lo. Essa última informação pode ser adquirida na tabela de elementos da aplicação que estavam executando (o servidor estava executando antes, não estava?) através do OID `SysAppElmtRunParameters`. De posse destas duas informações, podemos executar o comando que ativa o servidor de impressão.

Em adição à gerência de faltas e controle de configuração, podemos utilizar os OIDs `SysAppElmtRunCPU` e `SysAppElmtRunMemory` como parâmetros de desempenho. Na realidade, a métrica neste caso é a utilização de recursos.

3.3.3.3 Grupo para Mapeamento (System Application Map Group)

Esse grupo contém uma tabela com o propósito de fornecer um mecanismo de mapeamento para localizar, a partir de um identificador de processo, aplicações executadas, elementos instalados e pacotes de aplicações.

A idéia é simples. Quando um elemento que faz parte da aplicação é executado, o número do processo é armazenado nesta tabela junto com o índice da tabela de pacotes de software. A partir da tabela de software podemos explorar e acessar outras tabelas.

3.3.4 Application MIB (ApplMIB)

A MIB de aplicação representa um nível mais especializado na estrutura definido pelo AWG. Ela possui informações que não foram colocadas na `SysAppMIB`, porque para obtê-las precisaríamos instrumentar a aplicação. As informações contidas nessa MIB são utilizadas para fornecer os seguintes serviços:

- Dar suporte a medidas de tráfego na rede gerado pela aplicação (gerência de desempenho).
- Prover definições na MIB que permitam representar o que pode ser considerado unidade de trabalho para determinada aplicação (gerência contabilidade).
- Prover a capacidade para medir o desempenho de uma aplicação.
- Endereçar tópicos relacionados a recursos sendo utilizados. Exemplo: arquivos em uso, estatística de E/S, uso de recursos da camada de aplicação da rede.
- Facilidades para o controle da aplicação: parar elementos da aplicação, suspender e reativar elementos da aplicação, bem como, requisitar reconfiguração [KRUP 98].

3.3.4.1 Relacionamento com Outras MIBs

Além do relacionamento direto com SysApplMIB a MIB de Aplicação também se relaciona com outras MIBs que estão de certa forma envolvidas com a gerência de aplicação.

System Application MIB

O relacionamento entre a MIB de Aplicação e a MIB de Sistema ocorre através do escalar SysApplRunElmtIndex. É através deste índice que podemos acessar as informações sobre a aplicação e componentes que a instância da MIB representa. Lembre-se que possuímos uma instância da ApplMIB para cada aplicação sendo gerenciada.

É necessário então um cuidado redobrado quando implementando estas duas MIBs, principalmente no que diz respeito à política para definição e associação destes escalares (índices) nas duas MIBs.

Host Resource MIB

O ponto de relacionamento da *Host resource* MIB com as MIBs de aplicação estão nos grupos relacionados a software (hrSWRun, hrSWRunPerf, hrSWInstalled). O grupo de software define informações de gerência sobre as aplicações executando no sistema. Estas informações estão agrupadas em: elementos que formam as aplicações que estão executando (processos), desempenho destes elementos e aplicações instaladas.

A relação se encontra na tabela de aplicações sendo executadas dentro das MIBs Host Resource e SysApplMIB. Caso se esteja implementando a Host Resource MIB e a sysApplMIB, deve-se manter a consistência entre os índices que correlacionam estas tabelas; nesse caso hrSWRunIndex tem que ser consistente com o valor utilizado em sysApplRunElmtIndex.

3.3.4.2 Estrutura da ApplMIB

Esta MIB é organizada em vários grupos, cada grupo com tabelas que fornecem informações relevantes para o gerenciamento das aplicações (monitoração e controle). Estes grupos modelam:

- O nível de serviço da aplicação;

- Informações sobre pontos de comunicação em aberto (arquivos, conexões, etc.);
- Informações históricas sobre estes canais;
- Informação de *status* e controle a nível de processo.

Informações sobre Nível de Serviços

Esse grupo permite identificar uma ou mais instâncias de um determinado serviço (HTTP, Spooling, etc.) sendo utilizado no sistema e sua relação com elementos (processos) da aplicação que estão executando. As informações desse grupo servem como forma de mapeamento entre os serviços e as aplicações. Nesta tabela, o serviço é representado por uma *string*.

Um serviço é provido por um ou mais elementos da aplicação que está executando. As tabelas que seguem procuram relacionar serviços definidos pela aplicação e elementos da aplicação que estão executando. Por exemplo, suponha o nosso software de impressão. Para que uma impressora esteja fornecendo o serviço de impressão o servidor (chamado *despooler*) para esta impressora tem que estar executando. Se ele está executando, temos as seguintes informações sobre este elemento: o identificador que este processo tem na tabela de elementos executando, a aplicação à qual este processo pertence e o serviço que este processo presta (serviço de impressão). Podemos então organizar todas estas informações em tabelas que nos permitam navegar utilizando como índices: nome do serviço, instância do serviço e identificador de elementos que estão executando.

Este grupo fornece as seguintes informações na forma de tabelas:

Nome do serviço para instância do serviço. Facilita a localização e busca de uma da instância do serviço através do nome do serviço.

Instância do serviço para nome do serviço. Dado um identificador de instância do serviço, facilita a busca do nome do serviço sendo provido.

Instância do serviço para elemento da aplicação sendo executado. Facilita a identificação do conjunto de elementos da aplicação que está executando através do identificador de instância do serviço.

Elemento da aplicação que está executando para instância do serviço. Identifica o conjunto de serviços oferecidos por um dado elemento da aplicação que está executando.

De forma resumida, este grupo apresenta tabelas que mapeam serviços para elementos ativos da aplicação (processos) e vice-versa. Permite a monitoração dos serviços fornecidos pelas aplicações. Podemos por exemplo, dado um determinado nome de serviço monitorar as instâncias destes serviços, ou dado um determinado identificador de instância, verificar se os serviços providos por essa instância estão ativos, e assim por diante.

Grupo de Canais de Entrada/Saída

Esse grupo se destina a monitorar e apresentar informações estatísticas sobre recursos utilizados pela aplicação, estes recursos são na realidade canais de comunicação (arquivos abertos, conexões de rede) abertos pela aplicação com o intuito de processar informações. As informações que trafegam em tais canais devem ser estruturadas em transações de forma que seja mais fácil manter dados estatísticos. Fica mais simples entender quais informações podemos adquirir a partir dessa MIB quando verificarmos as diversas tabelas que a compõem.

Tabela de canais abertos. Disponibiliza informações como: tempo que o canal está aberto, número de requisições de leitura, número de bytes lidos, quando a leitura mais recente iniciou, número de requisições de escrita, número de bytes escritos e quando a última operação de escrita iniciou. Um canal pode ser um arquivo ou uma conexão.

Tabela de arquivos abertos. Contém uma entrada para cada arquivo aberto associado a um elemento de uma aplicação que esteja executando. As informações mantidas nessa MIB são: nome do arquivo, tamanho do arquivo, modo corrente do arquivo (*read* ou *write*).

Tabela de conexões abertas. Fornece informações sobre os canais que são conexões abertas ou processos que aguardam conexão. Para cada conexão temos as seguintes informações: identificação do protocolo de transporte em uso, identificação dos endereços e portas locais e remotas, identificação do protocolo da camada de aplicação em uso.

Tabela de sumário das transações. Contém sumários estatísticos sobre as transações por canal. Entre outras informações temos: identificação da conexão ou arquivo usado para essa sequência de transações, descrição do fluxo de informações (*stream*), tempo esperando por respostas, tempo gasto manipulando as respostas, tempo

gasto enviando respostas, número acumulado de transações iniciadas, número acumulado de transações processadas.

Tabela de estatísticas sobre o fluxo de informações. Contém estatísticas baseadas em número de pedidos e respostas transmitidas ou recebidas. As informações nessa tabela são: número de transações processadas, número de bytes processados, a hora de processamento da mais recente transação.

Tabela de estatísticas sobre tipos de transações. Mantém informações, organizadas por direção do fluxo de informações, sobre pedidos, respostas e tipo de transações.

A idéia nestas tabelas é modelar as aplicações através de canais de comunicação, quer seja arquivos abertos, ou conexões de rede. A partir destes canais determinamos dados estatísticos e utilização de recursos. A Figura 3.4 representa o relacionamento das tabelas deste grupo. Veja que as informações são modeladas em vários níveis, iniciando por transações, quebrando transações em canais genéricos de comunicação e sub-dividindo os canais de acordo com o tipo: arquivo ou conexão de rede.

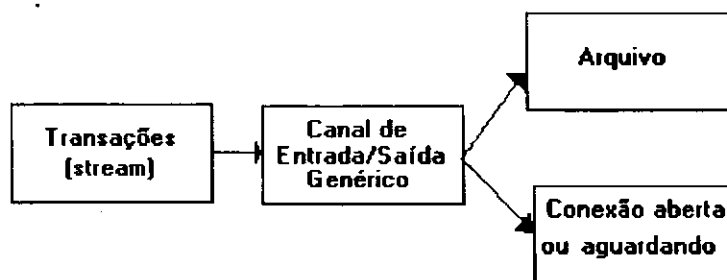


Figura 3.4. Relacionamento das Tabelas no Grupo de Canais

Tais informações satisfazem requisitos nas áreas de desempenho e contabilidade. Elas auxiliam no controle sobre o impacto que o uso de tais recursos podem causar ao sistema (planejamento de capacidade); monitoração do *stress* causado pelas aplicações; além do que as informações mantidas podem ser utilizadas na gerência de contabilidade. Maiores detalhes sobre as demais tabelas podem ser encontrados em [GREE 97].

Informações Históricas sobre Aplicações ou Instâncias em Execução

Este grupo é formado por duas tabelas que mantêm informações sobre os elementos das aplicações que estão executando. Estas tabelas apresentam informações sobre o *status* dos elementos da aplicação que estão executando (`applElmtRunStatusTable`) e outra tabela que permite o controle destes

elementos de forma rudimentar (`applElmtRunControlTable`). Os principais objetos destas tabelas são citados abaixo (Tabela 3.3):

<i>OID</i>	<i>Descrição</i>
<code>ApplElmtRunStatusSuspended</code>	Informa se o processo foi suspenso. A suspensão pode ter sido feita via uso de SNMP ou não.
<code>ApplElmtRunStatusHeapUsage</code>	Informa quanto do heap de memória esta aplicação está usando.
<code>ApplElmtRunStatusOpenConnections</code>	Número de conexões abertas sendo usada por essa aplicação.
<code>ApplElmtRunStatusOpenFiles</code>	Reporta o número de arquivos abertos e em uso pelo processo.
<code>ApplElmtRunStatusLastErrorMsg</code>	Reporta o erro mais recente. Este erro normalmente é escrito na saída padrão de erros pelo processo.
<code>ApplElmtRunStatusLastErrorTime</code>	Quando o erro foi armazenado.

Tabela 3.3 Tabela de Estados dos Elementos da Aplicação

A tabela acima representa a `applElmtRunStatusTable`. As informações nesta tabela auxiliam no diagnóstico de faltas, monitoração do estado da aplicação, bem como apresenta um resumo do que está sendo utilizado em termos de recursos pela aplicação.

<i>OID</i>	<i>Descrição</i>
<code>ApplElmtRunControlSuspend</code>	Atribuir o valor <i>true</i> a esta variável dispara a ação de suspensão do processo. Atribuir o valor <i>false</i> dispara a ação de prosseguir com a execução do processo.
<code>ApplElmtRunControlReconfigure</code>	A mudança do valor deste OID significa um pedido de reconfiguração do processo. Ou seja, o processo é reinicializado com a configuração inicial.
<code>ApplElmtRunControlTerminate</code>	Atribuir o valor <i>true</i> a este OID significa terminar o processo de forma coerente. Poderíamos chamar isso de um desligamento limpo.

Tabela 3.4 Tabela de Controle de Elementos da Aplicação

Esta MIB fornece um mecanismo para o controle de processos pertencentes a aplicações. Entre as principais ações temos: suspensão, reconfiguração e término do pedido. Tais OIDs estão diretamente relacionados com controle da configuração das aplicações.

3.4 Sumário

A importância da gerência de aplicações é incontestável. Aplicações são as entidades que fornecem os serviços necessários para a produtividade do usuário. Por serem as entidades que fornecem serviços, temos que aplicar a gerência sobre elas. Para isso, quando desenvolvendo uma solução de gerência, alguns requisitos têm que ser considerados: requisitos de desempenho, faltas, contabilidade, configuração além de requisitos para as aplicações gerenciadas. Soluções padrão que forneçam um patamar inicial para o desenvolvimento da solução também devem ser consideradas.

As MIBs sendo padronizadas pelo AWG fornecem informações de cunho genérico que podem ser utilizadas dentro de diversas áreas de gerência (configuração, desempenho, faltas). Essas MIBs representam o nível comum que toda aplicação que se destina a ser gerenciada deve ter. A partir desse nível, o desenvolvedor pode sentir-se à vontade para estender as MIBs, fornecendo informações específicas sobre suas aplicações.

Vistas as MIBs para a gerência de aplicações sendo padronizadas, chegamos à conclusão que elas não são suficientes para implementar todos os requisitos para a gerência de aplicações. A sua estrutura deixa a desejar por não contemplar *traps*, não permitir um controle mais preciso e que seja flexível o suficiente para se adaptar a qualquer aplicação; além do que, não prover nenhum esquema de segurança. Por exemplo, apesar de oferecer um controle rudimentar sobre a aplicação, como fazer para controlar completamente um serviço de impressão? Será que apenas suspender, continuar a execução e terminar uma aplicação são ações suficientes? O que faríamos para reconfigurar uma impressora (mudar parâmetros como formulários, número máximo de cópias, etc.)?

Fica claro a necessidade de se fornecer informações mais específicas sobre a classe de aplicações sendo gerenciada e mais ainda, sobre o produto a ser gerenciado. Para suprir essa necessidade, outras MIBs têm que ser definidas.

4 Gerência de um Sistema de Impressão

Neste capítulo, serão apresentados os principais requisitos para o desenvolvimento de uma solução de gerência baseada no protocolo SNMP (ou simplesmente gerência SNMP) para um sistema de impressão. Dividimos o capítulo em duas seções: a primeira seção apresentamos um estudo sobre os requisitos para a gerência de um sistema de impressão qualquer; inclui discussões sobre aplicações de monitoração e controle, desempenho, aspectos relevantes ao usuário, operador e segurança. A segunda seção, um apanhado bibliográfico, trata dos esforços da IETF para a padronização de uma MIB para a gerência de sistemas de impressão. Apresentamos as configurações para a gerência de um serviço de impressão e a MIB proposta pela IETF para a monitoração de pedidos de impressão.

O desenvolvimento de uma solução de gerência para um sistema de impressão envolve vários passos; porém a maior dificuldade está em definir quais informações devem ser fornecidas sobre a aplicação. Essa dificuldade está em descobrir o que se quer gerenciar e em definir quais MIBs são relevantes para a aplicação. No nosso estudo, integraremos diversas MIBs com o intuito de formar uma estrutura de dados concisa que permita a extração de informações relevantes sobre a aplicação. Propomos no capítulo 3 duas MIBs para gerência de aplicações em geral. Trataremos agora de MIBs específicas da aplicação.

No capítulo anterior, mostramos como podemos usar SNMP para simular uma hierarquia de informações através de tabelas, e apresentamos MIBs para a gerência de aplicações que se relacionavam pela distribuição das informações em níveis

hierárquicos. Esse capítulo segue a mesma idéia de especialização, mas fechamos o cerco sobre a aplicação alvo, sistemas de impressão. Extraímos informações específicas da aplicação, que não puderam ser adquiridas através de MIBs genéricas. A idéia é fornecer um conjunto de OIDs específicos que permitam gerenciar um sistema de impressão, ou seja, um tipo específico de aplicação, fornecendo, entre outras coisas, serviços de monitoração e contabilidade do uso de recursos.

4.1 O que é um Sistema de Impressão

Em nosso contexto, um sistema de impressão é um conjunto de software que permite o controle de várias impressoras, locais ou remotas. Este software oferece várias funcionalidades; entre elas podemos destacar: alocação de impressoras, a submissão de pedidos de impressão, obtenção do *status* de uma impressora ou de pedidos, cancelamento dos pedidos. Esse conjunto básico de funcionalidade foi definido em [RFC1179] para um protocolo de gerência de impressão conhecido como LPD/LPR.

Porém, vários produtos de software no mercado estendem essas funcionalidades básicas oferecendo ao usuário flexibilidade para o gerenciamento das impressoras e pedidos de impressão de seu ambiente. Podemos citar algumas das funcionalidade adicionais desses sistemas:

- Monitoração e controle do progresso dos pedidos de impressão através de interfaces gráficas baseadas em menu.
- Monitoração em tempo real das impressoras no ambiente.
- Automação de tarefas de impressão usando *scripts*.
- Controle de diversas impressoras cada uma com o seu formulário. Os pedidos podem ser impressos em qualquer impressora com qualquer formulário.
- Identificação da impressora menos ocupada.
- Integração com outros ambientes de impressão através de protocolos padrão como LPD/LPR.
- Inclusão de sistemas proprietários de segurança permitindo a restrição de uso das impressoras e a distribuição de níveis de acesso ao sistema de impressão.

A pergunta a ser respondida é: se um sistema de gerência de impressão oferece tantas funcionalidades, por que adaptar tais software para o SNMP? A resposta não é tão simples, mas a chave é integração. Por ser um protocolo aberto, SNMP permite a

integração do ambiente de impressão às interfaces de gerência existentes. Para ficar mais claro, o uso do protocolo aberto faz com que as aplicações proprietárias de gerência de impressão deixem de ser peças isoladas na rede e passem a ser funções integrantes de uma interface centralizada de gerência.

Passemos à pergunta: “o que se quer gerenciar com o SNMP?” Existem diversas áreas de gerência que podem ser utilizadas na gerência de sistemas de impressão. A monitoração dos pedidos de impressão, desempenho, faltas, controle das impressoras e pedidos de impressão. As possibilidades são muitas. Porém o foco, neste capítulo, está na monitoração dos pedidos de impressão durante o seu ciclo de vida e na contabilização do uso de recursos. Veremos nas seções seguintes os principais requisitos para uma solução de gerência que se destina à gerência de sistemas de impressão.

4.2 Requisitos para a Gerência de um Sistema de Impressão

A gerência de sistemas de impressão consiste da monitoração do progresso e estados dos pedidos de impressão durante o seu ciclo de vida, monitoração das filas de impressão, detecção de erros e contabilização do uso e necessidade de recursos e controle sobre os pedidos. Veremos quais requisitos devem ser considerados para a gerência de pedidos de impressão e como a IETF tenta resolver tais requisitos com uma proposta para uma MIB de monitoração de pedidos de impressão.

Utilizaremos alguns termos que merecem comentários [HAST 98]:

- **Aplicação de monitoração.** Aplicação implementada na estação de gerência que é utilizada para comunicação com um agente que implementa uma MIB. Esta aplicação utiliza o protocolo SNMP para comunicação com os agentes.
- **Aplicação de contabilidade.** Aplicação SNMP que copia as informações de uma MIB para uma forma de armazenamento mais permanente de forma que outras aplicações possam utilizar estas informações para contabilidade ou planejamento de capacidade.
- **Cliente.** O cliente representa a aplicação utilizada pelo usuário final para submissão de pedidos de impressão. Normalmente este cliente implementa um protocolo de submissão de pedidos de impressão.
- **Usuário final.** Pessoa que utiliza um cliente de impressão.
- **Operador ou administrador do ambiente de impressão.** Pessoa que utiliza uma aplicação de monitoração para gerenciar o sistema de impressão.
- **Servidor de impressão.** Aplicação responsável pela gerência da(s) filas de impressão. Normalmente um servidor é responsável por uma impressora.

Em uma aplicação proprietária o servidor de impressão utiliza o mesmo protocolo que o cliente que submete pedidos de impressão.

- Submissão do pedido. A submissão é feita por um protocolo de submissão de pedidos de impressão. Estes podem ser padrões como LPD/LPR [RFC1179] ou proprietários.

4.2.1 Independência do Protocolo de Impressão

Uma solução para gerência de sistemas de impressão tem que ser independente do protocolo para submissão de pedidos de impressão sendo utilizado. Este é um requisito importante e de certa forma lógico. Neste momento estamos definindo um conjunto de objetos para a gerência de um sistema de impressão qualquer. Ele deve apresentar um conjunto de informações genérico o suficiente para que possa ser aplicado a qualquer sistema de impressão.

Funcionalidades específicas de um produto de software que afetem a generalidade de uma MIB para um tipo de aplicação devem ser definidas e colocadas a um nível mais baixo, isto é, ao nível do produto de software a ser gerenciado. Por exemplo, uma MIB para sistemas de impressão pode ser utilizada para monitorar o *Print Manager* do Windows e o *OpenSpool*²² da HP, porém funcionalidades que pertencem apenas ao *OpenSpool* e não ao *Print Manager* e vice-versa deverão ser definidas em suas próprias MIBs.

4.2.2 Requisitos para o Controle do Sistema de Impressão

Toda a discussão coloca que a gerência de um serviço de impressão consiste basicamente de monitoração. Temos certeza que o usuário apenas quer “ver” os pedidos? E quanto a cancelar pedidos? Mudar prioridades de pedidos? Esta MIB permitirá isso? Veremos que a resposta é não. A princípio estamos modelando um tipo de aplicação, no nosso caso serviços de impressão. A nossa proposta inicial é termos uma MIB que atenda os requisitos de monitoração e contabilidade e que possa ser aplicada a qualquer sistema de impressão. Porém, nada impede que no desenvolvimento de uma MIB específica para o produto de software a ser gerenciado não possamos estender a(s) MIB(s) que serão discutidas aqui, para fornecer controle e funcionalidades intrínsecas da aplicação.

²² Produto para gerência de impressão do pacote OpenView da HP.

Se queremos prover funcionalidades de sistemas de impressão, utilizando SNMP, faz sentido que isso seja colocado ao nível do produto de software a ser gerenciado. Por exemplo, considere dois sistemas de impressão: SpoolView²³ (ambiente UNIX) e *Print Manager* (ambiente Windows). Ao nível do tipo de aplicação sendo gerenciada (sistemas de impressão) uma MIB que atenda aos requisitos definidos neste capítulo poderá ser utilizada para gerenciar ambas as aplicações. Porém, as funcionalidades oferecidas por ambas tendem a ser diferentes e dependentes de implementação. Por isso, elas são definidas nas MIBs específicas para o produto de software. Digamos, SpoolViewMIB e PrintManMIB.

4.2.3 Requisitos para a Monitoração

A monitoração em um sistema de impressão está relacionada com a observação e análise dos estados e comportamentos dos objetos que formam um sistema de impressão. Estes objetos incluem filas de impressão (JobSet ou JobQueue), pedidos de impressão e formulários associados às impressoras. Monitorar então significa acompanhar os pedidos de impressão desde a submissão até a sua impressão. Desse processo de análise e observação extraímos diversas informações ligadas diretamente às áreas funcionais de gerência. Logo, monitoramos desempenho, faltas, segurança, contabilidade e controlamos a configuração.

4.2.3.1 Monitoração dos Estados dos Pedidos

Uma solução para a gerência de um sistema de impressão deve prover ao usuário a capacidade de monitorar os estados dos pedidos de impressão durante seu ciclo de vida. Definimos o ciclo de vida de um pedido de impressão como sendo os possíveis estados pelos quais um pedido passa desde a sua submissão até a sua impressão ou remoção da fila. A Figura 4.1 mostra um diagrama sobre os possíveis estados que um pedido de impressão pode assumir e os seus relacionamentos. É importante salientar que o número de estados varia, dependendo do sistema de impressão.

²³ Produto para gerência de impressão desenvolvido e licenciado pela Light Infocon Technology, Inc.

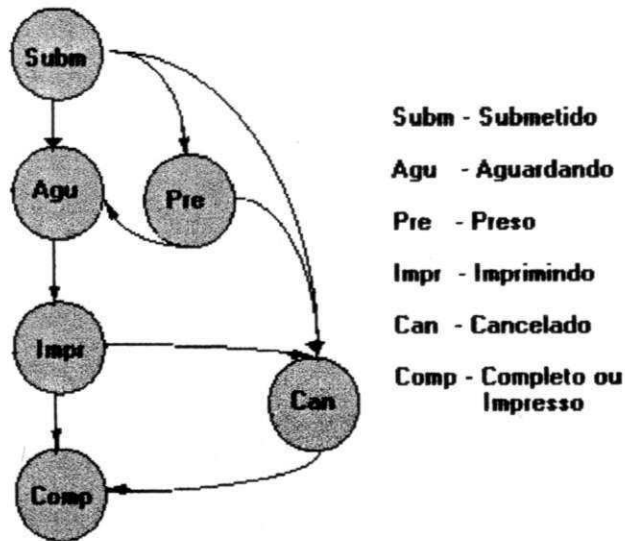


Figura 4.1. Ciclo de Vida e Estados dos Pedidos

- Submetido. O pedido foi submetido pela aplicação cliente. Normalmente esse pedido vai para o estado aguardando.
- Aguardando. O pedido de impressão está aguardando por algum recurso a ser liberado. Por exemplo, a impressora na qual ele deve ser impresso está ocupada imprimindo outros pedidos.
- Preso. O pedido de impressão está aguardando para ser impresso porém está preso. Cabe ao cliente de impressão liberar o pedido para que ele possa ser processado pelo servidor da fila de impressão.
- Imprimindo. O pedido está sendo processado pelo servidor de impressão. Como resultado temos a impressão do pedido.
- Cancelado. O usuário solicitou o cancelamento do pedido de impressão. O estado cancelado indica que houve um pedido de remoção, mas que ainda não foi concretizado.
- Completo. O pedido foi impresso com sucesso.

A monitoração dos estados dos pedidos de impressão nos dá a capacidade de saber como estão os pedidos, ou seja, em que estados eles estão e por quê. Tais informações nos auxiliam em outras áreas de gerência. Um exemplo típico de gerência de falta: existe um único pedido que está no estado aguardando apesar de aparentemente a impressora à qual se destina estar disponível. Existe algum problema (falta). Provavelmente o servidor responsável pela fila de impressão não está executando. Solução: monitorar os elementos que compõem a aplicação, verificar se o servidor de impressão está no ar e reinicializá-lo para a fila de impressão, caso necessário. Porém, além de podermos monitorar estados, precisamos saber como identificar os pedidos nas filas. Vejamos adiante tal requisito.

4.2.3.2 Identificação dos Pedidos de Impressão

Existem dois requisitos básicos com a identificação de pedidos de impressão: identificação dos pedidos de impressão dentro da MIB e como relacionar este identificador com o identificador do pedido gerado pela aplicação que submeteu o pedido de impressão.

Existem uma série de atributos, tais como, nome associado ao pedido de impressão, máquina que submeteu o pedido, nome do usuário, nome do pedido, arquivos que compõem o pedido, etc. que permitem ao usuário identificar pedidos que são de interesse. Em adição, uma solução de gerência deve possuir alguma forma de relacionar estes atributos com um identificador do pedido submetido pelo cliente. Tal índice permite à aplicação de monitoração localizar rapidamente e identificar um pedido de impressão em particular que foi submetido por um usuário de seus serviços sem ter que varrer toda uma tabela de pedidos.

Cabe ao agente que implementará esta solução atribuir um índice interno à tabela que contém informações sobre o pedido e ao mesmo tempo relacioná-lo com um índice que identifica este pedido na aplicação que o submeteu. Por exemplo, o *Print Manager* envia um pedido para ser impresso (este pedido é o pedido de número 100 para o *Print Manager*). Quando o agente identifica uma nova entrada na fila de impressão, este deve associar um identificador interno ao pedido, digamos 1. Temos então dois identificadores para o pedido: um interno à solução de gerência e um para o *Printer Manager*. Na monitoração o usuário deseja utilizar o número 100 (gerado pelo seu software) como identificador de pedido. A forma como esse identificador é mapeado internamente deve ser transparente para o usuário.

4.2.3.3 Requisitos de Monitoração Desempenho

Um solução para a gerência de pedidos de impressão deve ser projetada para fornecer informações que retratem métricas de desempenho para os componentes de um ambiente de impressão. Logo, é desejável termos disponíveis informações como: tempo de resposta, vazão dos pedidos de impressão, recursos sendo utilizados, além do que, informações que possam ser utilizadas no planejamento de capacidade do ambiente de impressão.

- Tempo de resposta em um ambiente de impressão é o tempo que um determinado pedido leva para passar do estado submetido para o estado

impresso. Permite ao administrador do ambiente determinar qual a impressora menos ocupada, ou com melhor performance.

- Recursos utilizados. Em um ambiente de impressão podemos listar diversos recursos que podem ser contabilizados e acompanhados durante o processo de impressão. Normalmente, o acompanhamento destes recursos é feito utilizando atributos que computam recursos requisitados e recursos consumidos. Por exemplo, o acompanhamento do número de cópias de impressão. O usuário requisitou 10 cópias, já foram impressas 5. Normalmente quando relacionados com o tempo estes atributos servem para mensurar o desempenho.
- Planejamento de capacidade. O administrador do ambiente de impressão deve ter informações sobre a utilização das impressoras em função do tempo. Este tipo de informação permite identificar quanto tempo um determinado pedido de impressão leva até iniciar a sua impressão. Podemos por exemplo, identificar que o volume de impressão para o departamento de Recursos Humanos é tão grande que leva em torno de 20 minutos para que pedidos iniciem a impressão. Por outro lado, no departamento Financeiro as impressoras estão sub-utilizadas. Esse tipo de informação permite que o administrador do ambiente possa remanejar impressoras e criar novas filas de impressão para resolver os problemas detectados.

4.2.3.4 Requisitos de Segurança

Projetar uma solução para a gerência de pedidos de impressão requer cuidados especiais, principalmente com a autorização do uso de uma determinada impressora e recursos associados. A solução deve considerar algum nível de segurança que permita identificar quem pode imprimir e em quais impressoras. A justificativa é simples: O ambiente não pode permitir por exemplo que notas fiscais ou documentos sem importância sejam impressos na impressora de alta qualidade do presidente da empresa.

O mérito da discussão recai novamente sobre o SNMP. A maioria dos desenvolvedores limitam-se ao uso do *get* em suas aplicações, utilizando as MIBs estritamente para a monitoração. Esquemas de segurança mais elaborados vêm sendo desenvolvidos e padronizados junto com o SNMPv3 [RFC2275]. Com a padronização da versão 3 do SNMP toda a parte de segurança da solução de gerência aqui apresentada deverá ser revista, e esquemas de segurança baseado nesta nova versão deverão ser propostos.

Outro fator de segurança está em limitar o acesso às informações por pessoas não autorizadas. Por exemplo, para um usuário comum (sem a função de administrador) apenas os pedidos que ele submeteu deverão estar visíveis.

4.2.4 Requisitos para a Contabilidade

Devemos incluir na gerência de serviços de impressão a capacidade de contabilizar o uso de recursos. A gerência de contabilidade permite a criação de registros ou históricos dos recursos consumidos no ambiente de impressão. Normalmente as informações são extraídas da base de dados após o término do processamento do pedido. Isto é diferente da gerência de desempenho que acompanha a utilização de recursos durante o processamento do pedido de impressão.

Um sistema de gerência de sistema de impressão deve fornecer informações sobre o uso de recursos (folhas, toner, cartucho de tinta, qualidade de impressão, etc.) e impressoras com o intuito de cobrar aos usuários ou a grupos de usuários os serviços prestados. Além do que, tais informações podem ser usadas para auxiliar o planejamento de capacidade do ambiente de impressão.

A partir do momento em que se monta uma base de dados sobre a utilização de recursos, uma série de informações estatísticas podem ser geradas. Estas informações auxiliam na estimativa do uso de recursos e a necessidade dos mesmos.

4.2.5 Requisitos para o Usuário Final

Até o momento, apresentamos requisitos genéricos para uma solução de gerência de sistemas de impressão. Porém, que tipo de funcionalidades os usuários finais esperam ter?

- Notificação que o pedido foi impresso. O usuário gostaria de ser avisado sobre o término da impressão de seu pedido. Existem várias formas de fazê-lo: notificação na interface, envio de e-mail para o usuário, etc.
- Estado corrente do pedido. A interface da solução de gerência deve permitir ao usuário acompanhar o progresso do pedido que foi submetido. Isto inclui o estado do pedido, número de cópias impressas, estimativa de término de impressão, quais arquivos foram submetidos, etc.
- Erros e diagnósticos para pedidos que não puderam ser impressos. A solução deve fornecer informações de tratamento e diagnóstico de faltas. Por exemplo, um usuário submeteu um pedido de impressão e este foi suspenso. O usuário necessita de informações sobre o tipo de falta e o motivo da falta. No exemplo sugerido o erro poderia ser “recurso não disponível” e o motivo “impressora está sem toner”.
- Habilidade de identificar a impressora menos ocupada. Através de informações de desempenho, o usuário deve ser capaz de identificar a impressora que está menos ocupada no momento ou a impressora que tem a melhor vazão de pedidos de impressão.

- Interface única. A solução deve oferecer aos usuários funcionalidades adicionais através de uma interface centralizada. Estas funcionalidades incluem a capacidade de monitorar e manipular os pedidos de impressão (controlar). Já discutimos esse assunto, porém o controle da configuração dos pedidos de impressão é um requisito desejável tanto a nível de usuário final quanto a nível de administrador do sistema de impressão e que deve ser considerado. Como exemplo de controle dos pedidos de impressão podemos citar: remover pedidos, mudar prioridades, prender, liberar, etc.

4.2.6 Requisitos para o Administrador do Sistema de Impressão

Os requisitos para o administrador são um super-conjunto dos requisitos de um usuário comum. Ao administrador cabem tarefas relacionadas à gerência dos pedidos de impressão. Podemos então listar os seguintes requisitos:

Capacidade de monitorar todos os pedidos de impressão através de uma interface centralizada usando o protocolo SNMP. Este requisito inclui a monitoração e gerência de todos os pedidos de impressão em todas as impressoras distribuídas na rede.

A monitoração inclui tarefas como identificar os usuários que submeteram os pedidos, quais os recursos que são necessários para a impressão dos pedidos, que fila de impressão o pedido está associado. Também deseja-se ter a capacidade de estimar quanto tempo o pedido levará para ser impresso.

Novamente, salientamos a necessidade do controle dos pedidos de impressão através da mesma interface de gerência. Caso o controle dos pedidos de impressão não seja possível, como é feita a manipulação dos pedidos de impressão em uma situação de erro? Normalmente, o operador utiliza a ferramenta que está sendo gerenciada como interface de controle. Neste caso, temos uma interface SNMP de monitoração e uma interface proprietária para o controle.

4.2.7 Notificações do Agente

Uma notificação é um evento que ocorreu no sistema de impressão e que deve ser reportado pelo agente à estação de gerência. Ou seja, um evento representa uma mudança de estado no pedido de impressão ou na impressora que possa vir a influenciar no processo de impressão dos pedidos. Alguns destes eventos merecem ser reportados para a estação de gerência. Normalmente os eventos reportados são aqueles que impossibilitam a impressão do pedido. Por exemplo, falta de *toner* na impressora.

Podemos classificar estas notificações em dois tipos:

- Alerta crítico. Um alerta crítico é disparado quando o sistema de impressão entra em um estado onde a impressão não pode prosseguir até que o problema seja resolvido. Por exemplo, falta de papel, falta de *toner*, etc.
- Alerta não crítico. É disparado quando o sistema de impressão entra em um estado em que a impressão não está parada, mas que poderá vir a sê-lo em um futuro próximo. Por exemplo, o *toner* está terminando, o papel está acabando, etc.

Um solução para a gerência de impressão deve definir tais *traps* e mecanismos para gerenciar os diversos alertas em um sistema de impressão correlacionando eventos quando possível.

4.3 Configuração para a Gerência de Serviço de Impressão

Definimos nas seções anteriores requisitos desejáveis em uma solução para a gerência de sistemas de impressão. Vamos discutir agora diferentes formas de organizar as peças em um sistema de impressão (impressora, cliente de impressão, agente, etc.) para que possamos extrair as informações e perfazer as funções de gerência. Estas configurações são apresentadas pela IETF em dois trabalhos: base de dados para a gerência de impressoras [RFC1759] e base de dados para a monitoração de pedidos de impressão [HAST 98].

Num ambiente de gerência de impressão podemos organizar as entidades participantes na gerência segundo três configurações. Estas configurações ilustram onde o agente, que implementa a MIB, estará localizado e de quem este agente coleta às informações para a MIB. A configuração que melhor se adequa ao ambiente de impressão deve ser escolhida. O principal fator que influencia nesta decisão é quem fornece as informações. Por exemplo, em um ambiente em que todas as impressoras são de rede²⁴ o agente deve coletar as informações diretamente da impressora. Este cenário pode diferir quando temos servidores de impressão no ambiente. Neste caso, o agente pode coletar informações tanto dos servidores quanto das impressoras.

²⁴ Impressora acoplada diretamente na rede e que implementa internamente um protocolo de submissão de pedidos de impressão.

Temos então as possíveis configurações: comunicação cliente/impressora com o agente localizado na impressora, comunicação cliente de impressão/servidor/impressora com o agente localizado no servidor e comunicação cliente de impressão/servidor/impressora com o cliente monitorando o agente na impressora, usando o protocolo SNMP e, o servidor, usando um protocolo proprietário [RFC1159, HAST 98].

4.3.1 Configuração Cliente/Impressora - Agente Localizado na Impressora

Nessa configuração, o cliente submete pedidos de impressão diretamente à impressora, seja por meio da rede ou por conexão direta.

A estação de gerência ou cliente de impressão monitora os pedidos de impressão através de um agente que esta localizado na impressora. Cabe a esse agente manter informações sobre os pedidos de impressão enquanto eles estão na impressora. Quando impresso, o pedido deve ser mantido pelo agente durante um tempo, o suficiente para que a estação de gerência possa acessar as informações no próximo ciclo de *polling* e copiar os dados para um sistema que possa contabilizar o uso de recursos. A Figura 4.2 ilustra este primeiro cenário.

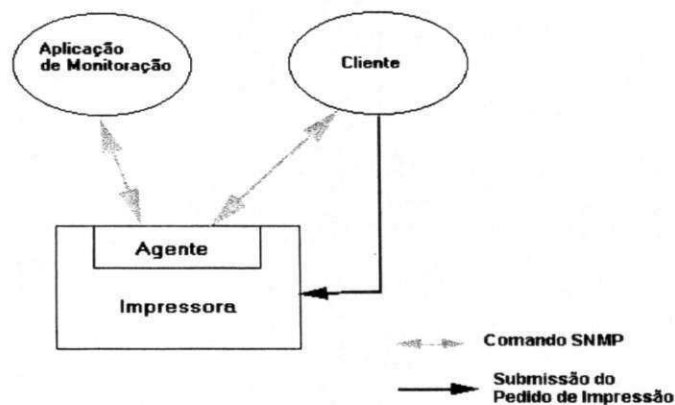


Figura 4.2. Configuração Cliente/Impressora – Agente na Impressora

4.3.2 Configuração Cliente/Servidor/Impressora - Agente Localizado no Servidor

Esta configuração é a mais comum quando estamos instrumentando um sistema de impressão proprietário. O cliente de impressão submete pedidos que serão tratados por um servidor de impressão (intermediário). Este servidor tem como função imprimir cada pedido enfileirado na impressora para a qual o pedido está destinado. O cliente ou a estação de gerência monitora os pedidos de impressão comunicando-se com o agente que é implementado dentro do servidor ou a frente do servidor.

Nesta configuração não existe um agente SNMP na impressora, pelo menos que o cliente de monitoração tenha ciência. O agente neste caso obtém informações dos pedidos de impressão através do servidor de impressão. Isto ocorre tanto para os pedidos que estão aguardando impressão como para os que já foram submetidos ou estão sendo processados. Estas informações são disponibilizadas para a estação de gerência. A Figura 4.3 ilustra este cenário.

Os seguintes relacionamentos são definidos para a MIB de monitoração:

1. Múltiplos clientes podem submeter pedidos em um servidor.
2. Múltiplos clientes podem monitorar um servidor.
3. Múltiplas estações de gerência podem monitorar um servidor.
4. Um cliente pode submeter pedidos a vários servidores.
5. Uma estação de gerência pode monitorar vários servidores.
6. Múltiplos servidores podem submeter pedidos a uma impressora.
7. Múltiplos servidores podem controlar uma impressora.

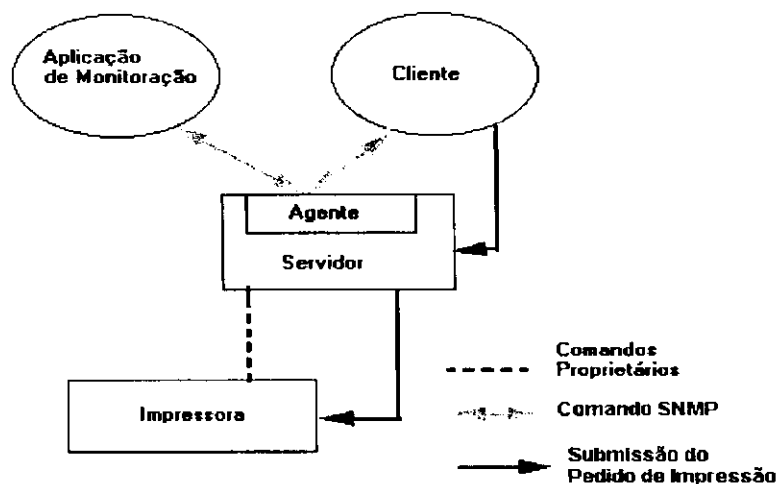


Figura 4.3. Configuração Cliente/Servidor/Impressora – Agente no Servidor

4.3.3 Configuração Cliente/Servidor/Impressora - Agente na Impressora - Cenário 2

Nesta configuração o cliente de impressão submete pedidos diretamente a um servidor, porém, este servidor não contém um agente para a monitoração de pedidos de impressão (ver Figura 4.4). A aplicação de monitoração ou cliente de impressão monitora os pedidos comunicando-se diretamente com:

1. O servidor utilizando algum protocolo proprietário para monitorar os pedidos. Neste caso os dados não estão sendo extraídos de uma MIB e sim de arquivos proprietários do servidor de impressão, e;
2. Um agente localizado na impressora que implementa uma MIB de monitoração de pedidos de impressão. Os pedidos sendo monitorados são aqueles que o servidor submeteu à impressora. A desvantagem nesse caso é óbvia. Pedidos que estão enfileirados nas estruturas do servidor não são monitorados.

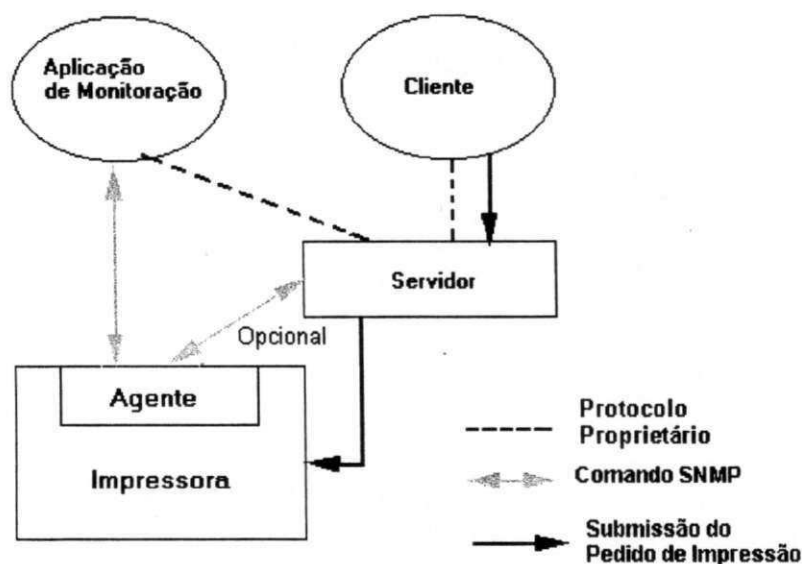


Figura 4.4. Configuração Cliente/Servidor/Impressora – Cliente Monitora Agente na Impressora e o Servidor.

4.4 Job Monitoring MIB

A MIB para gerência de impressoras, Printer MIB [RFC1759], definida pela IETF vem sendo extensivamente utilizada pelos fabricantes para facilitar o gerenciamento de impressoras via SNMP. Muitas impressoras contêm agentes SNMP

que respondem a softwares nas estações de gerência ou softwares proprietários para a configuração da impressora. A proposta da Printer MIB é auxiliar a gerência da impressora sem a preocupação com a gerência do processo de impressão propriamente dito.

Não havia então uma preocupação com o gerenciamento de pedidos de impressão, independente da impressora física. Este assunto foi abordado durante um ano pelo *Printing Working Group* (PWG) culminando com a definição de uma MIB para a monitoração de pedidos de impressão (*Job Monitoring MIB*). Devido estarmos focados na gerência de sistemas de impressão e do processo de impressão propriamente dito, está fora do nosso escopo entrar em detalhes sobre a Printer MIB.

A *Job Monitoring MIB* é uma solução genérica para sistemas de impressão. Ela pode ser implementada independentemente do protocolo ou aplicação utilizada para submeter pedidos. A sua função é prover informações sobre os estados dos pedidos de impressão, monitoração do progresso dos pedidos e informações que podem ser utilizados em softwares específicos para a contabilização do uso de recursos.

A *Job Monitoring MIB* atende a maioria dos requisitos apresentados. Está fora do escopo desta MIB: gerenciar operações (controle), cancelar pedidos, mudar prioridade, re-submeter pedidos de impressão, emitir *traps*, ou seja, esta MIB funciona como um “satélite com câmeras, e não como mísseis”[HAST 96]. Porém esta MIB permite que qualquer outra MIB que a estenda implemente requisitos que por ventura não tenham sido contemplados.

Como características fornecidas pela *Job Monitoring MIB* destacamos:

- Apresenta uma estrutura simples para a gerência de sistemas de impressão;
- Provê informações de forma que aplicações de contabilidade e estatística possam copiar as informações após os pedidos serem impressos;
- Pode instrumentar qualquer aplicação rodando qualquer protocolo de submissão de pedidos de impressão;
- Projetada para facilitar a expansão da MIB de forma a gerenciar dispositivos que não sejam impressoras (fax, *scanners*, etc.);
- Segurança: Permite que o agente implemente uma política para que o usuário só possa ver os seus pedidos; porém, ainda depende da segurança provida pelo SNMP.

Vejamos como a *Job Monitoring MIB* está organizada e como os seus objetos atendem aos requisitos definidos anteriormente.

4.4.1 Objetos da Job Monitoring MIB

A Job Monitoring *MIB* possui quatro grupos de objetos representados por quatro tabelas. Estes grupos são *General Group* (informações sobre filas de impressão), tabela JobID (mapea identificadores de pedidos proprietários em identificadores internos da MIB), tabela dos estados dos pedidos, tabela de atributos dos pedidos de impressão.

As sessões seguintes descrevem com detalhes os grupos que formam a Job Monitoring *MIB*. Discutiremos a importância dos elementos de cada tabela e como eles serão utilizados para a gerência de um serviço de impressão.

4.4.2 General Table (Tabela de Impressoras)

Esta tabela consiste de um conjunto de informações sobre um grupo de pedidos. Normalmente os grupos de pedidos são organizados por impressoras às quais os pedidos foram submetidos. Neste caso, podemos dizer que a General Table fornece informações sobre as filas de impressão. Esta tabela é composta por 6 objetos descritos abaixo:

- Índice da fila de impressão. Índice que permite a identificação da fila de impressão à qual os pedidos estão associados.
- Número de pedidos ativos. Contém o número de pedidos de impressão que estão aguardando impressão. Do ponto de vista do usuário este OID permite (junto com outras informações) identificar a fila de impressão menos ocupada.
- Índice do pedido mais novo na fila. Esse campo contém o valor do índice do pedido mais novo da fila. Esse pedido deve estar ativo²⁵, ou seja, aguardando impressão, preso ou imprimindo. Esse campo facilita a localização de pedidos ativos pela aplicação de monitoração. O processo é simples, a aplicação percorre a tabela de pedidos utilizando o índice do pedido mais antigo até o chegar ao índice do pedido mais novo.
- Índice do pedido mais antigo na fila. Esse OID contém o valor do índice do pedido mais antigo da fila.
- Persistência do pedido. Tempo em segundos que uma instância de um pedido de impressão para essa fila deve permanecer na fila de impressão após impresso. Após a impressão, um pedido processado para essa fila deverá permanecer na MIB durante o tempo indicado nesse campo. A razão para isso é simples: como a Job Monitoring *MIB* é totalmente baseada em *polling* (não define *traps*), temos que manter os pedidos por um certo tempo na MIB para que a estação de gerência possa coletar informações no *poll* subsequente.

²⁵ Um pedido ativo é aquele que está enfileirado no estado aguardando, preso ou já imprimindo. Em contra partida, um pedido é dito estar inativo quando ele se encontra no estado impresso ou cancelado.

- Persistência dos Atributos. Tempo em segundos que os atributos de uma instância de pedido devem ficar na tabela de atributos. A mesma idéia descrita acima aplica-se à tabela de atributos.
- Nome da Fila de impressão. Identifica a fila de impressão através de uma *string*.

A General Table contém informações sobre as impressoras e sobre a política associada às impressoras. Esta política inclui quanto tempo um pedido será mantido na tabela de identificação (JobID Table) e na tabela de atributos. Só existe uma entrada na General Table para cada impressora.

4.4.3 Job ID Table

Esta tabela auxilia a obtenção do índice de um pedido de impressão interno da MIB a partir de um identificador de pedido do cliente que submeteu o pedido de impressão. Vejamos como esse mapeamento funciona através de um exemplo. Considere que após a impressão de um documento o sistema de impressão associou como identificador para este pedido o valor 4. Quando o servidor informou ao agente sobre este pedido, o agente associou internamente o valor 55 para o mesmo pedido de impressão. Como se referir a esse pedido quando formos monitorar o seu progresso: 55 ou 4?

Para solucionar esse problema a tabela JobID foi definida. Ela permite que clientes possam se referir aos seus pedidos através do número de submissão e não identificadores internos. Normalmente, a necessidade de mapear tais pedidos acontece quando o agente gerencia um servidor que pode aceitar pedidos de múltiplos clientes. Como principais campos da tabela temos:

- Identificador externo do pedidos (`jmJobSubmissionID`). Identifica um pedido no ambiente de impressão.
- Índice da fila de impressão. Contém o valor do índice da fila, ou job set, à qual esse pedido pertence. Permite que informações sobre a fila de impressão sejam facilmente acessadas.
- Índice interno do pedido. Normalmente este índice é um número sequencial atribuído pelo agente que implementa a MIB. Permite a localização do pedido de impressão em outras tabelas.

Esta tabela contribui de forma efetiva na identificação de pedidos de impressão. A palavra chave é transparência. O usuário não precisa ter idéia do número do pedido

dentro da estrutura da MIB, basta para ele saber que identificador a sua aplicação associou ao pedido submetido.

Porém, como identificar pedidos com mesmo identificador gerado por diferentes clientes? `JmJobSubmissionID` é um octeto com 48 bytes que pode ser dividido para fornecer informações sobre o tipo de identificador sendo usado, a máquina na qual o pedido foi originado e o número do pedido no cliente. Tome como exemplo o protocolo LPD/LPR [RFC1179]. O mapeamento de um identificador de um pedido LPD/LPR é definido pelo PWG [BERG 98] como tipo 9. Veja a descrição de cada sub-divisão do campo `JmJobSubmissionID`:

Octeto 1: '9'	Define o tipo de mapeamento sendo usado para <code>JmJobSubmissionID</code>
Octetos 2-40	Contém o nome do host que submeteu o pedido de impressão.
Octetos 41-48	'0000XXX' ou '0000XXXX', onde XXX ou XXXX é o número (codificado em ASCII) representando o número do pedido LPD/LPR.

Tabela 4.1. Mapeamento de um Job LPD para `JmJobSubmissionID`

Maiores detalhes sobre os diversos tipos registrados pelo PWG podem ser encontrados em [BERG 98, HAST 98].

4.4.4 Job Table (Tabela de Pedidos)

Esta tabela contém informações sobre o estado e as características de cada pedido de impressão em uma fila. Contendo uma linha para cada pedido de impressão, a Job Table permite que a aplicação de gerência possa acessar informações sobre um pedido de impressão através de uma única operação *get*.

Através das informações contidas nesta tabela o usuário pode rapidamente identificar o estado em que os pedidos de impressão se encontram. Esta tabela fornece as informações para a implementação da monitoração dos estados dos pedidos de impressão. Como principais campos na tabela temos:

- Índice do pedido. Inteiro sequencial que identifica o pedido de impressão. Este índice permite a aplicação de gerência acessar outras tabelas para adquirir informações específicas (fila de impressão, atributos do pedido, recursos utilizados) sobre o pedido de impressão.
- Estado do pedido. Contém o identificador do estado corrente do pedido de impressão. Este estado pode assumir os valores:
 - `pending(3)` – aguardando impressão;

- `perndingHeld(4)` – preso;
 - `processing(5)` – imprimindo;
 - `processingStopped(6)` – suspenso;
 - `canceled(7)` – Impressão cancelada;
 - `completed(9)` – Impresso;
 - `aborted(8)` – removido.
- Prioridade do pedido. Representa a prioridade deste pedido na fila de impressão. Ele contém o número de pedidos de impressão que serão processados antes dele.
 - Tamanho do pedido de impressão. Esse tamanho é mensurado em Kbytes e não inclui multiplicativos, como o número de cópias solicitadas, ou seja, esse valor é absoluto. Permite estimar o quanto de recurso (espaço em disco) este pedido irá precisar.
 - Quantidade do pedido processado. Contém o número de Kbytes já processados pelo servidor de impressão. Este campo pode ser usado para auxiliar no acompanhamento do progresso do pedido de impressão, bem como, na estimativa de desempenho da impressora associada com a fila. Podemos relacionar quantos Kbytes por unidade de tempo a impressora está imprimindo (vazão de impressão).
 - Dono do pedido. *String* que identifica o usuário que submeteu o pedido. Este OID pode ser usado para associar os recursos utilizados com um usuário e possivelmente cobrar pelo uso de tais recursos.

4.4.5 Attribute Table (Tabela de Atributos)

A tabela de atributos é utilizada para a monitoração dos atributos envolvidos no processo de impressão, bem como por aplicações de contabilidade para obter informações sobre o uso de recursos pelos pedidos após a sua impressão. A aplicação de contabilidade deve monitorar a tabela de estados de pedidos para determinar quando um pedido foi impresso. A aplicação utiliza o índice adquirido na tabela de pedidos para acessar os atributos.

A tabela de atributos apresenta várias informações que podem ser agrupadas nas seguintes áreas: estados dos pedidos, atributos para identificação do pedido, parâmetros para os pedidos, atributos para a qualidade de impressão, atributos para o progresso do pedido, atributos de impressão, atributos para páginas, atributos para folhas, atributos

para recursos utilizados pelo pedido, atributos de tempo. Ao final da sessão apresentaremos uma tabela descrevendo os principais tipos de atributos.

4.4.5.1 O Mecanismo de Atributos

É interessante discutimos um pouco sobre a forma que a tabela de atributos foi definida. Ela representa uma ótima idéia que pode ser utilizada como exemplo em outras implementações. A tabela de atributos representa o que chamo uma tabela dinâmica. Comumente representamos atributos como colunas em uma tabela. Na tabela de atributos é diferente. Cada linha da tabela representa um atributo associado a um pedido. Não existe um tipo de atributo pré-definido, ou registrado como um OID na MIB. Os atributos são associados aos pedidos à medida que eles podem ser computados. Vejamos como esse mecanismo funciona.

A tabela de atributos é composta de 6 campos dos quais quatro são índices e dois contêm os valores do campo. Vejamos os campos desta tabela e como as linhas são criadas:

- Índice para o tipo de atributo (`JmAttributeTypeIndex`). Esse campo contém um valor que identifica o tipo de atributo que a linha representa. Um atributo pode representar um recurso necessário para imprimir o pedido, número de cópias, cópias completadas, etc. O tipo de atributo é definido na MIB através de uma enumeração. Por ser um inteiro único, o tipo de atributo pode ser usado como índice.
- Índice para instâncias dos atributos (`JmAttributeInstanceIndex`). Este índice permite identificar diferentes instâncias para o mesmo tipo de atributo. Alguns pedidos precisam de várias instâncias para o mesmo tipo de atributo. Por exemplo, nome dos arquivos que compõem o pedido de impressão. Em adição ao tipo de atributo associamos um inteiro sequencial a cada instância do mesmo tipo de atributo.
- Valor inteiro do atributo. Esse campo contém o valor do atributo caso o tipo deste atributo represente um inteiro.
- Valor do atributo como *string*. Esse campo contém o valor do atributo caso o tipo deste atributo represente um *string*.

Como deu para perceber, uma linha da tabela de atributos pode assumir um valor inteiro ou *string* dependendo do tipo de atributo utilizado. Esse mecanismo permite que o agente possa instanciar um tipo de atributo quando ele tem acesso ao mesmo. Outra característica ganha com este tipo de OID é flexibilidade. A capacidade de expansão nos diferentes tipos de atributos permite que possamos implementar a Job Monitoring MIB para diferentes tipos de sistemas de impressão.

Existem mais de cinquenta tipos de atributos definidos na Job Monitoring MIB. Estes atributos permitem entre outras coisas a monitoração do progresso dos pedidos de impressão. Identificamos atributos como número de Kbytes impressos, número de cópias requisitadas/impressas, número de páginas impressas, hora que foi submetido o pedido, entre outros. Estes atributos podem ser utilizado nas diversas áreas funcionais de gerência, destacando-se monitoração de desempenho (vazão, planejamento de capacidade), falta (estados dos pedidos, motivo para os estados) e contabilidade (recursos requisitados/consumidos).

4.4.5.2 Resumo dos Atributos e o seu Uso

A Tabela 4.2 apresenta os tipos de atributos definidos pela Job Monitoring MIB. Junto com cada tipo apresentamos uma breve descrição e em que área funcional de gerência cada atributo pode ser utilizado.

<i>Tipo de Atributos</i>	<i>Descrição</i>	<i>Área de Gerência</i>
<i>Atributos sobre o estado do pedido de impressão</i>		
JobStateReasons2	Informação adicional sobre o estado do pedido	F
<i>Atributos para a identificação do pedido de impressão</i>		
<i>Auxilia o usuário, operador ou aplicação de contabilidade identificar o pedido de impressão</i>		
JobAccountName	String que identifica uma conta à qual devemos computar o uso dos recursos.	C
ServerAssignedJobName	Número do pedido ou alguma forma de identificar o pedido de impressão. Auxilia o usuário identificar o pedido que ele submeteu. Usado quando o sistema não suporta jmJobSubmissionID.	M
JobName	Nome do pedido de impressão. Auxilia na localização do pedido de impressão.	M
SubmittingServerName	Nome do servidor que submeteu o pedido de impressão.	M
SubmittingApplicationName	Nome da aplicação cliente que submeteu o pedido.	M
JobOriginatingHost	Nome da máquina cliente que submeteu o pedido.	M
DeviceNameRequested	Nome do device requisitado. Normalmente o device associado a fila de impressão.	M
QueueNameRequested	Nome da fila de impressão	M
FileName	Nome do arquivo a ser impresso	M

Parâmetros do pedido de impressão

Especifica os parâmetros de entrada submetidos pelo cliente de impressão

JobPriority	Prioridade do pedido na fila	M
JobProcessAfterDateAndTime	Date e hora em que o pedido deve se tornar candidato a ser impresso.	M
JobHold	Se o valor é true o pedido está no estado preso até que seja liberado.	M,F

Atributos sobre o progresso do pedido de impressão (requisitados e consumidos)

Pares destes atributos podem ser usado para indicar relativo progresso ao usuário.

JobCopiesRequested	Número de cópias requisitados pelo usuário	D,P,C
JobCopiesCompleted	Cópias já impressas	D,P,C
JobKOctetsTransferred	Número de K transmitidos para o servidor ou dispositivo que o agente está suportando.	D,C

Atributos de impressão

Impressão em nosso contexto representa a face impressa da folha. No caso de um pedido de impressão com face única temos apenas uma impressão e uma folha impressa em um pedido de impressão dupla face temos duas impressões e uma folha (física) impressa

ImpressionsSpooled	O número total de impressões enviadas para o servidor ou dispositivo.	D,C
ImpressionsSentToDevice	Impressões enviadas ao dispositivo	D,C
ImpressionsInterpreted	O número de impressões processadas.	D,C
ImpressionsCompletedCurrentCopy	O número de impressões por cópia processada.	D,C

Atributos para a página

Uma página representa uma divisão lógica do documento sendo impresso.

PagesRequested	Número de páginas necessárias para a impressão do pedido.	D,C,P
PagesCompleted	Número de páginas impressas para esse pedido até o momento.	D,C,P
PagesCompletedCurrentCopy	Número de páginas para essa cópia do pedido.	D,C,P

Atributos de tempo

JobSubmissionToServerTime	Data e hora que o pedido foi submetido ao servidor.	D,M
JobSubmissionTime	Data e hora que o pedido foi submetido ao dispositivo ou servidor de impressão.	D,M
JobStartedBeingHeldTime	Última data e hora que o pedido entrou no estado suspenso.	D,M
JobStartedProcessingTime	Date e hora que o pedido começou a ser processado.	D,M

<i>JobCompletionTime</i>	Data e hora que o pedido terminou de ser impresso.	D,M
<i>JobProcessingCPUtime</i>	O tempo de CPU em segundos que o pedido ficou no estado processando.	D,M

Tabela 4.2. Resumo dos Atributos da Job Monitoring MIB

As letras apresentadas no campo Área de gerência representam:

- M – Monitoração (campos que de alguma forma auxiliam todas as áreas de gerência associadas com monitoração).
- D – Gerência de desempenho. Estes campos quando relacionados com tempo pode fornecer informações de desempenho.
- C – Gerência de contabilidade. Informações sobre consumo de recursos.
- P – Progresso do pedido de impressão. Auxiliam na monitoração do progresso de um pedido de impressão.
- F – Gerência de falhas. Apresenta informações sobre o estado do pedido e razões pela qual o pedido se encontra naquele estado.

4.5 Sumário

Definimos duas áreas específicas que devem ser consideradas para a gerência de aplicação. No capítulo três apresentamos duas MIBs genéricas que estavam totalmente voltadas para a gerência de aplicação na máquina hospedeira. Essas MIBs têm cunho genérico e permitem a monitoração e controle da aplicação a ser gerenciada.

Neste capítulo, continuamos explorando gerência de aplicações. Porém, onde antes gerenciávamos a aplicação como um todo, agora damos lugar a gerência das funcionalidade intrínsecas da aplicação. O estudo é feito em duas etapas, utilizando como exemplo um sistema de impressão. Na primeira etapa, estudada neste capítulo, definimos uma base de dados que fornece informações sobre o tipo de aplicação a ser gerenciada. No nosso caso, sistemas de impressão.

Os requisitos para esta base de dados incluem: ser genérica o suficiente para ser utilizada independente do protocolo de submissão de pedidos de impressão, apresentar informações que sirvam para a monitoração de pedidos de impressão, auxiliando assim no planejamento de capacidade do sistema e monitoração do progresso de um pedido, e, fornecer informações para a contabilização de recursos.

A Job Monitoring *MIB* sendo proposta pelo PWG fornece um conjunto de objetos úteis para a gerência de pedidos de impressão, porém, apesar de contemplar a

maioria dos requisitos apresentados, ainda se encontra restrita nos aspectos de segurança e controle. Como proposta de expansão da Job Monitoring *MIB* e implementação dos requisitos não contemplados pela mesma, definiremos um novo nível de informação. Este nível, fornece informações específicas sobre o produto de software a ser gerenciado. Esse assunto será alvo do capítulo 5 onde proporemos uma MIB para a gerência de um sistema impressão disponível no mercado.

5 Gerência de um Serviço de Impressão Via SNMP

Desde o início da dissertação descrevemos diversos níveis de informações para a gerência e discutimos o seu uso na gerência de aplicações. Neste capítulo, fechamos a discussão apresentando o último nível de informação necessário para efetivamente gerenciarmos uma aplicação, ou seja, informações específicas sobre o produto de software a ser gerenciado.

Esse capítulo apresenta um estudo de caso para o desenvolvimento de uma solução de gerência para um sistema de impressão. Especificamente, baseamos a solução em um produto de software para o controle de impressão: o SpoolView [SAUV 96]. Para o desenvolvimento da solução, descrevemos o SpoolView e as entidades envolvidas no processo de impressão, os requisitos necessários para gerenciar o SpoolView, o relacionamento da MIB proposta para o SpoolView com as MIBs definidas pela IETF. Por fim, descrevemos detalhes que devem ser considerados no momento de implementar a solução. Formalizamos a proposta com a definição da MIB SpoolView em ASN.1.

Além de fornecer ao desenvolvedor uma visão prática do processo de especificação de uma solução de gerência, de forma que os conhecimentos apresentados nos Capítulos 2, 3 e 4 possam ser validados, este capítulo apresentará como resultado final a proposta de uma MIB para a monitoração e o controle de pedidos de impressão, específicas para o SpoolView, totalmente integrada com as MIBs padrões para a gerência de aplicações.

5.1 SpoolView

SpoolView é um sistema de impressão originalmente concebido para prover controle de impressão em ambientes centralizados UNIX [GEDE 93]. Hoje, porém, com o crescimento dos sistemas distribuídos e alterações nas necessidades mercadológicas, o SpoolView apresenta-se como uma aplicação de controle de impressão em sistemas distribuídos.

O *SpoolView* oferece um conjunto de serviços que além de permitir a impressão de arquivos, fornece aos usuários informações detalhadas sobre: impressoras atualmente cadastradas na rede, pedidos sendo processados em todas as máquinas que estejam sendo controladas pelo *SpoolView*, gerenciamento de permissões de impressão por máquina, domínio ou impressora, dentre outras facilidades. Estes serviços são acessados pelo usuário através de uma interface interativa, baseada em menus, ou através de linhas de comando do sistema operacional UNIX (*shell*).

As principais mudanças em relação a versão centralizada, além da possibilidade de gerenciamento distribuído, foi a preocupação com a segurança do sistema. Vejamos como o SpoolView está organizado.

5.1.1 Arquitetura do SpoolView

O *SpoolView* foi desenvolvido para trabalhar em ambiente de rede sobre o protocolo TCP/IP. Ele permite a monitoração e controle de filas e pedidos de impressão. A troca de informações entre um cliente SpoolView e o servidor de impressão remoto, *Spoolnet*, é feito através de RPC (*Remote Procedure Call*) [STEV 90, SUNM 90, WW06 95].

O SpoolView foi dividido conceitualmente em duas partes: lado cliente e lado servidor (Figura 5.1). Do lado cliente, o SpoolView possui duas interfaces para o usuário, isto é, duas entidades clientes: o *sv*, interface interativa a caractere, e o SM (*Spool Master*), interface por linhas de comando. Do lado servidor, estão localizados os gerenciadores de impressão (*despoolers*) e duas interfaces para as aplicações: o

Spoolnet, servidor de impressão remota e o *Svlpd*, servidor de impressão para clientes que utilizem o protocolo de impressão LPD/LPR²⁶.

A arquitetura do SpoolView define algumas entidades que serão discutidas. A Figura 5.1 ilustra o processo de comunicação entre uma entidade cliente e o servidor de impressão remoto.

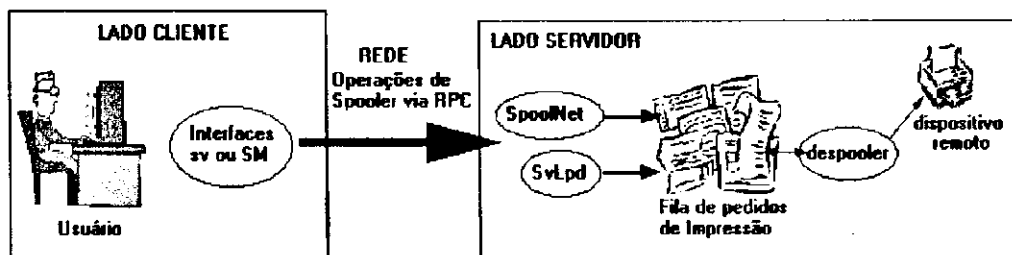


Figura 5.1. A Arquitetura do SpoolView

Considere um usuário submetendo um pedido de impressão a uma impressora remota. No momento em que a operação de impressão é solicitada através de uma das interfaces (sv ou SM), o SpoolView preenche as estruturas necessárias para a execução da operação e solicita um serviço remoto utilizando funções do RPC.

No lado servidor temos duas aplicações, *Spoolnet* e *Svlpd*, que têm como função prover os serviços remotos (*Spoolnet* e *Svlpd* são servidores). O *Spoolnet* responde a pedidos, transmitidos via RPC, oriundos do SpoolView. Ele executa o serviço solicitado e retorna um *status* (sucesso ou falha) para a máquina cliente que solicitou o serviço. O *Svlpd*, outro servidor de impressão, responde a pedidos oriundos de qualquer cliente LPD/LPR [RFC1179]. Ele traduz o protocolo LPD para o protocolo proprietário do *SpoolView*.

Analisando a Figura 5.1, nota-se que o *Spoolnet* e o *Svlpd* recebem os pedidos de impressão e os colocam na fila de pedidos (de impressão). Esta fila contém informações que representam pedidos, organizados por ordem de chegada, que serão tratados pelos *despoolers*. Cada impressora está associada a um *despooler*, aplicação responsável por ler as informações contidas nesta fila, selecionar os pedidos que pertencem a impressora a qual ele controla e efetivamente imprimí-los.

²⁶ LPD/LPR – É um protocolo de impressão largamente utilizado na Internet para comunicação entre clientes e servidores de impressão. LPD/LPR foi definido no UNIX (versão de Berkeley) e contém uma coleção de programas que interagem com um servidor de impressão chamado line printer daemon.

5.1.2 Objetos Manipulados pelo SpoolView e Seus Atributos

Definimos acima as entidades envolvidas no processo de comunicação do cliente SpoolView e servidores. Devemos destacar duas entidades que estão diretamente relacionadas com a impressão: impressoras e pedidos de impressão. Vejamos os principais atributos que representam impressoras e pedidos de impressão:

Pedidos de impressão

Um pedido de impressão é um unidade de trabalho na qual o resultado esperado é a impressão de arquivo(s) que estão neste pedido. Um pedido de impressão possui atributos que identificam que arquivo imprimir, para qual impressoras e com que características, etc.. Vejamos em detalhes os atributos utilizados pelo SpoolView para modelar um pedido de impressão:

- Número do pedido. Índice sequencial atribuído a cada pedido submetido ao SpoolView. Todo comando que manipula um pedido de impressão utiliza este índice para identificar o pedido.
- Número de cópias. Contém o número de cópias que deverão ser produzidas após o pedido ser processado. Normalmente, o número de cópias é requisitado quando o usuário submete o pedido de impressão.
- Página inicial. Este atributo contém o número da página na qual a impressão deverá iniciar. Por exemplo, podemos pedir para imprimir o arquivo contas.txt a partir da página 2.
- Página final. Usado em conjunto com a página inicial, fornece um intervalo de páginas que o *despooler* deverá imprimir. Por exemplo, página inicial 2, página final 8.
- Estado do pedido. Contém informação sobre o estado atual do pedido. O SpoolView define 3 estados para um pedido de impressão: aguardando, preso, imprimindo.
- Impressora. Este campo contém o nome da impressora, definida no SpoolView, no qual o pedido de impressão deverá ser impresso.
- Número de caracteres. Tamanho do pedido de impressão em bytes.
- Máquina. Nome da máquina na qual o pedido será impresso.
- Usuário. Nome do usuário que submeteu o pedido de impressão.
- Data. Este campo contém a data e a hora em que o pedido de impressão foi submetido.

Impressoras

Uma impressora é um dispositivo físico que produz marcas em papel. No SpoolView uma impressora é representada pela associação de um nome lógico com um

dispositivo físico. Abaixo uma lista dos atributos que fazem parte da entidade impressora:

- Nome da impressora. Nome lógico atribuído a uma impressora. Este nome serve como índice. Ele é usado em qualquer comando para a impressora.
- Dispositivo. Arquivo especial utilizado como ponte de comunicação entre o dispositivo físico (impressora) e o *despooler*.
- Número do pedido sendo processado. Identificador do pedido sendo atualmente impresso.
- Arquivo sendo impresso. Nome do arquivo que está sendo impresso pelo *despooler*.
- Estado. Estado atual da impressora. O SpoolView define 4 estados: livre, imprimindo, suspenso, aguardando.
- Número de caracteres impressos. Utilizado para o acompanhamento do progresso da impressão. Podemos derivar outras informações a partir desse atributo. Por exemplo, velocidade de impressão.
- Motivo. Este campo só contém algum valor quando a impressora se encontra no estado suspenso. Contém um texto que alerta o usuário sobre o motivo pela qual a impressora está suspensa.

5.2 Requisitos Específicos para a Gerência do SpoolView

Os requisitos específicos para a solução de gerência do SpoolView complementam os requisitos que não foram resolvidos pela Job Monitoring MIB (*traps* e controle), discutidos no capítulo 4. Uma solução de gerência para o SpoolView deve contemplar notificações pelo agente (*traps*), ou seja, deve ser capaz de identificar situações anormais com a impressora e reportar estas situações à estação de gerência, além disso, deve definir um esquema que permita o controle dos pedidos de impressão e impressoras do SpoolView de forma mais refinada, utilizando SNMP.

5.2.1 Controle do SpoolView

O SpoolView oferece, em sua interface, controle sobre os principais elementos que fazem parte de um sistema de impressão (impressoras e pedidos de impressão). O nível de controle é classificado segundo o tipo de usuário, podendo ser: usuário normal ou administrador.

O usuário normal basicamente controla os seus próprios pedidos de impressão (remove, examina, prende, libera., etc.). Em momento algum o usuário normal controla pedidos de outros usuários ou filas de impressão. O administrador, em adição aos comandos do usuário normal, tem acesso a recursos adicionais não disponíveis ao usuário normal. Estes recursos estão, em sua maior parte, relacionados às filas de impressão (desalocar impressoras, criar novas impressoras, suspender impressão, etc.)

Uma solução de gerência SNMP para o SpoolView deve fornecer um esquema que permita o controle do processo de impressão de forma tão transparente como se estivéssemos usando a própria interface SpoolView. A Tabela 5.1 contém os comandos que devem ser implementados pela solução de gerência para o SpoolView.

Comando	Descrição
Ações sobre os Pedidos de Impressão	
Remove	Remove o pedido de impressão
Prende	Prende o pedido de impressão. O SpoolView só submeterá este pedidos após a sua liberação.
Libera	Libera o pedidos de impressão.
Ações sobre as Filas de Impressão	
Suspende	Suspende a impressora. Permite o administrador armazenar o motivo da suspensão.
Prossegue	Prossegue a impressão. Utilizada para reinicializar o processo de impressão em uma impressora suspensa.
Desativa <i>despooler</i> de Impressão	Desativa (mata) o processo <i>despooler</i> para uma determinada impressora
Ativa o <i>despooler</i> de Impressão	Ativa (reinicializa) o <i>despooler</i> para uma determinada impressora.
Reinicializa o SpoolView	Permite limpar todos os arquivos que representam as entidades (pedidos de impressão e impressoras, por exemplo) no SpoolView. Reinicializando assim todo o sistema de impressão

Tabela 5.1. Comandos do SpoolView Implementados no Agente

Note que os comandos que devem ser implementados pela solução de gerência estão a nível de administrador. Isso faz sentido pois é o administrador do sistema que estará utilizando uma interface de gerência para controlar o SpoolView. Apesar da limitação de segurança imposta pela versão 1 do SNMP, optamos pelo controle do SpoolView a nível de administrador por dois motivos:

1. O administrador é o usuário alvo da solução de gerência. Por isso, temos que fornecer ferramentas (disponibilizar comandos) que permitam a ele controlar o SpoolView como se estivesse utilizando o protocolo nativo.
2. Acreditamos que com a padronização do SNMPv3 [RFC2275] os problemas de segurança serão resolvidos.

5.2.2 Traps (Notificações do Agente)

Discutimos o requisito de *trap* no capítulo 4, e definimos dois tipos de alerta: alerta crítico e alerta não crítico. Mostramos também que nenhuma MIB (Aplicações ou Job Monitoring MIB) define *traps*. A solução de gerência para o SpoolView deve implementar *traps*. Ela deve ser capaz de identificar problemas com as impressoras, determinar a gravidade dos problemas, determinar causas, quando possível, e alertar a estação de gerência.

Como o SpoolView se comunica com as impressoras? Existem duas situações em que o SpoolView relaciona-se diretamente com a impressora: abertura do dispositivo de impressão e *despooling* dos dados para a impressora.

Quando cadastramos uma impressora no SpoolView, um nome lógico é associado a um dispositivo físico do UNIX. O primeiro passo na impressão do pedido é a abertura deste dispositivo. Uma falha na abertura desse dispositivo significa que a impressora alocada não está disponível, ou seja, está *off line*, e que o pedido de impressão não será impresso.

Outra situação crítica acontece quando estamos escoando os dados para a impressora. Neste momento, qualquer problema com a impressora, como falta de papel, falta de toner, etc. acarreta a parada do escoamento dos dados. Podemos facilmente, neste tipo de situação, detectar problemas de impressão.

Baseado nas duas situações descritas acima definimos 6 tipos de alertas que podem ser detectados pelo SpoolView:

- Impressora *off line*. A impressora foi ou está desligada, ou seja, não está pronta para imprimir.
- Impressora *on line*. Impressora retorna ao estado aguardando pedidos.
- Problema na impressão. O fluxo de informações entre o *despooler* e a impressora foi interrompido.
- Prosseguir Impressão. Emitido quando o servidor de impressão retorna a escoar os dados para aquela impressora.

- Impressora lenta. A velocidade com que a impressora está imprimindo está abaixo da definida na tabela de limiares.
- “Estouro” na tabela. O número de pedidos para uma impressora excedeu o número definido na tabela de limiares.

O leitor deve estar se questionando sobre outros alertas que poderiam ser definidos para o SpoolView, como: toner acabou, a tampa da impressora está aberta, etc. Infelizmente, o nível de detalhes nos alertas é limitado pelo produto SpoolView. Este, não define um protocolo de comunicação refinado o suficiente para extrair informações detalhadas direto da impressora. Para aumentarmos a riqueza de informações nos alertas teríamos que melhorar o SpoolView como produto. Abaixo, apresentamos os principais alertas que poderiam ser detectados através de uma melhora no produto SpoolView.

- A tampa da impressora está aberta
- A tampa da impressora foi fechada
- Mudança de configuração na impressora
- O papel preso
- A impressora está desligada
- A impressora foi ligada
- Falta a bandeja de papéis
- O toner está vazio
- O toner está quase vazio

Apesar de não definir tantos alertas, a solução de gerência do SpoolView fornecerá informações importantes ao administrador do sistema. Este será capaz de identificar em sua interface problemas com as impressoras, e conseqüentemente nos pedidos de impressão, tão logo estes ocorram.

5.3 Base de Dados para a Gerência do SpoolView

A base de dados para a gerência do SpoolView (MIB SpoolView) define um conjunto de objetos que estende e adiciona novas capacidades à Job Monitoring MIB. Estas novas funcionalidades permitem o controle mais apurado do processo de impressão no SpoolView, através de uma interface centralizada. Além do que,

apresenta um esquema de alertas que auxilia o administrador do sistema detectar faltas nas impressoras da rede.

A MIB SpoolView está organizada em dois grupos que serão discutidos nas seções seguintes:

- Grupo de controle. Define um esquema para o controle dos pedidos e impressoras gerenciadas pelo SpoolView.
- Grupo de Alertas. Define objetos para a gerência de alertas do SpoolView.

A solução de gerência SpoolView é um esquema modular. Ela permite a integração de diversos níveis de informações, organizadas de forma hierárquica, com o objetivo de fornecer ao administrador ferramentas para a gerência dos sistemas de impressão no ambiente de rede. Por razões de modularidade, o agente SpoolView implementa as seguintes MIBs:

- ApplMIB. Informações genéricas da aplicação. Esta MIB possui uma instância por aplicação.
- Job Monitoring MIB. MIB para a gerência de um sistema de impressão. Esta MIB possui uma instância por aplicação.
- SpoolView MIB. Acrescenta funcionalidades com traps para impressoras e controle do SpoolView.

A solução de gerência SpoolView integra o que há de mais novo na área de gerência de aplicações via SNMP. Vejamos como a SpoolView MIB relaciona-se com as MIBs definidas pela IETF.

5.3.1 Relacionamento com as MIBs da IETF

A SpoolView MIB é a última etapa no desenvolvimento da solução de gerência proposta. Ela representa a base de dados com informações internas à aplicação de impressão. Recapitulando, começamos a nossa discussão pelos requisitos gerais para a gerência de aplicações e apresentamos a SysApplMIB e a ApplMIB, MIBs específicas para a gerência de aplicações em geral. Existe uma única instância da SysApplMIB por máquina enquanto que a ApplMIB apresenta uma instância por aplicação. Um agente que implemente a SpoolView MIB tem que implementar todos os objetos obrigatórios da ApplMIB. Em adição, o agente que implementa a MIB SpoolView deve acessar informações da SysApplMIB para manter a compatibilidade dos índices das tabelas na ApplMIB (ver relacionamento entre ApplMIB e sysApplMIB na seção 3.3.4.1).

Definimos no capítulo 4 requisitos específicos para sistemas de impressão e apresentamos a Job Monitoring MIB, proposta da IETF para a gerência de sistemas de impressão. Maioria dos atributos das entidades definidas no SpoolView podem ser mapeadas para a Job Monitoring MIB. Em adição, novas tabelas devem ser definidas na MIB SpoolView MIB para expandir a Job Monitoring MIB. A seção seguinte explana o mapeamento das estruturas do SpoolView para a Job Monitoring MIB.

5.3.1.1 Relação com a Job Monitoring MIB

Nesta seção, apresentaremos o mapeamento dos atributos dos objetos do SpoolView (pedidos e impressoras) para os grupos e tabelas da Job Monitoring MIB.

JmJobSubmissionID Mapeado para o SpoolView

Um pedido no SpoolView é identificado pelo número do pedido de impressão e a máquina que submeteu o pedido. Partindo deste pressuposto, podemos mapear jmJobSubmissionID da seguinte forma (ver Tabela 5.2):

Octeto 1	'9'
Octetos 2-40	Nome do host que originou o pedido de impressão. Qualquer parte dos octetos que não seja usada deve ser preenchida com espaços em branco.
Octetos 41-48	'0000XXX' ou '0000XXXX', onde XXX ou XXXX é o número do pedido representado em ASCII.

Tabela 5.2. Índice JmJobSubmissionID para o SpoolView

O primeiro octeto representa o tipo de endereço que estamos usando para representar um pedido SpoolView. O endereço tipo nove ('9') indica que os outros octetos serão preenchidos respectivamente com o nome da máquina que submeteu o pedido e o número atribuído pelo SpoolView para o pedido de impressão. Como foi discutido, esta tabela mapeará um pedido do SpoolView, representado por jmJobSubmissionID, para um índice interno da tabela de pedidos da Job Monitoring MIB (jmJobIndex). JmJobIndex é o número do pedido, interno, atribuído pelo agente SNMP.

General Table (Tabela de Impressoras)

Praticamente todas as informações da General Table podem ser obtidas a partir dos atributos da impressora. Porém, nem todos os atributos da impressora são mapeados para a General Table (ver Tabela 5.3).

<i>Job MIB</i>	<i>Atributo</i>	<i>Observação</i>
JmGeneralJobSetIndex		No SpoolView um jobset representa uma fila de impressão. Para cada impressora no Spoolview designaremos um número que servirá de índice na <i>general table</i> . Esse número é um inteiro crescente de acordo com a ordem da impressora na lista de impressoras do SpoolView.
JmGeneralNumberOfActiveJobs		O agente SpoolView deve gerar este valor a partir do número de pedidos na fila de impressão.
JmGeneralOldestActiveJobIndex		Índice do pedido que está a mais tempo na fila de impressão. O agente monitora as estruturas do SpoolView checando qual o pedido mais antigo. O atributo data pode ser usado para calcular este índice.
JmGeneralNewestActiveJobIndex		Índice do pedido de impressão que acabou de ser submetido. O agente atualiza este campo quando ele detecta que um novo pedido foi adicionado.
JmGeneralJobPersistence		Estes OID independe do processo de <i>desspooler</i> .
JmGeneralAttributePersistence		
JmGeneralJobSetName	Nome da impressora	Nome lógico associado a um dispositivo físico

Tabela 5.3. Mapeamento da General Table para o SpoolView

Job ID Table

Esta tabela mapea pedidos do cliente de impressão para índices internos da MIB.

Esta tabela está relacionada com o arquivo de pedidos da seguinte forma:

- JmJobSubmissionID. A lei de formação para esse índice foi definida na seção 5.3.1.1. Armazenamos na MIB a concatenação da máquina que originou o pedido mais o número do pedido gerado pelo SpoolView.
- JmJobIDJobSetIndex. Este campo contém o índice para a impressora à qual o pedido foi submetido. Este índice deve constar na General Table.
- JmJobIDJobIndex. O SpoolView não permite duplicação de números de pedidos em uma máquina. Vamos aproveitar esta regra e utilizar o próprio número do pedido do arquivo de pedidos do SpoolView como valor para este índice.

Job Table

A Job Table é uma das tabelas mais extensas da Job Monitoring MIB. Para preencher esta tabela, o agente SpoolView deve consultar ambos os arquivos: impressoras e pedidos de impressão. Nem sempre as informações coletadas são um mapeamento direto dos atributos das entidades para os campos da tabela. Em algumas situações, o agente que implementa a MIB deverá calcular os valores dos campos através de atributos fornecidos pelo SpoolView. Vejamos como (Tabela 5.4):

<i>Job MIB</i>	<i>Atributos</i>	<i>Observação</i>
JmJobState	Estado do pedido	Apesar do SpoolView só possuir três possíveis estados para os pedidos, podemos mapeá-los e gerar os estados necessários para alimentar a MIB. Processing – imprimindo Pending – aguardando PendingHeld – preso ProcessingStop – Gerado a partir da suspensão da impressora. Canceled – gerado quando o pedido é removido. Completed – gerado após a impressão do pedido. Aborted- gerado quando o comando remover é utilizado.
JmJobStateReasons1	Motivo da suspensão da impressora	Apesar de estarmos tratando de pedidos de impressão, o atributo motivo representa a razão porque um pedido pode estar suspenso. Utilizaremos o campo motivo para mapear Reasons1.
JmNumberOfInterveningJobs		O valor para este campo deve ser o número de pedidos que faltam para que esse seja impresso. O agente deve calcular este valor varrendo a fila de pedidos e determinando quantos pedidos existem para serem processados antes deste.
JmJobKOctetsPerCopyRequested	Número total de caracteres a serem impressos (tamanho do pedido).	O agente calcula este valor através do atributo tamanho do pedido. basta dividir o valor por 1024.
JmJobKOctetsProcessed	Número de caracteres impressos	O agente calcula este valor através do atributo da impressora, número de

caracteres processados. basta dividir o valor do atributo por 1024.

JmJobImpressionsPerCopyRequested

Esse valor representa para o SpoolView o número de páginas que forma o pedido. O SpoolView não armazena esse valor. Páginas são calculadas enquanto imprimindo o pedido.

JmJobImpressionsCompleted

Este campo representa o número de páginas impressas. Valor pode ser calculado pelo agente à medida que a impressão é feita.

JmJobOwner

Nome do usuário que submeteu o pedido de impressão.

Tabela 5.4. Mapeamento da Job Table para o SpoolView

Attribute Table

Os atributos de entidades SpoolView que não aparecem nos mapeamentos anteriores, são criados na tabela de atributos. Relacionamos cada atributo definido em uma entidade SpoolView com um atributo definido pela Job Monitoring MIB. A Tabela 5.5 ilustra este comparativo:

<i>Atributos dos Pedidos</i>	<i>Job MIB</i>	<i>Observação</i>
Número do pedido	JmJobIDJobIndex	
Número de cópias a serem impressas	JobCopiesRequested	
Estado do pedido	JmJobState	
Impressora a ser usada (Nome da impressora)	QueueNameRequested	
Prioridade do pedido	JobPriority	
Número total de caracteres a serem impressos.	JmJobKOctetsPerCopyRequested	
Nome do usuário que submeteu o pedido	JmJobOwner	
Data e hora que pedido foi submetido.		O SpoolView utiliza uma string contendo data e hora. Esta string pode ser mapeada para um tipo SNMP.

<i>Atributos das Impressoras</i>	<i>Job MIB</i>	<i>Observação</i>
Nome da impressora	JmGeneralJobSetName	
Número do pedido sendo processado.	JmJobIDJobIndex	
Número total de caracteres para o pedido de impressão.	JmJobKOctetsPerCopyRequested	O agente deve calcular o valor dividindo o valor do atributo por 1024
Número de caracteres já impressos	JmJobKOctetsProcessed	O agente deve calcular o valor dividindo o valor do atributo por 1024
Motivo da suspensão	JmJobStateReasons1	

Tabela 5.5. Attribute Table mapeada para o SpoolView

5.3.2 O Grupo de Controle

O Grupo de controle permite o controle dos pedidos de impressão e impressoras sendo gerenciadas pelo SpoolView, bem como a auditoria de comandos efetuados sobre estes pedidos e impressoras.

5.3.2.1 Tabela de Controle dos Pedidos (svJobControlTable)

Esta tabela é indexada por jmJobIndex. Uma entrada nesta tabela dá ao administrador, com apropriada permissão, a habilidade de remover um pedido de impressão, prender um pedido de impressão e liberar um pedido de impressão. Esta tabela complementa a tabela de pedidos de impressão definida na JobMonitoring MIB. Os campos desta tabela são definidos abaixo (ver Tabela 5.6):

<i>OID</i>	<i>Descrição</i>
SvJobControlRemove	Atribuir o valor true a esta variável dispara a ação de remoção do pedido de impressão.
SvJobControlHold	Atribuir o valor true a esta variável prende o pedido de impressão. Atribuir o valor false libera o pedido de impressão.

Tabela 5.6. Controle dos pedido de impressão

O processo de prender e liberar um pedido de impressão nada mais é do que uma mudança do estado do pedido na tabela de pedidos da Job Monitoring MIB e conseqüentemente no arquivo de pedidos do SpoolView. Por exemplo, considere que o

administrador do sistema precise colocar um pedido de número 100 no estado preso. Ele simplesmente manda um *set* para o agente que implementa a MIB SpoolView pedindo para modificar o valor de *svJobControlHold*, do pedido de índice 100, para o valor *true*. O agente por sua vez, interpreta este *set* como: coloque *true* nesta variável e mude o estado do pedido, para refletir o estado preso. Ou seja, podemos controlar o SpoolView manipulando campos do arquivo de pedidos de impressão através da SpoolView MIB.

5.3.2.2 Tabela de Controle das Filas (svPrinterControlTable)

Esta tabela é indexada por *jmGeneralJobSetIndex*. Uma entrada nesta tabela dá ao administrador, com apropriada permissão, a habilidade de suspender uma impressora, prosseguir com a impressão, matar o *despooler* de impressão, criar um *despooler* de impressão e reinicializar todo o SpoolView. Esta tabela complementa a tabela de impressoras (General Table) definida na JobMonitoring MIB. Esta tabela contém três campos: *SvPrinterControlSuspend*, *SvPrinterControlStop* e *SvPrinterControlReset* (ver Tabela 5.7).

OID	Descrição
<i>SvPrinterControlSuspend</i>	Atribuir o valor <i>true</i> a esta variável dispara uma ação para a suspensão da impressora. Atribuir o valor <i>false</i> libera a impressora.
<i>SvPrinterControlStop</i>	Atribuir o valor <i>true</i> a esta variável dispara uma ação para a parada do servidor de impressora. Para o servidor significa que iremos "matar" o processo <i>despooler</i> . Atribuir o valor <i>false</i> cria um novo <i>despooler</i> para a impressora.
<i>SvPrinterControlReset</i>	Atribuir o valor <i>true</i> a esta variável dispara uma ação para reinicialização de todos os arquivos de configuração do SpoolView.

Tabela 5.7. Controle de Impressoras no SpoolView

Apesar da limitação de segurança imposta pela versão 1 do SNMP, acreditamos que estes problemas serão resolvidos em breve com a versão 3. Como o usuário alvo da nossa solução é o administrador do sistema, optamos por fornecer opções de controle que o auxiliassem na gerência do produto. O intuito maior é exemplificar como controlar uma aplicação, definindo assim, um esquema que possa ser usado na instrumentação de outras aplicações.

5.3.2.3 Tabelas `svPastJobControlTable` e Tabela `svPastPrinterControlTable`

Estas tabelas mantêm um histórico dos comandos disparados sobre os pedidos de impressão e impressoras no SpoolView. Esta tabela é preenchida com informações vindas das tabelas de controle. Quando um pedido de controle é executado na tabela de controle de impressora ou controle de pedidos, uma linha em uma destas tabelas é criada. A principal razão para a definição de informações históricas é a necessidade de auditoria em caso de possíveis quebras de segurança.

Devido ao crescimento constante destas tabelas, dois escalares são definidos para controlar a idade das entradas nas tabelas e o tamanho das mesmas. O valor de `svPastControlMaxRows` especifica o número máximo de entradas que as tabelas podem conter, enquanto que `svPastControlTblTimeLimit` especifica a idade máxima para uma entrada nas tabelas.

5.3.3 O Grupo de Alertas

O agente SpoolView é responsável pela detecção de eventos que podem ser reportados à estação de gerência, pela inserção de uma linha na tabela de alertas e, se este alerta for crítico, por disparar uma *trap* para a estação de gerência. A tabela de alerta auxilia no processo de gerenciamento de alertas.

O grupo de alertas consiste de uma única tabela na qual todos os alertas são representados. Esta seção provê uma discussão sobre esta tabela e como os alertas são gerenciados. A tabela de alertas no SpoolView MIB contém apenas alertas críticos. Basicamente os campos nesta tabela são os apresentados na Tabela 5.8.

Como o SpoolView só detecta e armazena alertas críticos, cada linha inserida na tabela dispara um *trap*. Espera-se que estes alertas aconteçam em pares, ou seja, se algo acontece com a impressora, este acontecimento é reportado, a impressora é consertada e o processo de impressão é reinicializado. No momento da reinicialização outro alerta é gerado. Sendo assim, estamos sempre tratando com pares de alertas, normalmente problemas seguidos de soluções. Tendo essa idéia em mente, podemos gerenciar a tabela de alertas da seguinte forma: na primeira ocorrência do alerta inserimos uma linha na tabela de alertas e disparamos um *trap*. Quando o alerta complementar (conserto do problema anterior) acontece, ao invés de inserirmos outra linha na tabela, removemos o complementar. Desta forma se não existem problemas a tabela sempre estará vazia.

<i>OID</i>	<i>Descrição</i>
SvAlertIndex	Valor do índice utilizado para determinar se alertas foram adicionados ou removidos. Este é um inteiro positivo sequencial. É função da estação de gerência ler a tabela quando um <i>trap</i> ocorre ou quando a estação <i>poll</i> o gerente. A aplicação de gerência analisa a tabela de alertas para determinar se existem novas entradas. Isso é feito comparando o último índice conhecido com o índice de mais alto valor na tabela.
SvAlertPrinter	Índice da impressora onde o problema aconteceu.
SvAlertCode	Código que identifica o alerta. Este tipo é uma enumeração e pode assumir os valores: Other(1) – definido para possível expansão da MIB, unknown(2) – Problema desconhecido, Offline(3) – SpoolView não conseguiu abrir o dispositivo de impressão, online(4) – SpoolView conseguiu abrir o dispositivo após uma situação de erro, breakflow(5) – SpoolView estava transmitindo caracteres para a impressora e o fluxo foi interrompido, normalflow(6) – O fluxo de caracteres normalizou após um problema. Existem dois alertas relacionados a limiares para a impressora. São eles: lowspeed(7) – A impressora está imprimindo com velocidade abaixo da definida na tabela de limiares. requestoverflow – O número de pedidos para esta impressora excedeu o limite máximo de pedidos definidos.
SvAlertDescription	Descrição do tipo de alerta
SvAlertTime	Data e hora em que o alerta foi reportado

Tabela 5.8. Tabela de Alertas da SpoolView MIB

5.3.4 O Grupo de Limiares

O Grupo de limiares consiste de uma única tabela na qual todas as limiares são representadas. Esta seção provê uma discussão sobre como as limiares são utilizadas em conjunto com o grupo de alerta.

Na solução de gerência do SpoolView o agente é responsável por monitorar o comportamento dos *despoolers* e reportar situações anormais (ver seção 5.3.3). Entre estas situações estão os problemas com limiares. Estes problemas incluem:

- Impressora com desempenho baixo. A impressora está imprimindo abaixo da velocidade considerada “ótima” para impressão.

- Estouro da fila de impressão. O número de pedidos enfileirados para esta impressora está acima de um patamar pré-definido.

Nota-se que para determinarmos a existência de algum problema, no nosso caso, que comparar valores coletados do sistema (SpoolView) com variáveis totalmente dependentes da impressora em questão. Estas variáveis são: velocidade “ótima” de impressão e tamanho da fila de pedidos para a impressora. Onde o agente coleta tais parâmetros para decidir que uma impressora violou alguma limiar?

Estes parâmetros são definidos e armazenados como uma extensão da tabela de impressora. Esta extensão é representada por uma tabela no grupo de limiares (Ver Tabela 5.9).

<i>OID</i>	<i>Descrição</i>
SvPrinterThresholdSpeed	Contém a velocidade considerada ideal de impressão para cada impressora na tabela de filas de impressão. Esta velocidade é configurada pelo administrador do sistema de impressão.
SvPrinterThresholdNumberRequest	Número máximo de pedidos que podem ser enfileirados para uma determinada impressora.

Tabela 5.9. Tabela de Limiares da SpoolView MIB

A tabela de limiares é indexada por `jmGeneralJobSetIndex` expandindo assim a tabela de filas de impressão. Recapitulando, cabe ao agente que implementa a MIB SpoolView manter uma constante monitoração sobre o SpoolView e, em paralelo, comparar as informações coletadas do produto com valores armazenados na tabela de limiares. Quando um problema é detectado um alerta é produzido e conseqüentemente uma *Trap* é disparada.

5.4 Considerações na Implementação

No processo de especificação da MIB SpoolView nos deparamos com situações que merecem ser comentadas. Nesta seção, discutiremos aspectos relevantes ao desenvolvimento do agente que implementa a MIB SpoolView. Estes aspectos incluem: coleta de informações, o problema do compartilhamento de uma única porta para agentes SNMP e cuidados com tratamento de erros devido a unicidade de comando *set..*

5.4.1 Arquitetura para a Gerência SpoolView

No modelo SNMP, o agente desempenha um papel fundamental no processo de coleta e armazenamento de informações na MIB, servindo como mediador entre a aplicação de gerência e o dispositivo sendo gerenciado. Nesta seção, discutiremos aspectos inerentes ao agente. Destacamos uma proposta para o compartilhamento de uma porta de comunicação por vários agentes e apresentamos alternativas para a comunicação do SpoolView com o agente que implementa a SpoolView MIB.

Agentes SNMP são normalmente implementados como processos servidores [ROSE 91], os quais lêem informações de variáveis, tabelas, arquivos ou outros processos rodando na máquina. Cabe aos agentes disponibilizarem as informações adquiridas para as estações de gerências.

Como processos servidores, eles se prendem a um porta de comunicação (ver [STEV 90, WW06 95, GEDE 93]) bem definida, esperando pedidos dos gerentes. Porém, quando um agente se prende a uma única porta, significa que não podemos colocar mais de um agente na mesma máquina “pendurado” na mesma porta. Então, como fazer para gerenciar dispositivos que possuem seus próprios agentes?

5.4.1.1 Tecnologia AgentX

A tecnologia AgentX é uma proposta da IETF para a padronização de uma estrutura utilizada para a extensão de agentes SNMP. Ou seja, define um esquema que permite que um único agente SNMP, chamado agente mestre, possa ser usado para controlar e monitorar dispositivos de diferentes fabricantes através de sub-agentes.

O estrutura proposta pelo AgentX define os seguintes elementos [RFC2257]:

- Um agente mestre que é disponibilizado na porta default SNMP e que aceita mensagens SNMP.
- Um conjunto de sub-agentes que instrumenta os nós gerenciados.
- Um protocolo que opera entre o agente mestre e os sub-agentes, permitindo aos sub-agentes conectarem-se ao agente mestre, e ao agente mestre multiplexar mensagens SNMP para os sub-agentes.
- Por fim, um conjunto de ferramentas para facilitar o desenvolvimento de sub-agentes.

Conhecendo as peças envolvidas neste esquema, fica fácil mostrarmos como esta nova proposta resolve o problema de uma única porta para agentes SNMP. Vamos examinar um cenário típico que utiliza o agente SpoolView para exemplificar o uso do esquema AgentX. Existem três participantes nessa sessão: uma aplicação de gerência,

localizada na estação de gerência; o agente mestre SNMP; e o sub-agente SpoolView, usado para gerenciar as MIBs definidas para a solução de gerência SpoolView (ver Figura 5.2).

Vejamos o comportamento de cada elemento na arquitetura:

- A estação de gerência roda um cliente SNMP para enviar pedidos e receber respostas do agente SNMP.
- O agente mestre SNMP é um servidor rodando na máquina. O agente mestre age como um mediador entre a estação de gerência e o sub agente SpoolView: traduzindo mensagens SNMP da estação de gerência em mensagens AgentX para o sub-agente e vice-versa.
- O sub-agente é um servidor rodando na máquina; ele é responsável pelas MIBs que contém informações referentes ao produto SpoolView (SpoolView MIB), à sistemas de impressão (Job Monitoring MIB) e à aplicação em geral (AppMIB). O sub-agente usa o protocolo AgentX para comunicação com o agente mestre SNMP e um esquema proprietário para acessar os dados referentes ao serviço de impressão.

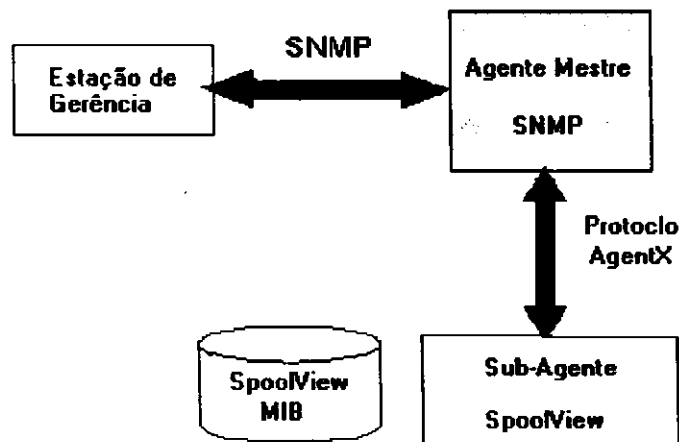


Figura 5.2. Arquitetura AgentX com um Sub-Agente

A vantagem dessa arquitetura, além do fato de utilizar um protocolo aberto padrão (AgentX), o que permite uma melhor integração de sub-agentes desenvolvidos por diferentes fabricantes, é o uso de um único *daemon* SNMP (agente mestre) em uma porta bem conhecida, para tratar mensagens SNMP. Resolvendo assim o problema de múltiplos agentes e uma única porta.

5.4.1.2 Comunicação Entre o Agente e o SpoolView

Definimos na seção anterior como o agente SpoolView irá se comunicar com a aplicação de gerência. Porém, como as informações do servidor de impressão são coletadas pelo agente?

Uma possibilidade é o próprio servidor de impressão conter o agente, ou seja, ele pode ser modificado para funcionar como o agente. Quando uma mensagem chega ao servidor, este mapea OIDs para campos dos arquivos de controle e retorna os pedidos com respostas adequadas.

Porém, se definirmos o agente como um processo separado do servidor de impressão temos então que lidar com comunicação entre processos ou IPC (*Inter Process Communication*). Neste caso podemos definir algumas alternativas:

- IPC via arquivo. Podemos desenvolver o agente SpoolView de forma que ele possa ter “conhecimento” sobre como arquivos de configuração do SpoolView funcionam. Dessa forma, o agente acessa diretamente os arquivos, armazenando informações geradas a partir destes arquivos de configuração em suas estruturas internas.
- IPC via qualquer mecanismo de IPC do UNIX (pipes, mensagens, etc.). Neste caso, teríamos que definir um protocolo de comunicação entre o agente e o SpoolView de forma que pudessemos coletar os dados necessários para a MIB.
- IPC via shared memory. Definimos uma área de memória compartilhada onde tanto agente como SpoolView terão acesso. A dificuldade neste caso está em definir a organização dos dados a serem compartilhados.
- Por fim, podemos definir um único processo, permanentemente no ar, que dispare como thread²⁷ ambos, agente e servidor de impressão. Desta forma, o agente seria também um thread do mesmo processo e portanto poderia acessar os dados, já que threads de mesmo processo compartilham os dados.

A alternativa escolhida, na nossa solução, para comunicação entre o agente e o SpoolView foi IPC através de arquivos. O principal motivo foi a facilidade de mapeamento entre os arquivos de configuração do SpoolView e os objetos definidos nas MIBs. No entanto, as alternativas apresentadas não devem ser desprezadas, elas representam possibilidades reais que devem ser consideradas no momento da implementação de uma solução de gerência para uma aplicação.

²⁷ Parte do código de um processo ao qual o sistema operacional aloca tempo de CPU para a execução do mesmo. Um Thread pode executar qualquer parte do código da aplicação. Todos os threads de um processo compartilham o mesmo espaço virtual de memória, variáveis globais, e recursos do sistemas operacional.

5.5 Resumo

O capítulo 5 representa o último passo na definição de uma solução de gerência para uma aplicação. O objetivo do capítulo foi mostrar como podemos complementar soluções existentes (MIBs padrões) para fornecer informações particulares da aplicação gerenciada. Para isso usamos um produto para a gerência de impressoras disponível no mercado, o SpoolView.

Com esse intuito apresentamos requisitos para a extensão da Job Monitoring MIB. O destaque foi dado aos esquemas definidos para o controle de pedidos de impressão do SpoolView utilizando SNMP e as tabelas de alertas que definem *traps* a serem enviados para a aplicação de gerência.

Por fim, discutimos detalhes e questões levantadas a serem consideradas no momento de implementar a solução de gerência. Apresentamos a proposta IETF para a extensão de agentes SNMP (AgentX) e como o agente SpoolView se comunica com o servidor de impressão para coletar os dados para a MIB.

6 Conclusão

Redes e sistemas distribuídos estão crescendo em importância e tornaram-se essenciais e críticos no mundo dos negócios. O controle e uso eficiente dos recursos distribuídos na rede levaram as empresas a investirem grande quantidade de dinheiro tanto na gerência de recursos de *hardware* quanto recursos de *software*.

Com o uso de ferramentas padronizadas de gerência, os usuários podem obter um maior retorno dos investimentos realizados e controlar custos inerentes ao crescimento de suas redes. Pelas vantagens que oferecem estas ferramentas de gerência, é importante dominar suas técnicas de projeto e implementação.

O maior problema enfrentado pelos programadores na área de gerência, atualmente, é que não existe um trabalho que discuta de forma integrada o processo de desenvolvimento de aplicações gerenciáveis. Esse problema tende a ser resolvido à medida que as pesquisas evoluírem, mas, por enquanto, uma forma de minimizá-lo é fornecer ao programador, em um único texto, informações que o auxiliem no processo de desenvolvimento destas aplicações.

Nos capítulos anteriores, foram apresentadas, com razoável nível de detalhes, um estudo sobre o desenvolvimento, adaptação e gerência de aplicações. Este estudo foi apresentado gradativamente, de forma hierárquica, introduzindo ao leitor, a cada capítulo, uma discussão mais especializada sobre o assunto de gerência, até culminar com uma proposta para a gerência de uma aplicação, o SpoolView. Neste sentido discutimos a gerência de rede de forma geral, introduzimos o conceito de gerência de aplicações, requisitos e MIBs propostas pela IETF para a gerência de aplicações.

Apresentamos a gerência de um tipo específico de aplicação, normalmente relacionado ao produto de software que está sendo adaptado ou desenvolvido, e mostramos uma MIB IETF para a gerência de sistemas de impressão. Por fim, apresentamos os requisitos e uma proposta para a gerência de um produto de software existente no mercado.

As principais contribuições desse trabalho foram: 1) organizar em um único texto as diversas propostas da comunidade científica para a gerência de aplicações; 2) efetivamente apresentar de forma gradual os conceitos, requisitos e alternativas para o desenvolvimento e adaptação de aplicações para SNMP; e 3) mostrar, de forma clara, que as propostas da comunidade científica não são suficientes para a gerência de aplicações e que muitas das vezes se faz necessário a extensão de MIBs existentes para efetivamente gerenciar uma aplicação específica.

Os conceitos foram fixados, através de um exemplo concreto, no estudo de caso, quando especificamos a adaptação de um produto já existente no mercado, SpoolView, para SNMP. Utilizamos todas as MIBs discutidas nos capítulos anteriores com o objetivo de apresentar uma solução modular, aberta e baseada em propostas a serem padronizadas. A seguir será feita uma avaliação dos objetivos propostos na introdução deste trabalho, bem como algumas conclusões e sugestões para trabalhos futuros.

6.1 Avaliação dos objetivos do Trabalho

Na seção 1.2, foram apresentados os objetivos desse trabalho. Nessa seção, será discutido até que ponto esses objetivos foram atingidos.

a) Apresentar de forma clara o que significa gerenciar aplicações

Através dos conceitos apresentado, dos requisitos levantados para a gerência de aplicações e a classificação dos mesmos em áreas de gerência (desempenho, faltas, contabilidade, segurança e configuração), demos ao programador uma visão geral do que é gerenciar aplicações e principalmente o que se pretende gerenciar nas aplicações.

b) Apresentar as soluções proposta pela IETF (*Internet Engineering Task Force*) para a gerência de aplicações.

Apresentamos as principais MIBs para a gerência de aplicações. SysAppMIB, que contém informações de cunho genérico e não necessita que as aplicações sejam

instrumentadas, e, ApplMIB que apresenta informações genéricas, mas que só podem ser adquiridas com instrumentação da aplicação.

Estas MIBs oferecem um primeiro nível para a gerência de aplicações e que deve ser utilizado pelo programador para integrar as aplicações ao ambiente de gerência.

Acreditamos que uma das limitações do trabalho foi ter se explorado apenas as propostas IETF, não discutindo propostas de outros grupos que estudam gerência de redes. Por exemplo, o grupo DMTF [WW13 98] (Desktop Management Task Force) que apresenta uma proposta para a gerência de redes via Web, chamada WBEM [WW11 98, WW13 98] (Web-Based Enterprise Management) ou a Sun Microsystems com a JMAPI (Java Management API) conjunto de objetos para o desenvolvimento de soluções de gerência [JMAP 98]. Um comparativo entre estas propostas teria dado ao programador maior flexibilidade na escolha do que melhor se adequasse ao seu projeto.

c) Apresentar os requisitos que nortearam o desenvolvimento das soluções para a gerência de aplicações;

A gerência de aplicações envolve várias fases. Estas fases estão relacionadas com o tipo e a que nível as informações têm que ser disponibilizadas. Temos então: informações genéricas, informações sobre o tipo de aplicação que se quer gerenciar, e informações sobre o software final a ser gerenciado. Em cada uma destas fases apresentamos os requisitos a serem considerados e como as soluções sendo padronizadas contemplavam estes requisitos.

Os requisitos sugeridos por este trabalho foram: requisitos para a gerência de desempenho (tempo de resposta, vazão, utilização de recursos e planejamento de capacidade), gerência de faltas (disponibilidade, monitoração dos estados da aplicação, detecção de erros pelo agente), gerência de contabilidade e gerência de configuração (controle da aplicação e relacionamento dos processos executando com pacotes de software instalados). Do ponto de vista do programador, estes requisitos cobrem satisfatoriamente as informações que podem ser extraídas, analisadas e apresentadas ao gerente de rede em seu sistema de gerência.

d) Mostrar que outras bases de dados de gerência são necessárias para a gerência de aplicações

Apresentamos as MIBs IETF para a gerência de aplicações e mostramos de forma clara que elas não são suficientes para a gerência de uma aplicação qualquer. Por serem soluções genéricas, elas não oferecem um mecanismo que permita controlar as funcionalidades particulares de uma aplicação, nem um esquema que permita a integração de *traps* na solução de gerência. Cabe ao programador buscar formas alternativas de expansão destas MIBs permitindo a gerência de aplicações específicas.

e) Apresentar os aspectos práticos de como tornar uma aplicação gerenciável

Vários aspectos práticos foram apresentados principalmente quando definimos MIBs para o tipo específico de aplicação e para o produto de software a ser gerenciado e as relacionamos com as MIBs para a gerência de aplicações. Por fim, foram abordados detalhes de implementação, no estudo de caso, destacando a tecnologia AgentX para a extensão de agentes SNMP e sugestões para comunicação entre agente e aplicações. Nós acreditamos que a seção que trata de detalhes de implementação poderia ser melhorada, fornecendo um nível mais detalhado sobre aspectos de implementação.

f) Avaliar o estudo apresentado para a adaptação e desenvolvimento de uma aplicação em SNMP, a partir de um estudo de caso.

A partir do estudo de caso pôde-se constatar que:

- Precisamos integrar as diversas bases de dados para a gerência de aplicações com o intuito de fornecer uma solução aberta e integrada.
- As MIBs IETF não são suficientes para gerenciar um sistema de impressão. Outras MIBs têm que ser definidas para suprir deficiências como controle refinado da aplicação e suporte a *traps*.
- O uso de tecnologias como AgentX é importante para a integração e distribuição da solução de gerência SpoolView em diferentes plataformas.
- Deve-se ter uma preocupação especial em como definir a comunicação entre o agente que instrumenta as MIBs e a aplicação efetivamente gerenciada.

Apesar de ter ilustrado muito bem o uso das MIBs de aplicação e MIBs desenvolvidas especificamente para sistemas de impressão, a proposta para a gerência do SpoolView não foi de todo completa. Apesar de termos descrito a MIB em ASN.1 (ver Apêndice A), não implementamos um agente que instrumentasse a MIB SpoolView. Essa tarefa é sugerida como trabalho futuro.

6.2 Conclusões Finais e Trabalhos Futuros

Cabe ao programador julgar a qualidade dos requisitos e considerações feitas no trabalho através do desenvolvimento de aplicações gerenciáveis. Contudo, acreditamos que os objetivos especificados no capítulo 1 foram atingidos de forma satisfatória.

Esse trabalho pode servir como uma referência bibliográfica ao programador. Uma literatura que esclarece pontos obscuros no desenvolvimento e gerência de aplicações com o protocolo SNMP.

Tendo como base os resultados deste trabalho, surgem inúmeras alternativas para o desenvolvimento de novos trabalhos que viriam a ampliar este estudo não só na área de serviços de impressão, mas também na área de gerência de aplicações. Algumas sugestões que contribuiriam para atingir essa meta são:

- Formulação de uma metodologia para o desenvolvimento de aplicações gerenciáveis.
- Considerações específicas de segurança com o uso do SNMPv3
- Considerações para a integração da solução de gerência SpoolView com a Printer MIB da IETF.
- Considerações para gerência do SpoolView via Web utilizando SNMP.
- Implementação da solução de gerência para o SpoolView.
- Análise comparativa entre diferentes propostas para a gerência de aplicações apresentada por diferentes grupos (IETF, DMTF).

A. A MIB SpoolView

```

SpoolViewMIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY, OBJECT-TYPE, enterprises, Integer32
FROM SNMPv2-SMI
TEXTUAL-CONVENTION
FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF;

-- As seguintes Convenções Textuais (textual-conventions)
-- são necessárias para implementar certos atributos, mas não são
-- necessárias para compilar esta MIB.
-- Elas são colocadas aqui por conveniência
-- DateAndTime
FROM SNMPv2-TC

SpoolViewMIB MODULE-IDENTITY
LAST-UPDATED "11/12/1998"
ORGANIZATION "Federal University of Paraiba UFPB"
CONTACT-INFO
    "Dalmer Azevedo Jr.
    Address: 541 Del Medio Ave. #134
    Mountain View, CA 94040
    USA

    Tel/Fax: (650)559-9818
    E-mail: dalmer@ix.netcom.com"

DESCRIPTION
    "Base de dados para a Gerência do SpoolView. SpoolView é um
    sistema de gerenciamento de impressão desenvolvido pela
    Light Infocon Technology, Inc."
 ::= { experimental 1 }

-- Convenções Textuais sobre esta MIB

```

```

-- *****
--
--svControlGroup - Este grupo contém duas tabelas que permitem
-- o controle dos pedidos de impressão e impressoras no SpoolView.
--
-- *****

-- *****
--
--      svJobControlTable - Tabela de controle de pedidos de
--      impressão.
--
-- *****

svControlGroup OBJECT IDENTIFIER ::= { SpoolViewMIB 1 }
svJobControlTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF svJobControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Esta tabela fornece a habilidade de controlar os pedidos
        de impressão do SpoolView, expandindo assim a tabela de
        pedidos de impressão da Job Monitring MIB através do índice
        jmJobIndex."

    ::= { svControlGroup 1 }

svJobControlEntry OBJECT-TYPE
    SYNTAX          SvJobControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Uma entrada svJobControlEntry contém informações para
        suportar o controle de um pedido de impressão no
        SpoolView."

    INDEX { jmJobIndex }
    ::= { svJobControlTable 1 }

SvJobControlEntry ::= SEQUENCE {
        svJobControlRemove      TruthValue,
        svJobControlHold       TruthValue }

svJobControlRemove OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Atribuir o valor true a esta variável dispara a ação de
        remoção do pedido de impressão"
    DEFVAL { false }
    ::= { svJobControlEntry 1 }

svJobControlHold OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION

```

```

    "Atribuir o valor true a esta variável prende o pedido de
    impressão. Atribuir o valor false libera o pedido de
    impressão"
 ::= { svJobControlEntry 2 }

-- *****
--
--     svPrinterControlTable - Tabela de controle das filas de
--     impressão.
--
-- *****

svPrinterControlTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF svPrinterControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Esta tabela fornece a habilidade de controlar filas de
        impressão do SpoolView, expandindo assim a tabela de filas
        de impressão (General Table) da Job Monitoring MIB através
        do índice jmGeneralJobSetIndex."
 ::= { svControlGroup 2 }

svPrinterControlEntry OBJECT-TYPE
    SYNTAX          SvPrinterControlEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Uma entrada nesta tabela dá ao administrador, com
        apropriada permissão, a habilidade de suspender uma
        impressora, prosseguir com a impressão, matar o despooler
        de impressão, criar um despooler de impressão e
        reinicializar todo o SpoolView."

    INDEX { jmGeneralJobSetIndex }
 ::= { svPrinterControlTable 1 }

SvPrinterControlEntry ::= SEQUENCE {
    SvPrinterControlSuspend TruthValue,
    SvPrinterControlStop    TruthValue,
    SvPrinterControlReset   TruthValue
}

SvPrinterControlSuspend OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Atribuir o valor true a esta variável dispara uma ação para
        a suspensão da impressora. Atribuir o valor false libera a
        impressora."
    DEFVAL { false }
 ::= { svPrinterControlEntry 1 }

SvPrinterControlStop OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION

```

```

    "Atribuir o valor true a esta variável dispara uma ação para
    a parada do servidor de impressora. Parar o servidor
    significa que iremos "matar" o processo despooler.
    Atribuir o valor false cria um novo despooler para a
    impressora."
 ::= { svPrinterControlEntry 2 }

SvPrinterControlReset OBJECT-TYPE
    SYNTAX          TruthValue
    MAX-ACCESS      read-write
    STATUS          current
    DESCRIPTION
        "Atribuir o valor true a esta variável dispara uma ação para
        reinicialização de todos os arquivos de configuração do
        SpoolView"
 ::= { svPrinterControlEntry 3 }

-- *****
--
--     svThresholdGroup - Este grupo contém informações sobre
--     limiaries que serão utilizadas em conjunto com a tabela de alerta.
--     O agente notificará a estação de gerência se qualquer limiar
--     for ultrapassada.
--     O processo é simples. O agente monitora o SpoolView e compara
--     algumas medidas de desempenho ou configuração com valores
--     armazenados na tabela de limiaries. Caso haja algum estouro
--     o agente adiciona uma linha na tabela de alertas. Conseqüentemente
--     uma trap é disparada.
--
-- *****
--
-- *****
--
--     svPrinterThresholdTable - Esta tabela contém limiaries sobre a
--     impressora. Estes escalares limitam o numero de pedidos
--     na fila e queda na velocidade de impressão.
--
-- *****

svThresholdGroup OBJECT IDENTIFIER ::= { SpoolViewMIB 2 }
svPrinterThresholdTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF svPrinterThresholdEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Esta tabela fornece a habilidade de definir limiaries para
        as filas de impressao do SpoolView, expandindo assim a
        tabela geral (General Table) da Job Monitoring MIB através
        do indice jmGeneralJobSetIndex."
 ::= { svThresholdGroup 1 }

svPrinterThresholdEntry OBJECT-TYPE
    SYNTAX          SvPrinterThresholdEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "Uma entrada nesta tabela contém informações sobre limiaries
        de desempenho e fila de impressao. Esta tabela define
        informações de configuração (limiaries) utilizadas em
        conjunto com a tabela de alerta. O agente monitora a

```


tabela de limiares. Caso uma limiar seja ultrapassada, o agente dispara uma notificação para a estação de gerência (Trap)."

```
INDEX { jmGeneralJobSetIndex }  
 ::= { svPrinterThresholdTable 1 }
```

```
SvPrinterThresholdEntry ::= SEQUENCE {  
  SvPrinterThresholdSpeed Integer32 (0..2147483647),  
  SvPrinterThresholdNumberRequest Integer32 (0..2147483647)  
 }
```

```
SvPrinterThresholdSpeed OBJECT-TYPE
```

```
SYNTAX Integer32 (0..2147483647)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"Contém a velocidade ideal de impressão. Caso a velocidade atual da impressora seja menor que esse limiar, uma notificação deverá ser disparada pelo agente. A velocidade corrente é um atributo da impressora no SpoolView"

```
 ::= { svPrinterThresholdEntry 1 }
```

```
SvPrinterThresholdNumberRequest OBJECT-TYPE
```

```
SYNTAX Integer32 (0..2147483647)
```

```
MAX-ACCESS read-write
```

```
STATUS current
```

```
DESCRIPTION
```

"Contém o número máximo de pedidos para essa fila de impressão. Caso o número de pedidos para essa fila ultrapasse este valor, uma notificação deverá ser disparada pelo agente."

```
 ::= { svPrinterThresholdEntry 2 }
```

```
-- *****  
-- The Alerts Group  
-- A tabela svAlertTable lista todos os alertas relacionados ao  
-- a alguma impressora do SpoolView. Dividimos os  
-- alertas em críticos e nao críticos.  
-- A tabela contém informações como a data e a hora que o alerta foi  
-- reportado junto com a descrição do alerta.  
-- *****
```

```
svAlertGroup OBJECT IDENTIFIER ::= { SpoolViewMIB 3 }
```

```
svAlertTable OBJECT-TYPE
```

```
SYNTAX
```

```
SEQUENCE OF SvAlertEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION ""
```

```
 ::= { svAlertGroup 1 }
```

```
svAlertEntry OBJECT-TYPE
```

```
SYNTAX SvAlertEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

DESCRIPTION

"Representa um alerta. Normalmente associado a uma impressora."

INDEX { jmGeneralJobSetIndex, SvAlertIndex }
 ::= { svAlertTable 1 }

SvAlertEntry ::= SEQUENCE {
 SvAlertIndex Integer32,
 SvAlertCode INTEGER,
 SvtAlertDescription OCTET STRING,
 SvtAlertTime OCTET STRING
 }

SvAlertIndex OBJECT-TYPE

SYNTAX Integer32
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"O valor do índice utilizado para determinar quais alertas foram adicionados ou removidos da tabela de alertas. Esse OID é um inteiro crescente inicializado com zero toda vez que a impressora é resetada.

::= { svAlertEntry 1 }

SvAlertCode OBJECT-TYPE

SYNTAX INTEGER {
 other(1),
 unknown(2),
 offline(3),
 online(4),
 breakflow(5),
 normalflow(6),
 -- Problemas de velocidade com a impressora
 lowspeed(7),
 -- overflow na tabela de pedidos
 requestoverflow(8)
 }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Codigo que descreve o tipo de alerta sendo adicionado a tabela de alertas."

::= { svAlertEntry 2 }

SvAlertDescription OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..255))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Descreve o tipo de alerta que esta sendo cadastrado."

::= { svAlertEntry 3 }

SvAlertTime OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(0..26))

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"String com data e hora em que a linha contendo o alerta foi adicionada."

```
 ::= { svAlertEntry 4 }

-- Conformance Information

SvMIBConformance OBJECT IDENTIFIER ::= { SpoolViewMIB 3 }

-- compliance statements
SvMIBCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Para que um agente esteja de acordo com esta MIB, ele deve
    implementar os seguintes grupos"
  MODULE -- este modulo
  MANDATORY-GROUPS {
    svControlGroup, svThresholdGroup, svAlertGroup
  }

-- Nap existem grupos condicionais ou opcionais

 ::= { SvMIBConformance 1 }

END
```

Bibliografia

- [ALLA 93] Leinwand, Allan - *Network Management: A practical perspective* - Addison-Wesley, 1993.
- [KRUP 98] Kalbfleisch, C., Krupczak, C., Presuhn, R., Saperia, J. - *Application Management MIB* - draft-ietf-aplmib-mib-08.txt, IETF 1998.
- [BARR 96] Bruins, Barry; - *Some Experiences with Emerging Management Technologies* - The Quaterly Newsletter of SNMP Technology, Comment, and Events (sm), Volume 4, Number 3, July, 1996.
- [BERG 98] Bergman, Ron - *Job Submission Protocol Mapping Recommendations for the Job Monitoring MIB* - draft-ietf-printmib-job-protomap-03.txt, Internet Draft, 1998.
- [BMC1 96] BMC Software - *Enterprise-Wide Analisis of Network-Based Application Performance* - <http://www.bmc.com/products/articles/a03wp.html>; White paper 1996
- [BMC2 96] BMC Software - *PATROL Architecture - Raising the Management Application Standard* - <http://www.bmc.com/products/pat/amo/patwp.html>; White Paper 1996.
- [BMC3 96] BMC Software - *Proactive Application Management for SAP R/3* - <http://www.bmc.com/products/pat/amo/pkrwp.html>; White Paper 1996.
- [BMC4 96] BMC Software - *Enterprise-Wide Analysis of Netwok-Based Application Performance* - <http://www.bmc.com/products/articles/a03wp.html>; White Paper 1996.
- [CARR 94] Carrilho, José A. Madeira, Edmundo R. M., *A - Scheme for FTP Management* - Proc. INET 94 / JENC5, 1994.
- [CGIS 97] *The CGI Specification* - <http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>, 1997
- [CHIU 92] Chiu, D., and Sudama, R. - *Network Monitoring Explained: Design and Application* - New York, Ellis Horwood, 1992.

- [COMM 91] Commer, D. E. & Stevens D. L. - *Internetworking with TCPC/IP – Design, Implementation, and Internals* - volume II, prentice Hall, 1991.
- [DAVI 93] Perkins, David - *Understanding SNMP MIBS* - September 1993.
- [DIVA 96] Udupa, Divakara - *Network Management Systems Essencials* - McGraw-Hill, 1996
- [DMI1 94] *The Desktop Management Interface (DMI) – Enabling the Next Generation of Desktop PC Systems*. The Desktop Management Task Force (DMTF), 1994.
- [DMI2 94] *The Desktop Management Interface and Management Information Format Files – Enabling Technology for Manageable, Intellingent PCs*. The Desktop Management Task Force (DMTF), 1994.
- [DOUG 93] Stevenson, Douglas W. - *Network Management, What is it and what isn't* - <http://smurfland.cit.buffalo.edu/Netman/Docs/Stevenson/>
- [DUPU 89] Dupuy, A. et al. - *Network fault Management: A User's View* - Proceedings, First International Symposium on integrated Network Management, New York, North-Holland, 1989.
- [FERG 98] Ferguson, Linda – DTMF Promotes Use of eXtensible Markup Language (XML) for Standards-based Management Solutions – DTMF Press Release, 1998.
- [FREI 98] Freire, Raissa Dantas - *Um Survey Sobre Aplicações de Gerência de Redes* - Relatório Técnico RT-GRC-981 - UFPB, 1998.
- [FRIE 90] Fried, S. and Tjong, J. - *Implementing Integrated Monitoring Systems for Heterogeneous Networks* - In *Networks Management and Control*, edited by A. Kershenbaum, M. Malek, and M. Wall, New York, Plenum. 1990.
- [GEDE 93] Santos, Gedeon José - *Considerações para o Desenvolvimeto de Aplicações Distribuidas em Ambientes Heterogêneos* - Campina Grande-PB, 1993.
- [GREE 97] Greene, M.;Gudur, S. - *Definitions of Managed Objects for Extensible SNMP Agents* - draft-ietf-agentx-mib-01.txt, November 1997.
- [HAST 96] Hasting, Tom – *Job Monitoring MIB Proposal for a New Standard Track Project* – Transparências apresentadas a IETF. Dezembro de 1996.
- [HAST 98] Hastings, Tom; Bergman, Ron; Isaacson Scott, Lewis, Harry – *Job Monitoring MIB* - draft-ietf-printmib-job-monitor-07.txt, Internet Draft, 1998.

- [IEEE 90] IEEE Std 802.6-1990 - *Local and Metropolitan Area Networks, Distributed Queue Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)* - Institute of Electrical and Electronics Engineers, Inc. , 1990
- [ISO 84] Information Processing Systems – Open System Interconnection Basic Reference Model. International Organization for Standardization and International Electrotechnical Committee, 1984. International Standard 7498.
- [JMAP 98] The Source for Java Technology - *JavaManagement API* – java.sun.com/products/JavaManagement
- [KRIS 96] Streeter, April Kristi; - *Management In Motion* - Lan Times, April 29, 1996 issue.
- [KRUP 95] Krupczac, Bobby - *Systems Management and the Internet Management Framework* - Connexions, Vol. 9, No. 8, August 1995.
- [KRUP 98] Krupczac, Cheryl - *Application & Systems Management. The New Challenge* - Empire Technologym Inc. 1998
- [LAND 98] Landmark - *Performance Management* - <http://www.landmark.com/PERFMAN/WhatisPM.html>, 1998.
- [LUCA 95] Deri. Luca – “Surfin’ Network Resouces Across the Web”, IBM Zurich Reasearch Laboratory, 1995
- [MEIR 97] Meira, Dilmar Malheiros - *Um Modelo para Correlação de Alarmes em Redes de Telecomunicações* - Tese apresentada ao Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito para à obtenção do título de Doutor em Ciência da Computação, UFMG 1997.
- [NEUM 95] NEUMAN, José de Souza; Oliveira, A. Mauro B. de - *Curso Avançado Sobre Gerênciametos de Redes* - LAR - Laboratório Multiinstitucional de Redes e Sistemas Distribuídos, 1995.
- [NORM 94] Normal, William B., *A Guided Tour Through SNMP version 2*, Merit Network Inc., 1994.
- [PATR 96] Mullaney, Patrick; - *Overview of a Web-based Agent* - The Quaterly Newsletter of SNMP Technology, Comment, and Events (sm), Volume 4, Number 3, July, 1996.
- [PRAS 95] PRAS, Aiko - *Network Management Architectures* - CTIT Ph. D-thesis series no. 95-02, Nethernlands, 1995.
- [RFC1155] Rose, M. T.; McCloghrie, K. - *Structure and identification of management information for TCP/IP based internets* – Maio 1990; 22 p.

- [RFC1156] McCloghrie, K., Rose, M. - *Management Information Base for network management of TCP/IP-based internets* - Maio 1990.
- [RFC1157] RFC 1157 - *A Simple Network Management Protocol (SNMP)* - Maio 1990.
- [RFC1179] RFC 1179 - *Line Printer Daemon Protocol* - Agosto 1990.
- [RFC1213] McCloghrie, K.; Rose, M.; - *Management Information Base for Network Management of TCP/IP-based internets: MIB-II* - Março 1991.
- [RFC1227] Network Working Group Rose, M. T. - *SNMP SMUX Protocol and MIB* - May 1991; 13p.
- [RFC1759] Simth, R; Wright, F.; Hasting, T.; Zilles, S.; Gyllenskog, J - *Printer MIB* - March 1995.
- [RFC2068] Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H. - *Hypertext Transfer Protocol - HTTP/1.1* - January 1997.
- [RFC2248] Freed, N., and S. Kille, "Network Services Monitoring MIB", RFC2248, January 1998.
- [RFC2257] Daniele, M., Wijnen, B., Francisco, D. - *Agent Extensibility (AgentX) Protocol* - RFC 2257, January 1998.
- [RFC2275] Wijnen, B., Watson, T. J. - *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)* - RFC 2275, January, 1998.
- [RFC2287] Krupczak, C., and J. Saperia, - *Definitions of System-Level Managed Objects for Applications* - RFC 2287, February 1998.
- [ROSE 91] Rose, Marshall T. - *The Simple Book: An Introduction of Management of TCP/IP-Based Internets* - Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
- [RUBE 96] Ruber, Peter; - *Apps Control from the Ground Up* - Lan Times, April 29, 1996 issue.
- [SAUV 96] Sauv e, Jacques Philippe - *SpoolView Users's Manual* - Infocon-PB, 1995.
- [SAUV 97] Sauv e, Jacques Philippe - *O Processo de Desenvolvimento de Software no Grupo de Redes de Computadores da UFPB* - Relat rio T cnico RT-GRC-98-1 - UFPB, 1998
- [SAUV 98] Sauv e, Jacques Philippe - *O Projeto WebManager do Grupo de redes de Computadores da UFPB* - Relat rio T cnico RT-GRC-98-2 - UFPB, 1998.

- [SAUV98A] Sauv , Jacques Philippe - *Ger ncia de Redes de Computadores* - DSC/UEPB/GRC, 1998
- [SCHI 94] Schiller, Jeffrey - *Distributed System Security* - Massachusetts Institute of Technology, 1994.
- [SHIM 96] Shimmim, Bradley F.; - *Roping and Runaway Apps* - Lan Times, April 29, 1996 issue.
- [SLOM 93] SLOMAN, M.; MAGEE J.; TWIDLE K.; KRAMER, J. - *An Architecture For Managing Distributed Systems* - Department of Computing, Imperial College, London - 1993.
- [STAL 93] Stallings, William. *SNMP, SNMPv2, and CMIP*. Don Mills: Addison-Wesley, 1993
- [STEV 90] Stevens, W. Richard - *UNIX Networking Programming*, Prentice Hall Software Series, 1990
- [STEV 94] Waldbusser, Steven. *SNMP and Distributed Systems Management* Carnegie Mellon University Press, 1994.
- [SUNM 90] Sun Microsystems - *Network Programming Guide* - 1990.
- [TERP 92] Terplan, K., *Communication Networks Management*. Englewood Clifff, N.J.: Prentice-Hall. 1992.
- [TOWN 95] Townsend, Robert L. *SNMP Application Developer's Guide*, VNR Communications Library. 1995.
- [WALL 94] Colyer, Wallace. Wong, Walter. *Managing Distibuted Systems*. Carnegie Mellon University Press, 1994.
- [WW01 96] World Wide Web - *Simple Management Protocol (SNMP)* - <http://cio.cisco.com/warp/public/535/3.html>, 1996.
- [WW02 96] World Wide Web - *Chapter 12 - Network Management* - <http://www.hsb.baylor.edu/html/ramsower/classes/ISY4310/olg/chapt-12/chapt-12.html>, 1996.
- [WW03 96] Word Wide Web - *Enterprise Networks:An Introduction to SNMP* - <http://www.anixter.com/snmp1.html>, 1995.
- [WW04 96] World Wide Web - *SNMP@INESC: Introdu o ao SNMP* - <http://cretina.inesc.pt/tfc/snmp/snmp.html>, 1996
- [WW05 95] World Wide Web - *Networking Guide* - <http://scohelp.tomcat.ru:457/NetAdminG/CONTENTS.html>, Santa Cruz Operation Inc.1995

- [WW06 95] World Wide Web – *Network Programmer's Guide and Reference* - <http://scohelp.tomcat.ru:457/cgi-bin/getnav/netguide/CONTENTS.html>, 1996
- [WW07 97] World Wide Web - *FAQ Simple Network Management Protocol* - [ftp://rtfm.mit.edu/pub/usenet/news.answers/snmp-faq/part2](ftp://rtfm.mit.edu/pub/usenet/news.answers.snmp-faq/part2), 1997.
- [WW08 98] FOLDOC - *Free On-line Dictionary of Computing* - <http://wombat.doc.ic.ac.uk/foldoc/index.html>, 1998.
- [WW09 98] Tivoli Systems - *Tivoli and Application Management* - http://www.tivoli.com/o_products/html/body_map_wp.html, 1998.
- [WW10 98] BMC Systems - *Achieving Higher Application Availability, Performance and Recover Across the Enterprise* - http://www.bmc.com/products/asa/asa_wp.htm, 1998.
- [WW11 98] Web based Enterprise Management Initiative - <http://wbem.freerange.com/default.htm>, 1998.
- [WW13 98] DMTF – *Desktop Management Task Force, Inc.* - <http://www.dmtf.org/>, 1998.
- [WW14 98] DMTF – *Driving Industry Standards for Systems Management to Reduce Total Cost of Ownership.* Desktop Management Task Force – 1998.