UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

SILVIO LUIS LEITE

Integrando Ferramentas de Software Livre para Gerenciamento e Monitoração de Redes Locais

Trabalho de Conclusão apresentado como requisito parcial para a obtenção do grau de Mestre em Informática

Prof. Dr. João Cesar Netto
Orientador

Porto Alegre, abril de 2004

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Leite, Silvio Luis

Integrando Ferramentas de Software Livre para Gerenciamento e Monitoração de Redes Locais / Silvio Luis Leite. – Porto Alegre: PPGC da UFRGS, 2004.

109 f.: il.

Trabalho de Conclusão (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2004. Orientador: João Cesar Netto.

1. Gerenciamento de redes. 2. SNMP. 3. Software Livre. I. Netto, João Cesar. II. Título.

“Para Claudia, Felipe e Mariana.”

AGRADECIMENTOS

Agradeço a minha esposa Claudia, meus filhos Felipe e Mariana pelo incentivo, colaboração e paciência; aos meus colegas de curso pela amizade e companherismo durante esta jornada; ao professor Dr. João Netto por sua dedicação, presteza, e sobretudo por sua orientação; aos meus pais e irmão pelo apoio em vários momentos. Obrigado.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BSD	Berkeley Software Distribution
CGI	Common Graphics Interface
CIR	Commited Information Rate
CMIP	Common Management Information Protocol
CMOT	CMIP over TCP/IP
CPU	Central Process Unit
DNS	Domain Name Services
DoD	Department of Defense
EGP	External Gateway Protocol
FDDI	Fiber-distributed Data Interface
FTP	File Transfer Protocol
GNU	Anacronismo recursivo para "GNU is not Unix"
GPL	General Public License
HTML	HyperText Markup Language
IAB	Internet Activities Board
ICMP	Internet Control Message Protocol
IMAP	Internet Mail Access Protocol
IP	Internet Protocol
ISO	International Standards Organization
LAN	Local Area Network

LGPL	Lesser General Public License
MIB	Management Information Base
MPL	Mozila Public License
MTBF	Mean Time Between Failure
MTTR	Mean Time To Repair
NNTP	Network News Transport Protocol
OID	Object ID
OSI	Open System Interconnection
PDC	Primary Domain Controler
POP3	Post Office Protocol v3
RMON	Remote Network Monitoring
SLA	Service Level Agreement
SMI	Struct of Management Information
SMP	Simple Management Protocol
SMS	Simple Message Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TI	Tecnologia da Informação
UDP	User Datagram Protocol
URL	Universal Resource Location
WAN	Wide Area Network

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

Este trabalho tem por objetivo o estudo e a integração, como forma de validação, de ferramentas de software livre para o uso em Gerência e Monitoração de Redes de computadores.

Com o crescimento das redes surgiu a necessidade por controle de seus recursos. Desta necessidade foi criado o protocolo SNMP, utilizado nos dias de hoje como padrão de fato na gerência e monitoração de redes.

Anteriormente ao surgimento do Software Livre, para a atividade de gerência e monitoração, existiam apenas produtos proprietários, os quais estavam restritos a poucas empresas que podiam arcar com seus custos. Com o surgimento do Software Livre, ferramentas simples de gerência e monitoração começaram a ser criadas. Estas ferramentas simples necessitam ser integradas de forma a prover maior quantidade de recursos.

O método proposto desenvolve um protótipo capaz de integrar várias ferramentas de gerência e monitoração utilizando exclusivamente ferramentas de Software Livre.

E para demonstrar a idéia na prática, um Estudo de Caso é apresentado utlizando o protótipo desenvolvido.

Palavras-chave: Gerenciamento de redes, SNMP, Software Livre.

Integrating Tools of Free Software for Management and Monitoring Computer Networks

ABSTRACT

The goal of this work is the integration and study, as a form of validation, of the application of Free Software tools to be used in monitoring and management of computer networks.

As a result of the fast pace growing of computer networks the need to control its resources is required. The Simple Network Management Protocol (SNMP) has been standardized and has become the *de facto* management protocol widely used.

Before the arising of Free Software there were just proprietary products for network monitoring and management and they were restricted only to a few companies that could afford its costs. With the sprouting of Free Software, simple tools of management and monitoring have been created. These simple tools need to be integrated to provide bigger amount of resources.

The prototype method developed is exclusively based on Free Software tools and its ability to integrate diverse tools for network monitoring and management of networks.

A case study is presented, in order to show how this idea works, using the prototype developed.

Keywords: Network Management, SNMP, Free Software.

1 INTRODUÇÃO

Com a popularização das rede e do seu crescimento, os serviços em geral, sejam eles bancários, setor elétrico, de saúde e outros; estão cada vez mais dependentes desta tecnologia.

Este crescimento deve-se a necessidade cada vez maior de comunicação entre as pessoas e empresas, e também da redução dos custos desta tecnologia.

Porém com o aumento do número de redes interligadas, seu gerenciamento está cada vez mais complexo. A necessidade de disponibilidade das redes é tal que podemos comparar com a necessidade que temos por energia elétrica.

O gerenciamento permite controle sob os recursos da rede assim como a identificação e prevenção de problemas. O investimento em gerenciamento justifica-se quando queremos controle dos recursos, de sua complexidade, serviços melhores, e controle de custos. O gerenciamento de redes engloba basicamente cinco áreas: gerenciamento de falhas, de contabilização, de configuração, desempenho e segurança.

Nos dias de hoje cresce a necessidade por ferramentas que permitam total controle destas redes. Estas ferramentas geralmente destinam-se ao gerenciamento de grandes estruturas de redes e por causa de sua complexidade tendem a ser muito caras, destinando-se apenas as grandes corporações.

Com o surgimento do Software Livre, novas ferramentas estão sendo criadas e aprimoradas todos os dias. Uma comunidade criou-se em torno do desenvolvimento de software no qual todos tem o direito de utilizarem seus produtos da melhor forma que lhes convierem.

Com isto o gerenciamento de redes tornou-se disponível a todos que necessitem. Grandes corporações utilizam-se de todas as áreas do gerenciamento, enquanto que para pequenas empresas desempenho e disponibilidade são as áreas mais importantes.

Este trabalho tem por objetivo apresentar um estudo no qual utilizando-se de ferramentas de software livre e integrando-as em um ambiente é possível termos um gerenciamento de desempenho e falhas eficiente e de baixo custo.

O primeiro capítulo apresenta os conceitos em Software Livre e Gerenciamento de Redes, e também o protocolo SNMP utilizado em gerência de redes. O segundo capítulo define o que é necessário para gerenciamento de desempenho e monitoração de falhas. O terceiro capítulo apresenta como o problema foi solucionado e o protótipo

desenvolvido. O quarto capítulo apresenta um estudo de caso no qual seu desempenho e falhas foram monitoradas em um ambiente de produção.

2 PRÉ-REQUISITOS

2.1 O que é Software Livre

Não é fácil definir o termo “Software Livre” ou “Software de Fonte Aberto” em poucas palavras, devido as múltiplas variantes que existem. Mas também não é complicado, já que a idéia em si próprio é simples.

Há um verdadeiro problema quando falamos em inglês o termo Software Livre (“Free Software”) há uma perigosa ambiguidade com relação ao termo FREE que significa tanto “livre” quanto “grátis”, por isso utiliza-se mais comumente o termo Fonte Aberto (ou “Open Source”), que em nossa língua não existe uma segunda interpretação para o termo. Portanto ao se falar em Free Software, devemos pensar em Software Livre e não grátis.

Os principais princípios que definem software livre são:

- Liberdade para os usuarios utilizarem o software como desejarem, para o que desejarem, em tantos computadores quanto quiserem e em qualquer situação tecnicamente apropriada.

- Ter o software a sua disposição para adequarem às suas necessidades. Com certeza, isso inclui melhorá-lo, corrigir os seus erros, aumentar a sua funcionalidade e estudar o seu funcionamento.

- Redistribuir o software a outros usuários, que poderão por sua vez utilizá-lo de acordo com as suas necessidades. Esta redistribuição pode ser gratuita. É importante esclarecer que estamos falando de liberdade e não de obrigações. Isto é, os usuários de um programa podem modificá-lo se acharem necessário.

- Os usuários de uma parte do software devem ter acesso ao código fonte. O código fonte de um programa, geralmente escrito em uma linguagem de programação de alto nível, é absolutamente necessário para poder entender a sua funcionalidade, para melhorar e/ou modificar o programa.

Licenças que regem o Software Livre

Ao se fazer um aplicativo é muito importante a escolha da licença a qual o seu programa irá melhor se adequar. Existem uma infinidade de licenças. Os desenvolvedores podem escolher proteger o seu software com diferentes licenças segundo o grau com que desejar cumprir os seus objetivos e os detalhes que queiram assegurar. As licenças mais comuns são: GPL, LGPL, Artísca, BSD, estilo MPL, etc.

- GPL ("GNU General Public License"): esta é a licença sob a qual é distribuído o software do projeto GNU. No entanto hoje é possível encontrar uma grande quantidade de softwares não relacionados com o projeto GNU distribuído sob a licença GPL, como por exemplo o kernel do Linux. A GPL foi concebida cuidadosamente para promover a produção de mais softwares livres, e por isso proíbe explicitamente algumas ações que possam levar a integração do software GPL em programas proprietários. A GPL está baseada na legislação internacional sobre o Copyright, o que assegura que seja efetiva. As principais características da GPL são: permite a distribuição de arquivos binários, mas apenas se garantir a disponibilização do código fonte; permite a redistribuição do código fonte; permite modificações sem restrições, se o trabalho realizado também se enquadrar na licença GPL; e só é possível a integração completa com softwares cobertos pela GPL.

- LGPL ("GNU Lesser General Public License"): também utilizada pelo projeto GNU, que permite a integração com quase qualquer outra classe de software, incluindo o software proprietário.

- MPL ("Mozilla Public License"): É a licença que a Netscape preparou para distribuir o código do Mozilla, a nova versão do seu navegador em redes. Em muitos aspectos é similar a GPL, porém mais orientada para empresas.

- BSD ("Berkeley Software Distribuition"): A licença BSD cobre, entre outros softwares, as distribuições do BSD. É um bom exemplo de licença permissiva, que não impõe quase nenhuma condição para aquilo que o usuário pode fazer com o software, incluindo cobrar por distribuições binárias, sem a obrigação de incluir o código fonte. Em resumo, os distribuidores podem fazer qualquer coisa com o software, incluindo utilizá-lo para produtos proprietários. Os autores só desejam que seu trabalho seja reconhecido. De alguma maneira esta restrição assegura uma certa "publicidade grátis". É importante observar que esta classe de licença não inclui nenhuma restrição orientada a garantir que os trabalhos derivados continuem a ser software livre.

- Outras licenças bem conhecidas são a licença QT, escrita pela Troll-Tech, os autores da biblioteca QT; a licença Artística, uma das licenças sob as quais é distribuído o Perl; e a licença do Consórcio X.

2.2 Introdução ao Gerenciamento de Rede.

Baseado no livro de William Stallings (SNMP, SNMPv2 and CMIP - The Practical Guide to Network-Management Standards) será apresentado a seguir um resumo sobre gerenciamento de rede utilizando-se o SNMP.

O Gerenciamento e Monitoração de Rede são atividades extremamente importantes para a boa operacionalidade de uma rede. Sem operações de gerenciamento, uma rede local não tem como evoluir. Redes corporativas sem estas funções estão condenadas ao caos.

As funções de Gerenciamento e Monitoração são atividades que visam prevenir e detectar problemas em redes locais e remotas.

As Redes e os Sistemas Distribuídos estão crescendo sua importância e, por consequência, tornando-se críticos para as empresas. Como as redes estão crescendo em escala, dois fatos tornam-se evidentes:

- As redes, com seus recursos associados, e as aplicações distribuídas estão se tornando indispensáveis às organizações.

- Mais coisas podem dar errado desabilitando a rede ou parte dela, ou degradando o desempenho a um nível inaceitável.

Uma grande rede não pode ser gerenciada por uma única pessoa. A complexidade deste sistema necessita de uma ferramenta automatizada de gerenciamento de rede. A urgência por estas ferramentas está aumentando, assim como a dificuldade de suprir tais ferramentas, se a rede incluí equipamentos de diferentes fornecedores.

Em 1992 uma pesquisa mostrou que 15% do total dos custos de sistemas de informação era gasto com gerenciamento de rede, uma média anual de 1,3 milhões de dólares. Para controlar estes custos, ferramentas padronizadas eram necessárias de forma a serem usadas por vários tipos de produtos - incluindo aplicações, roteadores, bridges e equipamentos de telecomunicações - e que pudessem ser utilizados em um ambiente heterogêneo. Em resposta a esta necessidade, dois padrões foram apresentados:

1. Família SNMP: o SNMP (Simple Network Management Protocol) define a um conjunto de padrões para gerenciamento de redes, incluindo um protocolo, uma especificação de estrutura de dados, e um conjunto de objetos. O SNMP foi adotado para ser o padrão para rede TCP/IP. Em 1993, recursos de segurança foram adicionados tornando-se conhecido por SNMPv2.

2. Família OSI: define um conjunto grande e complexo de padrões que definem aplicações de gerenciamento de propósito geral, gerenciamento de serviços e protocolos, uma especificação de estrutura de dados, e um conjunto de objetos.

2.2.1 A importância da gerência e monitoração de redes

Energia elétrica, luzes, telefone, nenhuma organização pode operar sem estes componentes da infra-estrutura. Felizmente, estas utilidades são confiáveis e seguras, quando entramos no escritório pela manhã sabemos, sem dúvida, que haverá luz, que o computador terá energia elétrica, e que o telefone terá tom de discagem.

Em muitas organizações a rede ainda não é confiável. Hoje em dia a rede é tão importante quanto a eletricidade ou o telefone. Confiamos na rede para acessar aplicações críticas para atender aos nossos clientes. Confiamos na rede para nos comunicar-mos. Na Era da Informação, a rede precisa ser estável.

Como em qualquer projeto, a melhor maneira de se começar é definindo as necessidades do usuários. Esta é com certeza a área mais complexa no gerenciamento de rede. Terplan (1992) listou as seguintes linhas mestras que justificam o investimento em gerenciamento de redes:

- Controle estratégico dos recursos: recursos de rede e computação distribuída são extremamente vitais para a maioria das organizações. Sem um controle efetivo, estes recursos não trarão de volta o investimento despendido.
- Controle da complexidade: o crescimento do número de componentes numa rede deve ser controlado assim como os recursos que estes componentes utilizam/necessitam.
- Serviços melhores: os usuários esperam a mesma ou melhor qualidade dos serviços com o aumento dos recursos computacionais.
- Balanceamento de necessidades: os recursos de computação de uma empresa devem ser distribuídos segundo as necessidades de cada grupo de usuários. A gerência de rede deve destinar e controlar os recursos de forma a balancear as várias necessidades.
- Reduzindo o *Downtime*: os recursos de rede de uma organização são cada vez mais importantes, o mínimo de disponibilidade requerido é de 100%. Com isto, a fim de prover um projeto redundante, o gerenciamento de rede torna-se a principal regra para esta Alta-Disponibilidade.
- Controle de Custos: a utilização dos recursos deve ser monitorada e controlada permitindo atingir a satisfação dos usuários com custos razoáveis.

Baseado na especificação funcional do modelo OSI, as funções a seguir definem os pré-requisitos para um sistema de gerenciamento de rede.

Gerenciamento de Falhas

Permite detectar, isolar, e corrigir uma operação anormal no sistema. Para manter a operacionalidade de uma rede complexa, cada componente do sistema deve ser cuidadosamente verificado para ver se sua operação está dentro do esperado. Quando uma falha ocorre, o mais importante a fazer é:

- Determinar exatamente onde a falha ocorreu.
- Isolar o restante da rede de onde ocorreu a falha a fim de evitar possíveis interferências da falha.
- Reconfigurar a rede de forma a minimizar o impacto do componente, ou componentes, que falharam.
- Reparar o componente que falhou e restaurar a rede ao seu estado inicial.

Os usuários de uma rede esperam uma resolução rápida para os problemas da rede. Muitos usuários toleram falhas ocasionais. Quando uma falha ocorre, os usuários geralmente esperam receber uma notificação imediata da solução do problema. Para prover este nível de solução de falhas, são necessárias funções de detecção de falhas e diagnóstico. O impacto e a duração das falhas podem ser minimizados pelo uso de componentes redundantes ou rotas de comunicação alternativas. Os usuários esperam ser informados de manutenções programadas e não programadas na rede.

Após a correção da falha e o retorno total do sistema, o gerenciamento de falhas deve certificar-se de que o problema foi totalmente resolvido e que nenhum outro problema foi introduzido. Como em outras áreas do gerenciamento de rede, a gerência de falhas deve ter um mínimo efeito no desempenho de rede.

Contabilização

Permite contabilizar o uso dos recursos de rede e identificar o custo deste uso. O gerente de rede deve ser capaz de rastrear o uso dos recursos de rede por usuário ou perfil de usuário pelos seguintes motivos:

- Um usuário, ou grupo de usuários, podem estar abusando de seus privilégios e tornando a rede cara para os demais usuários.
- Usuários podem estar tornando a rede ineficiente, e o gerente de rede deve agir para mudar os procedimento melhorando assim o desempenho.
- O administrador de rede é a melhor pessoa para planejar o crescimento da rede se as atividades dos usuários são conhecidas no detalhe.

Relatórios de contabilização devem ser gerados sob o controle do administrador da rede.

Gerenciamento de Configurações

Permite exercer o controle, identificar, coletar dados, e prover informações sobre os objetos da rede com o propósito de prover a operação contínua dos serviços. As redes modernas de hoje em dia são compostas por componentes individuais e subsistemas lógicos (por exemplo, o *driver* de um determinado hardware) que devem ser configurados para executar diferentes aplicações. O mesmo dispositivo pode ser configurado tanto como roteador, como servidor; ou ambos. Antes de decidir como

o dispositivo vai ser utilizado, sua configuração deve ser apropriadamente escolhida. A gerência de configuração é responsável também por inicializar e desativar partes ou toda a rede. Isto também significa manter, adicionar e atualizar as relações e estado dos componentes durante a operação da rede.

O gerente de rede deve ser capaz de mudar a configuração da rede quando as necessidades dos usuários mudam. A reconfiguração da rede é uma opção em resposta a uma avaliação de desempenho, *upgrade* na rede, recuperação de falhas, ou verificação de segurança.

Gerenciamento de Desempenho

Permite avaliar o comportamento dos objetos e a eficiência das atividades de comunicação. As redes são compostas por vários e variados componentes que se intercomunicam e compartilham dados e recursos. Em alguns casos, isto é crítico para a eficiência da aplicação que utiliza a rede. A gerência de desempenho se divide em duas categorias: monitoração e controle. Monitoração é a atividade que verifica as atividades na rede. A função de Controle permite a Gerencia de Desempenho fazer ajustes a fim de melhorar o desempenho da rede. Alguns tópicos relativos ao desempenho:

- Qual é o nível de utilização da rede ?
- Existe tráfego excessivo ?
- A vazão foi reduzida a níveis inaceitáveis ?
- Existe algum gargalo ?
- O tempo de resposta está aumentando ?

Para responder a estes questionamentos, a Administração de Rede deve focalizar um conjunto inicial de componentes a serem monitorados verificando-se os níveis de desempenho. Isto inclui associar métricas e valores aos recursos monitorados para indicar os diferentes níveis de desempenho. Por exemplo, qual a quantidade de retransmissões é considerada um problema de desempenho ?

Coletando-se estas informações, analisando-as, e usando o resultado da análise como parâmetro de valores, o administrador de rede torna-se cada vez mais apto a reconhecer situações que indicam a presença de degradação de desempenho. As estatísticas de desempenho podem ser usadas para reconhecerem potenciais gargalos que poderiam causar problemas aos usuários da rede.

Gerenciamento de segurança

Permite mapear os aspectos essenciais do modelo de gerenciamento OSI a fim de operar a rede corretamente e proteger os objetos gerenciados. A Gerencia de Segurança é responsável por criar, distribuir, e armazenar chaves criptografadas. Senhas e outras autorizações ou controles de acesso devem ser mantidos e distribuídos. A Gerencia de Segurança também está relacionada à monitoração e controle de acesso

aos computadores da rede. Os logs são uma importante ferramenta de segurança, a Gerencia de Segurança está muito mais envolvida em coletar, armazenar e auditar os registros e logs de segurança, do que habilitar ou desabilitar os logs.

2.2.2 Gerenciamento de Desempenho

Indicadores de Desempenho

Um pré-requisito absoluto para o gerenciamento de rede é a habilidade de mensurar o desempenho da rede. Não conseguiremos gerenciar e controlar um sistema ou uma atividade a menos que consigamos monitorar seu desempenho. Uma das dificuldades no gerenciamento de rede inclui selecionar e usar apropriadamente indicadores que consigam medir o desempenho da rede.

William Stallings cita alguns problemas:

- Existem muitos indicadores para usarmos.
- O significado da maioria dos indicadores ainda não é compreendido.
- Alguns indicadores são suportados por apenas alguns fabricantes.
- Frequentemente o indicadores estão corretos porém são interpretados de forma errônea.
- Em muitos casos, o cálculo dos indicadores leva muito tempo e o resultado final torna-se difícil de ser usado para controlar o ambiente.

Existem duas categorias para os indicadores de desempenho: medição orientada a serviço e medição orientada a eficiência. A tabela 2.1, baseada em estudo de Terplan (1992), nos dá um resumo dos maiores indicadores em cada categoria. O principal significado de julgar se a rede está dentro dos parâmetros requeridos é verificar se os níveis de serviço (SLA[1]) estão sendo mantidos para satisfação dos usuários.

Disponibilidade

Disponibilidade pode ser expressa como sendo o percentual de tempo que um sistema de rede, componente, ou aplicação está disponível para o usuário. Dependendo da aplicação, o valor financeiro da alta-disponibilidade pode ser significante. Por exemplo, no sistema de reserva de passagens de uma companhia aérea, uma hora sem sistema pode causar um prejuízo em torno de US$10.000,00.

A disponibilidade é baseada na confiabilidade de componentes individuais de uma rede. Confiabilidade é a probabilidade que um componente irá apresentar numa função específica por um determinado tempo em condições específicas. A falha de um componente é usualmente especificada pelo seu MTBF (*mean time between failure* - tempo-médio-entre-falhas). A disponibilidade pode ser formulada como:

$Disponibilidade = \frac{MTBF}{MTBF+MTTR}$

onde MTTR é o tempo-médio-para-reparo (*mean time to repair*).

[1]SLA: Service Level Agreement, Acordo de Nível de Serviço.

Tabela 2.1: Indicadores de Desempenho de Rede.

Medição Orientada a Serviço	
Disponibilidade	É o percentual de tempo que um sistema de rede, um componente, ou uma aplicação estão disponíveis para uma usuário.
Tempo de Resposta	Quanto tempo leva para uma resposta aparecer no terminal do usuário após este ter feito uma solicitação.
Precisão	O percentual de tempo em que nenhum erro de transmissão e recepção de informações ocorra.
Medição Orientada a Eficiência	
Vazão	A taxa na qual os eventos orientados a aplicação (transações, mensagens, transferência de arquivos) ocorrem.
Utilização	O percentual da capacidade teórica do recurso que está sendo utilizada.

A disponibilidade de um sistema depende da disponibilidade de componentes individuais mais a organização do sistema. Por exemplo, alguns componentes podem ser redundantes, assim a falha de um destes componentes não afeta o sistema. Ou a configuração de certo componente pode ser perdida resultando em redução da capacidade, porém o sistema continua funcionando.

Tempo de Resposta

Tempo de Resposta é o tempo que um sistema leva para reagir a uma entrada. Em uma transação interativa, pode ser definido como sendo o tempo entre a última tecla pressionada pelo usuário e início da apresentação da resposta ao usuário. Para aplicações diferentes existem significados distintos, mas geralmente é o tempo que o sistema leva para responder a uma requisição. Nós preferiríamos que os tempos de respostas fossem curtos, porém tempos de resposta curtos invariavelmente nos levam a custos altos.

Baseado em Martin (1988), a tabela 2.2 lista seis faixas de tempos de resposta.

O tempo de resposta é a chave para produtividade em aplicações interativas, confirmadas em vários estudos realizados (Shneiderman 1984; Thadhani 1981; Guynes 1988). Estes estudos mostraram que quando um computador e uma pessoa interagem fazendo com que nenhum dos dois espere pelo outro, a produtividade aumenta significativamente, o custo do trabalho realizado no computador diminui, e a qualidade tende a aumentar. Tempos de resposta até dois segundos são aceitáveis para a maioria das aplicações interativas porque uma pessoa pode pensar na próxima ação (Miller 1968).

Os resultados obtidos no tempo de resposta são baseados em análise de transações *on-line*. Uma transação consiste em um comando de usuário a partir de um terminal

Tabela 2.2: Faixas de Tempo de Resposta

Maior que 15 segundos	Uma demora maior que 15 segundos quebra as regras para uma interação conversassional. Para alguns tipos de aplicações, certos tipos de pessoas podem não se incomodar em ficar sentados num terminal por mais de 15 segundos esperando uma resposta do sistema. Porém, para pessoas ocupadas, esperar mais de 15 segundos é intolerável. Se este tipo de demora ocorrer, o sistema deve ser desenhado de forma a permitir ao usuário realizar outras atividades enquanto espera a resposta.
Maior que 4 segundos	Demoras maiores que 4 segundos são geralmente longas demais para que uma conversação requeira que o operador mantenha a informação na memória imediata (memória da pessoal). Esta demora não serve para aplicações onde é necessário entrada de dados, porém para certas aplicações, pode ser tolerado.
De 2 a 4 segundos	Uma demora de maior que 2 segundos pode inibir o operador quando for necessário operações que requeiram alto nível de concentração.
Menor que 2 segundos	Quando o usuário necessita lembrar-se de informações através de várias respostas, este tempo de resposta deve ser curto. Para atividades elaboradas, 2 segundos representam um importante tempo de resposta limite.
Mili-segundos	Certos tipos de trabalhos intensivos, especialmente com aplicações gráficas, requerem tempos de resposta muito curtos para que mantenham a atenção e o interesse por um longo período de tempo.
Micro-segundos	A resposta ao pressionamento de uma tecla e o aparecimento do caracter na tela do terminal, ou o clique em um objeto com o *mouse* precisa ser quase instantâneo - menos que 0.001 segundo após a ação.

e a resposta do sistema, é a unidade fundamental para trabalhos em sistemas *on-line*. Pode ser dividida em suas sequências:

1. Tempo de resposta do usuário: é o tempo entre o usuário receber uma resposta completa do sistema e acionar o próximo comando. Também chamado de "tempo de pensamento".

2. Tempo de resposta do sistema: é o tempo entre o momento em que o usuário aciona o comando e que a resposta completa é apresentada.

Para medir o tempo de resposta, um número de elementos precisam ser observados. Embora seja possível medir diretamente o tempo de resposta em um ambiente de rede, a figura 2.1 é usada para corrigir problemas ou planejar o crescimento da rede. Para isto, um detalhamento do tempo de resposta é necessário para identificar gargalos ou possíveis gargalos.

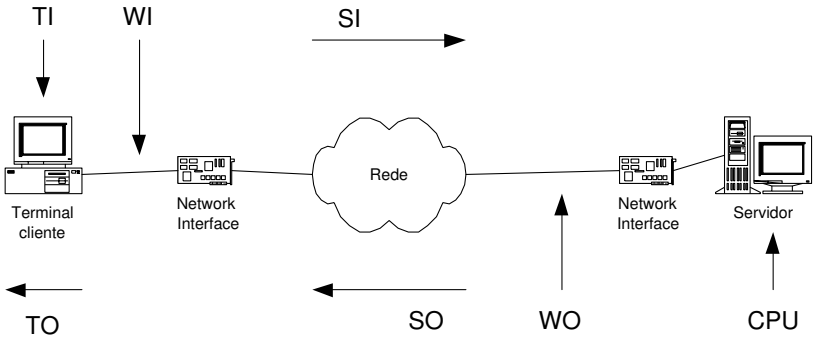

Figura 2.1: Elementos do Tempo de Resposta.

A figura 2.1 ilustra uma típica rede e indica os sete pontos de tempo de resposta comum para a maioria das aplicações interativas. Cada um destes elementos é um passo no caminho completo da comunicação e cada elemento contribui uma parte no cálculo total do tempo de resposta:

$RT = TI + WI + SI + CPU + WO + SO + TO$

1. *Inbound Terminal Delay (TI):* a demora entre pegar a entrada no terminal e colocar a informação na linha de comunicação.

2. *Inbound Queuing Line (WI):* é o tempo requerido pela controladora de comunicações.

3. *Inbound Service Time (SI):* é o tempo para transmitir na linha de comunicação, rede ou outro meio de comunicação, da controladora local até a controladora remota.

4. *Processing Time (CPU):* é o tempo de processamento, é o tempo do processador, discos, de receber a solicitação e preparar o resultado para ser enviado ao solicitante.

5. *Outbound Queuing Time (WO):* é o tempo para transmitir as informações do processador para a controladora de comunicações.

6. *Outbound Service Time (SO):* é o tempo para transmitir as informações da controladora remota à controladora local.

7. *Outbound Terminal Delay (TO):* é o tempo do terminal propriamente dito.

O tempo de resposta (RT) é relativamente simples de medir e é uma das mais importantes classes de informações para o gerente de rede.

Confiabilidade

A confiabilidade na transmissão de dados entre usuário
e máquina, ou entre máquinas, é essencial para qualquer rede. Por causa dos mecanismos de correção de erros embutidos nos protocolos de comunicação a precisão não é geralmente do conhecimento do usuário. Contudo ela é útil para monitorar a taxa de erros que podem estar ocorrendo. Ela pode nos dar uma indicação de falha intermitente ou a existência de ruído ou interferência que devem ser corrigidos.

Vazão (Throughput)

Vazão é uma medida orientada a aplicação. Dentre os exemplos podemos citar:

- Número de transações de um determinado tipo em um determinado período.
- Número de sessões de clientes a uma aplicação em um determinado tempo.
- Número de chaveamentos em um circuito do tipo comutado (*switch*).

Este é o tipo de rastreamento contínuo útil para avaliar uma demanda projetada e possíveis problemas de desempenho.

Utilização

A utilização é uma medida mais refinada que o *throughput.* Ela é utilizada para determinar o percentual em que um recurso é utilizado em um determinado período de tempo.

Talvez o mais importante na medida de utilização seja a procura por possíveis gargalos e áreas de congestionamento. A razão mais importante é que o tempo de resposta aumenta exponencialmente quando a utilização aumenta.

Procurando-se por um comportamento onde desejamos saber quais recursos estão alocados e quais não estão em um determinado período, seremos capazes de descobrir quais recursos estão sub-utilizados.

Baseado em Johnson 1985, existe uma técnica simples porém eficiente para medir a eficiência de uma rede. Esta técnica é útil por acessar a capacidade de vários links de

comunicação na rede. A idéia básica é observar as diferenças entre a carga planejada e a utilizada nos vários links de comunicação. A carga planejada é a capacidade da linha de comunicação, em bits por segundo, de cada link individual. A carga atual é a média de tráfego em cada link. Uma analogia pode ser realizada com a técnica do *cost-accounting* que verifica as proporções das despesas atuais com as planejada por departamentos dentro da empresa.

Considere, por exemplo, uma simples configuração de rede como a mostrada na figura 2.2. Ela expressa a carga em cada link assim como o percentual da carga total na rede e o fluxo em cada link, e o percentual do fluxo total. A tabela 2.3 mostra os resultados. Como podemos ver, a capacidade total da rede provém um margem confortável sobre a carga total na rede, e é claro, nenhum link está com a carga acima de sua capacidade. Entretanto, olhando-se a capacidade relativa e a carga relativa, nós podemos ver que alguns links estão carregando menos que a carga compartilhada proporcional e alguns estão carregando mais que a carga compartilhada. Isto indica uma ineficiência na alocação dos recursos. Ajustando-se estas proporções, tanto redirecionando o tráfego quanto mudando a capacidade de vários links, um balanceamento dos recursos mais preciso pode ser alcançado. Isto pode resultar numa redução na capacidade total requerida e num uso dos recursos mais eficiente.

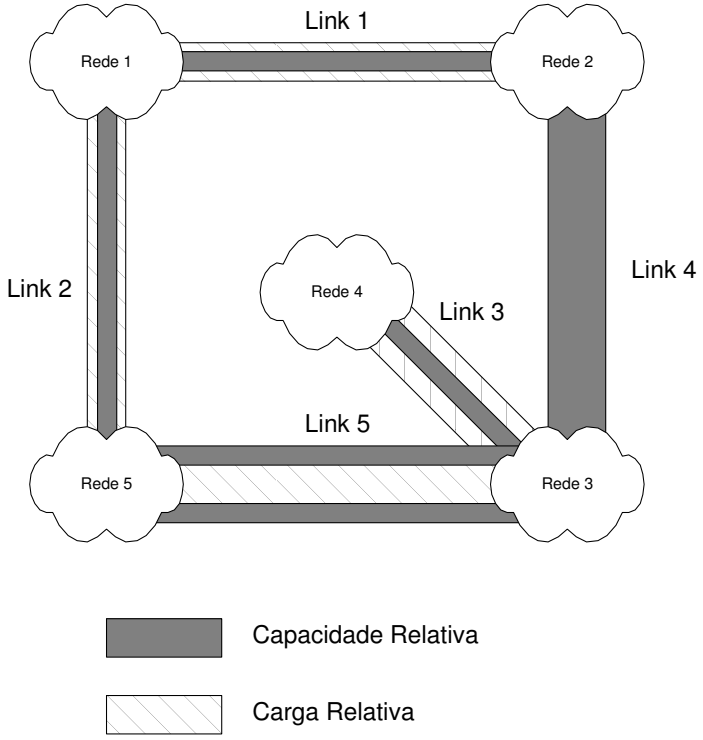

Figura 2.2: Exemplo simples de Análise de Eficiência.

Função de Monitoração de Desempenho

A monitoração de desempenho abrange três partes: medição de desempenho, que é a

Tabela 2.3: Fator de utilização para a Rede da fig. 2.2

	Link 1	**Link 2**	**Link 3**	**Link 4**	**Link 5**	**Total**
Carga (bps)	3.000	3.000	5.000	4.000	5.000	20.000
Capacidade (bps)	4.000	4.000	6.000	8.000	18.000	40.000
Perc. Total da Carga	15%	15%	25%	20%	25%	100%
Perc. do Total da Capac.	10%	10%	15%	20%	45%	100%
Proporção	1.50	1.50	1.67	1.00	0.55	

coleta de estatística sobre tráfego de rede e tempos de resposta; análise de desempenho, que consiste na apresentação gráfica da coleta de estatísticas; e, geração de tráfego sintético, que permite observar a rede sobre condições controladas.

A medida de performance é freqüentemente disponibilizada pelos agentes embutidos nos dispositivos de rede, como servidores, roteadores, hubs, etc. Estes agentes tem a função de observar o montante de tráfego que entra ou sai da interface, o número de conexões de rede, o tráfego por conexão, e outras medidas que provém uma imagem detalhada do dispositivo.

Em uma rede local, muitas das informações que necessitamos podem ser coletadas por um monitor externo simplesmente observando o tráfego na rede. A tabela 2.4 apresenta vários tipos de medidas que são possíveis coletar por um monitor de rede e nos dá uma idéia do tipo de medida que nos interessa.

Tabela 2.4: Relatórios de Medição de Desempenho

Nome	**Variável**	**Descrição**
Matriz de comunicação de servidor	Origem x Destino	Número ou % de pacotes, dados ou bytes.
Matriz de comunicação de grupo	Origem x Destino	Idem acima, porém consolidado para grupo de endereços.
Histograma de tipo de pacote	Tipo de pacote	Número ou % de pacotes por tipo.
Histograma de tamanho de pacote	Tamanho do pacote	Número ou % de pacotes por tamanho.
Vazão/Utilização	Origem	Total de pacotes/dados transmitidos.

- O tráfego está distribuído entre os usuários da rede, ou existe alguma comunicação origem x destino com tráfego mais pesado?

- Qual é o percentual de cada tipo de pacote? Pode algum tipo de pacote estar ocorrendo de forma não usual frequentemente, podendo indicar algum erro ou ineficiência do protocolo?

- Qual a distribuição dos tamanhos de pacotes?
- Qual é a utilização e vazão de um link?

Questões referentes ao aumento da carga e variação no tamanho dos pacotes incluem:

- Qual o efeito do aumento da carga de rede na utilização, vazão, e tempos de retardo (*delay*)? Quando este aumento de tráfego começa a degradar o sistema?
- Qual é a capacidade máxima de um canal de comunicação sobre condições normais de operação? Quantos usuários são necessários para alcançar o máximo do canal?
- Pacotes grandes aumentam ou diminuem a vazão e tempo de retardo?
- Como afetam a utilização e o tempo de retardo o uso de pacotes de tamanho constante?

Medição Estatística vs Exaustiva

Quando um agente em um nodo da rede, ou um monitor externo está monitorando um tráfego pesado, pode não ser muito prático coletar dados de forma exaustiva. Por exemplo, para que o monitor externo monte uma matriz que mostre o total de pacotes em um período de tempo, entre cada par de conexão, o monitor deverá capturar cada pacote transmitido na rede e obter as informações de origem x destino de cada pacote. Quando a rede estiver muito carregada, este monitor simplesmente não conseguirá mostrar a matriz.

A alternativa seria tratar cada parâmetro como uma variável aleatória e amostrar o tráfego de forma a estimar o valor destas variáveis aleatórias. Entretanto muito cuidado deve ser tomado na hora de empregar e interpretar este resultado estatístico. A pessoa responsável por projetar as funções de coleta e por interpretar os resultados obtidos, deve ter familiaridade com os princípios estatísticos.

***Baseline*: Definindo o perfil da Rede**

Uma *baseline* é um valor ou perfil da métrica de desempenho com o qual se pode comparar com os valores atuais da métrica de desempenho. Por exemplo, se conheçemos a média semanal de utilização de um *link*, esta média pode ser utilizada como *baseline* para comparações de mudanças futuras na utilização deste *link*.

Baselining é um termo amplamente utilizado para vários métodos de análise que comparam as mudanças nos dados atuais com sua *baseline*. O uso mais comum de *baselining* é como uma ferramenta de gerenciamento de desempenho para análise de tendências, comparando a métrica com o valor histórico para encontrar a tendência que pode ser usada para estimar desempenhos futuros ou necessidades.

Os termos Gerenciamento de Desempenho e *Baselining* muitas vezes tendem a significar a mesma coisa. Mas existe uma diferença. Gerenciamento de desempenho utiliza técnicas de *baselining* para análise de tendências, para desenvolver uma visão

futura das necessidades a partir das tendências passadas. Para efetivamente planejar os requisitos de capacidade de rede, é necessário levar em consideração futuras mudanças nas aplicações que podem causar mudanças na carga da rede.

Segundo W. Stallings, existem sete passos que devem ser trabalhados para desenvolver uma estratégia de Baseline:

1. Serviços Chaves: Quais são os serviços chaves que são providos para os clientes da redei, internos e externos.

2. Componentes do Serviço: Quais são os elementos discretos que são agregados para prover os serviços chaves.

3. Métricas: Quais são os serviços chave ou componentes do serviço necessários para medição e quais medidas são necessárias.

4. *Baseline*: Quais *baselines* estão associadas com cada métrica. São usados valores simples ou perfis nas *baselines*?

5. Necessidade de Análise e Instrumentação de dados: Quais dados serão coletados para medir o desempenho e que instrumentos serão necessários para que a rede nos dê estes dados.

6. Relatórios e Alarmes: Que ferramentas são necessárias para coletar, sumarizar, e interpretar os dados obtidos e comparar com as *baselines*.

7. Controle: Quem é responsável por identificar e providenciar as mudanças de capacidades requeridas.

Serviços chaves

O primeiro passo no processo é identificar o que queremos gerenciar. A resposta não é tão simples quanto "A Rede". Na realidade, a resposta pode variar dramaticamente de organização para organização. Por exemplo, um grupo responsável pela operação do *backbone* WAN terá requisitos diferentes de baseline daqueles que são responsáveis pela rede local (LAN).

Para entender o que são os Serviços Chave devemos perguntar-nos que serviços estão sendo providos para os clientes internos e externos, visto pela ótica dos clientes.

Componentes do Serviço

Quanto mais aprofundado for o estudo dos serviços chaves, mais fácil será determinar os elementos discretos que fazem parte do serviço. Por exemplo, no caso de um *backbone* WAN, os elementos discretos podem ser:

- Branch Routers
- Backbone Routers
- Serviços públicos de Frame-Relay e ATM

- Switchs Frame-Relay e ATM
- Linhas privadas
- Redes Virtuais Privadas

Já os administradores de redes locais (LAN) irão possuir elementos discretos diferentes, tais como:

- LAN Backbone e Switchs
- Roteadores locais
- Concentradores (HUBs e Switchs)
- Backcone Interswitch (ATM, Fast/Gigabit Ethernet, FDDI)
- Links de Servidores (ATM, Fast/Gigabit Ethernet, FDDI, Ethernet, Token Ring)
- Links de Estações de Trabalho (Ethernet, Fast Ethernet, Token Ring)
- Redes Virtuais (VLANs, ELANs, PVCs)

Para o propósito da gerência de desempenho, é necessário identificar quais componentes de serviço afetam o desempenho geral. Isto significa analizar um diagrama de rede para determinar quais são os componentes críticos. Lógico que isto varia de rede para rede, mas geralmente seguem as regras gerais:

- Para linhas privadas, os elementos críticos são os links do backcone, concentradores de links, e centrais de links.
- Para Frame-Relay, redes públicas ATM e VPNs os elementos críticos são aqueles entre os centrais de dados e a núvem de rede.
- Para redes locais, os elementos críticos são tanto as conexões entre hubs e switchs quanto as conexões com os servidores.

Métricas

Após ter determinado os serviços chave e os componentes críticos, devemos determinar o que medir. Diferentes medidas são usadas para diferentes propósitos, então é conveniente agrupar as medidas em categorias. As cinco categorias a serem consideradas são:

1. **Serviços proporcionados**. É a medida do acordo de nível de serviço (SLA) visto da perspectiva do cliente em termos de desempenho e disponibilidade. Métricas apropriadas incluem tempo de resposta round-trip, percentual de disponibilidade no mês ou ano, e o tempo médio para reparo.

2. **Serviços utilizados**. Para prover o nível de serviço apropriado, nós confiamos o serviço a provedores de serviço. A métrica será bem similar a utilizada nos Serviços Proporcionados mas mais específica a tecnologia de transporte.

3. **Planejamento de capacidade**. A meta principal no gerenciamento de desempenho é predizer as necessidades futuras a partir das tendências atuais de capacidade.

4. **Utilização da Rede**. Uma extensão do planejamento de capacidade é a analise de como o uso da rede está mudando. Algumas tendências podem ser analisadas ao nível do protocolo de rede. Por exemplo, gostaríamos de rastrear o tráfego HTML no backcone para saber em quais escritórios regionais seria interessante instalar servidores de *Proxy*.

5. **Saúde da rede**. Administradores de rede argumentam se a saúde da rede estaria relacionada a gerenciamento de desempenho ou gerenciamento de falhas, mas apesar de tudo, a aplicação de técnicas de tendências e baselining vão além de apenas medir velocidade e disponibilidade. Com a medição de indicadores de problemas específicos é possível detectar a ocorrência de problemas antes mesmo do problema ocorrer. A métrica mais comum para a saúde da rede é a taxa de erros associados a um link ou segmento de rede local.

Estabelecendo as Baselines

Tradicionalmente as baselines tem sido apenas um simples valor, geralmente representados como uma linha horizontal no gráfico da métrica pelo tempo. Este valor quase sempre deriva da coleta de dados em período específico e calcula a média como uma representação do estado atual, por exemplo, a média de utilização em uma semana.

Entretanto, este cálculo da média pode ser um pouco errôneo. Uma rede que está muito perto da ociosidade pode experimentar picos de utilização repentinos. Exemplos destes picos através do dia incluem picos de utilização de correio-eletrônico nos períodos da manhã e após o horário de almoço.

Porém utilizando-se uma baseline baseada em perfil ao invés da média, podemos verificar não apenas as grandes mudanças que ocorrem na rede, mas sim as pequenas mudanças no comportamento normal da rede. Estas pequenas mudanças podem indicar novos usos da rede ou algum problema pendente.

Análise e instrumentação dos dados

Se vamos produzir relatórios de tendências a partir de baselines, precisamos saber que tipo de informação de gerenciamento está disponível nos equipamentos de rede. Muitas das métricas requerem que os dados sejam obtidos apartir de diferentes partes da rede. O modo mais usual para obtenção de dados de dispositivos de rede é o SNMP.

Alarmes e Relatórios

Para os propósitos da gerência de desempenho, deveremos estar aptos a produzir

relatórios que descrevam como a métrica está se comportando referente a sua baseline. Para os serviços mais críticos devemos observar os relatórios diariamente ou semanalmente. Porém geralmente não temos tempo para observar todos os relatórios disponíveis, então precisamos de um mecanismo que nos informe quando uma métrica mudou de forma significante. Esta é a função dos *Thresholds* e Alarmes. Threshold é uma baseline atribuída a uma métrica. Quando um *threshold* é alcançado, seremos notificados via um alarme, e-mail, pager ou outro indicador de tela.

Existem duas classes de relatórios que nos interessam, A primeira classe de ferramentas é usada para coletar dados e reportar dados dos agentes SNMP. A segunda classe são os sistemas de *polling*. A primeira classe de ferramentas interroga a MIB a partir de uma larga variedade de agentes SNMP e provém uma grande variedade de relatórios os dados coletados pelo tempo. A segunda classe de ferramentas é um pouco diferente. Uma destas ferramentas consiste em uma aplicação que fica medindo o tempo de resposta de um determinado equipamento enviando-lhe pacotes ICMP (ping echo request). Outras aplicações vão mais a fundo na camada de aplicação e testam os tempos de resposta de uma determinada aplicação, por exemplo os servidor de correio-eletrônico (SMTP e POP3), servidores páginas WEB, servidores de FTP, etc.

Segundo Phillip Carden alguns valores podem ser utilizados para configurar os limites para valores de utilização média e pico:

- Linhas privadas: utilização média em 45% da velocidade da linha, pico em 70%, medição no período de 1 dia.

- Frame Relay: utilização média em 55% do CIR[2], pico em 80%, medição no período de 1 dia.

- Redes Locais: para redes compartilhadas (segmentos utilizando HUBs) utilização média em 15%, pico em 25%; e para redes comutadas (segmentos utilizando switchs - VLANs) utilização média em 25%, pico em 40%; medição num período de 15 minutos.

- Outras tecnologias de rede: Não existem problemas para Token-Ring e ATM. Utilização média em 50%, pico em 70%, medição no período de 15 minutos.

Controle

Não existe valor algum a menos que entendamos como utilizar as informações colhidas destas ferramentas. Para cada serviço chave e componente de serviço medido, deve-se definir quem é responsável por gerar e analizar os relatórios, quem receberá os alarmes, quem ajustará os thresholds e qual o mecanismo pelo qual os thresholds excedidos conduzirão a mudanças na capacidade ou topologia da rede.

[2]CIR: Commited Information Rate, é o indice que indica a taxa de entrega garantida em links Frame Relay.

2.2.3 Monitoração de Falhas (Detecção)

O objetivo da monitoração de falhas é identificar falhas tão rápido quanto possível após elas terem acontecido, identificar suas causas assim como as ações que serão tomadas.

Problemas da Monitoração de Falhas

Em um ambiente complexo, localizar e diagnosticar falhas pode ser difícil. A lista a seguir mostra os problemas que estão associados a observação e ao isolamento da falha. A observação da falha pode ter os seguintes complicadores:

- Falhas não observáveis: Certos tipos de falhas não são possíveis de serem observadas através da observação local. Por exemplo, a existência de *deadlock* entre processos distribuídos cooperativos não podem ser observados localmente. Outras falhas não podem ser observadas porque o fornecedor do equipamento não preparou-se para registrar a ocorrência de uma falha.

- Falhas parcialmente observáveis: Uma falha num nodo da rede pode ser observável, porém a observação pode ser insuficiente para identificar o problema. Por exemplo, um nodo da rede pode estar "travado" devido a falha em algum dispositivo periférico.

- Incerteza na observação: Sempre que for possível uma observação detalhada da falha, podem existir incertezas e inconsistências associadas a esta observação. Por exemplo, a falta de resposta de um dispositivo remoto pode significar que o dispositivo está travado, a rede pode estar interrompida, pode estar havendo congestionamentos que aumentam o tempo de resposta, ou falha no temporizador local.

Após a falha ser observada, será necessário isolar a falha de cada componente envolvido. Alguns problemas podem surgir, como os citados abaixo:

- Múltiplas causas potenciais: quando múltiplas tecnologias estão envolvidas, o potencial de pontos para falhas e de tipos de falhas aumenta consideravelmente. Isto torna difícil localizar a origem da falha. Por exemplo, se uma falha de transmissão ocorre em um ambiente fisicamente distribuído, a falha pode ter ocorrido em qualquer equipamento pelo qual esta transmissão passa.

- Muitas observações relatadas: um simples defeito pode afetar vários caminhos de comunicação. Por exemplo, a falha no link E1 de uma empresa pode ocasionar parada nos serviços de voz e dados desta empresa. Além disso, a falha em uma das camadas da arquitetura de comunicações pode causar degradações ou falhas em todas as camadas superiores, como mostra a figura 2.3. Assim, uma falha em uma linha E1 pode ser detectada nos roteadores como uma falha no link de comunicações e nas estações de trabalho como uma falha no transporte e aplicação.

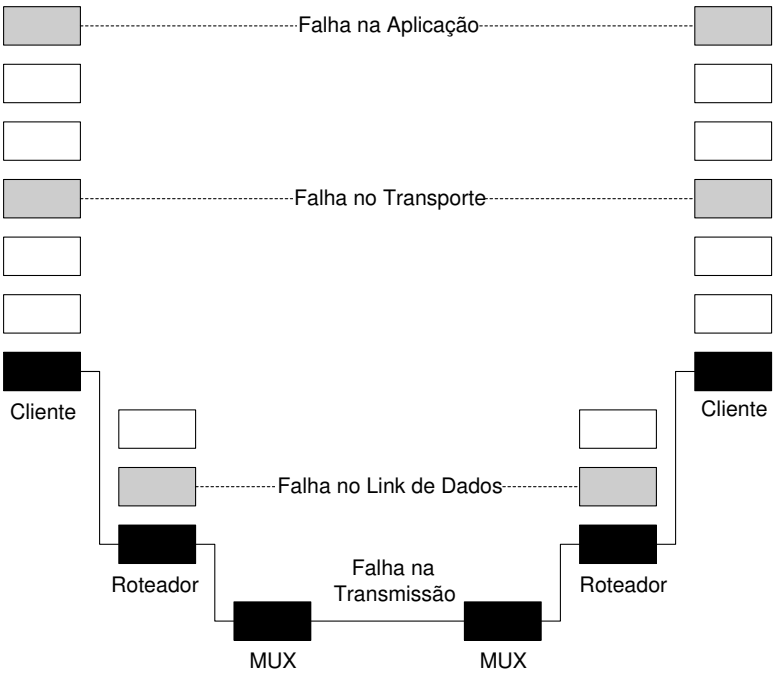

Figura 2.3: Propagação de Falhas nas Camadas Altas.

- Interferência entre diagnóstico e procedimentos de recuperação local: procedimentos de recuperação local podem destruir importantes evidências a respeito da natureza da falha, acabando com o diagnóstico.
- Ausência de ferramentas de testes automáticos: os testes para isolar falhas são difíceis e dispendiosos para os administradores.

Funções da monitoração de falhas

A função de sistema de monitoração de falhas é que ele seja capaz de detectar e reportar falhas. No mínimo o agente de monitoração de falhas deve manter um registro do eventos e erros mais significantes. Assim, um sistema que opera basicamente com *pollings* deve reportar estes registros de seus eventos.

Além disto, um bom sistema de monitoração de falhas deve ser capaz de antecipar as ocorrência de falhas. Por exemplo, se na fração de uma transmissão de pacotes pela rede acontecer um erro que excede a determindado valor, isto indica que um problema está se desenvolvendo através do caminho de comunicação. Se o limite (*threshold*) for configurado a um valor baixo o suficiente, o administrador da rede será alertado a tempo de tomar a medidas necessárias para evitar uma falha maior no sistema.

O sistema de monitoração de falhas deve também ajudar no isolamento e diagnóstico de falhas. Exemplos de testes que o sistema de monitoração de falhas deve ter:

- Teste de conectividade.

- Teste de integridade dos dados.
- Teste de integridade dos protocolos.
- Teste de saturação de dados.
- Teste de saturação de conexões.
- Teste de tempo de resposta.
- Teste de funções.
- Teste de diagnósticos.

Talvéz mais que em outras áreas da monitoração de redes, uma interface para usuário é necessária para a monitoração de falhas. Em situações complexas, falhas devem ser isoladas, diagnosticadas, e corrigidas através da cooperação do usuário e do software de monitoração.

2.3 Gerenciamento SNMP

O termo Protocolo de Gerenciamento de Rede é atualmente usado para se referir a uma coleção de especificações para gerenciamento de rede que incluem o protocolo em sí, a definição de uma base de dados, e conceitos associados.

2.3.1 Protocolo SNMP

As Origens do TCP/IP

O TCP/IP surgiu por volta do ano de 1969 quando o Departamento de Defesa (DoD) dos E.U.A. iniciou, através do Advanced Research Projects Agency (ARPA), o desenvolvimento da primeira rede comutada de pacotes, a ARPANET. O propósito da ARPANET foi estudar tecnologias relacionadas com o compartilhamento de recursos computacionais e tornar esta tecnologia utilizável no dia-a-dia do DoD. Com o crescimento da ARPANET, ela logo estava conectando centenas de computadores e milhares de terminais. O problema começou a aparecer quando a ARPANET evoluiu para uma Internet: um conjunto de redes espalhadas interligadas pela ARPANET.

Para resolver o problema de interoperabilidade, os pesquisadores da ARPANET desenvolveram, pelos idos de 1970, um conjunto padronizado de protocolos,
que evoluiu para o que hoje conhecemos como TCP/IP.

As Origens do Gerenciamento de Redes TCP/IP

Quando o TCP/IP foi criado, pouco se pensava em gerenciamento de Redes. No final dos anos 70, não existiam protocolos de gerenciamento como conhecemos hoje. A única ferramenta efetivamente usada para controle era o *Internet Control Message Protocol (ICMP)*. O ICMP era forma de transmitir mensagens de controle de roteadores

ou servidores a outros servidores para obterem um retorno sobre os problemas no ambiente. O ICMP está disponível em todos os dispositivos que suportam o protocolo IP. Do ponto de vista do gerente de rede, o recurso mais importante do ICMP é o par de mensagens *echo request/echo reply*. Estas mensagens provém um mecanismo para testar se é possível comunicação entre as entidades. Outro par de mensagens muito útil é *timestamp/timestamp reply*, que provêm um mecanismo para amostar os atrasos característicos de uma rede.

O número de máquinas conectadas na Internet tem crescido exponencialmente. Este crescimento vem acompanhado do aumento da complexidade. Com o número de máquinas na rede em torno de centenas de milhares e o número de redes individuais aos milhares, não seria possível deixar o gerenciamento desta rede a cargo de um pequeno grupo de especialistas. Foi necessário criar um protocolo padrão que fosse além da funcionalidade de um PING e que fosse de fácil uso e aprendizagem para um grande número de pessoas com responsabilidades de gerenciamento de rede.

Para alcançar estes requisitos, vários esforços foram iniciados para desenvolver um protocolo de gerenciamento de rede. Dentro destes esforços, os três que mais se destacaram foram:

1. *High-level entity-management system (HEMS):* Isto foi uma generalização que talvez tenha sido o primeiro protocolo de gerenciamnto de rede utilizado na Internet, o protocolo monitorador de servidor (ou *HMP - Host Monitoring Protocol*).

2. *Simple Network Management Prototol (SNMP)*: Esta foi uma versão estendida do protocolo de monitoração de *gateway* (SGMP - Simple Gateway Monitoring Protocol).

3. *CMIP sobre TCP/IP (CMOT)*: Esta foi uma tentativa de incorporar o máximo de extensões possíveis, o protocolo (*CMIP - Common Management Information Protocol*), serviços, e uma base de dados estruturada que foi padronizada pela ISO (*International Organization for Standization*) para o gerenciamento de rede.

Em meados de 1988, o IAB (Internet Activities Board) reviu estas propostas e aprovou o desenvolvimento do SNMP como sendo uma solução de curto prazo e o CMOT como sendo uma solução de longo prazo (Cerf 1988).

Surge o SNMP

Com os desenvolvedores do SNMP livres das restrições de compatibilidade da OSI, o progresso foi rápido espelhando-se na história do TCP/IP. O SNMP logo
ficou disponível em uma variedade de equipamentos. Com isto, logo o SNMP se tornou o padrão de protocolo para gerenciamento de rede.

A Evolução do SNMP

O básico do SNMP está amplamente utilizado hoje em dia. Praticamente todos os maiores vendedores de computadores, estações de trabalho, *bridges*, roteadores, e hubs

oferecem o SNMP básico. O trabalho continua progredindo para utilizar o SNMP sobre o modelo OSI e outros protocolos diferentes do TCP/IP. Com isto, melhoramentos no SNMP estão sendo realizados em várias direções.

Talvez a mais importante destas iniciativas seja o desenvolvimento da capacidade de monitoração remota (RMON) do SNMP. A especificação para a monitoração remota define complementos a MIB básica do SNMP assim como as funções que exploram a RMON MIB. O RMON permite ao gerente de rede a habilidade de monitorar subredes como se fossem dispositivos individuais. Tanto os vendedores quanto os usuários viram no RMON uma extensão substancial do SNMP.

Existe um limite de até onde o SNMP pode ser extendido, simplesmente definindo-se novas e mais elaboradas MIBs. O RMON talvéz represente o quão longe um é capaz de ir, tentando-se melhorar a funcionalidade do SNMP pela adição de objetos na MIB. Entretanto, como o SNMP está sendo utilizado em grandes e sofisticadas redes, as deficiências começam a tornar-se mais aparentes. Estas deficiências são nas áreas de segurança e funcionalidade.

Muito foi feito para remediar estas deficiências. Como primeiro passo, um conjunto de três documentos definindo melhorias na segurança do SNMP foram publicados em julho de 1992 propondo a padronização. No mesmo mês, surge a proposta para uma nova versão do SNMP chamada de SMP (Simple Management Protocol).

O SMP foi aceito como sendo a base para a definição de uma segunda geração do SNMP, conhecida como SNMP versão 2 (SNMPv2). Dois grupos foram formados para desenvolver as especificações do SNMPv2. Um grupo concentrou-se em todos os aspectos do SNMPv2, enquanto outro grupo concentrou-se nos recursos de segurança. O resultado foi a publicação, no início de 1993, de um conjunto de 12 documentos definindo os padrões do SNMPv2.

Arquitetura de Gerenciamento de Rede

O modelo de gerenciamento utilizado em redes TCP/IP incluem os seguintes elementos chaves:

- Estação de gerencia.
- Agente de gerenciamento.
- Base de informações gerenciais (MIB).
- Protocolo de gerenciamento de rede.

A estação de gerência é tipicamente um dispositivo *stand-alone* com capacidades de sistema compartilhado. A estação de gerência serve como interface entre o usuário e o sistema de gerenciamento da rede. Ela deve ter no mínimo as seguintes características:

- Um conjunto de aplicações para análise dos dados e recuperação de falhas.
- Uma interface de rede pela qual o gerente de rede possa monitorar e controlar a rede.

- A capacidade de traduzir as solicitações do administrador de rede em monitoração e controle de elementos remotos na rede.

- Uma base de dados extraída da MIB de todas as entidades gerenciadas na rede.

Somente os dois últimos elementos fazem parte do SNMP.

O elemento ativo no sistema de gerenciamento de rede é o Agente de Gerenciamento. Equipamentos como computadores, *bridges*, roteadores e hubs devem ser equipados com um agente SNMP para que possam ser gerenciados a partir da estação de gerência. O agente responde a requisições de informações e requisições de ações vindas da estação de gerencia e pode prover à estação de gerência informações importantes porém não solicitadas.

A estação de gerência e o agente estão ligados pelo protocolo de gerenciamento da rede, que no caso do TCP/IP é o SNMP. O protocolo inclui as seguintes funcionalidade:

- *Get:* permite a estação de gerência obter um valor de um objeto no agente de gerencia.

- *Set:* permite a estação de gerência enviar (definir) um valor para um objeto no agente de gerencia.

- *Trap:* permite ao agente de gerência enviar notificações à estação gerência sobre eventos importantes.

Protocolo para Gerenciamento de Rede

O SNMP foi projetado para ser parte do protocolo TCP/IP a nível de aplicação. Ele foi projetado para operar sobre UDP (*User Datagram Protocol*).

Cada agente deve implementar no mínimo o SNMP, UDP e IP.

A figura 2.4 mostra detalhadamente o contexto do protocolo SNMP. A partir da estação de gerência, três tipos de mensagens SNMP são emitidas para a aplicação de gerenciamento: *GetRequest*, *GetNextRequest* e *SetRequest*. Sendo que as duas primeiras são uma variação da função *Get*. As três mensagens são reconhecidas pelo agente na forma de uma mensagem *GetResponse*, que é passada para a aplicação de gerenciamento. Além disto, um agente ainda pode emitir uma mensagem do tipo *Trap* em resposta a um evento que afeta a MIB e seus recursos.

O SNMP por ser um protocolo baseado em UDP, que é um protocolo sem conexão, torna-o também sem conexão. Não são mantidas conexões entre o gerente e o agente no processo de gerenciamento. Em vez disto, cada troca de mensagens é uma transação separada entre a estação gerente e o agente.

Trap

Se a estação de gerência é responsável por um grande número de agentes, e cada agente mantém um grande número de objetos, torna-se impraticável para a estação gerente regularmente verificar em todos os agentes, todos os objetos. Assim, o SNMP

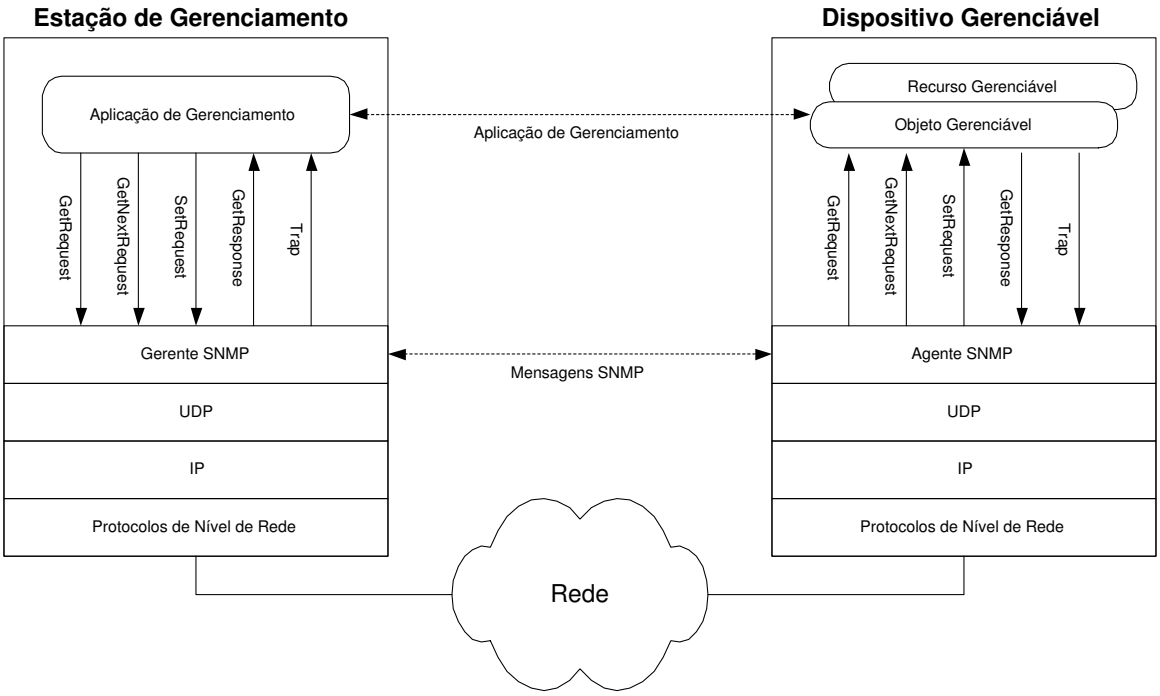

Figura 2.4: Comunicação SNMP.

e a MIB foram desenhados para permitir que o gerente use uma técnica chamada de *Trap-Directed Polling*.

A estratégia recomendada é: na inicialização, e talvez em intervalos aleatórios, como uma vez ao dia, a estação de gerência deve verificar todos os agente em busca de informações chaves, como carecterísticas de interface, e talvez algumas informações para estatísticas de desempenho. Em vez disto, cada agente é responsável por notificar a estação gerente por eventos não usuais, como por exemplo: uma reinicialização, falha de interface, sobrecarga de tráfego. Estes eventos são informados com mensagens SNMP conhecidas simplesmente como *Traps*.

No momento em que a estação de gerência é alertada de um evento excepcional, ela pode tomar algumas ações. Neste momento, a estação de gerência deverá verificar o agente em busca de informações que sirvam para diagnosticar o problema que ocasionou o *Trap*.

O *Trap-Directed Polling* pode resultar em uma substancial economia de capacidade de rede e processamento do agente.

2.3.2 A MIB

Assim como qualquer sistema de gerenciamento de rede, a base do gerenciamento TCP/IP é uma base de dados contendo informações sobre os elementos que serão gerenciados. Em ambos os ambientes TCP/IP e OSI a base de dados é chamada de MIB - *Management Information Base*. Cada recurso a ser gerenciado é representado

por um objeto.

De modo a fazer a MIB servir aos propósitos do gerenciamento de rede, deve-se definir dois objetos:

1. O objeto, ou objetos, utilizados para representar um recurso em particular deve ser o mesmo em cada nodo da rede.

2. Um esquema comum de representação deve ser usado para suportar interoperabilidade.

Estrutura da MIB

A estrutura do gerenciamento de informações (SMI[3]) é especificada na RFC 1155 que define um ambiente no qual a MIB pode ser definida e construída. O SMI identifica os tipos de dados que podem ser usados na MIB e como os recursos serão representados e nomeados.

Para padronizar o modo como serão representadas as informações de gerenciamento, a SMI descreve detalhadamente as técnicas padrões para:

- Definir a estrutura de uma MIB em particular.

- Definir os objetos individuais, incluindo a sintaxe e o valor de cada objeto.

- Codificar os valores dos objetos.

Associado com cada tipo de objeto na MIB existe um identificador do tipo OBJECT IDENTIFIER (ASN.1[4]). O identificador serve para nomear o objeto. Um OBJECT IDENTIFIER é um único identificador para um objeto em particular. Consiste de uma sequência de números inteiros. O conjunto de objetos possui a estrutura em forma de árvore, com a raíz desta árvore referenciando um objeto do padrão ASN.1. Começando pela raíz, existem três nodos no primeiro nível: iso, ccitt, joint-iso-ccitt. Abaixo do nodo *iso*, uma sub-árvore é utilizada pelo US-DoD[5] e outra utilizada por outras organizações.

A figura 2.5 representa a estrutura básica para o gerenciamento SNMP. Então o nodo *internet* possui o valor do seu OBJECT IDENTIFIER como 1.3.6.1. Este valor serve de prefixo para os nós nos próximos níveis da árvore. A SMI define quatro nodos abaixo do nó *internet*:

1. *directory:* reservado para uso futuro da ISO (padrão X.500).

2. *mgmt:* esta sub-árvore é usada pelos objetos definidos no IAB[6].

[3]SMI: Structure of Management Information
[4]ASN.1: Abstract Syntax Notation One.
[5]US-DoD: U.S. Department of Defense
[6]IAB: Internet Activities Board

3. *experimental:* esta sub-árvore é utilizada para identificar objetos usados em experiências.

4. *private:* esta sub-árvore é usada para identificar objetos definidos unilateralmente. Os fabricantes de equipamentos geralmente criam seus OBJECTS IDENTIFIERS neste nível da árvore.

Sintaxe dos Objetos

Os objetos na MIB do SNMP e toda a estrutura da MIB estão definidas utilizando-se o ASN.1. Com o objetivo de manter a simplicidade, somente um grupo restrito de elementos e recursos da ASN.1 foram utilizados.

Tipos Universais

Na classe UNIVERSAL, somente os tipos de dados abaixo são permitidos para definir objetos da MIB:

1. INTEGER (UNIVERSAL 2)

2. OCTET STRING (UNIVERSAL 4)

3. NULL (UNIVERSAL 5)

4. OBJECT IDENTIFIER (UNIVERSAL 6)

5. SEQUENCE, SEQUENCE OF (UNIVERSAL 16)

Tipos de Aplicação

A RFC 1155 lista os tipos de dados para aplicações, são eles:

- *NetworkAddress:* este tipo é definido usando a construção CHOICE, que permite a escolha do formato do endereço. Atualmente, somente está definido o IpAddress.

- *IpAddress:* um endereço de 32 bits usando o formato específico do IP.

- *Counter:* um inteiro não negativo que pode ser incrementado mas não decrementado. O valor máximo que pode alcançar é $2^{32} - 1$ ou (4.294.967.295), quando o contador chegar no máximo ele retorna novamente a zero.

- *Gauge:* um inteiro não-negativo que pode ser incrementado e decrementado. O valor máximo é $2^{32} - 1$, se o valor máximo é alcançado, o valor se mantém no máximo até ser reinicializado.

- *TimeTicks:* um inteiro não-negativo que conta o tempo em centésimo de segundos desde uma determinada época.

- *Opaque:* possui a capacidade para repassar dados. Os dados são codificados usando o tipo OCTET STRING.

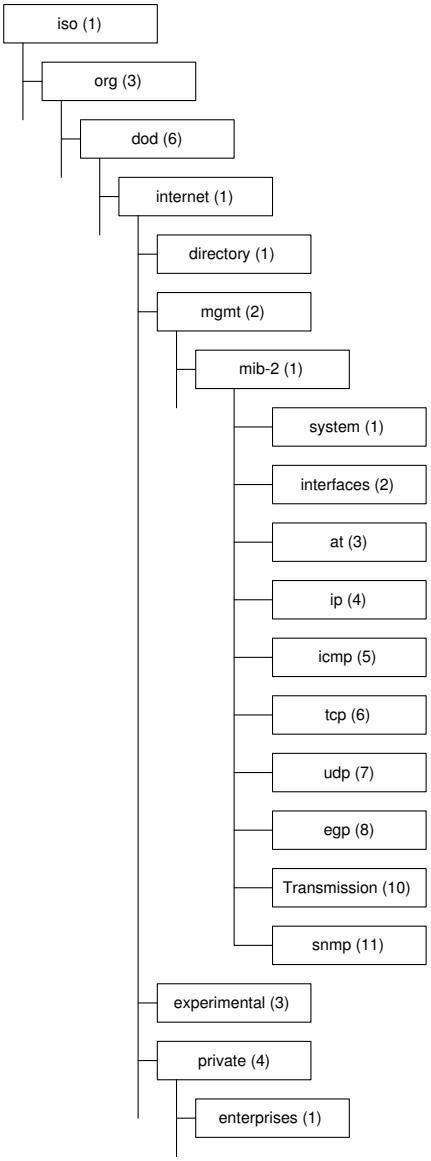

Figura 2.5: Grupos de Objetos da MIB-II.

Definindo Objetos

Uma MIB consiste de um conjunto de objetos. Cada objeto possui um tipo e um valor. O tipo do objeto define um tipo particular de objeto gerenciável. A definição do objeto é portanto uma descrição sintática.

Como podemos definir objetos para serem incluídos na MIB? A padronização utilizada é a ASN.1. Uma alternativa para definir objetos gerenciáveis é definir um novo tipo chamado Objeto. Então cada objeto da MIB será deste tipo. Precisamos permitir uma variedade de tipos de valores, incluindo contadores, *gauges*, etc. Além disto, a MIB ainda permite a definição de tabelas bi-dimensionais, ou *arrays*, de valores.

Uma alternativa mais atrativa é o uso de macros para definir um conjunto de tipos relacionados, usados para definir objetos gerenciáveis. Existem três níveis de definições para as macros:

- *Macro definition:* especifica a sintaxe do conjunto de tipos relacionados.
- *Macro instance*: uma instância gerada a partir de uma definição de macro específica informando-se os argumentos para os parâmetros da definição da macro.
- *Macro instance value*: representa uma entidade específica com valores específicos.

Definindo Tabelas

Até o momento a SMI suporta apenas tabelas bi-dimensionais. A definição de tabelas requer o uso dos tipos SEQUENCE e SEQUENCE OF da ASN.1 e IndexPart da macro OBJECT-TYPE.

MIB-II

A MIB-II (RFC 1213) define a segunda versão da MIB; a primeira versão, MIB-I, foi anunciada na RFC 1156. A MIB-II é um superconjunto da MIB-I com objetos e grupos adicionais.

Alguns critérios devem ser obedecidos para que um objeto pertença a MIB-II (RFC 1213):

- Um objeto deve ser essencial tanto para gerenciamento de falha quanto de configuração.
- Somente objetos fracos são permitidos devido aos protocolos ainda não serem muito seguros para operações de controle.
- Para evitar variáveis redundantes, é necessário que o objeto seja incluído como se fosse derivado de outro.
- Implementação de objetos específicos, para sistemas específicos.

A figura 2.5 mostra os objetos da MIB-II subdivididos nos seguintes grupos:

- *system*: informações gerais sobre o sistema.
- *interfaces*: informações sobre cada uma das interfaces do sistema.
- *at (address translation)*: descreve a tabela de tradução de endereços ARP[7].
- *ip*: informações relacionadas com a implementação e execução de experiências com o protocolo IP (Internet Protocol).
- *icmp*: informações relacionadas com a implementação e execução de experiências com o protocolo ICMP (Internet Control Message Protocol).
- *tcp*: informações relacionadas com a implementação e execução de experiências com o protocolo TCP (Transmission Control Protocol).
- *udp*: informações relacionadas com a implementação e execução de experiências com o protocolo UDP (User Datagram Protocol).
- *egp*: informações relacionadas com a implementação e execução de experiências com o protocolo EGP (External Gateway Protocol).
- *transmission*: provém informações sobre os esquemas de transmissão e protocolos de acesso de cada interface.
- *snmp*: informações relacionadas com a implementação e execução de experiências com o protocolo SNMP (Simple Network Management Protocol).

Aplicações Práticas

Como citado anteriormente, uma das duas funções básicas do gerenciamento de rede é a monitoração, que envolve medir certas quantidades de informações e reportar os resultados ao sistema gerenciador. No SNMP, a MIB suporta monitoração através do uso de contadores e *gauges*.

Por exemplo, considere um equipamento como uma *bridge* ou roteador cuja principal função é a de intermediar a comunicação entre outros dois equipamentos, capturando e reenviando os pacotes pela rede. A preocupação do gerente com o tamanho de sua implementação (quantos roteadores usar, qual a vazão cada roteador pode sustentar) está em saber quanto de tráfego passa por cada *bridge* ou roteador. O que pode ser mais simples?

Comparado com o modelo de gerenciamento da OSI, o SNMP sacrifica funcionalidade pela simplicidade. Uma das vantagens do SNMP é sua clareza e sutileza.

[7] ARP: Address Resolution Protocol - Protocolo de Resolução de Endereços

MIBs Privada

Outra das vantagens do SNMP é o modo como a MIB foi desenhada de forma a acomodar crescimento e flexibilidade para novos objetos. Extensões privadas podem ser acomodadas na sub-árvore *private* (Figura 1.5). Isto permite aos fabricantes criarem objetos para gerenciar aspectos de seus equipamentos e deixar visível estes obejtos à estação de gerenciamento. Por causa da padronização do SMI e dos Object Identifier, é possível gerenciar objetos privados de diferentes fabricante a partir da mesma estação de gerência.

Limitações dos Objetos da MIB

O gerenciamento de rede está limitado pelas capacidades do protocolo de gerência e pelos objetos utilizados para representar o ambiente a ser gerenciado. O padrão da MIB-II serve como base comum a todas as implementações, e é importante reconhecer que a MIB-II possui um limite na habilidade de monitorar e controlar uma rede.

3 NECESSIDADE POR GERENCIAMENTO

3.1 Gerência e monitoração usando software livre.

Em muitos ambientes de rede necessitamos de ferramentas que nos possam dar alguma informação sobre o que está acontecendo em nossa rede. Até pouco tempo atrás, se desejássemos monitorar uma rede éramos obrigados a escolher entre duas soluções possíveis:

1. Utilizar Ambientes de Gerenciamento de Rede comerciais, os quais invariavelmente são muito caros. Para uma rede de pequeno-médio porte (de 1 à 4 subredes interligadas) muitas vezes o custo elevado deste software não é compatível com o faturamento da empresa, portanto inviável. Outro ponto é que softwares deste tipo geralmente monitoram bem os equipamentos afiliados ao seu fornecedor, dando um suporte mais básico a outros equipamentos.

2. Utilizar pequenos softwares para monitoração de redes, softwares estes que possuem um conjunto limitado de ferramentas e que são praticamente "caixas-pretas", pois não sabemos como funcionam. Para gerenciamento de redes teremos que utilizar mais de uma ferramenta simultaneamente, o que dificulta a atividade do administrador de rede.

Hoje em dia temos uma terceira opção. Com a consolidação do TCP/IP como protocolo de rede, e por conseguinte do SNMP como protocolo de gerenciamento; e do surgimento do Software Livre, muitas ferramentas de gerenciamento e monitoração estão sendo desenvolvidas.

O desenvolvimento de aplicações utilizando software livre está se tornando estável ao ponto de se tornar viável. Outro fator que contribui é que não há custo de propriedade no uso das ferramentas. Embora elas ainda não estejam totalmente prontas, do ponto de vista de aplicações comerciais; seus resultados já são satisfatórios.

Quando necessitamos desenvolver algum produto de software utilizando-se do Software Livre temos que ter em mente algumas de suas características:

Simplicidade

As ferramentas de software livre para gerência e monitoração geralmente abordam um único tópico, como por exemplo a coleta de dados estatísticos, o que as torna extremamente simples de serem utilizadas. Por serem aplicações simples, não demandam grandes quantidades de recursos computacionais e podem ser utilizadas de forma escalar dividindo-se suas atividades em vários equipamentos.

Código Fonte

Como são ferramentas de software livre, o acesso ao código fonte das aplicações é garantido. Pode-se saber exatamente o que a aplicação está fazendo, e se
for necessário, alterá-la de forma a deixá-la personalizada ou adaptá-la as nossas necessidades.

Documentação

O acesso a documentação é fácil, além de contar com a documentação original, pode-se ainda acessar os grupos de usuários para conseguir informações sobre as capacidades das ferramentas. Cria-se também uma "Comunidade" de usuários em torno desta ferramenta, pessoas que tem necessidades comuns e buscam soluções parecidas.

Aperfeiçoamento

As ferramentas de software livre não são produtos estanques. Estão em contínuo aperfeiçoamento, devido também a esta comunidade que se cria em torno dela. Outro fator importante é que a depuração de problemas é permanente. Uma vez identificado um problema, este entra em uma lista de "afazeres" que será resolvido em versões futuras.

Custo

Não existe Custo de Propriedade associado ao Software Livre. Todos os produtos sob esta modalidade estão livres de custos. Porém o maior custo está no fator humano, no tempo dispendido para analisar e implementar soluções com estas ferramentas.

3.2 Descrição do Problema

O desafio está em criar um ambiente de gerenciamento e monitoração de baixo custo utilizando-se de ferramentas de software livre. Estas ferramentas devem trabalhar de forma integrada e transparente para o administrador da rede.

Este modelo deve ser modular de forma a possibilitar extensões futuras com o mínimo de remodelagem.

Este ambiente deve ser capaz de coletar e fornecer informações sobre os diversos objetos que serão monitorados. Estatísticas devem ser geradas a partir de métricas especificadas. Disponibilidade de serviços de rede devem ser monitorados. O acesso ao ambiente deve ser independente de local e plataforma.

Necessidade de um ambiente homogêneo

Uma aplicação de maior complexidade necessita de um ambiente homogêneo na qual se possa trabalhar. Este ambiente será uma interface entre o que necessitamos e as

várias ferramentas que nos dão apoio, funcionando de forma transparente na integração das várias ferramentas.

Necessidade de integrar ferramentas

As ferramentas disponíveis para gerência e monitoração em Software Livre são como peças de um produto maior. Geralmente necessitamos de mais de uma ferramenta para implementar uma solução de maior complexidade. Para isto precisamos integrar as ferramentas de forma que possam trabalhar juntas. Esta integração é possível devido a existência de uma boa documentação e do acesso ao código fonte, onde podemos analisar quais são as entradas necessárias para a aplicação e saídas desejadas.

3.3 O "Estado da Arte" em ferramentas de gerenciamento

Gerenciando o Desempenho

Os produtos mais avançados de gerência de desempenho possuem a inteligência de escolher o dispositivo certo para extrair a estatística certa do jeito certo. A gerência de desempenho busca a estatística a partir de pontos chaves na infra-estrutura. Este propósito não é só para produzir gráficos interativos em tempo real para diagnóstico imediato. Ao contrário, as informações são armazenadas em um banco de dados e processadas de forma a apresentar as *baselines*, tendências e violações de limites, que podem ser apresentadas como relatórios gerenciais ou relatórios detalhados de desempenho para diagnóstico e planejamento futuros.

Comparado com ferramentas tradicionais, que tentam rastrear cada dispositivo e interface de rede, estes produtos mais avançados apresentam informações apenas pela busca a dispositivos chaves. Os dados provêm da coleta a partir da MIB2, RMON, RMON2 e MIBs proprietárias.

Mas além de apenas coletar dados, os produtos adicionam valores de duas formas: primeiro provendo relatórios que identificam as necessidades específicas a diferentes usuários, departamentos, divisões e clientes. E segundo, eles organizam os relatórios baseados em certos critérios como tipo de dispositivo e aplicações.

Por muito tempo, as plataformas de gerência de desempenho baseadas em SNMP possuíam dificuldade em apresentar uma idéia clara sobre como o desempenho da rede estava. Agora, uma nova classe de ferramentas está “minerando” os dados da MIB2, RMON e MIBs proprietárias para criarem relatórios de alto nível e visões detalhadas para diagnóstico no uso das redes.

Monitorando a Rede

Gerenciar aplicativos de várias camadas ou com diferentes demandas
continua sendo um grande desafio. Afinal, as redes estão sob o comando de chefes diferentes, os quais, convenientemente, culpam uns aos outros quando surgem problemas. Uma nova geração de gerenciadores MoMs (*manager of manager* - gerenciadores de gerenciadores) tentam acabar com as brigas.

O coração dos produtos é o chamado “centro de excessão”. Quando erros ocorrem ou limites são ultrapassados, um evento é criado como uma mensagem formatada que

identifica a finalização do funcionamento de uma rede ou sistema. Enquanto algumas informações provêm diretamente dos dispositivos de rede, servidores e aplicativos, outros saem de sistemas de gerenciamento que monitoram as máquinas. Um MoM precisa, então, reunir e dar sentido a esse conjunto de informações.

Embora os domínios de gerenciamento em separado ofereçam dados de desempenho granulares e úteis, eles não têm idéia de como seu desempenho afeta outros dispositivos na cadeia. Sem a correlação, é impossível deduzir se um *switch* está ocasionando tempo longo de resposta dos aplicativos.

Não é papel do MoM fornecer diagnóstico granular e dados ajustados de como banco de dados, servidores WEB ou switchs de rede estão se saindo. As ferramentas atuais têm uma tarefa bem mais complexa do que as das gerações anteriores. Aplicativos multi-tarefas rodam em sistemas distribuidos forçam a solução a reunir e correlacionar uma rede mais ampla de eventos.

Os MoMs são gerentes de eventos. Se puderem alertar o setor de Gerência de Rede sobre uma interrupção antes que do chamado de um cliente ou de um usuário, será uma vitória. Isto soa como algo fácil, mas trafegar por milhares de ocorrências é como achar uma agulha em um palheiro. Porém, tal controle de eventos não é o suficiente, para construir relatórios é necessário obter diagnósticos para diminuir o tempo inativo. A maioria das soluções, entretanto, exige uma implementação front-end pesada e muito rigor e disciplina para realizar tal meta.

Gerenciar eventos era a grande parte do trabalho inicial do MoM, o qual necessitava desenvolver técnicas especiais. A automação posterior reduziu o número de eventos que exigiam uma resposta. A idéia era adotar regras que, quando atendidas, resultavam em algum tipo de ação. Estabelecer normas para eventos, os quais especificam reações e ações, empurraram os MoMs para frente. Um bom exemplo é a supressão de acontecimentos pela perda de contato entre o roteador e dispositivos de *downstream*. Se uma interface é desligada, o dispositivo ignora qualquer evento que surja a partir dos equipamentos que não podem ser contatados.

Os novos MoMs ainda administram o gerenciamento de eventos, mas precisam fazer mais. Uma solução sofisticada fornece uma idéia das origens dos erros e pode tomar uma ação corretiva automaticamente.

Infelizmente, a implementação de um MoM custa uma pequena fortuna. O produto mais simples no mercado norte-americano custa cerca de US$ 2 mil, com adicionais US$ 40 mil ao ano pela manutenção e outros vários milhares de dólares destinados a treinamento. É óbvio que quanto mais complexa a organização, menor o custo proporcional comparado com o orçamento de TI geral. Por este motivo corporações de grande peso usufruirão dos benefícios dos MoMs mais facilmente que as menores.

3.3.1 Ferramentas de Gerenciamento e Monitoração

Produtos

Existem no mercado vários produtos para gerenciamento de redes e sistemas. Dentre todo o espectro de gerência de redes - Falha, Configuração, Contabilidade, Desempe-

nho e Segurança - foram pesquisadas quatro ferramentas que atendam aos requisitos de detecção de falhas e monitoração de desempenho: duas em arquitetura proprietárias e duas baseadas em Software Livre.

Os critérios adotados para a avaliação são:

- **Interface WEB**: o sistema de gerenciamento deve possuir interface WEB que permita visualizar e administrar seus recursos;

- **Monitoração de Desempenho**: o sistema deve ser capaz de coletar dados de métricas especificadas gerando representações gráficas dos dados. Deve possuir recurso de verificação de estouro de limites de forma alertar o administrador da rede destas ocorrências;

- **Monitoração de Serviços de Rede**: o sistema deve ser capaz de verificar diversos serviços de rede TCP/IP. Dentre os serviços de rede estão: disponibilidade (PING), serviço WEB (HTTP), serviço de correio eletrônico (SMTP), serviço de terminal remoto (TELNET), serviço de nomes (DNS), e conexões em portas TCP, este último consistindo em verificar a conexão em uma porta TCP específica.

- **Análise de Desempenho On-Line**: o sistema deve ser capaz de monitorar de forma on-line o que esta ocorrendo nas métricas de gerenciamento, traçando um gráfico de seu desempenho.

- **Central de Alerta**: o sistema deve possuir um local onde todos os alertas sejam concentrados. Deve apresentar o evento de um alerta o mais rápido possível. Os alertas devem ser enviados para outros dispositivos além da interface da aplicação, tais como: e-mail, pager, e SMS. O sistema também deve armazenar o histórico de eventos dos alertas.

3.3.1.1 IBM Tivoli Netview

O IBM Tivoli Netview é um dos produtos da linha Tivoli da IBM responsável pelo gerenciamento de desempenho e disponibilidade de sistemas. O Tivoli realiza um "scanner" em redes TCP/IP, mostra a topologia das redes, correlaciona e administra eventos e problemas de SNMP, controla o estado da rede, e recompila dados sobre rendimento. Dentre suas características proporciona uma solução de administração distribuída escalável. Oferece relatórios de analises e tendências de redes.

Possui recursos para identificar a causa original de falhas na rede. Quando ocorre um problema em uma rede, o Router Fault Isolation (RFI) rapidamente se concentra no encaminhador com falha, e informa sobre o problema e quais os dispositivos afetados, descartando a avaliação desnecessária dos dispositivos afetados.

O Tivoli Netview Java Web Console oferece fácil acesso a uma completa gama de informações sobre redes procedente de localizações remotas.

O IBM Tivoli Enterprise Console é outro dos produtos da linha Tivoli especializado em eventos e alarmes. É um centro de controle para a totalidade do ambiente de rede. Ele processa e responde aos milhares de eventos e alarmes que ocorrem diariamente nos dispositivos de rede, sistemas de hardware, sistemas de administração de bases de dados relacionais e aplicativos. Integra completamente os eventos. Possui um potente processamento e correlação de eventos. Envia respostas e notificações de eventos de forma segura e automatizada.

3.3.1.2 Spectrum Infinity

O Spectrum Infinity é um produto para gerenciamento de redes da Aprisma Management Technologies no qual possui em seu núcleo um analisador de causa-raiz e uma série de aplicações adicionais. Ele possui três aspectos fundamentais para prover Garantia de Serviço: Monitoração pró-ativa, analise inteligente e restauração de serviços.

Monitoração Pro-ativa

O sistema inicia com uma modelagem automatizada dos dispositivos de rede, sistemas, aplicações e elementos de segurança no ambiente da rede. O Spectrum se comunica com os dispositivos e sistemas usando uma variedade de protocolos padrão de industria e protocolos proprietários para monitorar desempenho e disponibilidade. Possui módulos de gerenciamento que coletam dados e diagnosticam problemas com mais de 1000 dispositivos de fabricantes diferentes e aplicações. O Spectrum usa estes módulos de gerenciamento para prover acesso detalhado a informações de configuração de dispositivos, desempenho e operacionalidades.

Analise Inteligente

O Spectrum utiliza um modelo de infra-estrutura de tempo real e histórico de estatísticas criados durante a coleta dos dados. Normaliza grandes volumes de dados, correlacionando centenas de milhares de notificações de problemas, e suprimindo notificações repetidas ou sem sentido. O Spectrum possui habilidade para identificar problemas em tempo real o que permite aos administradores de rede antecipar problemas de infra-estrutura antes que os usuários sejam afetados.

Restauração de Serviços

O Spectrum utiliza os resultados de monitorações e analises prévias para resolver problemas de infra-estrutura já identificados. Ele primeiro procura automatizar uma ação corretiva através de uma comunicação direta com o componente afetado. Se a correção automática não é possível, ele sugere ações manuais.

Sistema de Gerenciamento de Falhas

Muitos provedores de serviços tem disponibilizado uma variedade de componentes de infra-estrutura consistindo de servidores, balanceadores de carga, LAN Switchs, roteadores e ATM Switchs. Problemas de rede podem acontecer em qualquer subcomponente destes serviços afetando muitos clientes em um curto período de tempo. Para manter a infra-estrutura ativa, os provedores precisam encontrar o causa raiz de

forma rápida, determinando o impacto que os clientes terão e resolvendo-os antes que sejam severamente afetados.

Sistema de Gerenciamento de Eventos

O Spectrum permite verificar métricas chaves e realizar uma correlação inteligente de dados de forma a permitir que um problema venha a impactar na rede.

Analise de Impacto de Alarmes

O Spectrum permite priorizar alarmes baseado na importância dos dispositivos gerenciados e no impacto que este alarme pode causar na rede.

WEB Operator Suite

O Web Operator Suite foi desenhado para prover operações de rede e atendimento interativo aos dispositivos de rede e aplicações através de uma simples interface WEB.

Gerenciador de Alarmes

O Gerenciador de Alarmes permite criar políticas de notificação através do notificador de alarmes para enviá-las por e-mail, enviá-las por Pagers, consolidar alarmes, filtrar alarmes, redirecionar e agendar alarmes.

Gerenciador de Desempenho de Serviços

O Gerenciador de Desempenho integra a habilidade de gerenciar falhas com gerenciamento de desempenho para assegurar a disponibilidade dos sistemas. Pela da detecção de problemas com tempo de resposta, estreitando o foco no processo de isolamento de falhas, e através de ferramentas para validar os problemas reportados e verificar sua resolução; o Spectrum ajuda a prevenir o impacto causado por um desempenho de rede pobre.

3.3.1.3 Open NMS

O OpenNMS é um dos tipos de sistema de software no mundo do software de fonte aberto. O OpenNMS é um sistema de gerenciamento centrado no usuário, colocando o administrador de rede como ponto focal para determinar os requisitos de funcionalidade do sistema. O OpenNMS monitora, controla e coleta dados através de um conjunto de atividades. A tabela 3.1 apresenta algumas de suas tarefas.

3.3.1.4 Big Brother System & Network Monitor

O Big Brother é um sistema de gerenciamento e monitoração de rede baseado em WEB. Consiste de simples scripts que periodicamente monitora a rede e o estado dos sistemas. Espaço em disco, utilização de CPU, servidores, e processos importantes podem ser monitorados.

O Big Brother apresenta através de uma página WEB uma matriz com as máquinas e funções que estão sendo monitoradas, com um código de cores que representam o estado atual.

O Big Brother pode notificar o administrador de rede através de e-mail, pager e SMS (através de software de terceiros).

Tabela 3.1: Tabela de Características do OpenNMS.

Tarefa	Descrição
Collection	Coleta dados de nodos gerenciáveis.
Discovery	Realiza uma varredura inicial nos nodos gerenciáveis.
Event Manager	Gerencia e armazena em um Banco de Dados os eventos originados nas outras tarefas.
Notification	Realiza a notificação aos usuários.
Outage Manager	Consolida eventos provendo um histórico de cada nodo ou serviço gerenciados.
Poller	Realiza uma verificação regular para determinar o estado operacional do nodo ou serviço.
Threshold Service	Monitora os nodos ou serviços gerenciados baseado em especificação de limites.

Através da adição em uma lista de serviços, o Big Brother pode monitorar a disponibilidade de serviços TCP específicos retornando os tempos de conexão.

O Big Brother também permite verificar os logs dos eventos (mudanças de estado) das últimas 24 horas ou das últimas 50 mudanças.

4 INTEGRAÇÃO DE FERRAMENTAS

4.1 Modelagem do Ambiente

A figura 4.1 apresenta o modelo proposto para a aplicação. Segundo nossas necessidades, a aplicação está dividida em sete módulos, descritos a seguir.

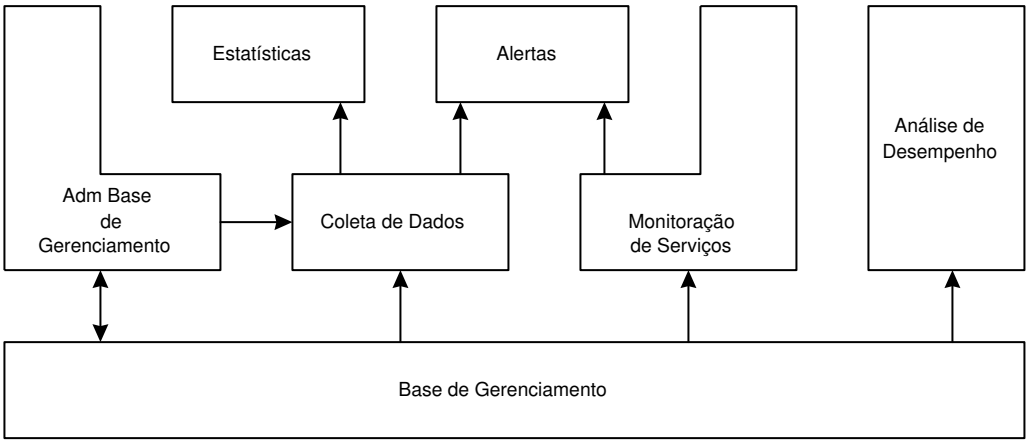

Figura 4.1: Modelo da Aplicação.

4.1.1 Base de Dados de Gerenciamento

A Base de Dados de Gerenciamento será responsável pela armazenagem de informações sobre os Objetos Gerenciáveis. Para cada objeto gerenciável poderão ser associadas Métricas de gerenciamento e Serviços de rede a serem monitorados. Os Objetos Gerenciáveis possuem também um histórico dos eventos de alertas registrados.

A base de dados é composta pelos seguintes objetos:

1. **Objeto Gerenciavel:** armazenar as informações sobre os objetos que serão gerenciados e monitorados. Cada Objeto Gerenciável possui uma ou mais Interfaces associadas. Ao Objeto Gerenciável serão associadas métricas para seu

gerenciamento. Um Objeto Gerenciável pode conter Serviços de Rede associados que devem ser monitorados. E finalmente, um Histórico é associado de forma a registrar os eventos que possa vir a ocorrer com este objeto.

2. **Interfaces:** armazenar informações sobre as interfaces dos Objetos gerenciáveis. Para a geração de gráficos estatísticos de desempenho das interface são associados Perfis de Gráficos. Também podemos associar Métricas para o monitoração das Interfaces.

3. **Métricas:** armazenar informações sobre as métricas utilizadas no gerenciamento dos Objetos Gerenciáveis. A Métrica especifica um agrupamento de Objetc ID (OIDs) da MIB que será utilizado. Especifica também valores que serão utilizados para verificação de limites (*threshold*). Quando um limite é excedido o sistema emite um alerta sobre o que ocorreu.

4. **Perfil de Gráfico:** armazenar parâmetros que definem o perfil dos gráficos que serão gerados pelas estatísticas.

5. **Object ID:** relaciona os OIDs da MIB que serão utilizados pelas métricas. Cada registro é composto por até dois OIDs possibilitando a coleta de dados comparativos (por exemplo: bytes enviados e bytes recebidos por uma interface de rede).

6. **Serviço:** especifica os serviços de rede que serão monitorados nos Objetos Gerenciáveis, especificando o Tipo de Monitor de Serviço que será utilizado, sua periodicidade e seus Alertas para falha e retorno. O tipo de monitor de serviço consiste em um script que realizará uma função específica dependendo do tipo de serviço a ser verificado.

7. **Monitor:** especifica os parâmetros que serão utilizados pelo script de monitoração de serviços de rede.

8. **Histórico:** armazenar os eventos de Alertas ocorridos com os Objetos Gerenciáveis.

9. **Alerta:** especifica como uma condição de alerta será informada para o gerente da rede, podendo ser por e-mail, pop-up de tela, pager, SMS, etc.

4.1.2 Administração da Base de Gerenciamento

A Administração da Base de Gerenciamento é responsável pela apresentação dos dados e pela manutenção da base de gerenciamento. Está dividida em dois módulos:

4.1.2.1 Módulo de Apresentação

Responsável pela apresentação das informações cadastradas na base de gerenciamento, das informações das atividades de monitoração, dos gráficos estatísticos das Métricas, do estado dos serviços monitorados e do histórico de alertas.

4.1.2.2 Módulo de Manutenção

Executa a manutenção dos dados e configurações da aplicação. Está subdividido no seguintes módulos:

- Manutenção de Objetos Gerenciáveis: responsável pela inclusão, alteração, exclusão de objetos gerenciáveis. Executa também a manutenção nas informações sobre as interfaces dos objetos gerenciáveis, métricas e serviços monitorados.

- Manutenção de Métricas: responsável pelo cadastramento, alterações e exclusões das métricas que serão utilizadas pelos Objetos Gerenciáveis. Cada métrica pode referenciar até dois OIDs. O uso de mais de um OID facilita a geração de gráficos estatísticos de utilização dos recursos dos Objetos Gerenciáveis, como por exemplo a utilização de rede onde pode-se confrontar a quantidade de bytes recebidos e enviados pela interface de rede.

- Manutenção do Perfil de Gráficos: responsável pela inclusão, alteração e exclusão dos perfis de gráficos utilizados para a geração de estatísticas. Para uma mesma métrica podem ser gerados diferentes gráficos.

- Manutenção dos Tipos de Alertas: responsável por cadastrar, alterar e excluir tipos de alertas. Os tipos de alertas definem a mensagem, o meio de envio de alerta e o nível de severidade.

- Monitoração de Serviços: responsável pelo cadastramento, alteração e exclusão dos monitores que executarão a verificação dos serviços disponíveis nos Objetos Gerenciáveis.

- Administração Módulo de Coleta de Dados e Estatísticas: responsável pela manutenção das configurações e parâmetros para os módulos de coleta de dados e geração de estatísticas.

- Administração Módulo de Monitoração de Serviços: responsável pela manutenção das configurações e parâmetros para o módulo de monitoração de serviços.

4.1.3 Coleta de Dados

Com base nas informações da Base de Dados de Gerenciamento este procedimento realiza a coleta de dados assinalado pelas métricas de gerenciamento. Este dados podem ser confrontados com parâmetros de limites máximos e mínimos definidos pelas métricas dos Objetos Gerenciáveis, sendo gerados alertas quando estes limites forem atingidos. Estes dados também serão processados pelo módulo de Estatísticas. Este procedimento será realizado periodicamente com base nos parâmetros de configuração.

4.1.4 Estatísticas

Responsável pela geração dos gráficos estatísticos. Com base nos dados obtidos na Coleta de Dados e de acordo com os Perfis de Gráficos, este procedimento processa os dados gerando os respectivos gráficos estatísticos.

4.1.5 Monitoração de Serviços

Este módulo é responsável pela monitoração da disponibilidade dos serviços e interfaces de rede. Este módulo verifica periodicamente se um determinado serviço de rede está disponível realizando uma tentativa de acesso ao serviço. Caso esta tentativa falhe, um alerta será gerado informando da indisponibilidade. O módulo continuará tentando verificar a disponibilidade do serviço, quando o serviço estiver normalizado, outro aviso poderá ser enviado informando do retorno a normalidade.

Este módulo também realiza a verificação do estado das interfaces de rede indicando se seu estado está ativo (*UP*) ou não ativo (*DOWN*). Este recurso permite monitorar o estado de links de comunicação, onde uma interrupção deste link (estado da interface em *DOWN*) pode ser alarmado indicando por exemplo o rompimento de cabo.

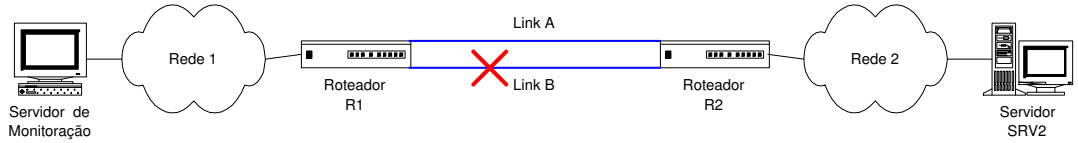

Figura 4.2: Exemplo de Monitoração de Comunicações.

No exemplo da figura 4.2 os serviços de rede do servidor SRV2 estão sendo monitorados pelo servidor de monitoração MON1 não apresentando qualquer anormalidade. Porém com o rompimento do Link B será gerado um alerta informando que ambas as interfaces B dos roteadores R1 e R2 estão em estado *DOWN*.

4.1.6 Alertas

Módulo responsável pelo registro de alertas e envio de mensagens ao Administrador de Rede. A cada evento que ocasione um alerta, o módulo verifica o tipo de alerta correspondente, grava a ocorrência do alerta no histórico e envia uma mensagem apropriada conforme os parâmetros do perfil de alerta utilizado.

4.1.7 Análise de desempenho

Módulo responsável pela coleta e apresentação de dados das métricas de gerenciamento de forma *on-line*. Permite verificar no momento atual o que está ocorrendo na métrica especificada. Apresenta os dados de forma gráfica permitindo uma comparação entre os valores das métricas que estão sendo monitoradas.

4.2 O protótipo desenvolvido

Desenvolvido a partir da linguagem de processamento de scripts PHP, o ambiente é composto de uma interface WEB que possibilita visualização de seus recursos.

Um banco de dados, utilizando-se o PostGreSQL, foi criado para armazenar as informações pertinentes aos objetos que serão gerenciados e monitorados. Alguns arquivos de configuração também foram criados para integrar as diversas ferramentas.

Para obtenção das informações de gerenciamento e monitoração utilizaremos a versão 2c do SNMP. Esta versão nos permite um nível básico de segurança. Não utilizaremos a versão 3 por ser recente e a maioria dos equipamentos verificados no Estudo de Caso não possuem suporte ao SNMP-v3.

Com o aplicativo MRTG, a aplicação é capaz de coletar dados e gerar estatísticas de diversas métricas, como: Utilização de Rede, Utilização de CPU, etc. Parâmetros máximos e mínimos podem ser atribuídos as métricas de forma a aplicação alertar quando estes forem ultrapassados.

A verificação da disponibilidade dos serviços é feita pelo Mon, verificando se algum serviço está fora do ar, e informando do seu retorno a normalidade.

Os Alertas gerados são gravados no Banco de Dados para futuras pesquisas, assim como enviados via e-mail, pop-up de tela, ou para pagers. Um módulo em Java alerta o ambiente WEB de novas ocorrências.

Outro módulo escrito em Java e C é responsável por traçar gráficos atualizados das métricas criadas. Assim é possível ver qual o tráfego de um segmento de rede quando um alerta de utilização de rede for alcançado.

4.2.1 Modelo de Informação - Base de Gerenciamento

O modelo de informação (figura 4.3) nos apresenta a estrutura de dados utilizadas para armazenar as informações necessárias ao funcionamento da aplicação. Utilizando-se o banco de dados PostGreSQL, foram criadas as tabelas a seguir.

4.2.1.1 Tabelas

A definição da estrutura das tabelas pode ser consultado no Anexo-I.

Objeto Gerenciável: Tabela para armazenamento dos Objetos Gerenciáveis. O Objeto Gerenciável é a base para o gerenciamento. A partir dele, e de seus recursos e características serão criados todos os scripts para gerência e monitoração. Ao criar um novo Objeto Gerenciável, algumas informações serão obtidas via consulta SNMP ao objeto criado, outras serão informadas pelo usuário. O objeto também será classificado segundo uma tabela de tipos.

Atributos:

- Nome do Objeto: é o *hostname* do objeto na rede.
- Endereço IP: endereço IP do objeto na rede.

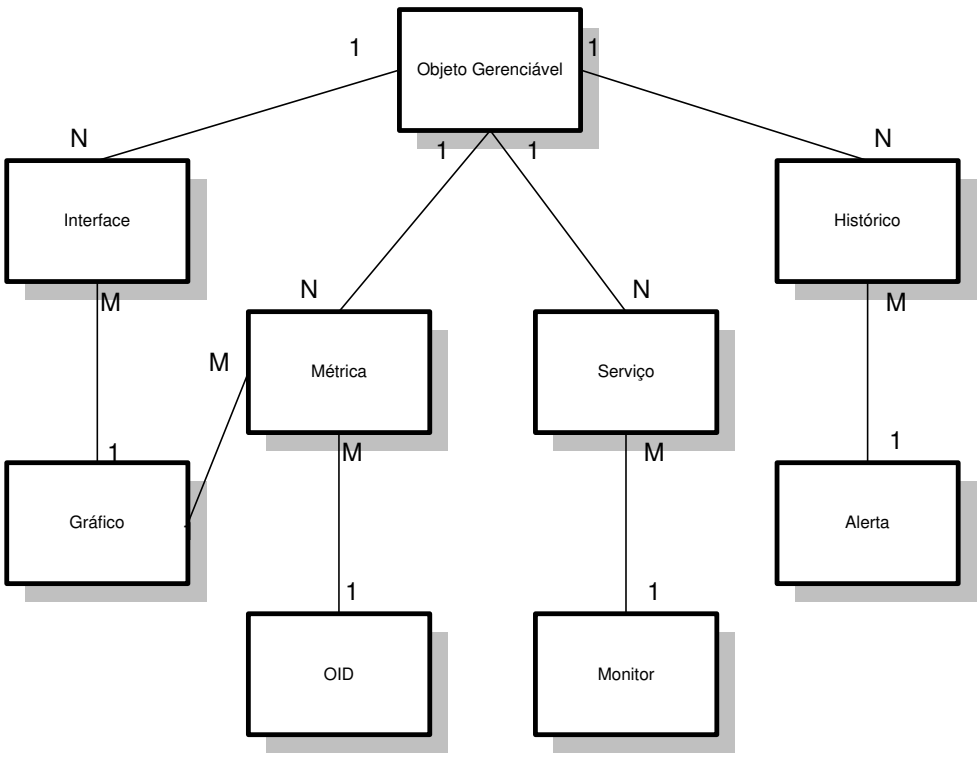

Figura 4.3: Modelo da Base de Dados de Gerenciamento.

- Localização: informação obtida apartir da MIB-II (system.sysLocation).
- Contato: informação obtida apartir da MIB-II (system.sysContact).
- Descrição Textual: uma descrição do que é o objeto.
- Tipo de Objeto: o objeto pode ser dos seguintes tipos: Servidor, Router, Switch, Printer, Undefined.
- Community de Leitura: community para leitura dos dados da MIB através do SNMP.
- Community de Escrita: community para escrita dos dados da MIB através do SNMP.
- URL: endereço para página de gerenciamento do objeto. Alguns equipamentos possuem interface WEB para sua administração.

Grupo OID: Tabela para armazenamento de grupos de ObjectID (OID) de gerenciamento. Esta tabela agrupa em número de dois os ObjectID's para posteriormente ser criada uma métrica para gerenciamento. Devido a uma limitação do software MRTG, apenas dois OIDs podem ser utilizados simultaneamente para gerar os gráficos comparativos.

Atributos:

- Descrição: descrição do grupo de OIDs.
- Object ID1: valor do Object ID para o primeiro parâmetro.
- Object ID2: valor do Object ID para o segundo parâmetro.
- Descrição do OID1: descrição textual para o primeiro OID.
- Descrição do OID2: descrição textual para o segundo OID.
- Sigla do OID1: sigla para o primeiro OID.
- Sigla do OID2: sigla para o segundo OID.
- Arquivos da MIB: localização do arquivo com a MIB para os respectivos OIDs
- Nome da MIB: nome da MIB para obtenção dos OIDs

Os atributos Descrição do OID1 e OID2, e Sigla do OID1 e OID2 serão utilizados na geração dos gráficos estatísticos.

Métrica: Tabela para armazenamento de quais métricas serão gerenciadas/monitoradas nos objetos. As métricas especificam qual quais os ObjectID's que serão utilizados para obter os dados, qual o tipo de gráfico que será gerado, quais os valores máximos e mínimos para verificação de limites (*Thresholds*), e qual tipo de alerta será utilizado. Com base nas associações destas informações a aplicação poderá gerar o script para o MRTG.

Atributos:

- Grupo OID: identificação do Grupo de OIDs que será utilizado para a obtenção de dados.
- Perfil de Gráfico: identificação do perfil do gráfico que será utilizado na apresentação das estatísticas.
- Interface: número da interface se a métrica se refere a alguma interface de rede, caso contrário é usado 0 (zero).
- Verificar Limites para OID1: verificar ou não limites de valores para o OID1.
- Valor máximo para OID1: valor máximo permitido para o OID1.
- Valor mínimo para OID1: valor mínimo permitido para o OID1.
- Verificar Limites para OID2: verificar ou não limites de valores para o OID1.
- Valor máximo para OID2: valor máximo permitido para o OID2.
- Valor mínimo para OID2: valor mínimo permitido para o OID2.
- Script de Alerta para OID1: script para acionamento de alerta para o OID1.
- Script de Alerta para OID2: script para acionamento de alerta para o OID2.
- Tipo de Alerta para OID1: tipo de alerta a ser enviado no caso de excesso de limite para o OID1.
- Tipo de Alerta para OID2: tipo de alerta a ser enviado no caso de excesso de limite para o OID2.

Gráfico: Tabela para armazenamento de perfis de gráficos. Especifica o tipo de gráfico que será gerado pelo aplicativo MRTG.

Atributos:

- Descrição do perfil: descrição simples do tipo de gráfico a ser gerado.
- Script: arquivo de *template* para geração do *script* do MRTG.
- MaxBytes: parâmetro de limite máximo para geração do gráfico.

- Opções: parâmetro para indicar opções de geração de gráfico.
- Unscale: gerar gráfico sem escala.
- Supress: parâmetro para suprimir a exibição de algum sub-tipo de gráfico.
- XSize: tamanho para o eixo X.
- YSize: tamanho para o eixo Y.
- ShortLegend: legenda simplificada para o gráfico.

Interfaces: Tabela para armazenar as Interfaces dos Objetos Gerenciáveis. Esta tabela armazena várias informações sobre as interfaces dos objetos gerenciáveis. Algumas características como geração de gráficos de utilização (MRTG) e monitoração de estado da interface (Mon) podem ser ativados para cada interface. Algumas informações desta tabela são obtidas pela consulta SNMP ao objeto associado. Foi criado um campo para uma descrição mais detalhada da interface, uma vez que alguns equipamentos não permitem mudar a descrição do OID *interface.ifEntry.ifDescr.x* na sua MIB.

Atributos:

- Número: número da interface na MIB, informação obtida apartir da MIB-II *interface.ifEntry.ifIndex.x.*
- Nome: descrição do que está ligado nesta interface.
- Descrição: descrição da interface na MIB, informação obtida apartir da MIB-II *interface.ifEntry.ifDescr.x.*
- Visualização: identificador para visualização na lista de interfaces.
- Velocidade: velocidade de operação da interface, informação obtida apartir da MIB-II *interface.ifEntry.ifSpeed*.x.
- Monitor: identificador para monitorar o estado da interface que pode ser *UP* ou *DOWN*.
- Gráfico: identificador para geração de gráficos de desempenho da interface utilizando-se os OIDs *interface.ifEntry.ifInOctets.x* e *interface.ifEntry.ifOutOctets.x*
- Perfil do Gráfico: tipo de gráfico que será utilizado para geração dos gráficos estatísticos.

Histórico: Tabela para armazenamento dos históricos de alertas ocorridos.
Armazena os eventos ocorridos com os objetos gerenciáveis para futuras consultas e relatórios.

Atributos:

- Tipo de Alerta: tipo de alerta utilizado.
- Horário: horário em que ocorreu o alerta
- Descrição: descrição detalhada do alerta.
- Pendência: flag para identificação de pendências.

Alerta: Tabela de tipos de alerta. Os tipos de alertas especificam como um evento será informado ao usuário, por e-mail, pop-up de tela, pager, etc.

Atributos:

- Descrição: descrição do tipo de alerta.
- Mensagem: mensagem padrão que será enviada.
- Alertas: especifica os tipos de meios que serão utilizados para enviar os alertas: e-mail, pager ou pop-up.
- Nível: indica o nível de severidade do alerta: *ready*, *warning* ou *critical*.
- E-mails: Endereços de e-mails para envio do alerta.
- Pop-up: Endereços das máquinas para envio de pop-up.
- Pager: Endereços de Pager para envio de mensagem.

Serviço: Tabela de serviços a serem monitorados. Especifica os serviços que serão monitorados em um objeto gerenciável. São especificados o tipo de serviço, sua periodicidade, seus alertas para falha e retorno, e parâmetros adicionais. Estas informações serão utilizadas pela aplicação para gerar os scripts para o software Mon.

Atributos:

- Tipo: tipo de monitor de serviço.
- Descrição: descrição do serviço a ser monitorado.
- Parâmetros: Parâmetros adicional.
- Intervalo: intervalo de verificação do estado dos serviços.
- Período1: Período de Verificação-1

- Falha1: tipo de alerta a ser enviado em caso de problemas no serviço.
- Retorno1: tipo de alerta quando o serviço retornar ao estado normal.
- Tempo1: intervalo de envio de mensagens do Alerta1.
- Período2: Período de Verificação-1
- Falha2: tipo de alerta a ser enviado em caso de problemas no serviço.
- Retorno2: tipo de alerta quando o serviço retornar ao estado normal.
- Tempo2: intervalo de envio de mensagens do Alerta2.

Monitor: Tabela de especificação de Monitores de Serviços. Especifica os monitores de serviço, qual a prioridade que serão executados, e os arquivos e parâmetros necessário para a geração do script para monitoração.

Atributos:

- Tipo: tipo de monitor de serviço que será utilizado. Atualmente são suportados os seguintes monitores: *ping*, *dns*, *http*, *ftp*, *pop3*, *smtp*, *tcp por*t, *smb* e *interface*.
- Prioridade: prioridade na execução dos monitores. Podem ser utilizado mais de um monitor ao mesmo tempo para verificar os serviços de um servidor.
- Script: script de execução do monitor.
- Parâmetros: parâmetros adicionais do monitor.

4.2.2 Interface e Controle da Aplicação (Apache e PHP)

O protótipo foi desenvolvido apartir da linguagem de processamento de Scripts PHP 4.0 o qual é executado sob a plataforma do servidor WEB Apache 2.0.

O PHP foi escolhido por proporcionar uma interface simples com o servidor WEB Apache, possuir uma a API necessária ao desenvolvimento do protótipo com funções para banco de dados e SNMP, além de ser de fácil aprendizado.

A interface com a aplicação esta dividida em dois módulos: um módulo de Apresentação e um módulo de Manutenção.

Visão geral sobre o PHP O PHP é uma linguagem que permite a criação de sites WEB dinâmicos, possibilitando uma interação com o usuário através de formulários, parâmetros de URL e links. A diferença do PHP com relação a linguagens semelhantes a Javascript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas o HTML puro. Desta maneira é possível interagir com banco de dados e aplicações existentes no servidor, não expondo assim o código fonte. Isto pode ser útil quando o programa está lidando informações confidenciais.

O que diferencia o PHP de um script CGI escrito em C ou Perl é que o código PHP fica embutido no próprio HTML, enquanto no outro caso é necessário que o

script CGI gere todo o código HTML, ou leia de um outro arquivo. Basicamente, qualquer coisa que pode ser feita por algum programa CGI pode ser feito também com o PHP, como coletar dados de um formulário, gerar páginas dinamicamente ou enviar e receber cookies.

O PHP possui suporte a um grande número de banco de dados e também tem suporte a outros serviços através de protocolos como o IMAP, SNMP, NNTP, POP3, além do HTTP.

A linguagem PHP foi concebida durante o outono de 1994 por Rasmus Lerdorf. As primeiras versões não foram disponibilizadas, tendo sido utilizadas em sua home-page apenas para que ele pudesse ter informações sobre as visitas que estavam sendo feitas. A primeira versão utilizada por outras pessoas foi disponibilizada em 1995, e ficou conhecida como "Personal Home Page Tools" (ferramenta para páginas pessoais).

Em meados de 1996 o interpretador foi reescrito, e ganhou o nome de PHP/FI (onde, FI: Form Interpreter), um pacote que interpretava formulários HTML. Combinando os scripts do pacote Personal Home Page com o FI e adicionando suporte a mSQL, foi criado o PHP/FI.

Em 1997 houveram mudanças no desenvolvimento do PHP. O interpretador foi reescrito por Zeev Suraskin e Andi Gutmans, e este novo interpretador foi a base para a versão 3. O lançamento do PHP4, em maio de 2000, trouxe muitas novidades aos programadores de PHP. Uma das principais foi o suporte a sessões. Além das mudanças referentes a sintaxe e novos recursos de programação, o PHP4 trouxe como novidade um otimizador Zend, que permite a execução muito mais rápida de scripts PHP.

Módulo de Apresentação O módulo da apresentação é responsável pela apresentação de todos os Objetos Gerenciáveis e suas respectivas interfaces,
métricas e serviços. Neste módulo também podem ser verificados os Históricos de Alertas e verificados os estados dos serviços de rede.

Apresentação do Gerenciamento de Performance A visualização dos Objetos Gerenciáveis (O.G.) (figura 4.4) inicia-se pela apresentação dos tipos de objetos que estão catalogados (servidores, switchs, roteadores, impressoras e outros). Escolhendo-se um dos tipos, é apresentada uma lista com todos os objetos daquele tipo.

Ao escolher um Objeto Gerenciável, o módulo realiza uma consulta via SNMP ao objeto em questão. Estando o objeto ativo, os dados serão apresentados na tela. Se o objeto não estiver ativo, uma mensagem de erro será exibida e os dados apresentados serão obtidos do banco de dados. Os dados apresentados são:

- Nome
- Endereço IP
- Localização
- Contato
- Tempo desde a última reinicialização (*UpTime*)

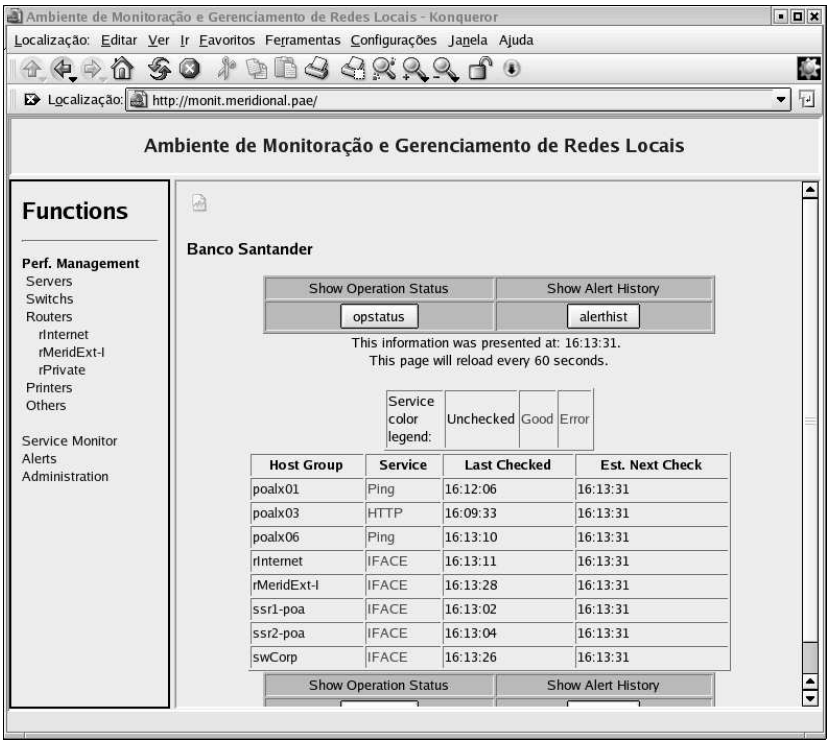

Figura 4.4: Visualização dos Tipos de Objetos.

- Descrição
- URL
- Links para os Interfaces, Métricas e Alertas.

A figura 4.5 mostra um exemplo das informações apresentadas.

A seguir podemos visualizar o estado das interface dos Objetos Gerenciáveis verificando a descrição das interfaces, quais interfaces estão ativas (UP ou DOWN), qual a velocidade de operação, link para o gráfico de utilização (figura 4.5).

Quando as interfaces do objeto são visualizadas, o sistema faz uma consulta via SNMP ao OG para obter as seguintes informações:

- Estado de Operação (*interface.ifTable.ifEntry.ifOperStatus*) que pode ser *UP* ou *DOWN*;
- Estado Administrativo (*interface.ifTable.ifEntry.ifAdminStatus*) que pode ser *UP* ou *DOWN;*
- Velocidade de Operação (*inteface.ifTable.ifEntry.ifSpeed*); e
- Endereço Físico (*interface.ifTable.ifEntry.PhysAddress*)

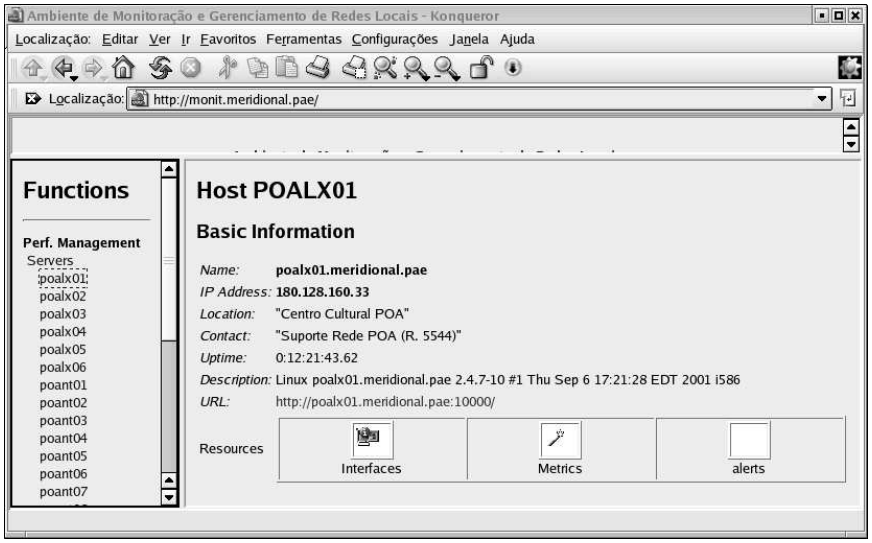

Figura 4.5: Exemplo de Informações de Objetos Gerenciáveis.

O sistema então verifica na base de dados quais as interfaces do OG estão disponíveis para apresentação (parâmetro *Visualização* da tabela de *Interfaces*) e apresenta-as respectivamente. Caso o estado de operação seja "down" o sistema ainda verifica se esta interface está administrativamente "ativa" ou "inativa" (*up* ou *down*).

Na verificação das métricas (figura 4.7, o sistema executa uma consulta SNMP ao Objeto Gerenciável referenciando-se as métricas do objeto, apresentando os valores correntes na tela e as referências para os gráficos das métricas especificadas (figura 4.8).

Por último temos a visualização dos eventos de alertas ocorridos no Objeto Gerenciável. Ao acessar o link, o sistema busca o Histórico dos eventos de alertas no banco de dados e apresenta-os em ordem decrescente de data.

Apresentação do Monitor de Serviços Ao acessar o Monitor de Serviços, o sistema executa um CGI para a apresentação do estado dos serviços monitorados (figura 4.10). Os dados apresentados são Host, Serviço, Horário da última checagem e horário da próxima checagem. Os serviços que estiverem disponíveis serão apresentados em verde, e os serviços que estiverem indisponíveis serão apresentados em vermelho. Clicando no serviço indisponível, podemos verificar qual o monitor que identificou a falha.

Apresentação dos Alertas A visualização dos alertas apresenta o histórico dos eventos ocorridos (figura 4.11). A listagem apresenta qual O.G. que gerou o evento, a data e hora do evento, e qual o tipo de mensagem. Clicando-se no O.G. um *link* leva a página de Apresentação do Objeto Gerenciável. Clicando-se sobre a mensagem de alerta, esta nos apresenta a página das métricas do Objeto Gerenciável. Pode-se nesta página realizar alguns filtros. Podem ser filtrados o nome do O.G., data do evento e Tipo de Alertas, ou uma combinação entre eles.

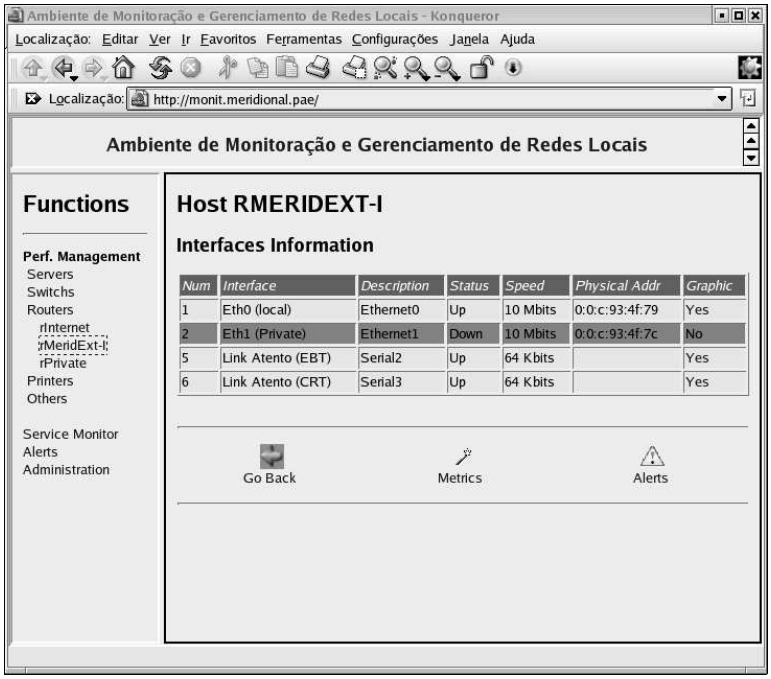

Figura 4.6: Exemplo do estado das Interfaces de um Objeto Gerenciável.

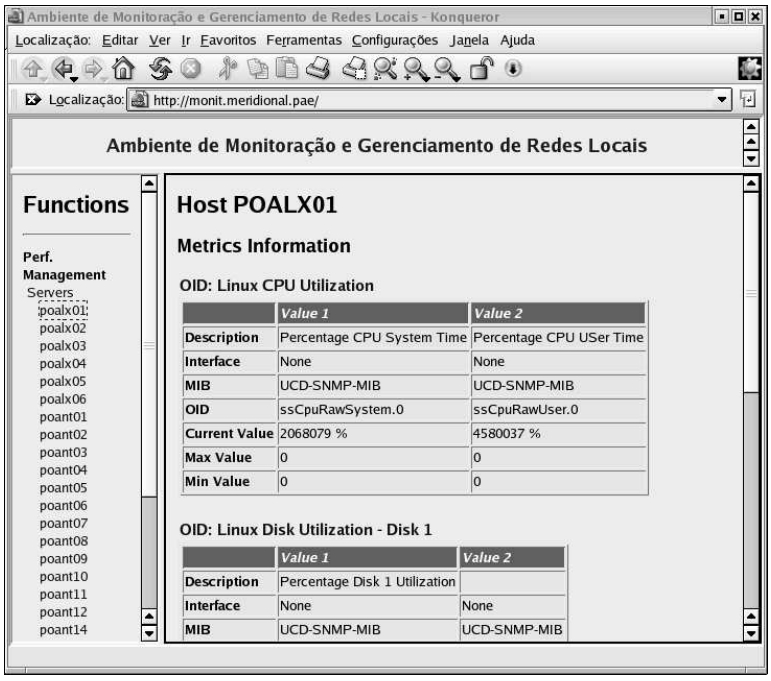

Figura 4.7: Exemplo de Visualização das Métricas dos Objetos Gerenciáveis.

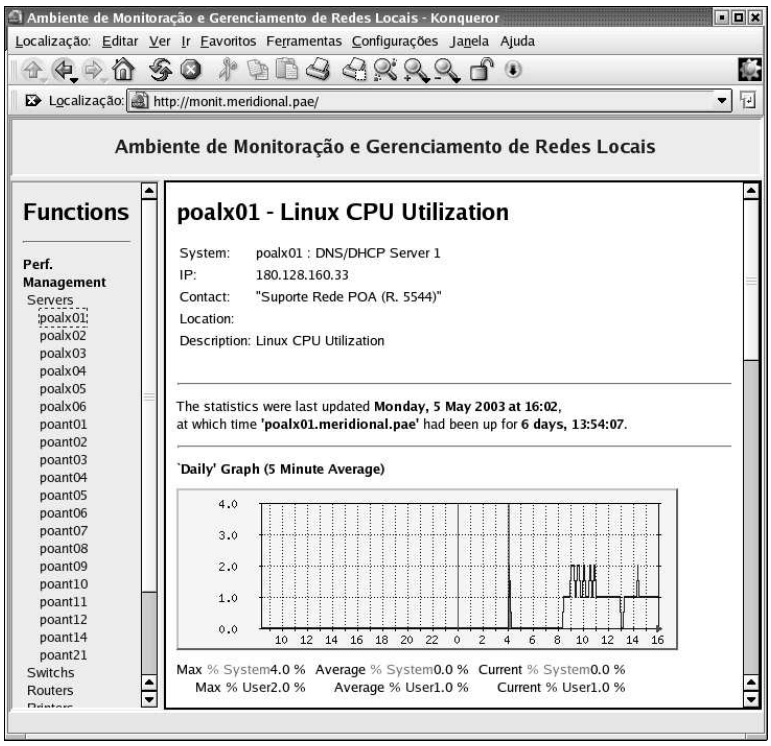

Figura 4.8: Exemplo de Gráfico de Métrica para Utilização de CPU.

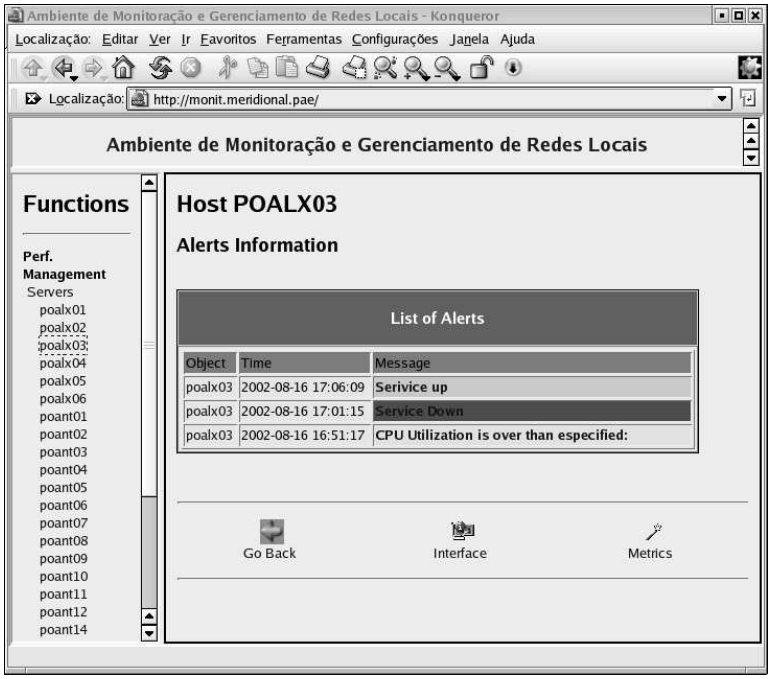

Figura 4.9: Exemplo de Alertas para um Objeto Gerenciável.

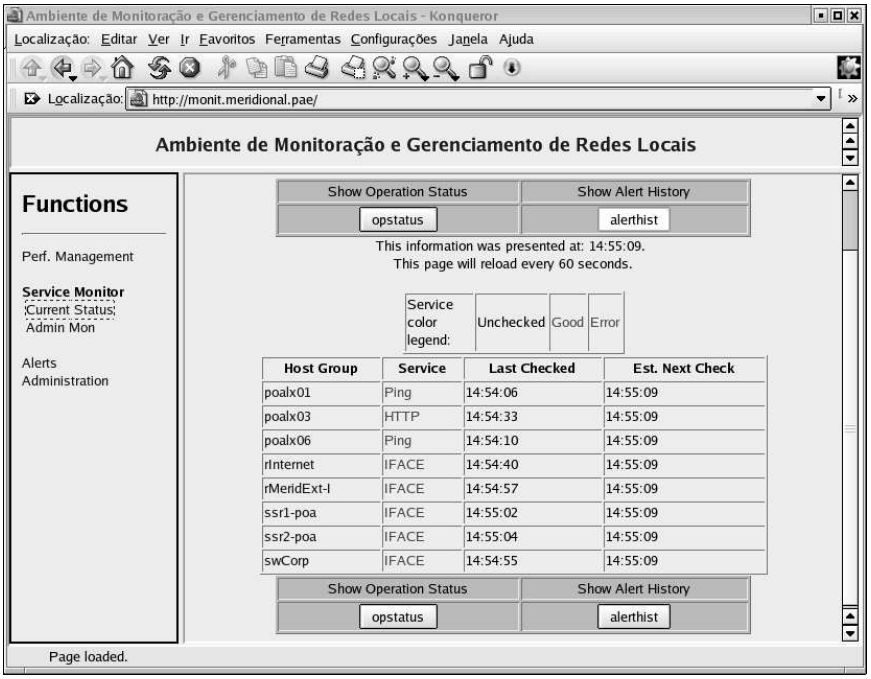

Figura 4.10: Monitor de Serviços.

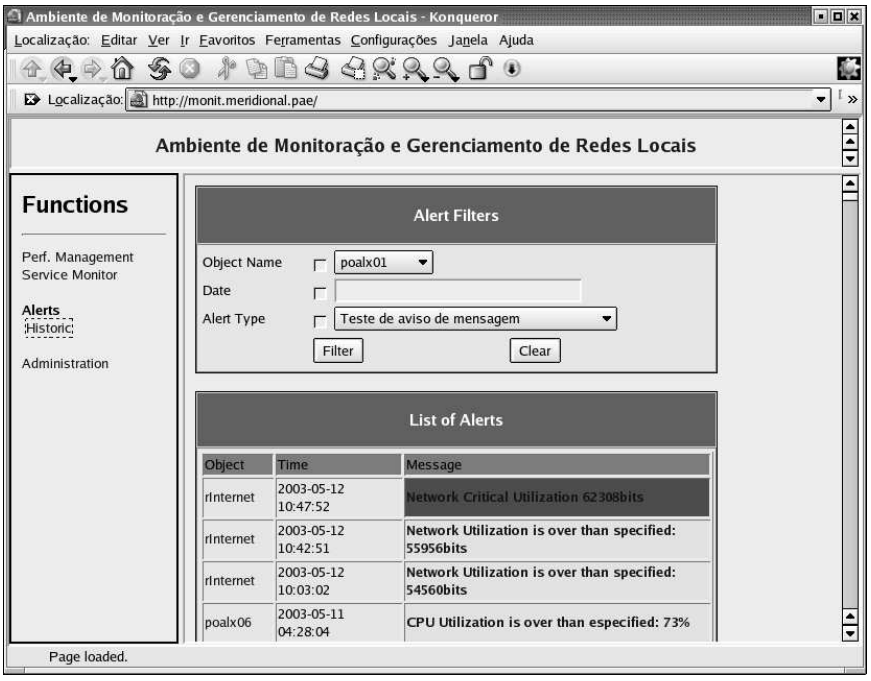

Figura 4.11: Histórico de Alertas.

4.2.2.1 Módulo de Manutenção

O Módulo de Manutenção é responsável por manter toda a configuração do ambiente de gerenciamento.

Manutenção de Objetos Gerenciáveis Este módulo é responsável pela administração dos dados dos Obejtos Gerenciáveis. As funções executadas pelo módulo são:

1. Listagem dos Objetos:
   lista todos os objetos da base de dados.

2. Inserção de um novo objeto:
   busca informações e insere um novo objeto.

3. Alteração do Objeto Gerenciável:
   altera os dados do O.G.

4. Manutenção dos dados das interfaces do Objeto Gerenciável:
   insere ou altera dados das interfaces.

5. Manutenção dos Serviços associados ao Objeto Gerenciável:
   insere ou altera dados dos serviços a serem monitorados.

6. Manutenção das Métricas associadas ao Objeto Gerenciável:
   insere ou altera dados das métricas de gerenciamento.

O gráfico da figura 4.12 apresenta a modelagem das funções da Administração de Objetos.

Listagem dos Objetos Quando a Administração de Objetos é chamada é apresentada a lista de todos os objetos da base de gerenciamento(figura 4.13). São apresentadas as seguintes informações: Nome, Endereço IP, Descrição, Tipo e URL. Clicando-se sobre o nome do O.G. podemos entrar no módulo de alteração dos dados deste objeto. Muitos equipamentos possuem uma interface WEB própria para administração de seus recursos. Podemos acessar este endereço clicando sobre o campo URL do O.G..

Inserção de um novo objeto Para inserir um novo O.G. devemos fornecer o seu hostname e a Community de leitura do SNMP. Caso nosso objeto não esteja catalogado em um DNS, devemos fornecer também o endereço IP do objeto.

Com base nas informações fornecidas, o sistema executa uma consulta SNMP ao Objeto Gerenciável retornando as informações de *sysLocation* e *sysContact* que são gravadas na tabela de objetos, executa também uma consulta às interfaces do objeto extraindo as seguintes informações: *ifNum, ifIndex*, *ifDescr* e *ifSpeed*; armazenando estas informações na tabela de interfaces. Após buscar as informações o sistema apresenta os dados consultados, objeto e interfaces, juntamente com novas informações

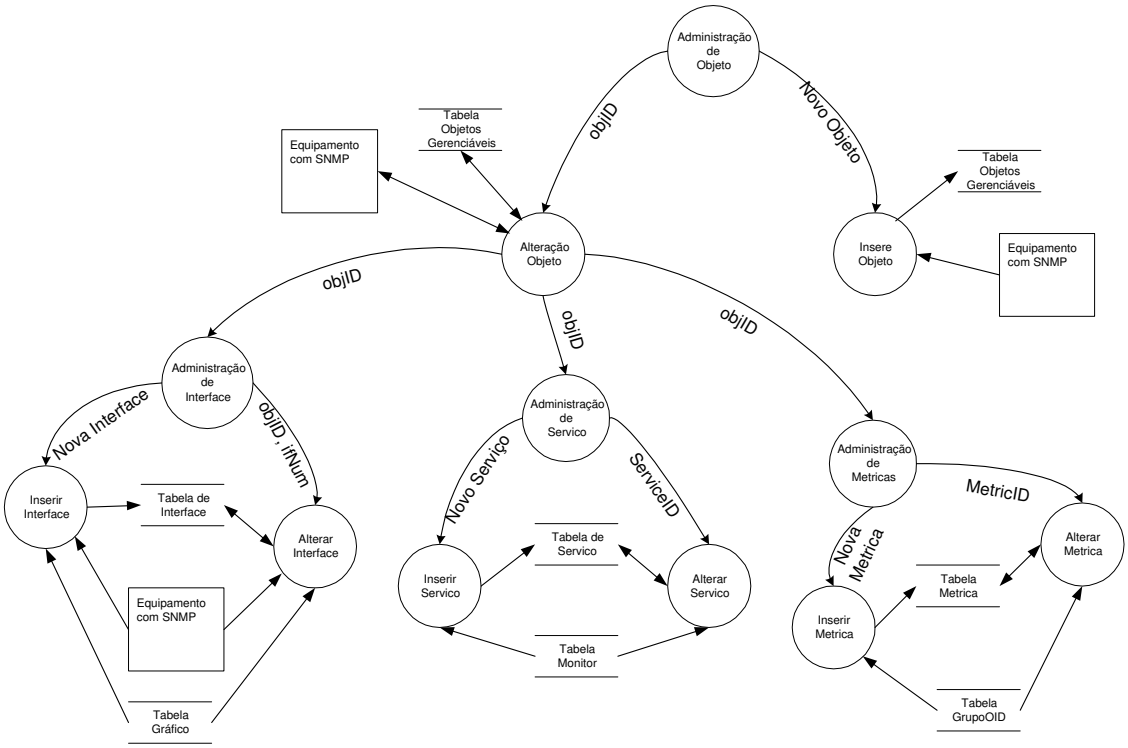

Figura 4.12: Modelagem das Funções de Manutenção de Objetos.

que devem ser registradas no mesmo formulário de Alteração do Objeto Gerenciável (figura 4.14).

O próximo passo é verificar os parâmetros das interfaces, inserir os Serviços a serem monitorados e as Métricas de gerenciamento.

Alteração do Objeto Gerenciável Quando um objeto é inserido ou selecionado da Lista de Objetos (figura 4.13), seus dados são lidos da base de dados e apresentados na tela. As informações da base de dados poderão ser alteradas, e os parâmetros sysLocation e SysContact poderão ser re-sincronizados com o Objeto Gerenciável. A partir deste ponto podemos realizar manutenções nas Interfaces, Métricas e Serviços assoassociadosciados ao Objeto Gerenciável.

Manutenção dos dados das interfaces do Objeto Gerenciável A manutenção das interfaces consiste em manter as informações adicionais das interfaces dos Objetos Gerenciáveis.

A figura 4.15 apresenta a tela para alteração dos dados da Interface.

As informações que podem ser alteradas são:

- Nome (*Name*): nome identificativo da interface, não é o *ifDescr* da MIB, porque alguns equipamentos não permitem que seja alterada.
- Apresentação (*Show*): indica se a interface será listada quando for visualizado o

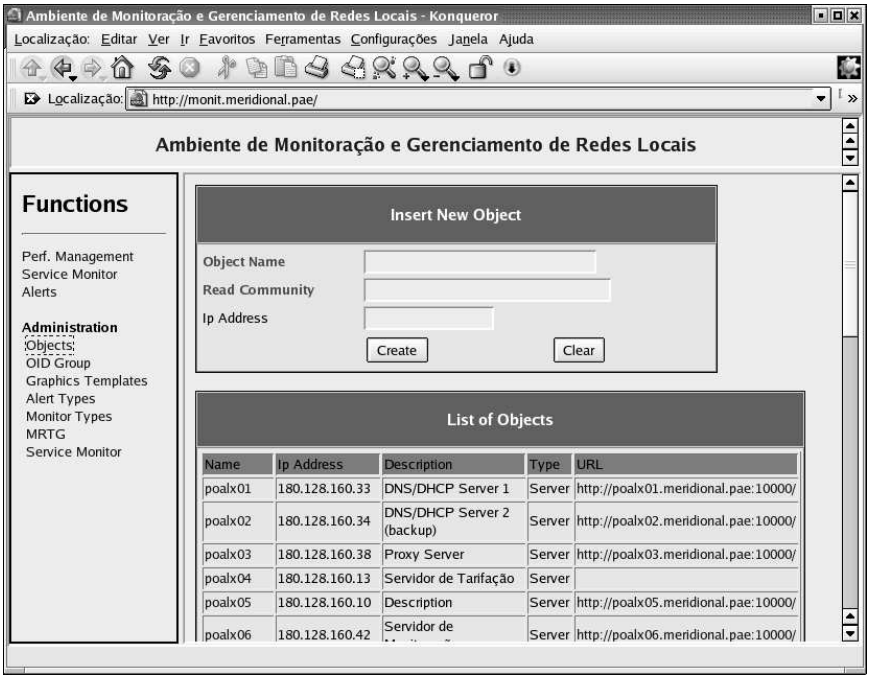

Figura 4.13: Administração de Objetos Gerenciáveis.

Objeto Gerenciável no módulo de Apresentação de Objetos.

- Monitoração (*Monitoring*): habilita a monitoração do estado da interface (*UP* ou *DOWN*) realizado pelo módulo de Monitoração de Serviços.

- Gráfico (*Grafics*): habilita a geração de gráficos estatísticos de utilização que será realizado pelos módulo de Coleta de Dados e Estatísticas.

- Tipo de Gráfico (Graphic Skill): seleciona qual o tipo de gráfico será gerado pelo módulo de Estatísticas.

Manutenção dos Serviços associados ao Objeto Gerenciável Os serviços de rede que serão monitorados pelo Mon são mantidos por este módulo. Quando um objeto é criado não existe nenhum serviço associado. Os serviços podem ser agregados aos Objeto conforme as especificações dos Monitores de Serviços disponíveis (ver. Tipos de Monitores).

A figura 4.16 apresenta os parâmetros que devem ser informados para a criação de um serviço a ser monitorado.

Os parâmetros a serem informados são:

- Tipo de Monitor: tipo de monitor de serviço que será executado.

- Descrição: descrição do serviço que será monitorado.

Figura 4.14: Alteração de Dados dos Objetos Gerenciáveis.

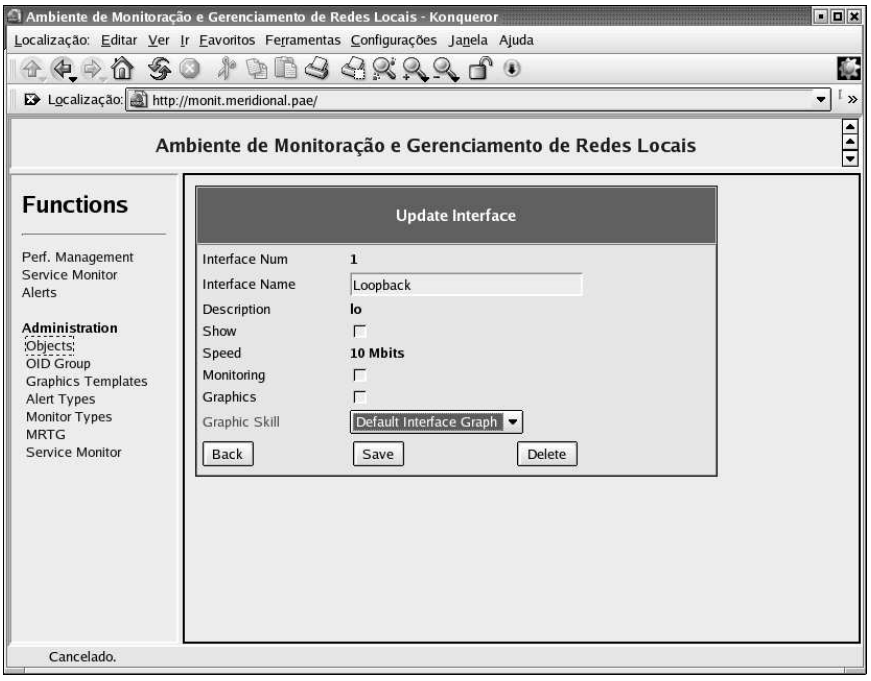

Figura 4.15: Alteração dos dados da Interface.

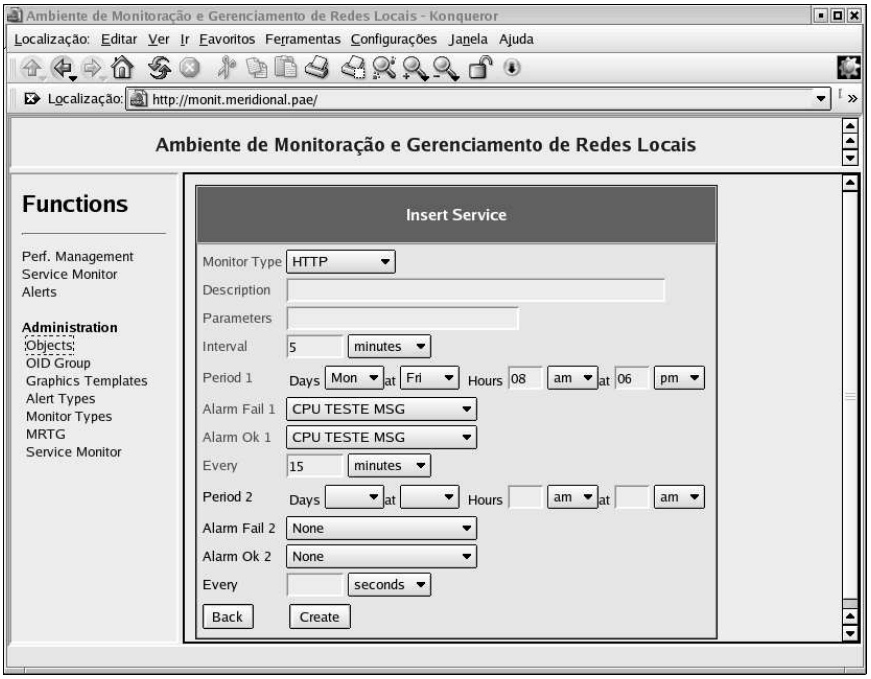

Figura 4.16: Inserção de Serviço a ser monitorado.

- Parâmetros: parâmetros adicionais a serem passados para os monitores de serviço. Para cada monitor podemos ter parâmetros diferentes.

- Intervalo: especifica qual o intervalo de verificação do serviço.

- Período: especifica qual o período de tempo que o serviço será monitorado.

- Alerta de Falha: especifica qual o alerta que será gerado em caso de falha.

- Alerta de Retorno: especifica qual o alerta que será gerado para o retorno a normalidade do serviço.

- Intervalo de Alerta: intervalo de tempo em que será enviado os alertas de indisponibilidade do serviço.

Podem ser especificados até dois períodos diferentes para monitoração dos serviços. A figura 4.16 mostra um exemplo de serviço WEB (HTTP) que será monitorado a cada 5 minutos, num período de segunda à sexta-feira dás 8:00h às 18:00 horas. Em caso de indisponibilidade do serviço HTTP um alerta do tipo *Service Down* será gerado a cada 15 minutos até o retorno a normalidade, onde será gerado um alerta do tipo *Service Up*.

Manutenção das Métricas associadas ao Objeto Gerenciável Este módulo administra as métricas de gerenciamento para os Objetos Gerenciáveis. Assim como os serviços, ao ser criado um novo objeto, não é associada nenhuma métrica a este. As métricas podem ser associadas aos objetos informando os seguintes parâmetros:

- Grupo de OID: especifica qual o grupo de OID que será utilizado para gerar a métrica (ver Adm. de Grupos OID).
- Perfil de Gráfico: especifica qual o perfil de gráfico será utilizado para a geração dos gráficos.
- Interface: número da interface do objeto a qual a métrica está se referindo, caso não exista uma interface associada este parâmetro receberá o valor *None* (ou zero).
- Valor Máximo: especifica qual o maior valor possível a ser aceito pela métrica.
- Limites: especifica se haverá verificação de limites para a métrica.

Caso exista verificação de limites, os parâmetros a seguir são obrigatórios:

- Valor máximo: Valor para o limite máximo, se a métrica ultrapassar este valor será gerado um alerta.
- Valor Mínimo: Valor para o limite mínimo, se a métrica ultrapassar este valor será gerado um alerta.
- Tipo de Alerta: tipo de a ser enviado no caso de excesso no limite especificado.

A figura 4.17 apresenta uma associação de métrica para verificação de carga de processamento em CPU de servidores Linux.

Administração de Grupo OID Este protótipo foi desenvolvido utilizando agrupamento de métricas para até dois OIDs. Esta característica é em função do aplicativo MRTG utilizar até dois OIDs para suas coletas de dados.

A administração de Grupos de OID consistem em especificar grupos de Object ID. Estes Grupos serão utilizados para a coleta de dados realizada pelo MRTG, sua verificação de Limites e geração de Gráficos.

A criação de um grupo de OID consiste em especificar os seguintes parâmetros:

- Descrição: especifica uma descrição para o grupo de OIDs.
- Valores: especifica os Object IDs da MIB-II que serão utilizados nas métricas.
- Descrição do OID: descreve o significado do OID.
- Legenda: descreve uma legenda abreviada para ser utilizada na geração dos gráficos estatísticos.

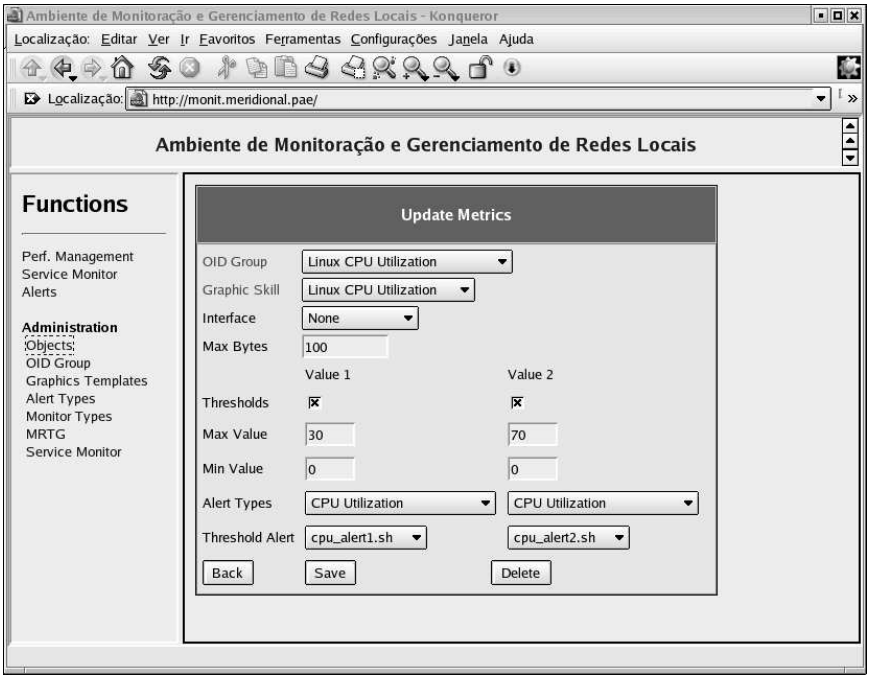

Figura 4.17: Associação de Métrica para Objeto Gerenciável.

- Arquivo da MIB: especifica em qual arquivo da MIB está o Object ID especificado.

A figura 4.18 mostra um exemplo de grupo OID especificando métricas de gerenciamento para carga de CPU de servidores Linux:

Administração de Perfil de Gráficos Este módulo tem por objetivo administrar os parâmetros dos gráficos que serão gerados pelo MRTG.

Podem ser especificados vários perfis de gráfico para serem utilizados com as métricas. Cada métrica irá se referir a um único perfil.

Administração dos Tipos de Alertas Especifica os tipos de Alertas que poderão ser enviados em caso de algum evento ocorrer (excesso de Limite, serviço indisponível, etc).

Os parâmetros que devem ser especificados são:

- Descrição: descreve o tipo de Alerta.
- Mensagem: especifica qual a mensagem que será enviada ao administrador da rede em caso de alerta.
- Tipo de Alerta: especifica a severidade do alerta. Pode ser *Ready* (quando o serviço está disponível), *Warning* (aviso de algum problema), ou *Critical* (quando ocorrer algum evento severo).

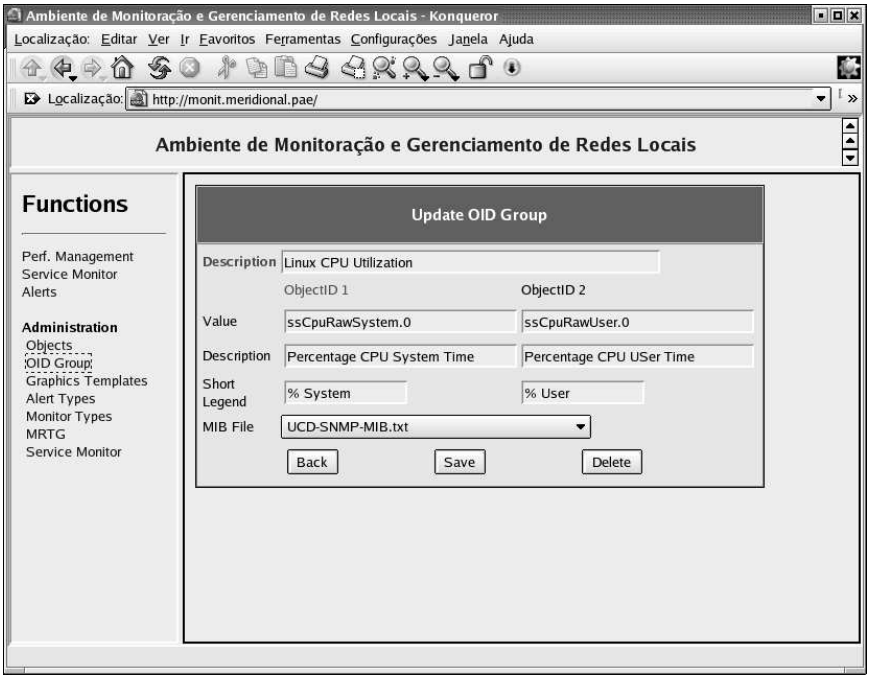

Figura 4.18: Administração de Grupos de OID.

- Destinatários: especifica para onde será enviado os alertas. Pode ser:
  via e-mail - envia uma mensagem por correio-eletrônico, deve-se especificar os endereços eletrônicos dos destinatários;
  Popup - envia uma mensagem para a tela do gerente de rede, deve-se especificar as estações da rede que receberão o aviso; e
  Pager - envia uma mensagem para equipamentos Pager, deve-se especificar o endereço do Pager de destino.

Administração de Tipos de Monitores Executa a administração dos tipos de monitores que serão utilizados na monitoração de serviços. Os tipos de Monitores especificam o tipo de serviço que será monitorado, o arquivo de script que será utilizado pelo aplicativo Mon, a prioridade na monitoração dos serviços e parâmetros adicionais. Um Objeto Gerenciável pode pode ter vários monitores associados, o parâmetro prioridade irá indicar qual a sequência de monitores será utilizada para monitorar este equipamento. Quanto menor a prioridade, maior a precedência de execução.

4.2.3 MRTG para coleta de Dados e Estatísticas

4.2.3.1 Visão geral sobre o MRTG

O MRTG é uma ferramenta utilizada para monitorar a carga de tráfego em links de rede. O MRTG gera páginas HTML com imagens GIF que representam visualmente este tráfego. O MRTG é baseado em PERL e C, rodando em Unix e Windows NT.

O MRTG é totalmente livre sobre os termos de licenciamento da GNU - General Public License.

Criado em 1994 por Tobias Oetiker <oetiker@ee.ethz.ch> no intuito de gerenciar um link de dados de 64 kilobits/s, um pequeno script escrito em linguagem PERL foi desenvolvido para tal função. Em 1996, com a contribuição
de Dave Rand <dlr@bungi.com> para melhorar a sua performance, algumas rotinas em C foram inseridas dando por iniciada a versão 2 do MRTG.

Consistindo de um script em PERL, o MRTG utiliza o SNMP para ler as informações de tráfego das interfaces dos roteadores, e um programa em C grava os dados e gera os gráficos correspondentes. Estes gráficos são diagramados em páginas WEB que podem ser visualizadas por qualquer navegador WEB moderno.

O MRTG cria quatro representações do tráfego, um gráfico diário, com as amostras dos últimos cinco minutos; um gráfico semanal, com as médias das amostras de 30 minutos; um gráfico mensal, com as médias das duas horas; e um gráfico anual, com as médias de um dia. O log mantém todos os dados relevantes dos últimos dois anos.

O MRTG não está limitado a monitorar somente o tráfego de rede, ele é capaz de monitorar qualquer variável SNMP. Pode-se ainda utilizar programas externos para obter dados que serão monitorados pelo MRTG.

Adicionalmente a função de geração de gráficos, podemos definir limites máximos e mínimos para cada variável monitorada. Quando um dado obtido estiver abaixo ou acima do especificado, um programa externo pode ser acionado, por exemplo enviar um e-mail avisando que um determinado link está sobrecarregado.

Recursos do MRTG

- Portável: o MRTG roda em plataforma Windows e Unix.

- Código Fonte: rodando através de um interpretador PERL, seu código fonte está aberto.

- SNMP: utilizando a biblioteca de SNMP do PERL, não há necessidade de instalar outro pacote SNMP. Atualmente suporta a versão SNMPv2c.

- Identificação de Interfaces: as interfaces dos equipamentos podem ser identificadas através de seu endereço IP, Descrição ou endereço Ethernet, além da identificação normal do número da interface.

- Configuração Automática: o MRTG possui algumas ferramentas que tornam a sua configuração mais simples.

- Performance: rotinas mais críticas foram escritas em C melhorando seu desempenho.

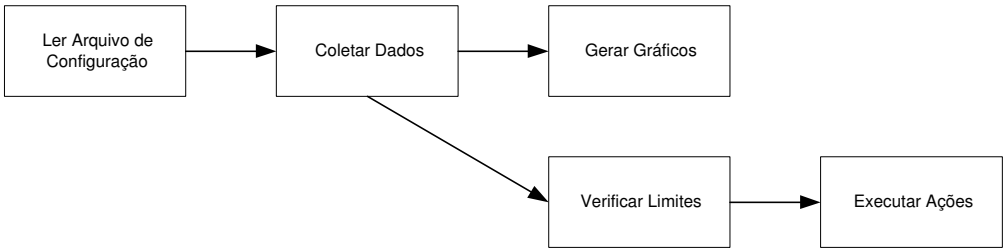

Figura 4.19: Funcionamento do MRTG

Funcionamento do MRTG

Para a execução do MRTG é necessário criar um arquivo com as Métricas de Gerenciamento e alguns outros parâmetros de configuração. O MRTG pode ser executado apartir da linha de comando. Ele irá coletar os dados, gerar os respectivos gráficos e verificar os limites, se programado para isto. Para implementar a coleta dos dados de forma periódica, pode-se utilizar do agendador de tarefas do sistema (*crontab* no Unix). Desta forma os dados serão coletados em intervalos determinados.

O MRTG pode ser executado como serviço (ou modo *daemon*). O propósito deste modo é que o MRTG é iniciado uma única vez e não repetidas vezes como no modo de linha de comando. Isto minimiza a utilização dos recursos computacionais na carga e interpretação dos arquivos de configuração. Porém é necessário reinicializar o processo para ativar as mudanças realizadas nos arquivos de configurações.

Parametrização

O arquivo de configuração do MRTG é composto de alguns parâmetros obrigatórios e outros opcionais. Os arquivos de configurações do MRTG estão organizados em três: Parâmetros Globais, Lista de Arquivos de Objetos e Arquivo de Configuração para Objeto Gerenciável.

Parâmetros Globais O arquivo de configuração dos Parâmetros Globais será utilizado por todos os outros arquivos de configuração. Os parâmetro Globais utilizados são:

- HtmlDir: especifica o diretório onde as páginas HTML serão armazenadas.
- ImageDir: especifica o diretório onde as imagens serão armazenadas. Devem ficar abaixo do diretório especificado em *HtmlDir*.
- LogDir: especifica o diretório onde os logs dos dados coletados serão armazenados.
- ThreshDir: especifica o diretório onde os limites (thresholds) serão armazenados.

- Forks: número de sub-processos para aquisição dos dados via SNMP. Em ambientes cuja a latência dos dados é relativamente grande, este parâmetro auxilia no desempenho geral da aplicação.

- Refresh: especifica em quantos segundos o *browser* será instruído a realizar uma atualização da página.

- Interval: especifica qual o período em que o MRTG será executado.

- Options [_]: especifica opções globais para a geração dos gráficos.

Lista de Arquivos de Objetos Gerenciáveis Este arquivo contém alguns parâmetros extras além da relação de todos os arquivos dos Objetos Gerenciáveis que serão monitorados. Os parâmetros utilizados neste arquivo são:

- RunAsDaemon: especifica que o MRTG será executado no modo *daemon*.

- Include: especificam os arquivos que serão incluídos na monitoração.

Este é o arquivo base para toda a monitoração realizada pelo MRTG.

Arquivo de Configuração para Objeto Gerenciável Para cada Objeto Gerenciável que for monitorado pelo MRTG será criado um arquivo de configuração com as respectivas Métricas de Gerenciamento. No início deste arquivo é incluído o arquivo de definições globais, de forma que este arquivo possa ser executado independentemente da Lista de Arquivos de Objetos Gerenciáveis. Isto permite que o Objeto seja testado antes de ser incluso na lista de objetos gerenciados.

Este arquivo está dividido em duas seções. A primeira seção especifica os parâmetros para as métricas de monitoração de desempenho das interfaces do Objeto Gerenciável. E a segunda secção especifica os parâmetros para as demais Métricas que serão gerenciadas pelo MRTG.

Os parâmetros para as Métricas de Desempenho das Interfaces são:

- Target: especifica um nome para a Métrica de gerenciamento.

- XSize: especifica o tamanho do eixo X do gráfico de desempenho.

- YSize: especifica o tamanho do eixo Y do gráfico de desempenho.

- MaxBytes: especifica o valor máximo para a Métrica de gerenciamento, será utilizado como referência para cálculo de percentual.

- Options: especifica opções extras para a Métrica.

- Title: especifica o título da página HTML gerada para a Métrica.

- PageTop: especifica as informações que irão aparecer no topo da página gerada. As informações apresentadas são: Nome do Objeto, Endereço IP, Nome para Contato, Localização, Descrição da Interface e Velocidade.

Para a segunda secção, demais Métricas, temos os seguintes parâmetros:

- Target, XSize, YSize, MaxBytes, Options e Title: seguem a mesma descrição dos parâmetros para Métricas de Desempenho de Interfaces.
- ShortLegend: unidade de medida para o gráfico (por exemplo: bits/s)
- Supress: Por default o MRTG gera quatro tipos de gráficos (diário, mensal, semanal ou anual). Com esta opção podemos suprimir a geração de alguns destes gráficos.
- Unscale: Por default cada gráfico é escalado verticalmente de forma a tornar visível os dados que sejam bem menores que o valor máximo especificado em MaxBytes. Com este parâmetro podemos desabilitar a escala para o um tipo de gráfico específico (diário, mensal, semanal ou anual).
- YLegend: Legenda do eixo Y do gráfico.
- Legend1 e Legend 2: Descrição para os dois OIDs da Métrica de Gerenciamento.
- LegendI e LegendO: Descrição resumida para os dois OIDs da Métrica de Gerenciamento.
- ThresMaxI e ThreshMaxO: valores máximos para os dois OIDs da Métrica de Gerenciamento.
- ThresMinI e ThresMinI: Valores mínimos para os dois OIDs da Métrica de Gerenciamento.
- ThresProgI e ThresProgO: scripts que serão acionados para o caso de ocorrerem excessos de limites.
- PageTop: especifica as informações que irão aparecer no topo da página gerada. As informações apresentadas são: Nome do Objeto, Endereço IP, Nome para Contato, Localização, Descrição da Métrica.

4.2.3.2 Integração do MRTG

Após a análise dos parâmetros e funcionamento do MRTG, ele foi integrado o ambiente utilizando-se scripts em PHP que executam funções de Manutenção de Parâmetros Globais, Geração de Scripts de Objetos, Geração da Lista de Objetos, e reinicialização do MRTG.

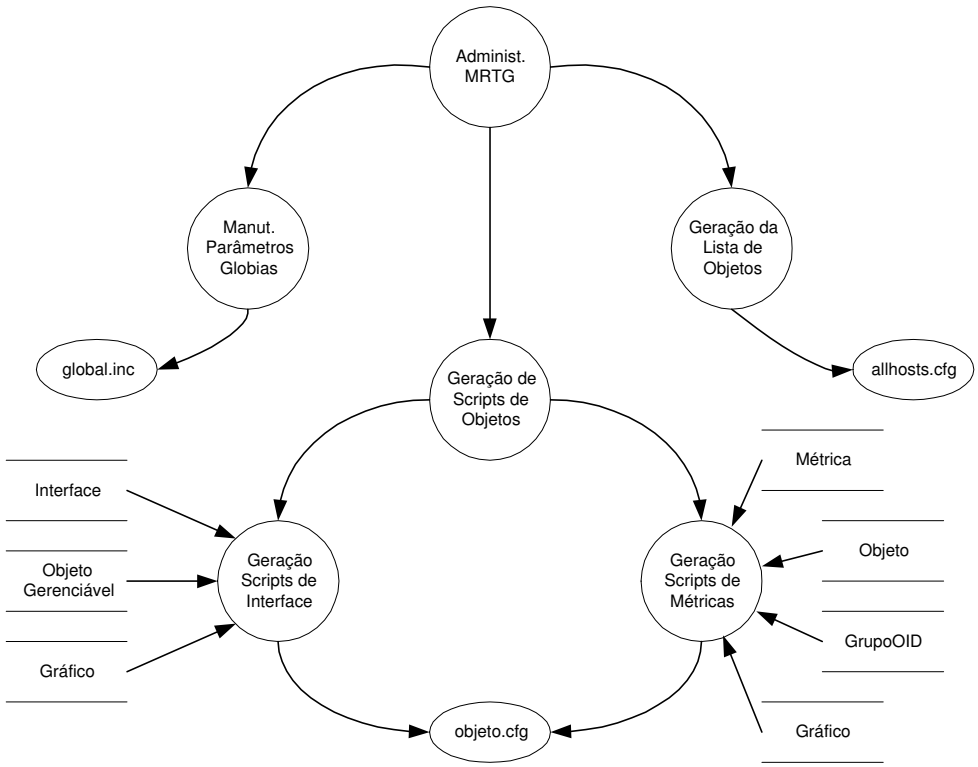

Figura 4.20: Administração do Módulo do MRTG.

Estrutura de Diretórios do MRTG O aplicativo MRTG foi instalado segundo suas instruções normais. Foi criado uma estrutura de diretórios para armazenar os arquivos de configuração e logs do MRTG. Os subdiretórios são:

- conf: diretório para armazenar os arquivos de configuração gerados pelos scripts.
- logs: diretório para armazenar os logs do MRTG. Os Logs do MRTG são os arquivos com os dados coletados periodicamente.
- thresholds: diretório para armazenagem dos scripts de alerta para as condições de excesso de limites.

Função de Manutenção de Parâmetros Globais Na função de Manutenção de Parâmetros Globais são definidos pelo usuário as informações para a execução do MRTG. Estas informações após serem atualizadas são salvas no arquivo "global.inc" no diretório de configurações do MRTG. As informações são:

- Diretório HTML: especifica o caminho completo do diretório onde as páginas geradas serão armazenadas.
- Diretório Imagens: especifica o caminho completo do diretório onde as imagens das páginas geradas serão armazenadas. Deve ficar em um diretório abaixo do especificado para o HTML.

- Diretório de Logs: especifica o caminho completo para o diretório onde os Logs serão armazenados.
- Diretório de Scripts de *Thresholds*: especifica o caminho completo para o diretório onde os Scripts de Thresholds serão armazenados.
- Número de Sub-Processos: número de sub-processos que serão executados para a coleta e geração dos gráficos das Métricas, quando executado em modo *daemon.*
- Tempo de Refresh: tempo de atualização para as páginas HTML geradas.
- Intervalo de Execução: periodicidade no qual os dados serão coletados.

Função de Geração de Scripts de Objetos Esta função realiza a geração dos scripts que serão executados pelo MRTG. Este procedimento se divide em dois sub-processos no qual o primeiro executa a geração dos parâmetros para monitoração das interfaces e o segundo executa a geração dos parâmetros para monitoração das métricas relacionadas.

O sistema apresenta uma lista com todos os Objetos Gerenciáveis, o administrador da rede seleciona para qual, ou quais, objetos será realizado a geração do script. A geração do script começa criando um arquivos com o nome do objeto onde serão gravados os parâmetros. Selecionando em cada Objeto Gerenciável as interfaces a serem monitoradas, gera as respectivas entradas no arquivo de script. Após ler todas as interfaces, faz uma seleção nas Métricas do Objeto Gerenciável e grava também as respectivas entradas no arquivo de script. Repete o processo para todos os objetos selecionados da lista.

Função de Geração de Lista de Objetos Esta função permite inserir ou retirar da lista de objetos (arquivo allhosts.cfg) aqueles que devem ou não ser monitorados. Quando acionado, o sistema busca a lista de todos os Objetos Gerenciáveis e verifica quais objetos estão na lista de monitoração. Permite que o administrador da rede inclua ou exclua objetos desta lista. Ao salvar, o sistema gera o arquivo com os objetos selecionados.

Função de Controle de Processo Este procedimento permite reinicializar o processo do MRTG apartir do próprio ambiente. Ele executa uma chamada a um script que verifica qual o processo do MRTG, encerra-o e inicializa outro processo. Devido ao uso do modo daemon, o MRTG precisa ser encerrado para que as novas configurações sejam carregadas. O MRTG lê o arquivo *allhosts.cfg* e reprocessa os scripts de todos os objetos que compõem a lista.

4.2.4 MON para Monitoração de Serviços

O Mon é um sistema de monitoração de serviços de propósito geral escrito inteiramente em PERL, que pode ser utilizado para monitorar links de rede, disponibilidade de servidores, condições ambientais e muito mais.

A monitoração de recursos deve ser vista como duas atividades separadas: o teste da condição, e o tipo de ação a ser tomada após uma falha. O mon foi projetado de forma a deixar o teste e a ação como tarefas separadas. O mon implementa um escalonador que executa os Monitores (testes), e dispara os Alertas apropriados em caso de falha do Monitor.

Monitores e Alertas não são partes internas do mon. Isto significa que se um novo serviço precisa ser monitorado, ou um novo tipo de alerta é necessário, o mon não precisa ser alterado.

4.2.4.1 Monitores de Serviços Inclusos

- Ping: verifica a disponibilidade de um servidor através de um "ICMP echo request (Ping)".
- SNMP: A conectividade ao SNMP é testada conectando-se na porta SMTP (nº 25) e esperando por um código de retorno "220" como resposta, enviando um "Quit" e esperando a resposta "221", e então encerrando a conexão.
- Telnet: determina se o serviço está ativo aguardando o "Prompt" do servidor.
- FTP: verifica se o serviço está disponível conectando-se a porta 21, e enviando um "Quit" e aguardando a resposta apropriada.
- NNTP: similar ao SMTP, porém utilizando-se o protocolo NNTP.
- HTTP: verifica o serviço utilizando o protocolo HTTP para obter uma página do servidor WEB. O monitor do http pode mensurar o tempo de latência na entrega da página HTML e reportar falhas se a velocidade de transferência mínima não for atingida.
- POP-3: similar ao SMTP, porém utilizando o protocolo POP3.
- IMAP: similar ao SMTP, porém utilizando o protocolo IMAP.
- Serviços TCP: um serviço que utiliza protocolo TCP pode ser monitorado. Ele abre e fecha uma conexão TCP com o servidor, se não conseguir realizar a conexão, retornará uma condição de erro.
- Espaço em Disco: o espaço em disco pode ser monitorado através de NFS ou "mount points" locais.
- HP Printers via SNMP: utilizando-se o SNMP, impressoras HP com interface JetDirect podem detectar condições de erro da impressora como falta de papel, atolamento de papel, problemas com toner, e outros.
- Processos via SNMP: o monitor verifica no agente UCD SNMP por determinados processos verificando-se estão ativos ou não.

- DNS: este monitor questiona algumas informações da zona primária de um domínio (master) e questiona as zonas secundárias verificando o número serial do domínio.

Existem ainda outros monitores disponíveis para os mais variados serviços. O mon foi escolhido para a atividade de monitoração de serviços de rede devido a sua grande variedade de monitores disponíveis.

Os alertas disponíveis são:

- Email
- PopUp de tela
- Pager alfanumérico via modem.
- Mensagem para dispositivos portáteis usando SNPP.
- Traps podem ser enviados a servidores remotos.

4.2.4.2 Funcionamento do Mon

A execução do Mon inicia pela carga e interpretação do arquivo com as configurações para monitoração. Em seguida realiza o escalonamento dos serviços a serem monitorados iniciando pela verificação de todos os serviços cadastrados. Um relógio interno é associado a cada serviço para que seja refeita a verificação do serviço no período pré-determinado.

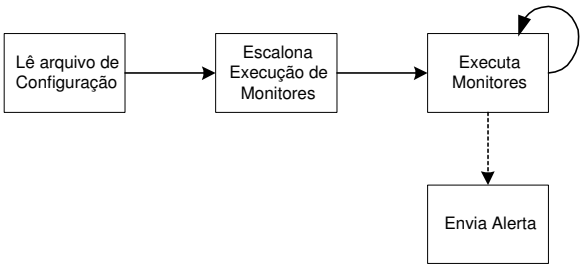

Figura 4.21: Funcionamento do Mon.

No caso de falha de algum serviço, um alerta é acionado informando da falha do serviço. O alerta é acionado periodicamente até o re-estabelecimento do serviço. Ao restabelecer o serviço, outro alerta é enviado para informando do retorno a normalidade.

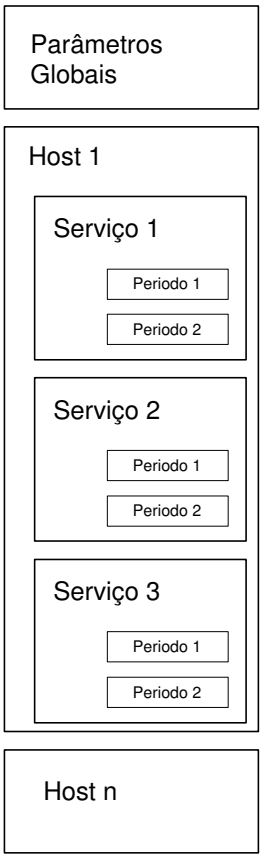

Figura 4.22: Estrutura do Arquivo de Configuração do Mon.

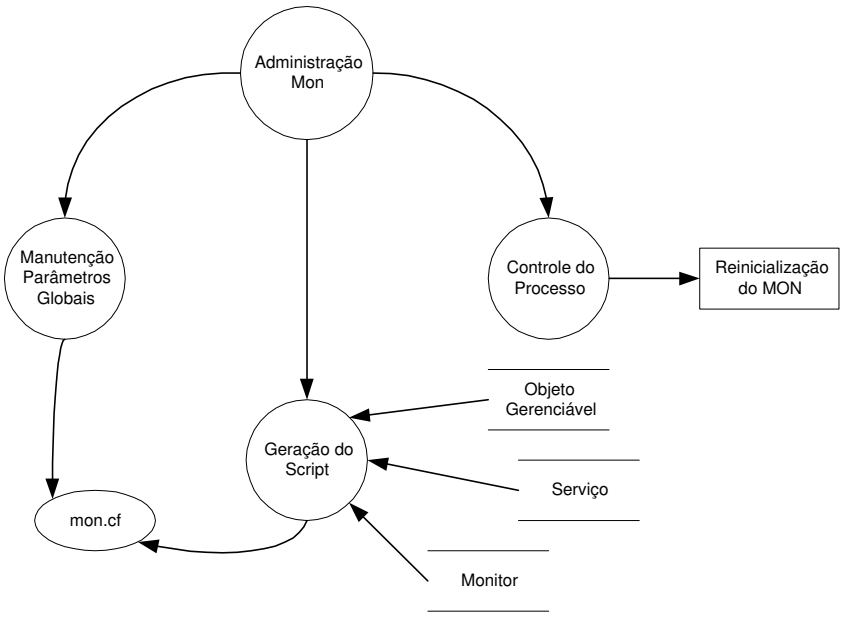

Figura 4.23: Administração do Mon.

4.2.4.3 Parametrização

O arquivo de configuração do Mon é composto de forma hierárquica de serviços. Desta forma os serviços podem ser monitorados em cascata. O Mon possibilita várias combinações para a monitoração de seus serviços. De forma a integrar com o ambiente proposto, o arquivo de configuração ficou estruturado conforme a figura 4.22.

Para cada Objeto Gerenciável temos os serviços associados a ele. Um parâmetro de prioridade indica a ordem de execução dos monitores do Objeto. Para cada serviço monitorado do Objeto podem ser especificados até dois períodos de monitoração distintos, com suas respectivas ações.

4.2.4.4 Integração Módulo de Monitoração de Serviços

Assim como no MRTG, o módulo de administração do Mon administra os parâmetros e funções necessárias para a execução da atividade de monitoração de serviços (figura 4.23). O mon executa a partir de um arquivo de parâmetros que informam quais os monitores e alertas que serão executados para cada Objeto Gerenciável.

O mon possui uma hierarquia para execução dos monitores, fazendo com que os testes realizados pelos monitores funcionem em cascata. Por exemplo, quando for monitorado algum serviço de um Objeto Gerenciável, é possível verificar se este objeto está alcançável (através de um *ping*), assim somente será executado o teste no serviço se o objeto estiver disponível na rede.

Estrutura de Diretórios Foi criada uma estrutura de diretórios seguindo a estrutura básica criada na instalação do mon. Os diretórios que compõem esta estrutura são:

- alert.d: especifica o diretório onde estão os scripts de alertas.
- conf: especifica o diretório do arquivo de configuração do mon.
- log.d: especifica o diretório para armazenagem de logs.
- mon.d: especifica o diretório onde estão os scripts de monitoração (monitores).
- state.d: especifica o diretório onde estão armazenadas informações de estado dos serviços.

Função de Manutenção de Parâmetros Globais A tela de administração do mon apresenta os parâmetros iniciais para sua execução:

- Diretórios: especifica os diretório citados acima.
- Processos: número máximo de processos de monitoração que serão executados simultaneamente.
- Histórico: tamanho máximo do histórico de eventos registrados pelo mon.
- Autenticação: tipo de autenticação utilizada no mon. Necessário para a execução de processos de reinicialização do processo do mon.

Função de Geração Automática de Monitoração de Interface Este procedimento automatiza a geração do serviço de monitoração de interfaces. São definidos alguns parâmetros para a monitoração e podem ser escolhidos na lista de Objetos quais serão monitorados. O processo verifica na lista de Objetos, quais as interfaces deverão ser monitoradas gravando as informações do serviço no Banco de Dados.

Função de Geração do Script Esta função realiza a geração do script de configuração que será utilizado pelo Mon. O processo inicia selecionando todos os Objetos Gerenciáveis que possuem serviços a serem monitorados. Realiza nesta mesma pesquisa uma relação com os monitores que serão utilizados pelos serviços. Cria o arquivo *mon.cf* no diretório de configuração, inserindo nas primeiras linhas os parâmetros globais. Gera então as linhas de script para cada serviço a ser monitorado em cada Objeto.

Função de Controle do Processo Este procedimento realiza a reinicialização do serviço de monitoração com o propósito de recarregar as definições do arquivo de configuração.

4.2.5 Alertas

O módulo de Alertas é responsável por concentrar e dispachar todos os eventos de alertas gerados. O módulo de Alertas unifica as mensagens geradas pelas aplicações e é responsável também por armazenar os históricos dos eventos ocorridos. Eventos estes que podem ser visualizados pelo Módulo de Apresentação de Alertas.

Funcionamento dos Alertas O módulo de alertas é acionado toda vez que um evento exija que uma mensagem seja enviada ao Administrador da Rede. A aplicação que gerou o evento aciona o módulo passando como parâmetros o nome do Objeto Gerenciável que foi alvo do evento, o tipo de Alerta, e informações complementares para compor a mensagem do alerta. A figura 4.24 apresenta o diagrama de funcionamento do módulo de Alertas.

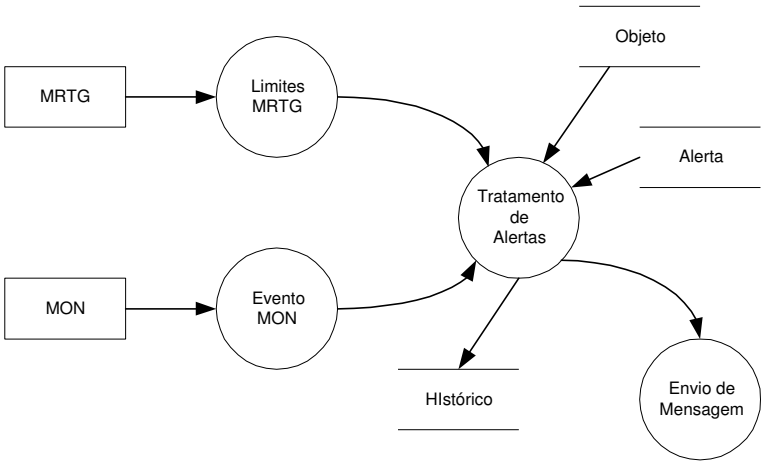

Figura 4.24: Funcionamento do módulo de Alertas.

Após receber a informações, o módulo verifica nos Tipos de Alertas qual a mensagem padrão e como e para onde deve ser enviada (e-mail, pop-up de tela, pager). O módulo registra no histórico o evento ocorrido e despacha a mensagem.

Integração com as demais ferramentas O módulo de alertas está integrado com as ferramentas MRTG e Mon. No MRTG quando um limite é alcançado, um script é acionado de forma a compor os parâmetros que serão enviados para o módulo de Alertas. Este script organiza os dados enviados pelo MRTG (métrica, limite e valor corrente) e aciona o módulo de Alertas.

No Mon foi criado um script especial que realiza a integração com o módulo de Alertas. Este script recebe as informações passadas pelo Mon, organiza e envia as informações: nome do Objeto, serviço e complemento; para o módulo de Alertas.

4.2.6 Análise de Desempenho (gráficos on-line)

Desenvolvido no modelo Cliente-Servidor, este recurso do ambiente permite que sejam coletados dados de métricas e visualizados graficamente. A captura destes dados pode ser programada para ser executada em intervalos de tempo curtos, como por exemplo a cada segundo. Dois módulos foram construídos para este recurso. Um protocolo de mensagens foi definido para permitir a comunicação entre os dois módulos.

Módulo Servidor O módulo servidor foi construído em linguagem C e é responsável pela coleta periódica dos valores das métricas do Objeto Gerenciável.

O módulo servidor inicia aguardando por uma conexão do cliente. Ao conectar o cliente envia uma mensagem ao servidor com os parâmetros para a monitoração. O servidor então inicia a coleta de dados via protocolo SNMP enviando os resultados ao cliente. O processo repete-se até o cliente encerrar a operação.

Módulo Cliente O módulo cliente é responsável por informar ao servidor que métricas serão monitoradas e depois apresentar os dados graficamente.

O módulo cliente inicia solicitando algumas informações para iniciar a monitoração das métricas. os dados informados são:

- Objeto Gerenciável
- Intervalo
- Grupo de OIDs

Após informar os dados, o Cliente conecta ao Servidor, envia os parâmetros e aguarda pelos valores coletados. Ao receber os dados, gera o gráfico de desempenho.

Protocolo de Mensagens O protocolo de mensagens é bem simples e permite o controle de fluxo entre o Cliente e o Servidor.

A figura 4.25 apresenta a comunicação entre Cliente e Servidor.

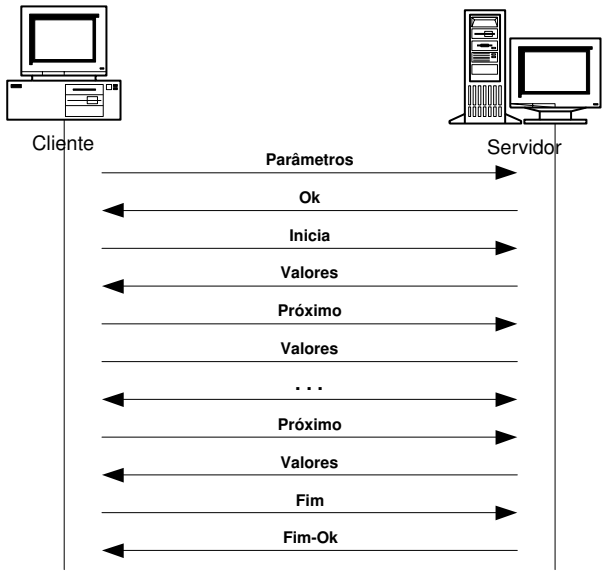

Figura 4.25: Troca de Mensagens entre Cliente-Servidor.

As mensagens que definem o protocolo são:

Tabela 4.1: Definição do Protocolo para troca de mensagens

Mensagem	Descrição	Origem	Destino	Conteúdo
Parâmetros	Especifica os parâmetros que serão enviados ao servidor para iniciar as coletas. Os parâmetros são: Endereço IP do Objeto Gerenciável, Community de leitura, Intervalo (em segundos), ObjectID 1 e ObjectID 2	Cliente	Servidor	host: xxx.xxx.xxx.xxx comm:<community> int:<segundos> oid1:<ObjectID1> oid2: <ObjectID2>
Confirma	Especifica se os parâmetros foram recebidos e interpretados de forma correta (OK) ou Incorreta (NOK).	Servidor	Cliente	ok ou nok
Inicia	Indica que o cliente está pronto para receber os valores.	Cliente	Servidor	inicia
Valores	Especifica os valores que foram coletados pelo servidor	Servidor	Cliente	val1:<valor1> val2:<valor2>
Próximo	Especifica que os valores foram recbidos Ok e indica que o cliente está pronto para receber os próximos valores.	Cliente	Servidor	proximo
Fim	Indica ao servidor que o cliente está finalizando a aplicação.	Cliente	Servidor	fim
Fim Ok	Indica que o servidor finalizou a coleta de dados dos Objetos Gerenciáveis	Servidor	Cliente	fim-ok

5 ESTUDO DE CASO

O protótipo foi testado com sucesso no ambiente de rede da Administração Central Porto Alegre do Banco Santander, antigo Banco Meridional. Durante o período de teste foram identificados alguns comportamentos que serão descritos abaixo.

5.1 Características da rede analisada

A rede analisada é baseada em rede local, composta por mais de 350 estações de trabalho, 15 servidores, impressoras, roteadores e *switchs*. A rede está sub-dividida em seis segmentos Ethernet de 100 Mbits/s sendo um segmento corporativo, onde localizam-se os servidores, roteadores e switchs principais; e cinco segmentos departamentais, destinados aos usuários da rede. Possui um *backbone* de tecnologia Gigabit Ethernet que interliga os seus segmentos. Utiliza-se cabeamento estruturado Categoria 5 e o protocolo de rede é exclusivamente TCP/IP.

A infra-estrutura da rede é composta por três *switchs* corporativos e cinco *switchs* departamentais. Para acessos externos são utilizados três roteadores: um para Internet, um para acesso a empresas externas (parceiras), e outro para acesso de voz e dados à rede corporativa Santander Brasil. Os servidores estão assim distribuídos:

- 5 Servidores de Arquivos
- 3 Servidores de Aplicativos
- 2 Servidores DHCP/DNS
- 2 Servidores de Impressão
- 1 Servidor WEB
- 1 Servidor Proxy de Internet
- 1 Firewall para Internet e Empresas

A figura 5.1 apresenta a estrutura de rede.

O sistema de impressão possui 26 impressoras em rede distribuídas por todo o ambiente. A impressão é realizada via TCP/IP (protocolo LPD).

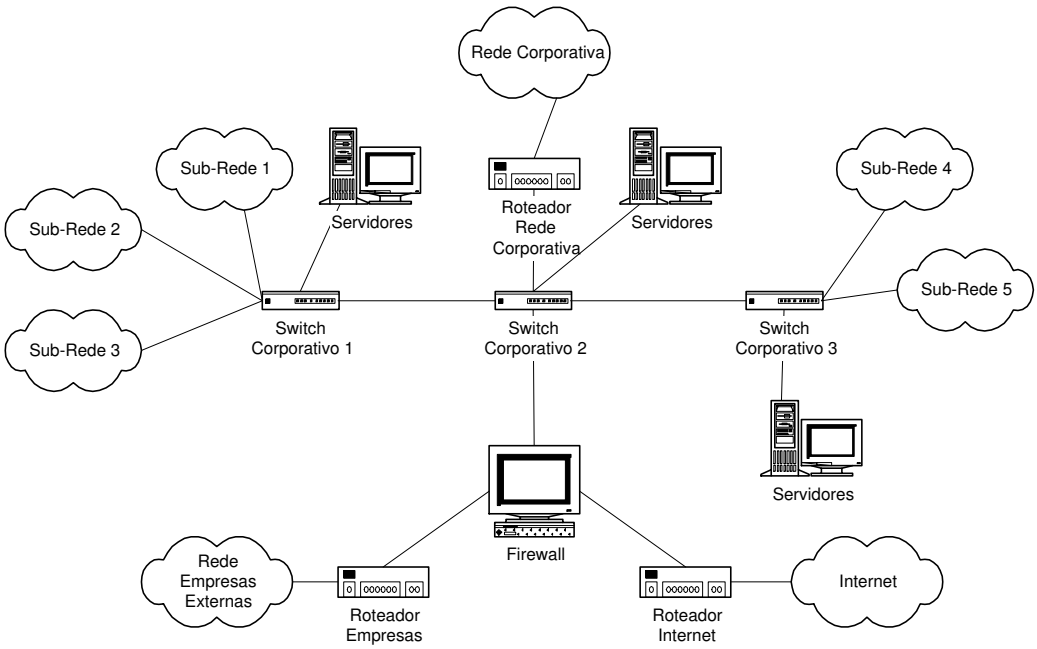

Figura 5.1: Diagrama da Rede Analisada.

5.2 Recursos mapeados

Os equipamentos de rede estão com seus recursos sendo monitorados diariamente. Os recursos de rede foram divididos em classes e foram monitorados quanto a desempenho e detecção de falhas.

5.2.1 Monitoração de Desempenho

Os equipamentos a seguir foram monitorados quanto ao desempenho de seus recursos:
Servidores Windows NT:

- Percentual de utilização de CPU para processos de usuários e processos internos do sistema.
- Percentual de espaço disponível de disco para partições de sistema e área de dados.
- Quantidade de memória disponível e utilizada.
- Utilização de Rede.

Servidores Linux:

- Percentual de utilização de CPU para processos de usuários e internos do sistema.

- Percentual de utilização de disco para partições específicas.
- Quantidade de memória real e virtual disponível.
- Utilização de Rede.

Switchs:

- Utilização de Rede.

Roteadores:

- Utilização de Rede.

5.2.2 Detecção de Falhas

A detecção de falhas verifica os seguintes recursos:

1. Interfaces de Rede: foram monitoradas minuto-a-minuto o estado das principais interfaces de rede de *switchs* e roteadores com o intuito de informar alguma indisponibilidade na rede.
2. Serviços: foram monitorados a disponibilidade dos servidores através de processos de Ping, e verificando-se serviços de WEB, DNS e conexões TCP a Banco de Dados.

5.3 Métricas usadas

O gerenciamento desta rede pode tornar-se uma tarefa árdua se tudo não estiver bem configurado. Outra dificuldade é a diversidade de equipamentos/fornecedores. Cada equipamento/fornecedor possui sua própria solução para gerência e monitoração. Para focalizar nosso gerenciamento, evitando gasto de tempo com análises de informações desnecessárias, foi definidas Métricas de Gerenciamento. Para cada objeto gerenciável foram definidos seus parâmetros críticos.

5.3.1 Métricas para Servidores Windows NT

Para os servidores Windows NT foram especificadas as seguintes métricas para monitorar suas características de utilização de CPU, Memória, Rede e Disco.

As métricas definidas são:

1. CPU
   cpuPercentProcessorTime.0 : especifica o percentual de utilização de CPU para processos do Sistema Operacional.
   cpuPercentUserTime.0 : especifica o percentual de utilização de CPU para processos de usuários.

2. Memória
   memoryAvailableBytes.0: especifica a quantidade de memória disponível.
   memoryCommittedBytes.0: especifica a quantidade de memória virtual que está em uso.

3. Disco
   ldiskPercentFreeSpace....: especifica o percentual de espaço livre no disco. Devido a construção da MIB para cada partição de dados do servidor são definidos sub-ítens individuais.

4. Rede
   ifInOctets: especifica o número total de bytes recebidos pela interface de rede
   ifOutOctets: especifica o número total de bytes enviados pela interface de rede

5.3.2 Métricas para Servidores Linux

Para os servidores Linux as métricas para monitorar a utilização de CPU, Memória, Rede e Disco são:

1. CPU
   ssCpuRawSystem.0 : especifica o percentual de utilização de CPU para processos do Sistema Operacional.
   ssCpuRawUser.0 : especifica o percentual de utilização de CPU para processos de usuários.

2. Memória
   memAvailReal.0: especifica a quantidade de memória real disponível.
   memAvailSwap.0: especifica a quantidade de memória virtual disponível.

3. Disco
   dskPercent.x: especifica o percentual de utilização de uma partição (.*x*) do disco.

4. Rede
   ifInOctets: especifica o número total de bytes recebidos pela interface de rede.
   ifOutOctets: especifica o número total de byes enviados pela interface de rede.

5.3.3 Métricas para Switchs e Roteadores

Para os switchs e roteadores foram especificadas as seguintes métricas para todas as interfaces do equipamento:

1. Utilização de Rede
   ifInOctets.x e *ifOutOctets.x*: especifica o número total de bytes recebidos e enviados pela interface de rede *x*.

2. Pacotes Descartados
   ifInDiscards.x e *ifOutDiscards.x*: especifica a quantidade de pacotes recebidos e enviados que foram descartados pela interface *x*.

3. Erros na Interface
   ifInError.x e *ifOutError.x*: especifica a quantidade de erros de recebimento e envio identificados pela interface *x*.

5.3.4 Especificação de Valores Limites

Com o objetivo de manter o nível de disponibilidade da rede o mais elevado possível foram definidos pontos críticos de monitoração para os equipamentos de rede. Os valores foram especificados com base em observações dos equipamentos de rede e da experiência dos administradores da rede. Os servidores de arquivos possuem como ponto crítico a utilização de seu espaço em disco. Neste caso foram especificados valores limites ocupação de 80% do espaço da área em disco. Isto significa que caso a utilização do disco ultrapasse os 80% um alerta de aviso será enviado para o administrador da rede.

Os servidores de aplicativos tem seu ponto crítico em utilização de CPU e memória. Valores limites de utilização em 30% e 70% foram definidos respectivamente para os processos de usuário e sistema. Para a memória disponível foram especificados como limite mínimo 2 MBytes disponíveis.

Roteadores e switchs tem todas suas interfaces monitoradas. Porém somente para algumas interfaces foram definidos limites para verificação. São elas:

- Interface de roteador para acesso a Internet: Definidos dois níveis de alertas para utilização das interfaces sendo o primeiro alerta de aviso enviado quando o link atinge 70% de utilização para envio ou recebimento. O segundo nível de alerta definido como crítico ocorre quando o link atinge 95% de sua capacidade.

- Interface de roteador para acesso de empresas externas: definido como nível crítico de utilização se o link atingir 80% de sua capacidade.

- Interfaces de Switchs para o backbone: o nível crítico para o backbone foi definido em 40% de utilização.

5.4 Análise dos resultados obtidos

Durante o período de teste do protótipo foram identificados problemas de desempenho e falhas em alguns equipamentos.

Um dos problemas de desempenho mais crítico identificados é o tráfego de Internet. A Rede possui um link de dados de 512 Kbits/s sendo que sua utilização apresenta níveis críticos diariamente. O gráfico da figura 5.2 apresenta a utilização em um único dia, sendo que seu comportamento se repetiu nos demais dias de testes do protótipo.

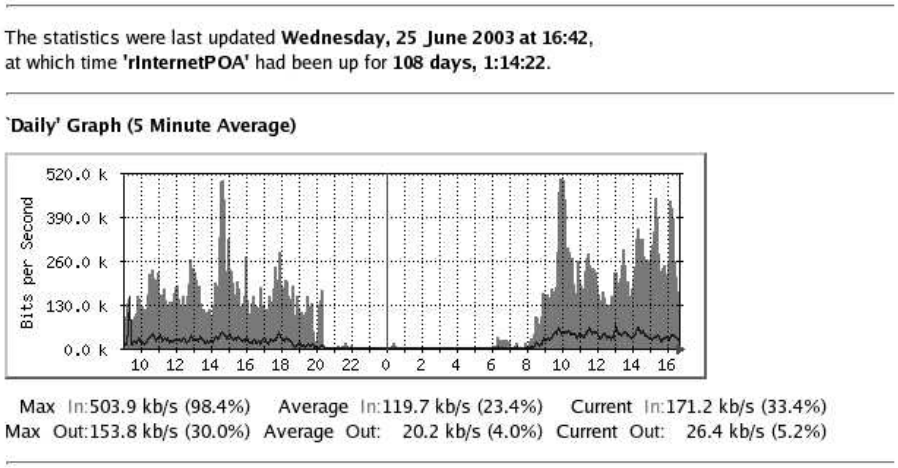

Figura 5.2: Tráfego de Utilização de Internet.

Outro problema de desempenho verificado foi que a rede fora submetida a novas configurações de Domínio NT onde o Controlador de Domínio Primário (PDC) recebeu novas Relações de Confiança com outros domínios remotos. Isto acarretou uma sobrecarga de processamento e rede neste servidor. Os gráficos das figuras 5.3 e 5.4 demonstram o aumento da demanda por processamento e o acréscimo do tráfego de rede para o Controlador de Domínio Primário.

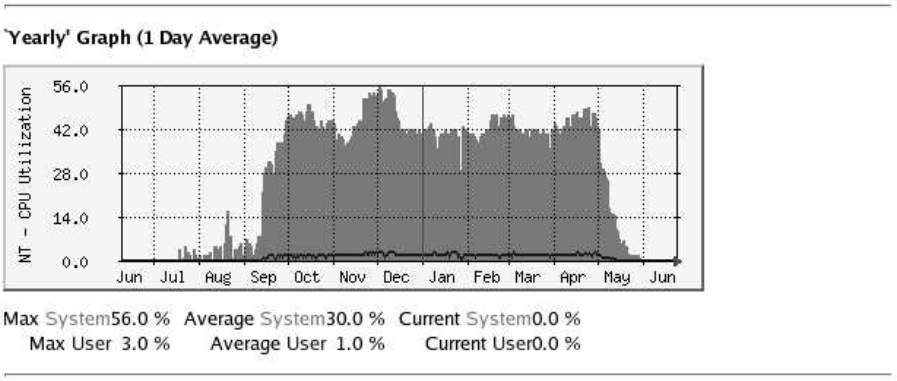

Figura 5.3: Utilização de CPU no PDC.

O problema foi resolvido trocando o equipamento Controlador de Domínio Primário para um de maior capacidade.

Alguns serviços também apresentaram-se indisponíveis. Um dos casos mais críticos foi a indisponibilização de dois servidores de arquivos devido a troca do software de Anti-vírus. Esta troca causou incompatibilidade entre o software de Anti-vírus e o software de controle de quotas de disco de usuários. O problema apresentava-se por um aumento exagerado da utilização de memória, tornando em determinado momento,

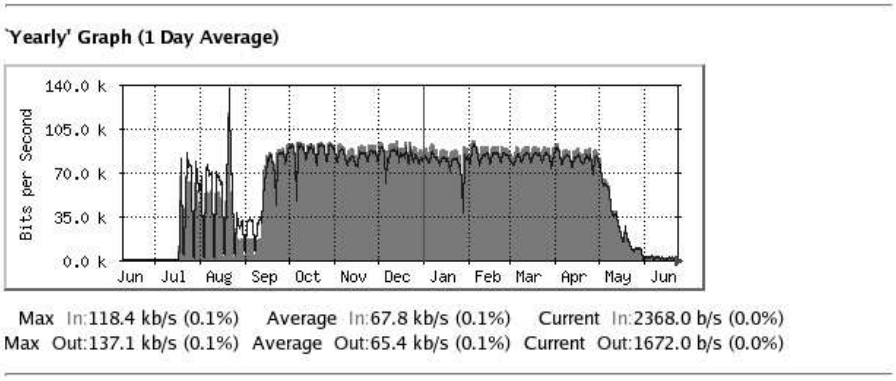

Figura 5.4: Tráfego de Rede no PDC.

os serviços dos servidores indisponíveis causando sua parada.

Alguns links de comunicação apresentaram paradas em virtude de falta de energia-elétrica. No entanto a maioria dos links permaneceu estável, somente apresentando quedas em momentos de manutenção nos equipamentos de comunicação.

5.5 Recomendações

Como itens a serem revistos como problemas em potencial temos:

1. Roteador Internet: como foi identificado nas análises o link de dados para a Internet, embora sua utilização não esteja sobrecarregada, em momentos de pico de utilização, ou durante alguma transferência de arquivos mais demorada, o desempenho de utilização da Internet fica comprometido. Um aumento da largura de banda seria recomendável.

2. Switch da Sub-Rede 5: este equipamento possui sua infra-estrutura deficitária pois não está ligado a uma fonte de energia ininterrupta (*no-break*). Quando ocorre uma queda de energia, o equipamento é desligado, e isto pode comprometer sua configuração e por conseguinte sua disponibilidade.

3. Servidor de Aplicação: para este servidor seria apropriado rever suas características de processamento de dados de forma a melhorar seu desempenho de CPU.

6 CONCLUSÃO

Este trabalho foi realizado visando integrar várias ferramentas de Software Livre proporcionando ao Administrador de Rede uma ambiente no qual ele possa verificar o que está ocorrendo em sua rede, verificando necessidades e reduzindo e otimizando custos.

A gerência de desempenho apresenta dados estatísticos de utilização dos recursos possibilitando uma análise mais apurada e identificando pontos críticos da rede.

A monitoração de falhas trabalha como um "cão de guarda" alertando qualquer violação dos parâmetros especificados e vigiando a disponibilidade dos recursos.

A análise de desempenho on-line não foi finalizada faltando a rotina para a apresentação dos dados coletados.

O uso de ferramentas de Software Livre em gerência de redes é um processo contínuo, as ferramentas de apoio estão constantemente sendo atualizadas, seja pela presença de bugs ou pela adição de novos recursos. Cabe ao administrador da rede avaliar se há necessidade ou não de manutenção nas ferramentas de apoio.

O Software Livre permite expandir a gerência de rede dependendo das necessidades. O uso de Software Livre muitas vezes é trabalhoso, para adicionar uma nova funcionalidade, o desenvolvedor deve primeiro analisar o ambiente atual de forma a não inserir alguma incompatibilidade no sistema. A atualização de versão de uma ferramenta de apoio deve ser muito bem analisada e testada pois poderá causar atualizações em todo o sistema.

Este ambiente, da forma em que foi concebido permite a inclusão futura de outras ferramentas como: Detector de Intrusão, tratamento de *traps* de dispositivos de rede, expansão do ambiente para gerenciamento distribuído, integração dos dados com geração de relatórios mais complexos.

A utilização de Software Livre não acarreta necessariamente uma solução de custo-zero. Embora não exista custo de propriedade sobre as ferramentas, existe o custo em aprendizagem, avaliação, testes e integração das ferramentas. Contudo, soluções utilizando Software Livre são viáveis para a maioria das pequenas e médias empresas que queiram gerenciar seus recursos e não tenham condições de arcar com os custos de uma ferramenta proprietária.

REFERÊNCIAS

ADVENTNET. **AdventNet SNMP API.** Disponível em: <http://www.adventnet.com/products/snmp/index.html>. Acesso em: setembro 2001.

APRISMA. **Spectrum Infinity - Service Provider Solutions.** Disponível em: <http://www.aprisma.com/support/tech-specs.shtml>. Acesso em: julho 2003.

BRASIL, I. **Produtos e Serviços - Software- Tivoli.** Disponível em: <http://www.ibm.com/br/software/tivoli/performance/>. Acesso em: julho 2003.

CARVALHO, R. M. de. **O que é software livre ?** Disponível em: <http://www.xlinuxnews.com.br/>. Acesso em: outubro 2001.

COMER, D. **Internetworking with TCP/IP**: principles, protocols and architecture. 3rd.ed. New Jersey: Prentice Hall, 1995. 447-463p.

COMER, D. **Internetworking with TCP/IP**: design, implementation and internals. 3rd.ed. New Jersey: Prentice Hall, 1999. 449-578p.

COMPUTING, N. **Os melhores do mercado - Gerenciamento.** Disponível em: <http://www.revistanetwork.com.br/estudos/artigo.asp?id=5190>. Acesso em: julho 2003.

FILIPO, R. **Curso de PERL/Basico.** Disponível em: <http://www.cipsga.org.br>. Acesso em: agosto 2001.

FLATIN, J.-P. M. **A Simple Typology of Distributed Network Management Paradigms.** Switzerland: Swiss Federal Institute of Technology, 1997.

GÜRER, D. **An Intelligent-Agent-Based Architeture for the Management of Heterogeneous Networks.** Menlo Park, CA, USA: SRI International, 1998.

HAYES, S. Analyzing Network Performance Management. **IEEE Communication Magazine**, [S.l.], v.31, n.5, May 1993.

KONETY, M. **Mobile Components to Manage the Heterogeneous Internet.** USA: HCI Technologies, 1997.

LINUX-HA. **High Availability Linux Project.** Disponível em: <http://www.linux-ha.org>. Acesso em: setembro 2001.

LINUX., R. do. **Filosofia da Liberdade.** Disponível em: <http://www.revistadolinux.com.br>. Acesso em: agosto 2001.

MON. **mon - Service Monitoring Daemon (v. 0.38.21).** Disponível em: <http://www.kernel.org/pub/software/admin/mon/html/>. Acesso em: setembro 2001.

MRTG. **MRTG - Multi Router Traffic Grapher.** Disponível em: <http://people.ee.ethz.ch/ oetiker/webtools/mrtg/>. Acesso em: outubro 2001.

OPENNMS. **OpenNMS - Network Management System.** Disponível em: <http://www.opennms.org>. Acesso em: julho 2003.

PERL. **Perl Mongers.** Disponível em: <http://www.perl.org>. Acesso em: setembro 2001.

PHP. **Personal Home Page.** Disponível em: <http://www.php.net/docs.php>. Acesso em: setembro 2001.

QUEST SOFTWARE, I. **The Big Brother System and Network Monitor.** Disponível em: <http://www.bb4.com/bb/help/>. Acesso em: julho 2003.

SIENA, C. **Curso de PHP. Introdução - Aula 1.** Disponível em: <http://www.xlinuxnews.com.br/>. Acesso em: outubro 2001.

STALLINGS, W. **SNMP, SNMPv2 and CMIP**: the practical guide to network management standards. 1st.ed. Massachusetts: Addison-Wesley Publishing Company, 1993.

WEBALIZER. **Home of WebAlizer.** Disponível em: <http://www.mrunix.net/webalizer>. Acesso em: outubro 2001.

APÊNDICE A BASE DE DADOS DE GERENCIAMENTO

A.1 Objeto Gerenciável: tb_object

Coluna	Tipo	Descrição
id_obj	serial	Identificação do Objeto
obj_name	vchar(30)	Nome do Objeto (hostname)
obj_ipaddr	char(15)	Endereço IP do Objeto [1]
obj_location	vchar(50)	Localização do Objeto [2]
obj_contact	vchar(50)	Informação para Contato [3]
obj_descr	vchar(50)	Descrição textual do Objeto
obj_type	char(1)	Tipo de Objeto [4]
obj_readcommunity	char(32)	Community de leitura do SNMP
obj_writecommunity	char(32)	Community de escrita do SNMP
obj_http	vchar(50)	URL para página de gerenciamento

Observações:

1. O endereço IP será armazenado como string.

2. Informação obtida apartir da MIB-II (system.sysLocation).

3. Informação obtida apartir da MIB-II (system.sysContact).

4. Tipo de objeto pode ser:
   S - Servidor
   R - Router
   W - Switch
   P - Printer
   U - Undefined

A.2 Grupo OID: tb_oidmib

Coluna	Tipo	Descrição
id_odi	serial	Identificação do ObjectID
oid_descr	vchar(50)	Descrição do ObjectID
oid_val1	vchar(250)	ObjectID 1
oid_val2	vchar(250)	ObjectID 2
oid_descr1	vchar(50)	Descrição do ObjectID 1
oid_descr2	vchar(50)	Descrição do ObjectID 2
oid_simple_desc1	vchar(15)	Sigla para o ObjectID1
oid_simple_desc2	vchar(15)	Sigla para o ObjectID2
oid_mibfile	vchar(50)	Arquivo com a MIB
oid_mibname	vchar(80)	Nome da MIB

A.3 Métrica: tb_metr

Coluna	Tipo	Descrição
id_metr	serial	Identificação da métrica
id_oid	integer	ObjectID
id_obj	integer	Identificação do Objeto
id_graph	integer	Identificação do perfil de gráfico
mtr_threshold1	boolean	Verificar intervalo de valores para OID1
mtr_max1	integer	Valor máximo para OID1
mtr_min1	integer	Valor mínimo para OID1
mtr_threshold2	boolean	Verificar intervalo de valores para OID2
mtr_max2	integer	Valor máximo para OID2
mtr_min2	integer	Valor mínimo para OID2
mtr_alert1	vchar(50)	Atributo ThresholdProgI [1]
mtr_alert2	vchar(50)	Atributo ThresholdProgO [2]
mtr_ifnum	smallint	Número da Interface [3]
id_alert1	integer	Tipo de Alerta para OID1
id_alert2	integer	Tipo de Alerta para OID2

Observações:

1. ThresholdProgI: script de alerta para acionamento do Threshold no parâmetro input. Buscar lista de arquivos do diretório de Thresholds.

2. ThresholdProgO: script de alerta para acionamento do Threshold no parâmetro output. Buscar lista de arquivos do diretório de Thresholds.

3. Número da Interface se a métrica se refere a alguma interface, caso contrário é usado 0 (zero).

A.4 Gráfico: tb_graph

Coluna	Tipo	Descrição
id_graph	serial	Identificação do Perfil
grp_desc	vchar(30)	Descrição do perfil
grp_template	vchar(50)	Arquivo de script [1]
grp_maxbytes	smallint	Atributo MaxBytes [2]
grp_options	vchar(170)	Atributo Options [2]
grp_unscale	char(5)	Atributo Unscale [2]
grp_supress	char(5)	Atributo Supress [2]
grp_xsize	smallint	Atributo XSize [2]
grp_ysize	smallint	Atributo YSize [2]
grp_shortlegend	vchar(10)	Atributo ShortLegend [2]

Observações:

1. Arquivo de template para geração do script do MRTG. Buscar lista de arquivos do diretório de Templates.

2. Atributos do MRTG.

A.5 Interface: tb_interface

Coluna	Tipo	Descrição
id_interface	serial	Identificação da Interface
id_obj	integer	Identificação do Objeto
if_num	smallint	Número da Interface [1]
if_name	vchar(30)	Nome da Interface [2]
if_descr	vchar(50)	Descrição SNMP da Interface [3]
if_show	boolean	Flag de visualização [4]
if_speed	vchar(30)	Velocidade de operação da Interface [5]
if_mon	boolean	Monitorar o estado da interface [6]
if_graph	boolean	Gerar gráfico de utilização da Interface
id_graph	integer	Perfil de geração do Gráfico

Observações:

1. Informação obtida apartir da MIB-II (interface.ifEntry.ifIndex.x).

2. Descrição do que está ligado nesta interface.

3. Informação obtida apartir da MIB-II (interface.ifEntry.ifDescr.x).

4. Flag para visualização na lista de interfaces.

5. Informação obtida apartir da MIB-II (interface.ifEntry.ifSpeed.x).

6. Monitorar o estado da interface, pode ser UP ou DOWN.

A.6 Histórico: tb_hist

Coluna	Tipo	Descrição
id_hist	serial	Identificação do Alerta
id_obj	integer	Identificação do Objeto
id_alert	integer	Tipo de Alerta
hst_timestamp	timestamp	Horário do alerta
hst_desc	vchar(50)	Descrição Completa do Alerta
hst_pending	boolean	Verificação de Pendência[1]

Observações:

1. Flag para verificação dos alertas ocorridos. Indica se o alerta já foi verificado ou não.

A.7 Alertas: tb_alert

Coluna	Tipo	Descrição
id_alert	serial	Identificação do Tipo de Alerta
ale_descr	vchar(30)	Descrição do Alerta
ale_message	vchar(250)	Mensagem de Alerta Padrão
ale_mail	vchar(50)	Endereços de e-mails para envio de alerta
ale_popup	vchar(50)	Endereços de máquinas para envio de pop-up
ale_pager	vchar(50)	Endereços de Pager
ale_opt	char(3)	Enviar Alertas [1]
ale_type	char(1)	Nível de Alerta [2]

Observações:

1. Enviar alertas para :
   m - Email
   p - Pager
   w - Windows Popup

2. Nível de alerta:
   w - Warning
   c - Critical
   r - Ready

A.8 Serviço: tb_service

Coluna	Tipo	Descrição
id_service	serial	Identificação do Serviço
id_obj	integer	Identificação do Objeto
id_mon	integer	Tipo de monitor de serviço
srv_descr	vchar(50)	Descrição do Serviço
srv_param1	vchar(15)	Parâmetro adicional
srv_interval	char(10)	Intervalo de verificação
srv_period1	vchar(15)	Período de Verificação-1
srv_alert1_fail	integer	Alerta1 para falha [1]
srv_alert1_ok	integer	Alerta1 para retorno [2]
srv_time1	vchar(15)	Tempo entre Alertas1
srv_period2	vchar(15)	Período de Verificação-2
srv_alert2_fail	integer	Alerta2 para falha
srv_alert2_ok	integer	Alerta2 para retorno
srv_time2	vchar(15)	Tempo entre Alertas1

Observações:

1. Tipo de Alerta a ser enviado em caso de problemas no serviço.
   - Service [xxxxx] off-line.
   - [CPU, Network, Memory, Disk] overloaded.
   - etc.

2. Tipo de Alerta quando o serviço retornar ao estado normal.

A.9 Monitor: tb_monitor

Coluna	Tipo	Descrição
id_mon	serial	Identificação do Serviço
mon_type	char(05)	Tipo de Monitor de Serviço[1]
mon_priority	smallinteger	Prioridade de verificação
mon_file	vchar(50)	Arquivo PERL do monitor
mon_param	vchar(50)	Parâmetros do Monitor

Observações:

1. Tipos de Monitores de serviços:
   ping - Verificação de objeto
   dns - Servidor de Nomes;
   http - Servidor WEB;
   ftp - Servidor de FTP;

pop3 - Servidor POP3;
smtp - Servidor SMTP;
tcp - Serviço em determinada porta TCP;
smb - Serviço na porta 139 (Windows Network);
iface - Verificação das interfaces do Objeto.

A.10 Tipos de Dados

Tipo	Descrição
serial	Número interido incrementado automaticamente
char(n)	Tipo de dado caracter de tamanho fixo (n)
vchar(n)	Tipo de dado caracter de tamanho variável (máximo n caracteres)
integer	Tipo de dado numérico inteiro de 32 bits
smallinteger	Tipo de dado numérico inteiro de 16 bits
boolean	Tipo de dado lógico (t - verdadeiro / f - falso)
timestamp	Tipo de dado para datas