

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Desenvolvimento de Componentes para Facilitar a Monitoração  
Remota de Redes de Computadores Usando a Ferramenta  
WebManager

**Exson Machado Souza**

**Área de concentração: Ciência da Computação**

**Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos**

Campina Grande – PB

Julho - 2002

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
CURSO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**Desenvolvimento de Componentes para Facilitar a  
Monitoração Remota de Redes de Computadores Usando a  
Ferramenta WebManager**

**Exson Machado Souza**

Dissertação submetida à Coordenação de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal de Campina Grande como requisito parcial para a obtenção do grau de Mestre em Ciências (MSc)

Jacques Phillippe Sauvé  
(orientador)

Campina Grande – PB

Julho - 2002

A minha família, amigos e a Deus, pela  
presença constante e ajuda nos  
momentos difíceis

# Sumário

<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1 OBJETIVOS DA DISSERTAÇÃO .....	12
1.2 ESCOPO E RELEVÂNCIA .....	12
1.3 ESTRUTURA DA DISSERTAÇÃO .....	13
<b>INTRODUÇÃO À GERÊNCIA DE REDES .....</b>	<b>14</b>
2.1 O QUE É GERÊNCIA DE REDE? .....	14
2.2 ARQUITETURA DE GERÊNCIA.....	16
2.3 O PADRÃO SNMP DE GERÊNCIA .....	17
2.3.1 <i>Origem</i> .....	17
2.3.2 <i>Características e Funcionamento</i> .....	18
2.3.3 <i>Troca de informações</i> .....	20
2.3.4 <i>Base de informação Gerenciável (MIB)</i> .....	21
2.4 RMON - FUNCIONAMENTO E PRINCIPAIS CARACTERÍSTICAS DO RMON .....	22
2.4.1 <i>Probes RMON</i> .....	23
2.4.2 <i>Descrição dos Grupos do RMON</i> .....	25
2.4.2 <i>Necessidade de Ferramentas</i> .....	27
<b>COMPONENTES PARA GERÊNCIA RMON .....</b>	<b>29</b>
3.1 REQUISITOS DOS COMPONENTES .....	29
3.2 ARQUITETURA DOS COMPONENTES .....	30
3.2.1 <i>Coleta</i> .....	31
3.2.2 <i>Armazenamento</i> .....	34
3.2.3 <i>Processamento das informações</i> .....	35
3.2.4 <i>Comunicação</i> .....	37
3.2.4 <i>Configuração</i> .....	37
3.3 ARQUITETURA DOS COMPONENTES RMON ACOPLADOS A UMA SOLUÇÃO DE GERÊNCIA. ...	38
3.3.1 <i>WebManager</i> .....	38
3.3.2 <i>Arquitetura do WebManager</i> .....	39
3.3.3 <i>Arquitetura RMON incorporada ao WebManager</i> .....	41
<b>IMPLEMENTAÇÃO .....</b>	<b>43</b>
4.1 PACOTES.....	43

4.2 APIs UTILIZADAS.....	44
4.2.1 API MonarchCharts .....	44
4.2.2 API AdventNet SNMP.....	45
4.3 CLASSES DESENVOLVIDAS .....	45
4.3.1 Pacote rmon.config.....	45
4.3.2 Pacote rmon.monitor.....	46
4.3.3 Pacote rmon.log .....	46
4.3.4 Pacote rmon.loader .....	47
4.3.5 Pacote rmon.addressT.....	47
4.3.6 Pacote rmon.graphs .....	48
4.4 MODIFICAÇÕES NO WEBMANAGER.....	50
<b>ESTUDO DE CASO NO USO DOS COMPONENTES .....</b>	<b>51</b>
5.1 A REDE DO CAMPUS II UNIT .....	51
5.2 INTERFACE DA GERÊNCIA RMON .....	53
5.3 GERÊNCIA RMON NA REDE DO CAMPUS II - UNIT .....	57
5.3.1 Verificando estatísticas de erros entre os equipamentos do Bloco A.....	58
5.3.2 Descobrimo o Equipamento que mais transmite no Bloco C.....	58
5.3.3 Descobrimo os maiores transmissores e seus destinos na rede da Biblioteca.....	59
5.3.4 Comparando o funcionamento de duas interfaces do switch S2BIB001.....	60
<b>CONCLUSÕES.....</b>	<b>62</b>
6.1 AVALIAÇÃO DOS COMPONENTES.....	62
6.2 DIFICULDADES ENCONTRADAS .....	64
6.3 TRABALHOS FUTUROS .....	65
<b>DESENVOLVIMENTO DE SOFTWARE COM COMPONENTES.....</b>	<b>68</b>

## Lista de siglas

API	Application Programming Interface
CIMP	Common Management Information Protocol
DNS	Domain Name Service
HEMS	High–Level Entity Management System
IP	Internet Protocol
ISO	International Organization for Standardization
JSP	Java Server Page
LAN	Local Area Network
MAC	Medium Access Control
MIB	Management Information Base
OSI	Open System Interconnection
RFC	Request for Comments
RMON	Remote Monitoring
SGMP	Simple Gateway Monitoring Protocol
SNMP	Simple Network-Management Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WAN	Wide Area Network
XML	Extensible Markup Language

## Lista de Figuras

1	Arquitetura de Gerência	16
2	Arquitetura de Gerência com SNMP	19
3	Árvore Mib RMON	27
4	Arquitetura dos Componentes	31
5	Funcionamento do Monitor	32
6	Tabelas de Controle e dados	33
7	Armazenamento dos dados	34
8	Processamento das informações RMON	35
9	Arquitetura do WebManager	39
10	Framework de Gerência	40
11	Arquitetura RMON adaptada ao modelo WebManager	41
12	Mapa da rede do Campus II da UNIT	53
13	Exemplo de Página de Gerenciamento do Equipamento S2BIB001	54
14	Página de Gerenciamento Modificada do S2BIB001	54
15	Lista de Datas com Gráficos	55
16	Lista de Gráficos para o S2BIB001	56
17	Lista de Interfaces do equipamento S2BIB001	56
18	Exemplo de um gráfico para o elemento S2BIB001	57
19	Estatísticas de pacotes de broadcast	58
20	Maiores Transmissores do Bloco C	59
21	Tráfego entre equipamentos da Rede da Biblioteca	60
22	Comparação entre interfaces do Switch S2BIB001	61

## Lista de Tabelas

1	Lista de Grupos RMON	25
2	Componentes Desenvolvidos	44
3	Elementos Ativos do Campus II	52
4	Requisitos alcançados pelos componentes	63



# Resumo

Os equipamentos de redes estão implementando probes RMON (monitoramento remoto) e oferecendo essa ferramenta como opção para melhorar a gerência da rede. O RMON fornece dados que possibilitam a detecção de erros e verificação de performance das redes no nível de enlace, no entanto a quantidade de informações é muito grande e sua análise é muito difícil. Nesse trabalho são desenvolvidos componentes para facilitar a utilização da tecnologia RMON, possibilitando a coleta e processamento dos dados disponíveis e apresentação gráfica dos mesmos. Como avaliação e teste dos componentes desenvolvidos, foi executada a incorporação dos mesmos na aplicação de gerência WebManager, desenvolvida e utilizada pela Universidade Federal da Paraíba, e apresentado um exemplo prático de gerenciamento de redes utilizando a ferramenta modificada.

# Abstract

Network equipments are implementing RMON(Remote Monitoring) Probes and offering this tool as an option to improve the Network Management. The RMON supplies data that makes possible to detect errors and to verify the network performance at the Data Link layer, however the quantity of information is too high and it's too difficult to analyze it. This work involves the description and implementation of software components to facilitate the use of RMON technology, making possible to collect and processes the available data and present this data in a graphical way. As evaluation and test of the components, they were incorporated to the network management tool WebManager, developed and used by the Federal University of Paraíba, and presented a practical example of network management using this modified tool.

# Capítulo 1

## Introdução

As redes de computadores estão cada vez mais presentes na sociedade, seja interligando computadores pessoais em uma casa ou interligando toda a estrutura de comunicação de empresas multinacionais. Utilizadas para facilitar o desenvolvimento dos trabalhos e para possibilitar o acesso a informações *on-line* na Internet, estão cada vez mais influenciando os serviços oferecidos pelas empresas e instituições, e desta forma tornou-se vital mantê-las sempre disponíveis, com uma boa performance e com um baixo custo de manutenção.

A dependência das empresas em relação ao funcionamento das redes de computadores faz com que haja uma busca pela utilização de ferramentas que possibilitem executar a gerência dessas redes. A Gerência de Redes representa a utilização de tecnologias de hardware e software com o objetivo de monitorar e modificar o funcionamento dos recursos que compõe a infraestrutura da rede.

Várias ferramentas estão disponíveis para executar essas tarefas de gerenciamento, como por exemplo, o Open View da HP e o Unicenter TGN da Computer Associates, o diferencial entre elas é a quantidade e qualidade dos recursos disponíveis e principalmente o preço para compra e implantação.

Várias dessas ferramentas de gerência de redes têm como base de sua implementação o protocolo SNMP(Simple Network Management Protocol) para executar troca de informações com os equipamentos de rede e desta forma analisar o funcionamento desses equipamentos. O software presente nos equipamentos é capaz de fornecer as informações de

gerência é chamado de Agente, e suas características determinam a qualidade e quantidade de informações coletadas e disponíveis para os softwares de gerência.

Uma das características que determina a qualidade de um Agente é se ele implementa o RMON (Remote Monitoring). Os Probes RMON embutidos em dispositivos como hubs, switches e roteadores fornecem uma vasta gama de dados. Só através do que um probe RMON fornece, pode-se fazer toda a gerência de desempenho e falhas de uma rede local. Mas, apesar de rica, essa informação não é muito simples de ser utilizada, pois é de muito baixo nível. Uma pessoa comum teria muita dificuldade de extrair informações úteis apenas lendo os dados que estão presentes nas tabelas da MIB RMON. Por exemplo, o probe mantém uma matriz de tráfego entre os pares fonte-destino, essa matriz pode ser muito grande e só pode ser analisada se for mostrada de alguma forma gráfica para o usuário, ou se for utilizada por um programa para responder a perguntas do tipo "quem anda conversando muito entre si recentemente?".

O maior problema da utilização do RMON é a visualização dos dados coletados pelos probes, pois se faz necessária a utilização de ferramentas que possibilitem o tratamento desses dados para assim facilitar o entendimento pelo administrador da rede.

## **1.1 Objetivos da Dissertação**

Desenvolver componentes de alto nível que apresentem a riqueza de informações presentes no Probes RMON de uma forma mais simples de ser entendida pelo gerente da rede. Estes componentes serão descritos e como forma de validação serão desenvolvidos e incorporados à ferramenta de gerência de rede da Universidade Federal da Paraíba o WebManager[Sauvé, 1999].

## **1.2 Escopo e Relevância**

A contribuição desse trabalho apresenta-se primeiramente na forma de um documento descrevendo as informações mais relevantes presentes na MIB RMON e na descrição de componentes para o tratamento dessas informações. Este documento servirá de base para o desenvolvimento de alguns dos componentes propostos e como fonte de informação para desenvolvedores de softwares de gerência rede.

Como segunda contribuição tem-se a integração desses componentes à ferramenta de gerenciamento de redes Webmanager [Sauvé, 1999], desenvolvida por pesquisadores da Universidade da Paraíba, expandindo suas funcionalidades para utilização do RMON.

É importante ressaltar que essa proposta não se limita a criar uma ferramenta capaz de tratar os dados RMON, mas sim propor e desenvolver componentes reutilizáveis e capazes de possibilitar o desenvolvimento de aplicações de gerenciamento de redes com informações RMON sem a necessidade de implementar o tratamento dos dados da MIB RMON e ainda criar a integração desses componentes a uma ferramenta já funcional.

### **1.3 Estrutura da Dissertação**

No Capítulo 2 apresentamos os conceitos envolvidos com a gerência de redes, abrangendo uma explicação da arquitetura de gerência, o padrão SNMP e o RMON. Demonstramos também o porquê da complexidade de utilizar os dados RMON e da necessidade de ferramentas para aproveitar esses dados.

No Capítulo 3 apresentamos um levantamento das necessidades e problemas que precisam ser resolvidos pelos componentes RMON e descrevemos uma arquitetura utilizada como proposta para solucionar esses problemas.

No Capítulo 4 demonstramos como os componentes foram desenvolvidos, dando ênfase aos pacotes e classes utilizados e desenvolvidos no processo.

No Capítulo 5 apresentamos um exemplo prático utilizando os componentes desenvolvidos e incorporados à ferramenta WebManager para executar a gerência da rede da Universidade Tiradentes.

No Capítulo 6 apresentamos os problemas enfrentados, propostas de trabalhos futuros e as conclusões do trabalho.

No Anexo A apresentamos uma introdução sobre o desenvolvimento de softwares utilizando componentes.

# Capítulo 2

## Introdução à Gerência de Redes

Nesse capítulo apresentamos uma explicação sobre a gerência de redes, mostrando sua necessidade e principais aplicações. Na Seção 2.2 demonstramos como está organizada a arquitetura de uma solução de gerência de redes. Na Seção 2.3 explicamos o padrão de gerenciamento utilizando o protocolo SNMP, descrevendo de forma sucinta seu funcionamento, quais seus comandos, os agentes e a MIB. Na seção 2.4 abordamos o Monitoramento Remoto (RMON), explicando sua funcionalidade, aplicabilidade, estrutura e a necessidade de ferramentas para sua utilização.

### 2.1 O que é gerência de rede?

A gerência de rede é uma atividade complexa que envolve o domínio e controle sobre diversos aspectos distintos da administração da rede e dos seus equipamentos, e engloba toda uma arquitetura de hardware e software. Os requisitos para uma boa gerência podem ser divididos em 5 áreas de aplicação segundo o Modelo OSI da ISO [Harnedy, 1998]:

- Gerência de Configuração

Envolve a manutenção dos elementos da rede:

- Coleta de informações de configuração;
- Geração de eventos;
- Atribuição de valores iniciais aos elementos gerenciados;
- Registro de informações;

- Alteração de configuração dos elementos gerenciados;
- *Upgrade* de versões de softwares;
- Início e desligamento dos elementos gerenciados.

○ Gerência de Falhas

Envolve a detecção, registro e correção de falhas na rede:

- Responsável pela detecção, isolamento e conserto em falhas na rede;
- Detecção, isolamento e antecipação de falhas;
- Supervisão de alarmes;
- Ações necessárias ao restabelecimento de elementos com problemas;
- Testes de funcionamento;
- Registro de ocorrências e geração de relatórios.

○ Gerência de Performance

Envolve a manutenção do desempenho, planejamento de capacidade e estimativa do comportamento da rede:

- Seleção de indicadores de desempenho;
- Monitoração e análise de desempenho;
- Planejamento de capacidade;
- Alteração de um modo de operação;
- Estabelecimento de um padrão de normalidade;
- Determinação de tempo de resposta;
- Detecção de tendências e prever o comportamento da rede.

○ Gerência de Segurança

Envolve a proteção dos elementos da rede, garantido que seja mantida a política de segurança estabelecida:

- Criação e manutenção de serviços de segurança;
- Monitoração dos serviços de segurança;
- Manutenção de logs, envolvendo coleta, armazenamento e análise.

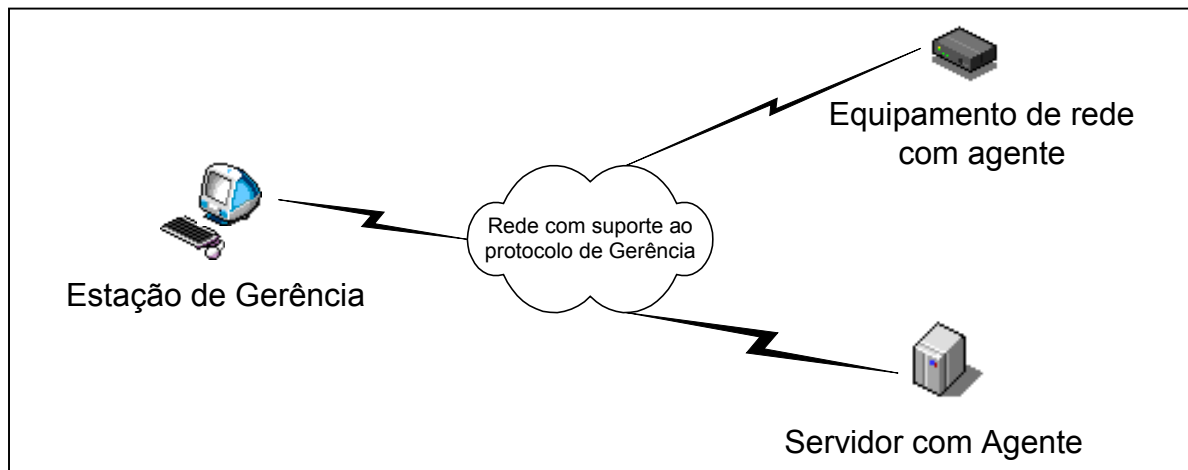
○ Gerência de Contabilidade

Envolve a manutenção de limites de utilização da rede:

- Coleta de informações de utilização;
- Estabelecimento de cotas de utilização;
- Estabelecimento de escalas de tarifação;
- Aplicação das tarifas e faturamento.

## 2.2 Arquitetura de Gerência

Uma solução de gerência envolve a utilização de um conjunto de elementos de hardware e software que devem ser configurados para interagir e possibilitar a implementação desejada. Na Figura 1 podemos identificar um exemplo da arquitetura de uma solução de gerência.



**Figura 1 - Arquitetura de Gerência**

Pode-se separar esses elementos em quatro componentes distintos [Miller,1993] :

### **Elementos gerenciados**

Objeto presente na rede e que será alvo da gerência, deve possuir um software específico para possibilitar seu gerenciamento, o Agente. Os elementos podem ser roteadores, comutadores, hubs, servidores, estações de usuários e qualquer outro equipamento ou software que implemente um agente.



### **Estação de gerência**

Estação de trabalho onde se localizam os softwares de controle dos equipamentos gerenciados, possibilitando uma interface entre o administrador da rede e esses elementos. A estação de gerência é o ponto central da arquitetura de gerenciamento, pois processa e dá acesso às informações da rede. Geralmente é um computador com bastante recursos, como memória e velocidade de processamento, e com várias plataformas e aplicações instaladas como o OpenView da HP [OpenView, 2002] e o Transcend da 3Com [Transcend,2002].

### **Protocolo de Gerência**

Protocolo responsável pela comunicação entre a Estação de Gerência e os agentes.

### **Informações de Gerência**

Dados que podem ser acessados e obtidos utilizando-se os três componentes anteriormente citados. Envolvem, por exemplo, informações de configuração e estatísticas de tráfego.

## **2.3 O Padrão SNMP de Gerência**

### **2.3.1 Origem**

No início, para a detecção de certas falhas o administrador da rede utilizava um comando bastante simples chamado PING, presente no diversos equipamentos que implementam o TCP/IP, e seu funcionamento resume-se a informar quanto tempo leva o envio de pacotes de dados até um determinado elemento da rede. Desta forma a gerência resumia-se a executar pings e tentar achar os problemas analisando o tempo ou ausência de resposta.

Com o crescimento das redes e o conseqüente aumento do número de roteadores e outros equipamentos, esse tipo de técnica de detecção de falhas tornou-se ineficaz. Para resolver essa deficiência no gerenciamento surgiram algumas soluções [Miller,1993]:

- CMIP sobre TCP/IP : o Common Management Information Protocol foi fruto das especificações e estruturas de bancos de dados criados pela ISO.

- High-Level Entity Management System (HEMS) : baseado no modelo OSI, esse sistema utilizava como alicerce o SGMP (Simple Gateway Monitoring Protocol). Protocolo em uso pelos equipamentos que faziam parte da Internet.
- Simple Network-Management Protocol (SNMP) : em maio de 1990 surge o RFC (Request for Comments) 1157, definido o protocolo que é uma melhoria do SGMP.

O SNMP foi implementado em equipamentos de diversas empresas e seu uso foi crescendo cada vez mais, até que o protocolo tornou-se o padrão para troca de informações de gerência.

### **2.3.2 Características e Funcionamento**

Um das características marcantes do SNMP é a definição de uma forma de comunicação simples, padronizada e independente de fabricantes. O protocolo é considerado simples por implementar uma estrutura de pedidos e retorno de informações. A estação de gerência requisita dados aos agentes que se encarregam de enviar uma resposta.

A troca de dados é implementada utilizando um número limitado de comandos Get, GetNext, Set e GetResponse. Além disso, todo o processamento das informações fica a cargo da estação de gerência, o que torna possível criar agentes com pouca complexidade de software.

Os comandos e as informações de gerência trafegam pela rede utilizando o UDP (User Datagram Protocol) e o IP (Internet Protocol). O UDP, é um protocolo de comunicação que executa a troca de mensagens sem muitas informações de controle, o que torna os pacotes transferidos menores e por isso mais rapidamente entregues. A escolha do UDP deve-se basicamente a preocupação com o tamanho dos pacotes e ausência de conexão. Se os pacotes fossem grandes, uma rede com muitos equipamentos poderia ter seus enlaces de dados ocupados apenas com informações de gerência. Já ausência de conexão é relevante pois diminui o overhead com a comunicação evitando também a ocupação dos canais de comunicação.

O UDP, apesar de apresentar essas vantagens, traz como característica a não garantia de entrega dos pacotes, o que não é considerado um problema grave pois como toda a

estrutura baseia-se em solicitações e respostas, caso algum pacote seja perdido nada impede de uma nova solicitação ser emitida pela estação de gerência.

Os agentes mantêm uma base com as informações relevantes sobre o funcionamento do equipamento ou software, fica a cargo de cada fabricante definir e publicar que dados são mantidos pelos seus agentes. A estação de gerência tem a função de coletar e processar esses dados para possibilitar o gerenciamento e o SNMP entra como o meio de comunicação para a transferência dos dados entre a estação de gerência e os agentes. A organização dessa arquitetura pode ser visualizada na Figura 2.

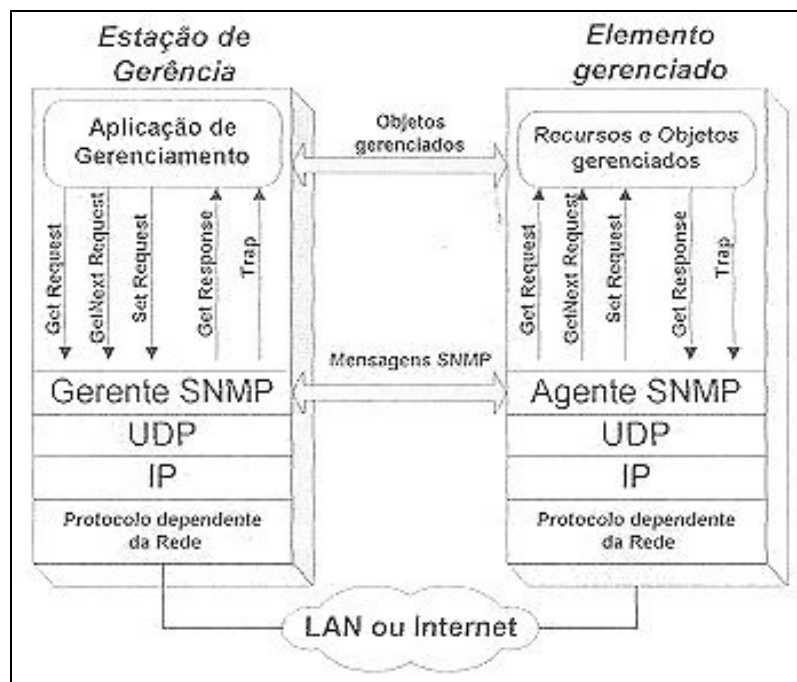


Figura 2 - Arquitetura de Gerência com o SNMP [Miller,1993]

Pode-se então destacar as seguintes características do SNMP:

- Simplifica o desenvolvimento de agentes com suporte ao protocolo;
- Ocupa pouca largura de banda da rede com a transmissão de informações de gerência utilizando pacotes pequenos e sem confirmação de entrega;

- É o mais abrangente possível evitando utilizar operações complexas que dificultem o desenvolvimento de ferramentas de gerência, limitando o número de comandos disponíveis para troca de dados;
- É capaz de transportar vários tipos dados de gerência, possibilitando que as empresas incorporem qualquer informação que queira nos agentes de seus equipamentos e softwares preocupando-se apenas em utilizar os tipos predefinidos pelo protocolo;
- Mantém uma estrutura independente e padronizada para o gerenciamento.

### 2.3.3 Troca de informações

O modelo de implementação de gerência utilizando o SNMP é do tipo cliente/servidor, onde o cliente é a estação de gerência e o servidor é o agente presente no elemento gerenciado.

O agente mantém uma base de dados, a MIB (Management Information Base), contendo vários objetos referentes ao funcionamento e configuração do equipamento ou software da rede. O SNMP trabalha como uma interface entre a estação de gerência e o agente, transportando as informações de gerência contidas na MIB.

Para acessar a MIB, a estação de gerência envia comandos SNMP ao agente que os processa e envia uma resposta. Essas instruções podem ser divididas em três grupos básicos, um de leitura e resposta, um de gravação e um de alarme[Miller, 1993].

#### Comandos de Leitura e Resposta

Servem para a estação de gerência solicitar alguma informação ao agente, são eles :

- *GetRequest*: solicitação de valor de um determinado objeto da MIB.
- *GetNextRequest*: solicita o nome e o valor do próximo objeto da MIB, permitindo uma navegação seqüencial entre os objetos.
- *GetResponse*: é o comando de resposta que o agente envia a estação de gerência após ter recebido um comando de leitura

### **Comando de Gravação**

- *SetRequest*: comando enviado pela estação de gerência solicitando que o agente modifique o valor de determinado objeto da MIB.

### **Comando de Alarme**

- *Trap*: são notificações enviadas pelo agente à estação de gerência com o objetivo de informar que determinado limiar de funcionamento, com valor previamente configurado, foi ultrapassado.

## **2.3.4 Base de informação Gerenciável (MIB)**

Até agora vimos que a gerência de redes é implementada utilizando um protocolo simples e funcionando apenas com troca de comandos de leitura e gravação entre a estação e o agente. Podemos questionar então como esse protocolo vai disponibilizar maneiras de executar todas as tarefas inerentes ao gerenciamento dos equipamentos. Essas tarefas podem ser executadas com auxílio do agente e da MIB ou Base de Informação Gerenciável.

A MIB é uma base de dados virtual das informações de gerência[Harnedy,1998], atualizados pelo o agente, os dados nela refletem o estado atual e histórico de funcionamento do elemento da rede. Nesta base encontram-se informações como: endereços de rede, configurações de funcionamento e contadores de erros. Com base nos dados presentes na MIB a estação de gerência pode criar gráficos, controlar versão de softwares, monitorar erros, alterar valores de configuração ... ou seja, executar as funções de gerenciamento.

Como a MIB reflete o estado de funcionamento do elemento gerenciado, seus valores podem ser alterados para executar configurações. Por exemplo, existe um objeto que representa o endereço IP do equipamento, caso seja necessário modificar esse endereço a estação de gerência envia um *SetRequest* com um novo valor e o agente executará a alteração tanto na MIB como na configuração do equipamento.

## 2.4 RMON - Funcionamento e principais características do RMON

A idéia principal da utilização do RMON é configurar um ou vários agentes com a função específica de coletar dados da rede, e, quando houver necessidade, a estação de gerência acessa esses agentes em busca das informações armazenadas. Pode-se notar que esse modelo é bastante parecido com o que é utilizado pelo gerenciamento convencional, no entanto a coleta de informações é feita de forma independente da estação de gerenciamento.

O equipamento de coleta de informações recebe o nome de **Probe (sonda)** e nele é configurado um Agente RMON. Toda essa troca de informação entre agente e estação de gerência é executada utilizando o protocolo SNMP.

O funcionamento do monitoramento remoto baseia-se na implementação da MIB RMON, na qual estão objetos organizados em tabelas cujos valores são constantemente atualizados pelo agente. Cabe à estação de gerência buscar essas informações utilizando as mensagens SNMP. Na verdade a solução RMON se encaixa como uma extensão à Arquitetura de Gerência com SNMP. A troca de informações é executada da mesma forma, mas o agente é mais funcional pois incorpora algumas tarefas novas e outras antes executadas apenas pela estação de gerência. Tarefas como a manutenção de dados históricos e de informações referentes à troca de pacotes entre hosts.

Existe também um aumento do gerenciamento proativo, pois os probes implementam alarmes que podem ser configurados para alertar situações adversas e com isso prever possíveis erros.

O administrador passa então a utilizar Probes em locais estratégicos da rede, como em sites separados por um enlace de dados WAN ou em partes da rede com falhas no funcionamento. Com essa estratégia são solucionados problemas como [Perkins, 1999]:

### **Enlaces de dados lentos**

Não é mais necessário que a estação de gerência fique ocupando constantemente os enlaces lentos. Agora ela pode coletar os dados da MIB RMON e também configurar limiares de funcionamento para geração de alarmes.

### **Gerenciamento *off-line***

Caso falhe a comunicação entre os elementos gerenciáveis e a estação de gerência, os agentes RMON vão continuar coletando informações e cadastrando eventos. Quando a comunicação for restabelecida a estação de gerência pode verificar o que ocorreu consultando as informações armazenadas nos Probes.

### **Históricos de funcionamento**

Dados históricos agora são mantidos pelos probes, liberando a estação de gerência de ficar constantemente solicitando informações aos agentes para conseguir criar e manter uma coleção desses dados. Quando necessário, a estação pode solicitar todo o histórico mantido pelo agente RMON.

### **Visão geral da rede**

Com as informações mantidas pelo Probe RMON, o gerente de rede consegue mais facilmente visualizar o funcionamento da rede como um todo. Identificando por exemplo, as áreas com maior colisão, quais os enlaces mais problemáticos e onde estão os gargalos.

## **2.4.1 Probes RMON**

O Probe é o principal elemento da gerência baseada em RMON, esse equipamento é responsável por coletar e manter os dados das redes em que está conectado, dados que posteriormente poderão ser acessados e analisados através da estação de gerência.

O probe funciona com uma interface de rede em modo promiscuo, ou seja, capturando todos os pacotes que trafegam pelo canal de comunicação. Esses pacotes então são transferidos para o Agente RMON, que utilizando o processador e a memória do equipamento, processa os pacotes coletados e mantém as variáveis da MIB RMON. É a capacidade da memória que determina a quantidade de entradas que podem ter as tabelas com as variáveis RMON.

O Probe pode ser implementado basicamente de duas maneiras:

## **Standalone**

São utilizados equipamentos com a função específica de coleta de dados RMON, geralmente não passam de caixas com uma ou mais interfaces de rede onde todo o acesso e configuração é feito utilizando SNMP. Algumas das principais empresas do ramo como a HP, Axon Networks e Armon Networking oferecem equipamentos desse tipo no mercado.

## **RMON incorporado em elementos ativos**

Os elementos da rede que geralmente atuam transferindo os dados, como comutadores, roteadores e hubs, passam a implementar a MIB RMON e começam a trabalhar também com a função de probe.

A principal diferença entre os dois equipamentos é em relação ao processamento das informações e ao custo de implementação:

- Processamento das informações

Quando colocamos um equipamento de rede para também funcionar como probe RMON, seus recursos passam a ser compartilhados entre a sua real função e a função de probe, o que diminui sua performance. Com probes standalones todos os recursos do equipamento destinam-se ao processamento dos dados de gerência e portanto não há problemas em diminuição de performance da rede.

- Custo de implementação

A principal desvantagem do equipamento standalone é que eles são caros e tem “apenas” a função de gerenciamento. Os administradores de rede tendem a preferir comprar elementos ativos com capacidade adicional de implementar o RMON, pois aumentam o custo/benefício com a compra de um equipamento com duas funções.

Para uma solução ideal com RMON deveríamos colocar um Probe em cada segmento da rede, só que devido ao custo de implementação os administradores definem locais específicos para instalar esses equipamentos, geralmente os mais críticos, como por exemplo o backbone da rede e o segmento com os servidores.



## 2.4.2 Descrição dos Grupos do RMON

Os dados da MIB RMON foram organizados em tabelas e essas tabelas em 10 grupos distintos, cada um com uma função específica. Podemos verificar esses grupos na Tabela 1 [Perkins, 1999]:

Grupo	Função	Elementos
Statistics	Contém estatísticas de rede para cada interface do equipamento	Número de Pacotes descartados e enviados, numero de bytes enviados, numero de pacotes de broadcast e de multicast, Quantidade de erros de CRC, runts, giants, fragmentos, jabbers, colisões, e contadores de pacotes com tamanhos entres 64-128, 128-256, 256-512, 512-1024, e 1024-1518 bytes
History	Grava periodicamente amostras da rede, criando um histórico de funcionamento	Período de amostragem, quantidade de amostras e itens da amostra
Alarm	Periodicamente compara dados da MIB com limites previamente estabelecidos, se esse limite for ultrapassado um evento é gerado	Inclui a tabela de alarmes contendo tipo do alarme, intervalo entre testes, limite inferior e limite superior. Obs. : Requer que o grupo Events seja implementado
Host	Contem estátísticas relacionadas a cada host descoberto na rede.	Endereço do host, pacotes e bytes recebidos e transmitidos, bem como pacotes de broadcast, multicast e erro.
HostTopN	Mantém os hosts que estão no topo de uma lista ordenada por alguma variável presente na tabela Host.	Estatísticas, hosts, inicio e fim da amostra e duração
Matrix	Mantém estatísticas de conversação entre dois endereços. A cada nova conversa identificada uma nova entrada na tabela é	Pares de endereços fonte e destino e pacotes, bytes e erros para cada par

Grupo	Função	Elementos
	criada.	
Filters	Possibilita a criação de filtros para a captura de pacotes	Tipo e expressão de filtro e expressão condicional para ordenar os pacotes coletados
Packet Capture	Possibilita a captura de pacotes	Tamanho para os pacotes capturados, status, numero de pacotes capturados
Events	Controla a geração e notificação de eventos	Tipo de evento, descrição e ultima vez que foi enviado

**Tabela 1 – Lista de Grupos RMON**

Não é necessária a implementação de todos os grupos para considerar um equipamento capaz de possibilitar o gerenciamento RMON. E mesmo que um equipamento implemente todos os grupos, é comum existir configuração onde o agente mantém apenas algumas das tabelas possíveis, isso se deve ao fato de que a manutenção dessas tabelas ocupa memória e recursos de processamento. Cabe ao administrador da rede escolher que tabelas o probe deve manter.

Na Figura 3 pode ser identificada a MIB RMON e onde ela se encaixa na árvore da MIB de gerenciamento SNMP.

Diagrama da árvore MIB do RMON1

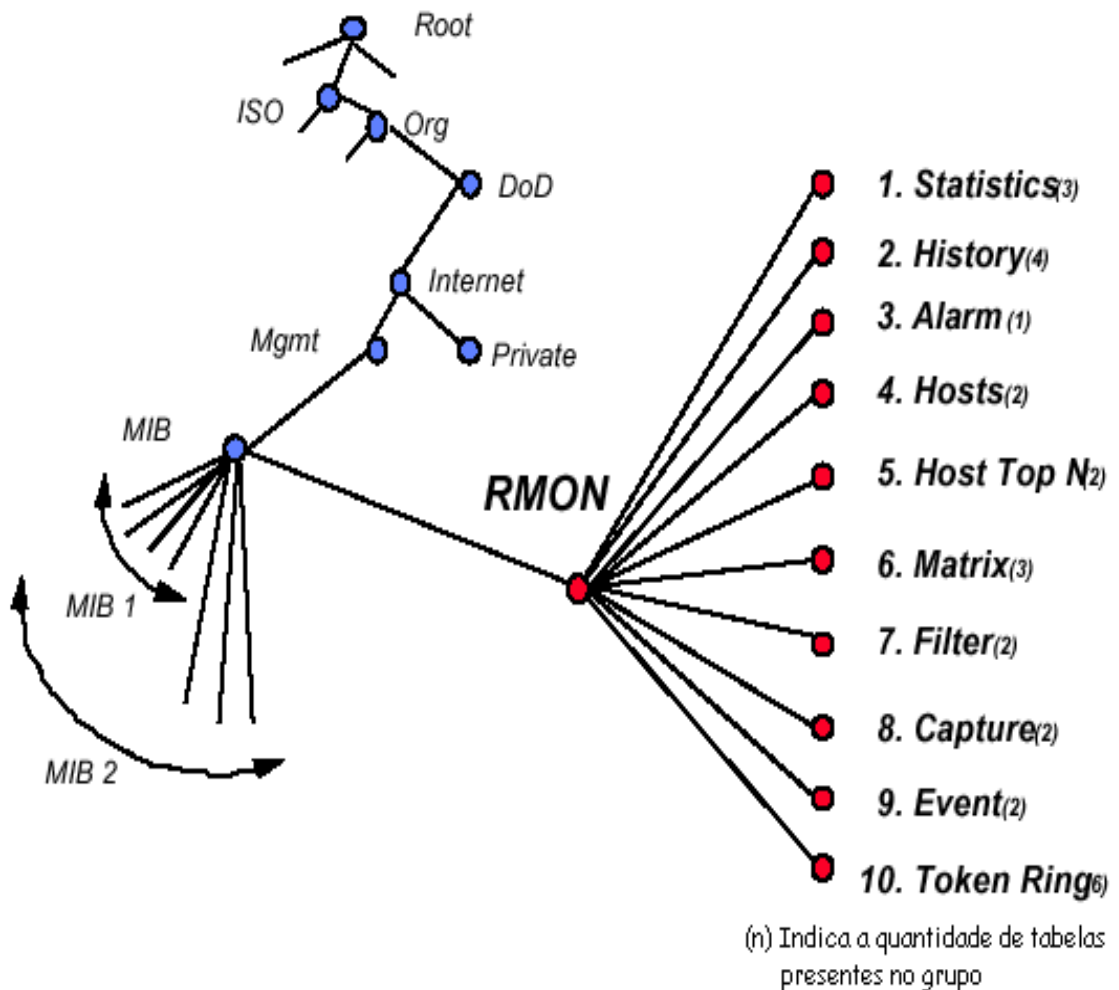


Figura 3 – Árvore Mib RMON [3Com, 1999]

### 2.4.2 Necessidade de Ferramentas

A gerência de redes é executada com o auxílio de ferramentas, como sistemas de representação de mapas da rede, geradores de gráficos, programas para configuração dos elementos ativos e outras. Essas ferramentas são instaladas na estação de gerência, e possibilitam uma interface entre o administrador e as informações gerenciadas. Em se tratando do RMON, essas ferramentas são essenciais devido aos seguintes pontos:

- Grande quantidade de dados

Dividido em 10 grupos e 27 tabelas, e essas tabelas em várias colunas e linhas, o RMON é uma fonte de muitos dados. Por exemplo, uma tabela de estatísticas de um computador com 26 portas modelo 1100 da 3Com pode chegar a ter 4255 linhas ou mais para cada interface.

- Informações de baixo nível

As informações presentes no probes são de baixo nível, uma coleção de vários números que para serem entendidos precisam ser analisados e tratados para ter valor como informação de gerência. Por exemplo, o endereço MAC de uma interface de redes é apresentado como um grupo de valores hexadecimais, o que torna a identificação da interface mais difícil do que se fosse mostrado ao usuário o seu endereço IP.

- Difícil de analisar apenas executando MIB Browser

MIB Browser é a técnica de verificar os dados de gerência navegando pela árvore de informações da MIB lendo variável a variável. Como a MIB RMON está organizada em tabelas e utiliza índices para relacioná-las, essa técnica é muito improdutiva.

Por exemplo, para ler todos os dados de uma tabela MatrixSDTable de 4255 linhas utilizando MIB browser e considerando que para cada coluna, 6 ao todo, de todas as linhas teria que ser executado pelo menos um comando getnext, ao final seriam necessários 25530 comandos.

- Necessidade de uma forma de tratar os dados para facilitar a compreensão

O ser humano consegue visualizar e extrair conclusões mais facilmente de um gráfico do que de um grande número de valores, e para melhor aproveitar as informações contidas no RMON devem existir ferramentas que tratem os dados e gerem gráficos para representá-los.

# Capítulo 3

## Componentes para Gerência RMON

Como vimos no capítulo passado, a gerência RMON necessita de ferramentas para conseguir acessar e tirar proveito da vasta gama de informações encontradas na MIB RMON. Neste capítulo enumeramos os requisitos que devem ser implementados e mostramos também a arquitetura de componentes que irão facilitar a criação dessas aplicações.

Os detalhes da implementação serão mostrados no Capítulo 4.

### 3.1 Requisitos dos componentes

Com base nos problemas levantados no Item 2.4.2, podemos identificar os seguintes requisitos necessários à uma aplicação de gerência RMON:

- R1. Para uma maior abrangência e funcionamento em arquiteturas abertas, deve ser capaz de acessar os dados da MIB seguindo o padrão Internet de gerenciamento, ou seja utilizando o protocolo SNMP para a comunicação com os agentes;
- R2. A coleta dos dados deve ser altamente configurável, podendo definir as interfaces alvo, intervalo de tempo entre amostras e variáveis específicas de um determinado grupo e tabela RMON;
- R3. Manter um histórico dos dados coletados, armazenando dados passados para possibilitar a busca de informações em períodos específicos;

- R4. Gerenciar as tabelas de controle dos grupos, mantendo a correlação entre os índices RMON e as Interfaces de rede dos equipamentos;
- R5. O armazenamento dos dados deve ser feito de forma organizada e capaz de gerenciar informações como: equipamento, data, hora, interface, índice, grupo, tabela, nome da variável e valor de um dado coletado;
- R6. Apresentar informações sobre dados coletados em um determinado intervalo de tempo de um dia.
- R7. Gerar gráficos históricos ou atuais de forma que possam ser acessíveis através dos mais diferentes tipos de aplicações. Os gráficos devem representar os dados RMON em um nível mais elevado, ou seja processando e interligando valores de variáveis para gerar informações úteis.

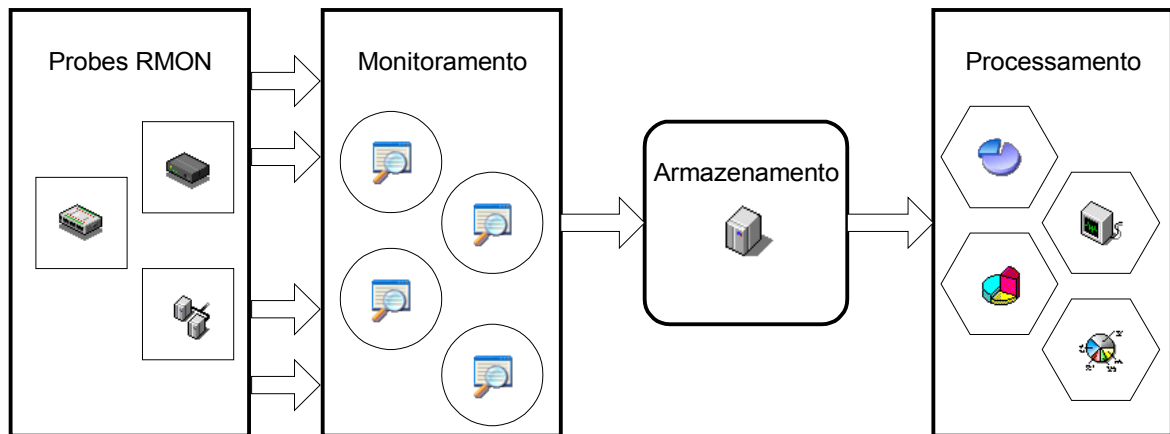
## **3.2 Arquitetura dos Componentes**

Seguindo os requisitos apresentados dividimos as atividades envolvidas no monitoramento RMON em:

- Coleta;
- Armazenamento dos dados;
- Processamento das informações;
- Comunicação;
- Configuração.

Um equipamento RMON teria vários monitores coletando seus dados e, depois de coletados, os mesmos seriam passados para componentes responsáveis pela manutenção do armazenamento.

Uma outra parte da arquitetura seria destinada ao tratamento dos dados coletados, contendo componentes para processamento desses dados e geração de gráficos. De uma forma geral e simplificada a arquitetura pode ser representada pela Figura 4.



**Figura 4. Arquitetura dos Componentes**

### 3.2.1 Coleta

A etapa de coleta tem como principais componentes o Probe RMON e os Monitores de dados. Cada um deles apresenta também um módulo de configuração para definir seus parâmetros de funcionamento.

O probe RMON representa o equipamento de rede e seus parâmetros de configuração são:

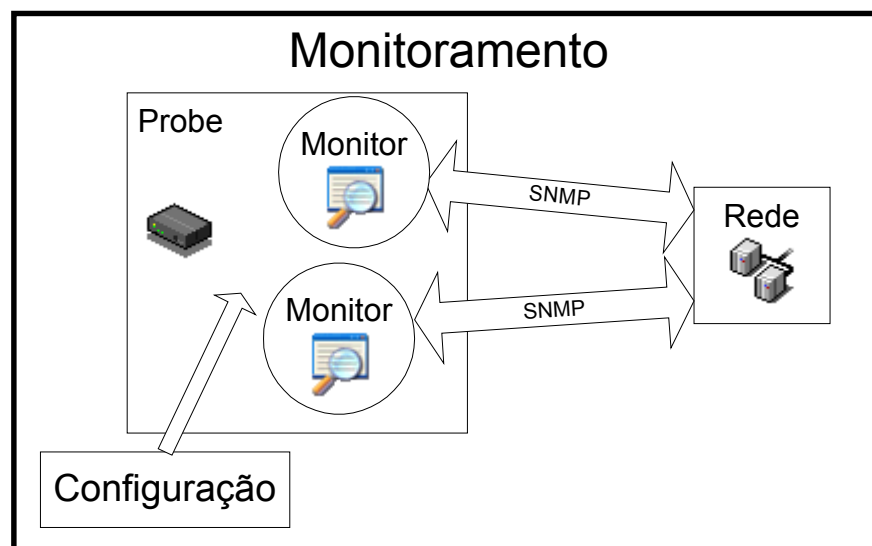
- Nome do equipamento;
- Endereço IP;
- Comunidade SNMP;
- Grupos RMON que implementa;
- Informações de identificação.

Para cada probe RMON temos associado um ou vários Monitores para a coleta de dados. Esses monitores apresentarão as seguintes informações de configuração:

- Probe RMON a que está associado;
- Grupo RMON;

- Tabela RMON;
- Variáveis;
- Interfaces;
- Período entre coletas.

Desta forma, em intervalos de tempo predeterminados, o Monitor pega as informações de acesso com o componente que representa o probe e utilizando SNMP envia uma solicitação para o equipamento de rede. Quando recebe o retorno da solicitação, o monitor envia os dados aos componentes de armazenamento. Representamos esse funcionamento com a Figura 5.



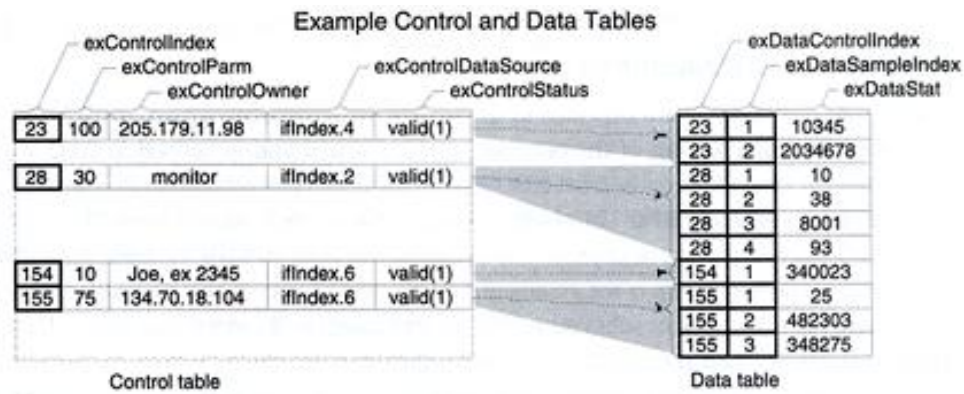
**Figura 5. Funcionamento do Monitor**

### 3.2.1.1 Funcionamento do Monitor

Os grupos RMON apresentam sempre dados de controle para definir seu funcionamento enquanto que as variáveis coletadas ficam distribuídas em uma ou mais tabelas de dados.



O dados de controle podem ou não ficar em uma tabela separada e especificam os parâmetros do monitoramento, como o período entre amostras, número de linhas nas tabelas de dados e o qual interface deve ser monitorada.



**Figura 6 – Tabelas de Controle e Dados [Perkins, 1999]**

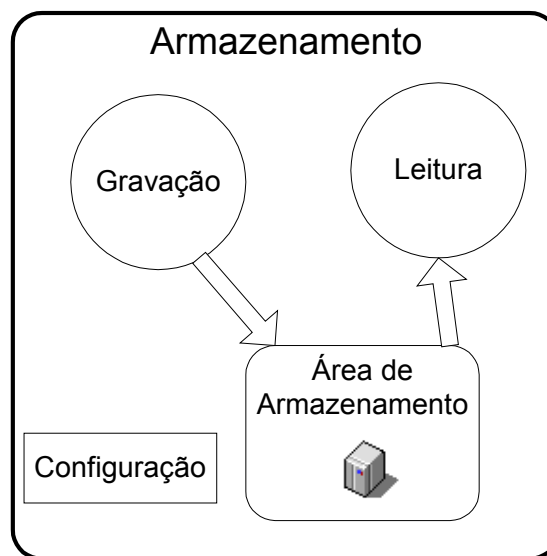
Na figura 6 podemos verificar um exemplo de relacionamento entre as tabelas RMON. A primeira delas é uma Control Table, armazena os dados de configuração do probe, e a segunda é uma Data Table, utilizada para os dados. Para acessar os valores coletados para a interface 2 do equipamento, faz-se necessário identificar qual o ControlIndex desta interface (28) na Control Table e depois executar a leitura das linhas com esse mesmo índice (linhas 3 a 6) na data table.

Desta forma ao escolher uma interface para monitoramento de um determinado equipamento, a ferramenta de gerenciamento RMON deve ter o cuidado de localizar o índice específico da interface na tabela de controle e utilizar esse índice para a coleta das informações.

O componente RMON de monitoramento deve ser capaz de tratar essas tabelas de controle de uma forma genérica, para conseguir trabalhar independente do grupo que está sendo monitorado.

### 3.2.2 Armazenamento

Os componentes de armazenamento devem ser implementados com a função de gravar e ler os dados RMON monitorados e também devem ser passíveis de configuração, podemos ver uma representação dessa estrutura na figura 7.



**Figura 7. Armazenamento dos Dados**

O ponto crucial para o armazenamento é a organização dos dados, pois deve ser utilizada uma estrutura padronizada capaz facilitar o acesso e identificação do que foi monitorado.

O armazenamento deverá sempre envolver os seguintes dados :

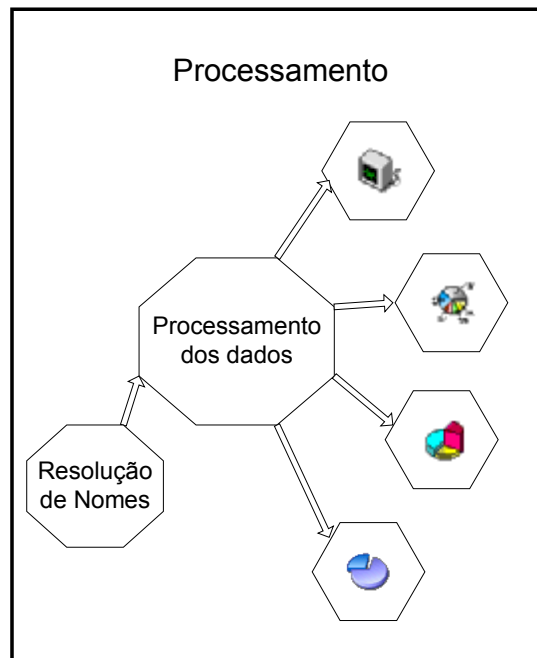
- Equipamento gerenciado;
- Data da Coleta;
- Grupo RMON;
- Tabela;
- Variável;
- Hora;
- Interface;
- Índice de Controle;
- Valor.

### 3.2.3 Processamento das informações

O processamento das informações é um modulo bastante importante pois será o responsável por tratar os dados coletados e transformá-los em informações úteis para o gerente da rede.

O tratamento em relação aos grupos RMON agora será feito de forma diferenciada, pois cada grupo pode gerar um tipo diferente de informação.

Como podemos visualizar na Figura 8, esse grupo de componentes será dividido em três: Processamento dos dados, Resolução de nomes e geração de gráficos.



**Figura 8. Processamento das Informações RMON**

#### 3.2.3.1 Processamento dos Dados Coletados

Como as informações são armazenadas e lidas de forma genérica esses componentes devem tratá-las para extrair informações relevantes específicas por grupos.

Os dados são solicitados ao componente de Armazenamento e, dependendo das variáveis e do grupo envolvido, diversos tipos de algoritmos podem ser desenvolvidos para processar esses dados.

Por exemplo, podemos implementar um componente de processamento para os dados do grupo *statistics* capaz de analisar as informações coletadas e extrair uma média de erros em um determinado intervalo de tempo.

### 3.2.3.2 Resolução de Endereços

Os endereços mantidos pelo RMON são todos MAC, valor numérico que identifica uma interface de rede ethernet, o que dificulta o trabalho do gerente de rede. Esse componente é responsável por manter a resolução dos endereços MAC para IP ou Nome do equipamento e fornecer essa informação para os componentes de processamento.

As informações mantidas são:

- Endereço MAC;
- Endereço IP;
- Nome do equipamento;
- TimeStamp.

Os endereços e o nome devem ser mantidos de maneira tal que possam ser executadas pesquisas onde a partir de qualquer um dos três seja possível chegar nos outros dois. E o *TimeStamp* servirá para controle de expiração e atualização do cadastro dos endereços coletados.

Para executar essa tarefa os dados de endereços serão mantidos de duas formas: coletados dos equipamentos de rede ou cadastrados manualmente.

Os equipamentos de roteamento mantêm uma tabela chamada *atTable* (.iso.org.dod.internet.mgmt.MIB-2.at.atTable), contendo como variáveis listas de pares de endereços MAC e IP. O componente de resolução deve manter uma lista desses equipamentos e utilizando SNMP coletar os dados encontrados nessas tabelas. Desta maneira serão conseguidos pares de endereços MAC e IP, e os nomes dos equipamentos são configurados através de pesquisas no servidor de DNS da rede.

O cadastro manual dos dados de endereço (Nome, IP e MAC) pode ser feito através de um arquivo de configuração em formato texto, onde as informações são digitadas utilizando um editor qualquer, ou de ferramentas desenvolvidas com esse propósito específico, como por exemplo uma página Web para entrada dos dados.

O componente deve implementar as seguintes informações de configuração:

- Lista de equipamentos para coleta;
- Intervalo entre coletas;
- Tempo expiração de um endereço coletado.

O componente também deve implementar a persistência dos dados.

### **3.2.3.2 Geração de Gráficos**

Utilizando as informações geradas pelos componentes de processamento dos dados, podemos criar gráficos para facilitar o entendimento por parte do gerente da rede.

Os gráficos também são específicos por grupos e devem ser apresentados de maneira tal que sua reutilização seja facilitada. A melhor opção seria gerar arquivos de figuras, pois esses arquivos podem ser reaproveitados em diversos tipos de aplicações.

Por exemplo, podemos criar um componente para gerar um gráfico como 10 maiores transmissores de uma rede, esse gráfico seria apresentado em forma de uma figura que poderíamos aproveitar em uma aplicação de gerência, documento ou página web.

### **3.2.4 Comunicação**

A troca de dados entre os processos de Coleta e Armazenamento será executada de forma assíncrona. A idéia é utilizar uma estrutura de Produtores e Consumidores, onde os Monitores funcionam como os Produtores dos dados e o Armazenamento como Consumidor.

A comunicação deve então ser estabelecida através de um canal inteligente onde os produtores e consumidores possam ser acoplados. O canal funcionaria como um gerenciador das trocas de dados, ou seja, quando um monitor produzir um dado novo e lançá-lo no canal o consumidor (Armazenamento) é avisado e retira esses dados.

### **3.2.4 Configuração**

Como foi visto nos itens anteriores, os componentes da arquitetura têm várias informações de configuração para determinar seu funcionamento. O processo de configuração

deve ser de responsabilidade de um componente específico e capaz de ler os parâmetros de configuração e definir o funcionamento dos outros componentes da arquitetura.

### **3.3 Arquitetura dos Componentes RMON acoplados a uma solução de Gerência.**

Os componentes RMON devem possibilitar que aplicações de Gerência de Redes possam utilizá-los para incorporar funcionalidade RMON à sua estrutura. Para garantir que essa funcionalidade possa ser alcançada, a arquitetura dos componentes seguirá como modelo um Framework de aplicação de gerência já existente e comprovadamente funcional, o WebManager.

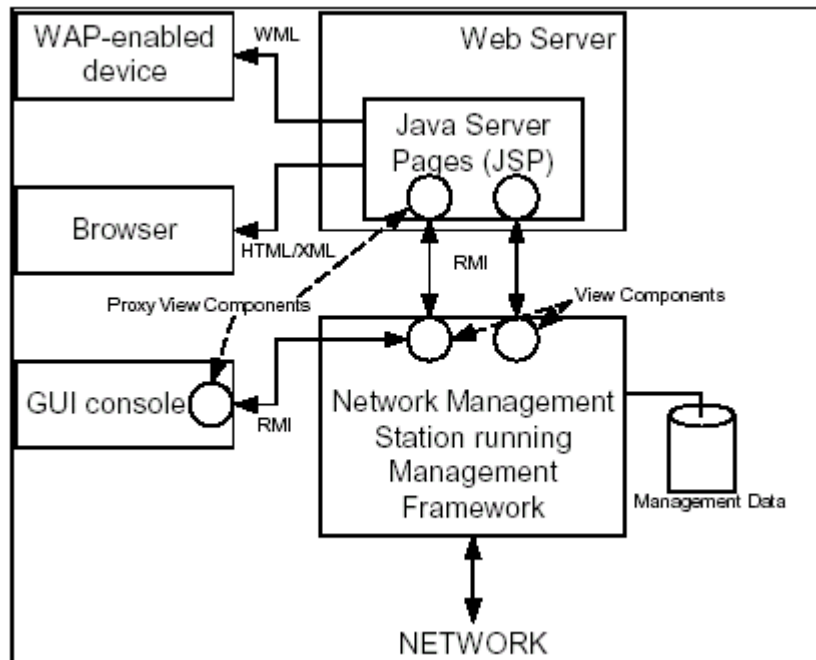
#### **3.3.1 WebManager**

O Webmanager é uma ferramenta de gerência desenvolvida pelo Departamento de Sistemas e Computação da Universidade Federal da Paraíba.

Construída como um Framework baseado em componentes, a aplicação tem como principais características:

- Interface Web;
- Suporte ao SNMP;
- Navegação através de mapas hierárquicos das redes;
- Apresentação do status dos equipamentos nos mapas através de cores;
- Alarmes configuráveis;
- Portabilidade;
- Apresentações gráficas dos dados coletados;
- Altamente configurável;
- Extensível.

A aplicação apresenta-se em um modelo de camadas.



**Figura 9. Arquitetura do WebManager [Sauvé, 1999]**

Como pode ser visto na Figura 9, o software cliente, caracterizando-se como a primeira camada, apresenta-se na forma de um Browser ou ferramenta Wap acessando um servidor web capaz de compilar páginas jsp utilizadas para disponibilizar dados conseguidos através de componentes de visualização (View Components). Podemos considerar como a segunda camada o conjunto formado pelo servidor web e a estação de gerência rodando o webmanager. Por fim tem-se a rede como terceira camada.

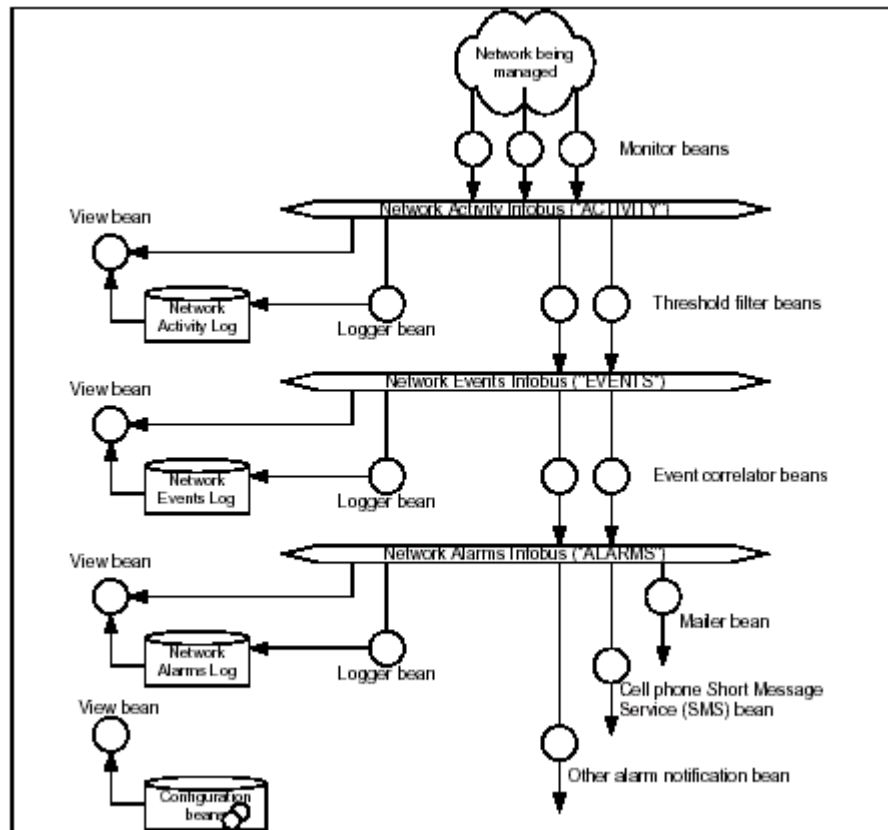
A ferramenta é desenvolvida sobre a tecnologia Java e tem JavaBeans como os componentes do framework.

### 3.3.2 Arquitetura do WebManager

A arquitetura do WebManager é baseada em um modelo criado a partir de componentes produtores e consumidores interligados entre si através de barramentos.

O barramento que interliga os componentes é um infobus, e funciona como um retransmissor de informações para todos os componentes que estão conectados a ele. Desta forma um componente produtor lança dados no barramento e os consumidores conectados recebem esses dados.

A arquitetura pode ser visualizada utilizando a Figura 10.



**Figura 10. Framework de Gerência. [Sauvé, 1999]**

Componentes monitores utilizando SNMP acessam os equipamentos de rede e coletam dados que são lançados no barramento “Activity”.

Conectados ao “Activity” estão o Loggers e o Threshold. O Logger é responsável por gravar os dados coletados e informações geradas pela aplicação e aparecem conectados a todos os barramentos. O Threshold processa os dados verificando limites predefinidos de funcionamento, caso algum limite seja ultrapassado ele gera um ou mais eventos e os lança no infobus “Events”.

Conectado ao “Events”, além dos Loggers, estão os Event Correlator Beans, esses componentes são responsáveis por tentar identificar situações críticas através da interligação de eventos. Caso uma situação crítica seja identificada um alarme é gerado e lançado no barramento “Alarms”.

Conectados ao “Alarms” estão os componentes responsáveis por notificar os alarmes, seja por meio de correio eletrônico ou mensagens para celulares.

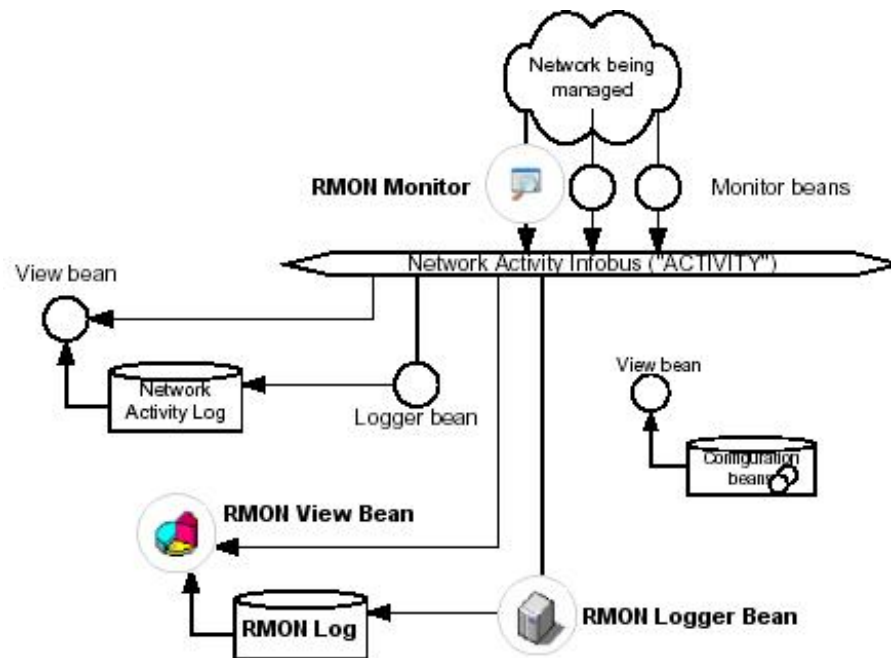
Na estrutura existem ainda os View Beans e os Configuration Beans. O view bean é componente que acessa as informações gravadas pelos Loggers e apresenta essas informações



aos clientes. O configuration Bean é o componente responsável por ler as informações de configuração armazenadas em um arquivo XML e criar os componentes da arquitetura no momento da inicialização da aplicação.

### 3.3.3 Arquitetura RMON incorporada ao WebManager

Adaptando as necessidades dos componentes RMON à estrutura do WebManager, temos uma configuração como a da Figura 11.



**Figura 11. Arquitetura RMON adaptada ao Modelo do WebManager**

Herdando as características do WebManager, os componentes se adaptarão a sua estrutura e passarão a funcionar como uma extensão RMON da Aplicação de Gerência. Essas características serão:

- As etapas de coleta, armazenamento e processamento dos dados RMON se tornarão Beans Monitor, Logger e View respectivamente;

- As configurações dos componentes RMON serão feitas pelo configuration Bean do WebManager utilizando XML;
- Os componentes serão desenvolvidos na linguagem JAVA;
- Os gráficos serão gerados pelos componentes View Bean, e utilizarão como padrão de resposta a criação de arquivos de figuras;
- Deverão ser criadas páginas Web para interação com o usuário utilizando JSPs;

# Capítulo 4

## Implementação

Nesse capítulo apresentamos detalhes sobre a implementação dos componentes. Na Seção 4.1 demonstramos como foram organizados os componentes desenvolvidos, na Seção 4.2 apresentamos as APIs desenvolvidas por outros e utilizadas na implementação, na Seção 4.3 descrevemos as classes desenvolvidas e mostramos os detalhes sobre a implementação e na seção 4.4 apresentamos as modificações executadas nos arquivos do WebManager para conseguir incorporar os componentes RMON.

### 4.1 Pacotes

No capítulo passado descrevemos a arquitetura base para o desenvolvimento dos componentes e também demonstramos a maneira que essa arquitetura se encaixa no Framework de Gerência de redes desenvolvido para o WebManager. A organização dos componentes foi criada seguindo as atividades levantadas no item 3.1, Coleta, Armazenamento, Processamento, Comunicação e Configuração. Na Tabela 2 temos a lista dos pacotes desenvolvidos para essas etapas.

<b>Etapa</b>	<b>Pacote</b>	<b>Função</b>
Configuração	rmon.config	Configuração dos equipamentos RMON gerenciados
Coleta	rmon.monitor	Coletar os dados RMON dos Equipamentos gerenciados
Armazenamento	rmon.log	Gravar as informações coletadas
Processamento	rmon.loader	Ler dados RMON previamente armazenados
Processamento	rmon.addressT	Fornecer a funcionalidade de mapear endereços MAC, IP para o nome do equipamento
Processamento	rmon.graphs	Criar gráficos com os dados RMON

**Tabela 2 – Componentes Desenvolvidos**

## 4.2 APIs Utilizadas

Neste item, descrevemos as APIs que foram utilizadas no processo de desenvolvimento e que foram desenvolvidas por outros. Estas APIs estão disponíveis para download da Internet e têm sua utilização liberada sem custos.

### 4.2.1 API MonarchCharts

Para auxiliar na criação dos gráficos desenvolvidos, empregamos a API MonarchCharts fornecida pela empresa Singleton Labs [Singleton, 2000]. Desta API, os seguintes pacotes foram utilizados:

- It.monarch.chart: disponibiliza classes para o tratamento das informações que serão utilizadas para criar os gráficos.
- It.monarch.chart.chart3D: disponibiliza classes para a criação de gráficos em 3D, incluídos gráficos em Barras, em Linhas e em torta.
- It.monarch.chart.legend: disponibiliza classes para a inclusão de textos e legendas nos gráficos

## 4.2.2 API AdventNet SNMP

Para executar a monitoração dos equipamentos de rede com o protocolo SNMP, empregamos a API fornecida pela empresa AdventNet, Inc. [AdventNet, 2000]. Os pacotes utilizados foram:

- `com.adventnet.snmp.snmp2`: provê classes para a utilização dos comandos do protocolo SNMP, possibilitando também a definição de parâmetros de configuração da comunicação entre o equipamento e o componente de monitoração.
- `com.adventnet.snmp.mibs`: disponibiliza classes para o suporte de MIB/SMI. Possibilitando interpretar as informações sobre os tipos e organização dos dados disponíveis para o Agente SNMP.
- `com.adventnet.snmp.beans.SnmpTarget`: provê uma representação de um equipamento com agente SNMP, facilitando a comunicação e coleta de dados entre o monitor de rede e o equipamento.

## 4.3 Classes Desenvolvidas

Neste item, mostramos as classes desenvolvidas e descrevemos alguns detalhes da implementação para cada um dos pacotes dos componentes.

### 4.3.1 Pacote `rmon.config`

Este pacote envolve classes para representar a configuração dos probes rmon. As seguintes classes foram desenvolvidas:

- `RMONConfig` – executa funções relacionadas à configuração do probe RMON. O principal método implementado é o `getControlIndex`, como explicamos no Capítulo 3 Seção 3.2.1.1 os grupos RMON utilizam tabelas de controle para manter as configurações de coleta, esse método pesquisa nessas tabelas qual o índice de dados de uma determinada interface. Esse índice é utilizado pelo monitor para executar a coleta dos dados.

- RMONDevice – representa o equipamento com suporte ao RMON, trazendo informações de configuração comuns a elementos de rede, como nome e endereço IP, e mais dados específicos sobre o RMON, como os grupos implementados pelo equipamento e uma lista de objetos(RMONMonitor) representando quais os monitores que estão apontando para ele.
- RMONDeviceTemplate – template, tipo de classe que armazena um conjunto de propriedades comum a um grupo de objetos, que representa equipamentos com suporte ao RMON.

### 4.3.2 Pacote rmon.monitor

Responsável pela coleta dos dados RMON utilizando o protocolo SNMP. As classes desenvolvidas foram:

- RMONDadosMonitor – representa as configurações de um monitor RMON, como qual o grupo monitorado, quais as interfaces e que variáveis devem ser coletadas.
- RMONMonitor - representa um monitor RMON e é responsável por executar a coleta de dados utilizando SNMP. Essa classe tem sempre um objeto (RMONDadosMonitor) com suas configurações e outro objeto (RMONDevice) representando o equipamento que é responsável por monitorar.

### 4.3.3 Pacote rmon.log

Responsável por armazenar os dados RMON coletados pelo Monitores. Neste pacote a única classe desenvolvida foi o RMONActivityLogger.

A classe utiliza o diretório “webmngn\data\activity\iso\org\dod\internet\mgmt\MIB-2\rmon” para gravar as informações coletadas pelos monitores e dentro desse diretório, como forma de organização, são criadas subpastas da seguinte forma:

*“nome do equipamento monitorado”\”data da coleta”\”Grupo RMON”\”Tabela Monitorada”\”variável monitorada”.mon*

Por exemplo, para gravar dados da variável de pacotes da tabela matrixSDTable do grupo Matrix para um equipamento com o nome de S2BIB001, o seguinte arquivo seria utilizado:

*webmngr\data\activity\iso\org\dod\internet\mgmt\MIB-2\rmon\swBiblioteca\2001-10-16\matrix\matrixSDTable\matrixSDPkts.mon*

Dentro do arquivo, as informações foram organizadas de forma que cada linha representa um valor coletado. A estrutura da linha fica a seguinte :

*“Hora da coleta”*: *“interface”*: *“índice RMON da interface”*: *“Valor Coletado”*

Organizado desta forma, garantimos que os requisitos de armazenamento levantados no Capítulo 3 seção 3.2.2 serão atingidos e que o componente se adapta a maneira como o WebManager armazena os dados coletados em seu diretório Data.

#### **4.3.4 Pacote rmon.loader**

Esse pacote é responsável pela leitura dos dados gravados pelo logger e é bastante utilizado pelos pacotes gráficos. As classes desenvolvidas são:

- RMONData : classe para representa um valor RMON coletado, inclui as informações de monitoração que são : variável, interface, índice, valor e hora da coleta.
- RMONViewBean : classe que executa a leitura de uma variável coletada e retorna um conjunto de objetos RMONData. Recebe por parâmetro o equipamento, a variável, a interface, a data e o intervalo de tempo que deverão ser lidos.

#### **4.3.5 Pacote rmon.addressT**

Esse pacote tem a função de executar a resolução dos endereços IP, MAC e Nome dos equipamentos da rede. As classes desenvolvidas nesse pacote são:

- AddressMapData : classe que representa uma entrada de valor mapeado formada pelo endereço IP, endereço MAC, Nome e timestamp da coleta.
- AddressMap : classe que executa todo o processamento relacionado com a coleta e tradução de endereços. Utilizando SNMP e seguindo o que foi definido no Capítulo 3 Seção 3.2.3.2, os valores relacionando endereço MAC e IP são coletados da tabela atTable presente nos equipamentos de roteamento e a resolução

de IP para nome é feita utilizando o DNS. A classe também disponibiliza métodos para a pesquisa de valores coletados por IP, MAC ou Nome.

### 4.3.6 Pacote **rmon.graphs**

Esse pacote apresenta classes que possibilitam a geração de gráficos com os dados processados e está subdividido em mais 3 pacotes, `hosts`, `matrix` e `statistics`. Todos os gráficos apresentam as seguintes características em comum:

- São gerados para imagens em arquivos em disco;
- Podem ser configurados para utilizar informações de um período específico do dia através de dos paramentos de hora inicial e hora final.
- Quando necessário, utilizam o pacote `rmon.addressT` para criar os gráficos com os nomes dos equipamentos, traduzindo o endereço MAC para Nome.
- Recebem como parâmetro o número da interface do equipamento para que deverá gerar o gráfico.

#### 4.3.6.1 **rmon.graphs.hosts**

Possibilita criar gráficos para as variáveis do grupo `hosts`. Os gráficos criados representam os dez mais de uma determinada variável em um período de tempo. Utilizando esse pacote podemos criar gráficos como os dez host que mais transmitem na rede ou os dez equipamentos que mais geram pacotes com erros. As classes implementadas foram:

- `RMONHostsData`: estende as funções do `RMONData` para melhor representar as informações do grupo `Hosts` e para facilitar a criação dos gráficos. Apresenta métodos para comparar os valores coletados e possibilitar a ordenação dos mesmos.
- `RMONHostsViewBean`: estende a classe `RMONViewBean` para possibilitar a transformação dos objetos lidos dos arquivos de log em `RMONHostsData` e para executar a ordenação desses valores.
- `RMONGraphicPlotter`: cria um gráfico representando os dez campeões em determinada variável.



#### 4.3.6.2 rmon.graphs.matrix

Possibilita a criação de gráficos apresentando a interligação de transmissores e receptores de pacotes na rede, demonstrando os equipamentos que mais transmitem e mais recebem pacotes em determinado ponto da rede. As classes desenvolvidas foram:

- RMONMatrixData: estende as funções do RMONData para melhor representar as informações dos pares de transmissores e receptores.
- RMONMatrixViewBean: estende as funções do RMONViewBean para conseguir extrair as informações do grupo matrix. A principal função da classe é extrair os pares de transmissores e receptores através da formatação do índice presente no RMONData, pois o mesmo é formado pelos endereços MAC dos dois equipamentos.
- RMONMatrixGraphicPlotter: de posse dos objetos RMONMatrixData, processa quais os dez maiores transmissores e cria um gráfico interligando esses transmissores aos equipamentos que cada um mais enviou pacotes.

#### 4.3.6.3 rmon.graphs.statistics

Fornecer gráficos referentes ao grupo statistics. As classes desenvolvidas foram :

- RMONStatsErrosViewBean: executa a leitura das variáveis de quantidade de pacotes, broadcast, colisões e erros.
- RMONStatsErrosGraphicPlotter: cria um gráfico em barras demonstrando qual o percentual de erros, broadcasts e colisões relativos à quantidade de pacotes transmitidos.
- RMONStatsErrosSecViewBean: executa a leitura das variáveis de broadcast e colisões.
- RMONStatsErrosSecGraphicPlotter: cria um gráfico apresentando a variação de quantidade de pacotes de broadcasts e colisões através do tempo.
- RMONStatsSizeViewBean: Lê as variáveis do grupo statistics referentes ao tamanho de pacotes.
- RMONStatsSizeGraphicPlotter: Cria um gráfico em forma de pizza apresentando o percentual dos pacotes separados por seus tamanhos : 65-127, 128-255, 256-511, 512-1023 e 1024-1518

- RMONStatsRadarViewBean: executa a leitura das variáveis de broadcast, colisões e erros para as interfaces envolvidas no radar.
- RMONStatsRadarGraphicPlotter: Cria um gráfico em forma de radar apresentando uma sobreposição de percentuais de erros, broadcasts e multicasts de duas interfaces.

## 4.4 Modificações no WebManager

Para incluir o suporte aos componentes RMON no WebManager, necessitamos executar alterações em alguns arquivos da aplicação. As alterações executadas foram nos seguintes arquivos:

- webmngnconfig.dtd: esse arquivo é responsável por definir as tags que são utilizadas para configurar o WebManager. Foi modificado para adicionar tags para os componentes RMON, incluindo opções de configuração para o RMONMonitor, RMONDevice, addressT e RMONLogger;
- JConfigSetup.xml: Quando o WebManger inicia, os arquivos de configuração são lidos e os Beans da aplicação são criados. Este arquivo é utilizado para criar uma relação entre as tags de configuração e as classes correspondentes dos beans. A modificação executada foi para incluir as classes dos componentes RMON;
- WebSystem.Java: Classe que representa o sistema. Foi modificado para incluir os métodos para adicionar o RMONLogger e addressT à estrutura da aplicação;
- WebManager.Java: Classe principal do sistema e responsável por iniciar os processos, como monitoração e logger, da aplicação. Foi modificado para iniciar a coleta de endereços (addressT);
- WebMngrConfig.Java: Classe de configuração do sistema. Foi modificada para incluir os métodos de criação do RMONDeviceTemplate;
- NetWork.Java: Classe que representa uma rede de elementos gerenciáveis. Foi modificada para adicionar um método para incluir um RMONDevice.

# Capítulo 5

## **Estudo de caso no uso dos Componentes**

Nesse capítulo demonstramos um exemplo prático de gerenciamento de redes utilizando os componentes desenvolvidos em conjunto com a aplicação WebManager. Como laboratório de testes utilizamos a rede do Campus II da Universidade Tiradentes, devido à facilidade de acesso e por trabalhar envolvido com o grupo de pessoas responsáveis pela administração dessa rede. Na seção 5.1 apresentamos uma descrição dessa rede e de seus equipamentos, na seção 5.2 explicamos a interface da aplicação para a interação com as informações RMON e por fim na seção 5.3 apresentamos exemplos possíveis de utilização da aplicação.

### **5.1 A rede do Campus II UNIT**

O Campus II fica localizado na cidade de Aracaju e concentra quase toda a estrutura administrativa da Universidade Tiradentes. Pode ser considerado o maior dos quatro Campi da instituição e nele estão localizados os prédios da Reitoria, Biblioteca, Laboratórios e blocos de aulas.

Gerenciada pelo Departamento de Tecnologia e Informática, a infraestrutura de rede do Campus II disponibiliza serviços cruciais para o funcionamento da Universidade. Serviços que envolvem:

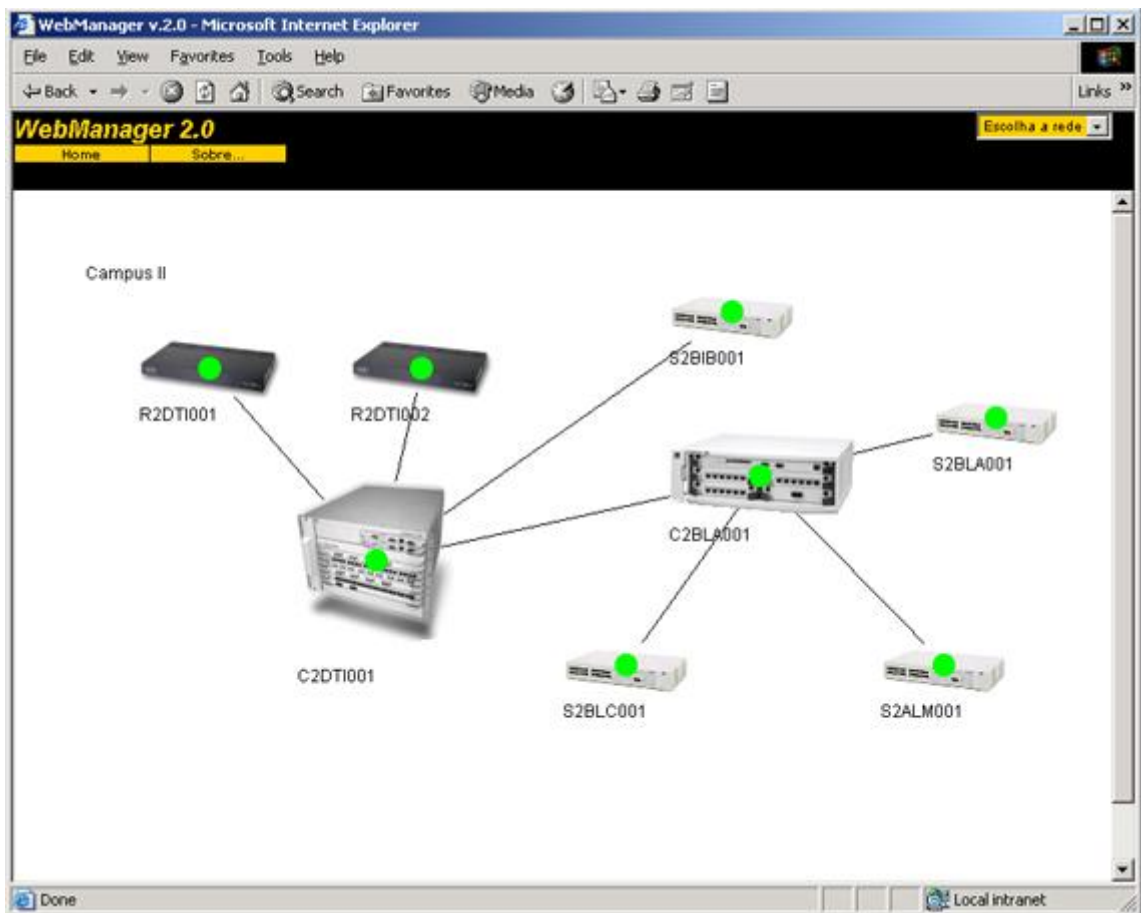
- Correio eletrônico corporativo;
- Acesso a Internet via link dedicado;
- Servidores de arquivos;
- Infraestrutura de comunicação para aplicações Cliente Servidor;
- Serviços disponibilizados na Web, como aplicação para Matrícula On-Line e para acesso ao Sistema de Biblioteca;

Devido a essa gama de serviços, o bom funcionamento da rede é imprescindível para a instituição, o que cria a necessidade de utilização de equipamentos modernos e gerenciamento proativo. Logo abaixo, na Tabela 3, podem ser vistos os principais equipamentos da rede e algumas de suas características.

Nome	Modelo	Local	Interligação	Descrição
R2DTI001	Cisco 2500	Reitoria	C2DTI001	Wan com o Campus I
R2DTI002	Cisco 2600	Reitoria	C2DTI001	Wan com os Campi III e IV
C2DTI001	Corebuilder 9000 3Com	Reitoria	R2DTI001, R2DTI002, C2BLA001 e S2BIB001	Switch Layer 3 e que compota os Servidores da Universidade, os Links para os roteadores da rede e os equipamentos da Reitoria.
C2BLA001	Corebuilder 3500 3Com	Bloco A	C2DTI001	Switch Layer 3 que gerencia os Links para os outros blocos e para Reitoria.
S2BLA001	Switch 1100 3Com	Bloco A	C2BLA001	Comutador que possibilita o acesso à rede dos equipamentos dos Blocos A e B
S2BLC001	Switch 1100 3Com	Bloco C	C2BLA001	Comutador que possibilita o acesso à rede dos equipamentos do Bloco C
S2BIB001	Switch 3300 3Com	Biblioteca	C2DTI001	Comutador que possibilita o acesso à rede dos equipamentos da biblioteca.
S2ALM001	Switch 1100 3Com	Almox	C2BLA001	Comutador que possibilita o acesso à rede dos equipamentos da Gráfica e Laboratórios de Design

**Tabela 3 – Elementos Ativos do Campus II**

O WebManager utiliza arquivos escritos em xml para configurar as redes e os equipamentos gerenciados. Seguindo a estrutura definida para esses xmls de configuração, foi modificado o arquivo webmngnrcfg.xml para descrever a rede e a organização dos equipamentos da Tabela 2 . O resultado dessa configuração pode ser visto na Figura 12.

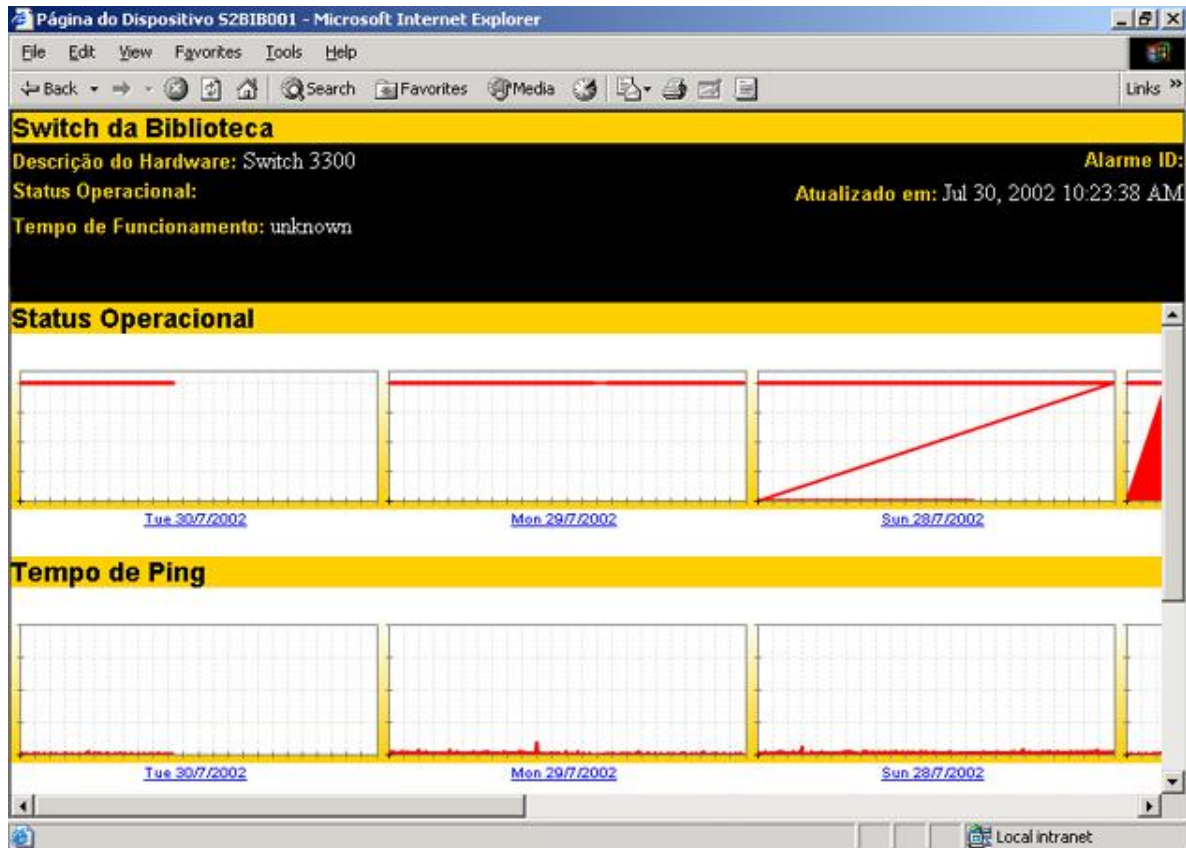


**Figura 12 – Mapa da Rede do Campus II da UNIT**

## 5.2 Interface da Gerência RMON

A interface Web do WebManager funciona com uma série de Mapas representando a rede e o status dos elementos gerenciados. Ao clicarmos em um elemento do mapa, temos acesso a uma página com as configurações desse elemento e com os gráficos disponíveis para ele. Na Figura 13 pode ser visualizado um exemplo de uma dessas páginas, no caso em

questão o elemento ativo é o S2BIB001. Como podemos notar não são apresentadas nenhuma informações RMON.



**Figura 13 – Exemplo de Página de Gerenciamento do Equipamento S2BIB001**

Para disponibilizarmos o acesso às informações RMON, foram criadas e incorporadas ao WebManager novas páginas JSP. Nessas novas páginas, anexamos menus que definem parâmetros de chamada dos gráficos, executando os componentes de visualização RMON. Na Figura 14 tem-se a interface de gerenciamento do equipamento S2BIB001 com as alterações.



**Figura 14 - Página de Gerenciamento Modificada do S2BIB001**

Ao todo adicionamos cinco opções de configuração dos gráficos:

- Data;
- Gráfico;
- Interface;
- Horário Inicial;
- Horário Final.

Quando um equipamento de rede é selecionado o WebManager agora faz um teste a mais para saber se ele tem informações RMON sendo gerenciadas, determinando que página fornecer ao usuário. Os menus de configuração apresentados acima só são disponibilizados se realmente existirem gráficos RMON para o equipamento em questão.

Depois de verificado que os menus devem ser apresentados, é executada uma pesquisa dos dados RMON disponíveis e são geradas as opções de Data, Gráfico e Interface através de um cruzamento desses dados. Por exemplo, se existem dados para o equipamento S2BIB001 do grupo Matrix no dia 10 de Julho, esse dia é disponibilizado no primeiro campo (Data) e ao ser escolhido, os gráficos de Matrix aparecem como opção no segundo campo (Gráficos). Desta maneira fica fácil identificar as informações disponíveis para a consulta.

Na Figura 15 pode ser visualizado um exemplo contendo a lista de datas com gráficos disponíveis para o elemento S2BIB001.

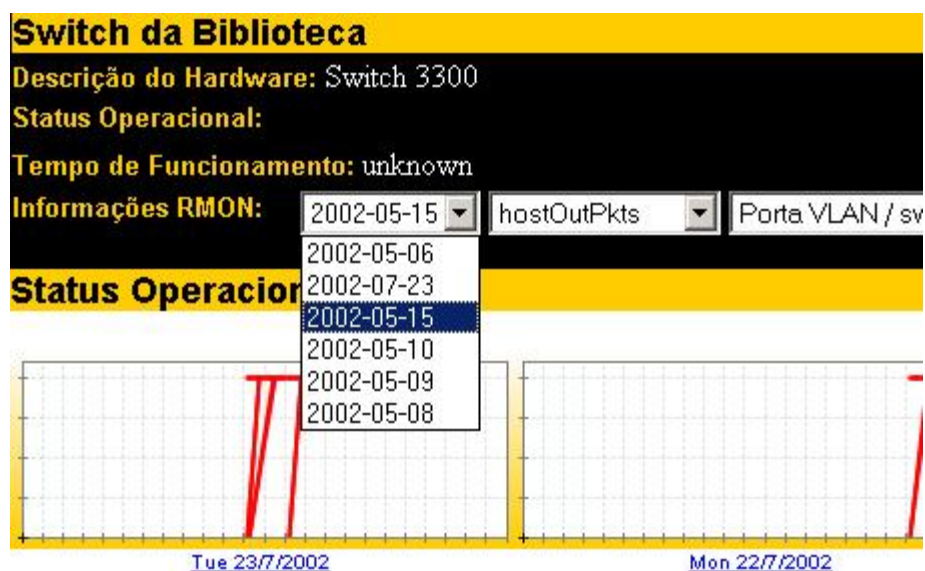


Figura 15 – Lista de Datas com Gráficos

Na Figura 16 pode ser visualizado um exemplo contendo a lista de gráficos disponíveis para o S2BIB001 no dia 10 de maio de 2002 (escolhida no 1ª campo), no caso estão disponíveis 3 gráficos diferentes um do grupo Hosts e outros dois do grupo Matrix.

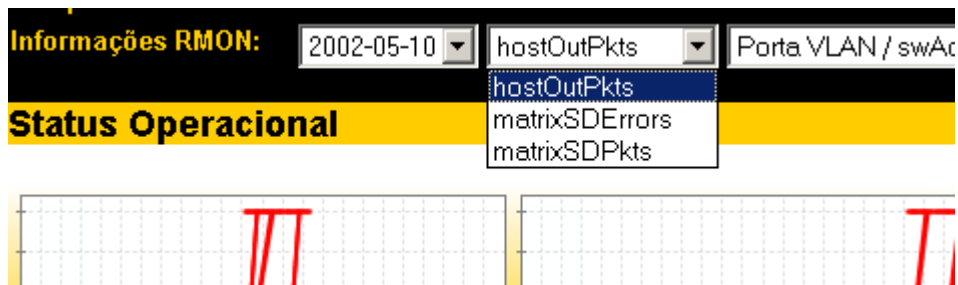


Figura 16 – Listas de Gráficos para o S2BIB001

O terceiro campo determina qual a interface que deve ser utilizada para gerar os dados do gráfico. Os valores disponibilizados para a escolha nesta opção são obtidos através do cadastro do equipamento no WebManager. Para o elemento S2BIB001 existe apenas uma interface, no caso uma porta representando uma VLAN, como pode ser visto na Figura 17.

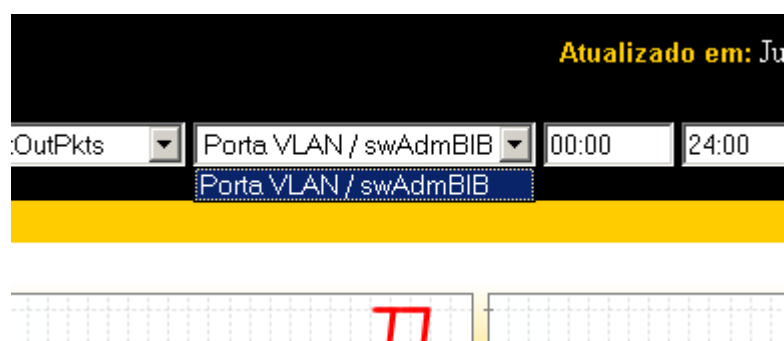


Figura 17 – Lista de Interfaces do equipamento S2BIB001

Por fim existem os dois últimos campos que representam o horário inicial e o horário final para as informações apresentadas no gráfico. Na Figura 17 pode-se notar que esses valores estão com 00:00 e 24:00 respectivamente, desta forma o gráfico terá como faixa todas



as horas do dia. Esses dois campos são bastante úteis para isolar a situação da rede em intervalos específicos.

Depois de definidos os cinco campos de configuração, basta o usuário clicar no botão gráfico que um novo é gerado de acordo com os parâmetros definidos. Na Figura 18, por exemplo, pode-se visualizar a demonstração de um gráfico de Top Hosts gerado para o S2BIB001 com informações do dia seis de maio de 2002 para a interface de rede da VLAN.

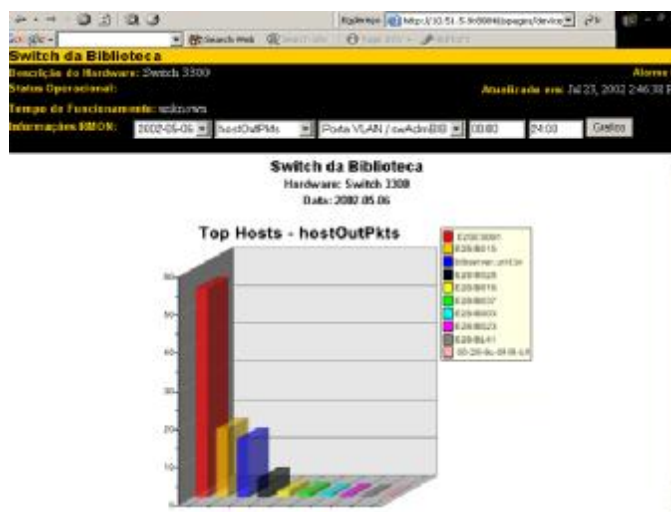


Figura 18 – Exemplo de um gráfico para o elemento S2BIB001

### 5.3 Gerência RMON na rede do Campus II - UNIT

Fazendo uso das informações RMON acessíveis através dos componentes criados e incorporados com a aplicação do WebManager, algumas informações relevantes para o funcionamento da rede podem ser mais facilmente identificadas. Nesta seção descrevemos 4 casos comuns de consultas para a rede do Campus II da UNIT.

### 5.3.1 Verificando estatísticas de erros entre os equipamentos do Bloco A

Selecionando o principal equipamento da rede do Bloco A, o Corebuilder 3500 – C2BLA001 pode-se escolher o gráfico de estatísticas de erros e visualizar o que está ocorrendo em um determinado período de tempo. Como pode ser visto na Figura 19, os valores margeiam entre 0.2 e 0.5 pontos percentuais o que pode ser considerado perfeitamente normal. Desta forma entre 16:20 e 17:30 pode-se garantir que o funcionamento da rede do bloco A apresentava níveis aceitáveis de broadcast.

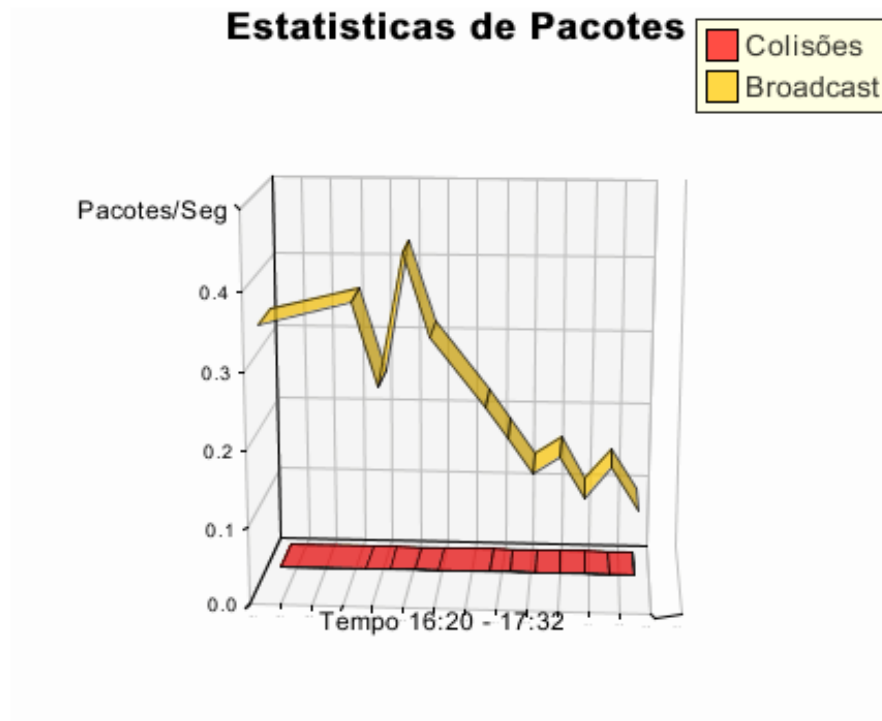
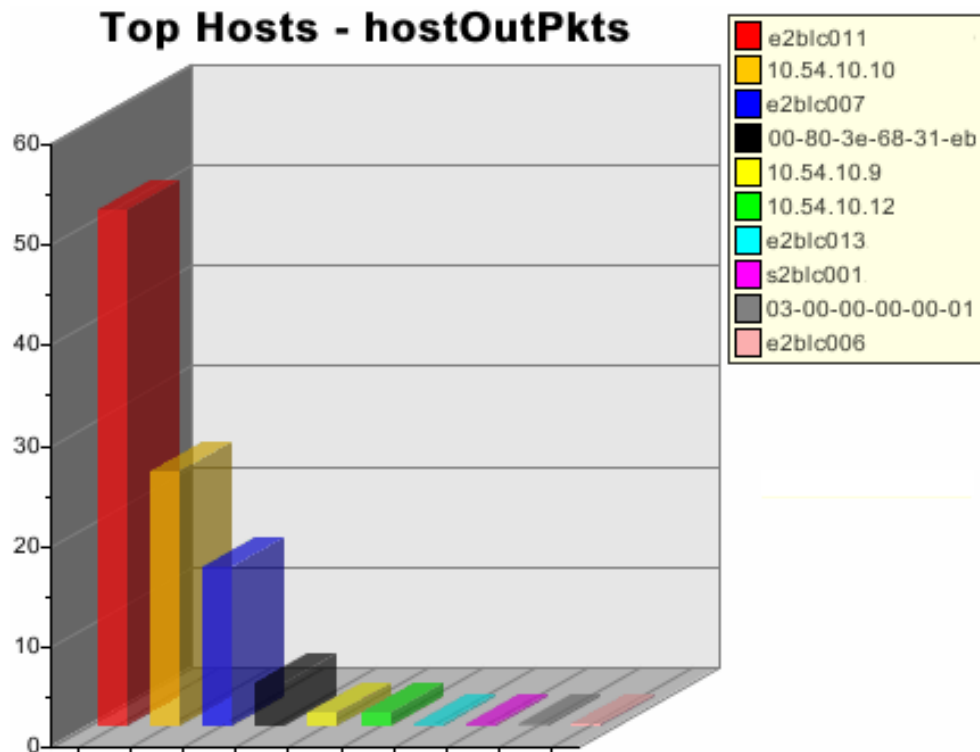


Figura 19 – Estatísticas de pacotes de broadcast

### 5.3.2 Descobrimo o Equipamento que mais transmite no Bloco C

A verificação do maior transmissor de pacotes no Bloco C fica bastante fácil utilizando a ferramenta. Selecionando o Switch 1100 S2BLC001, uma data determinada e o gráfico de Top Hosts por Pacotes, tem-se como resultado a Figura 20. Analisando a figura

pode-se notar que o maior transmissor é disparado o e2blb011, esse equipamento é uma das estações de trabalho dos Coordenadores de Curso.



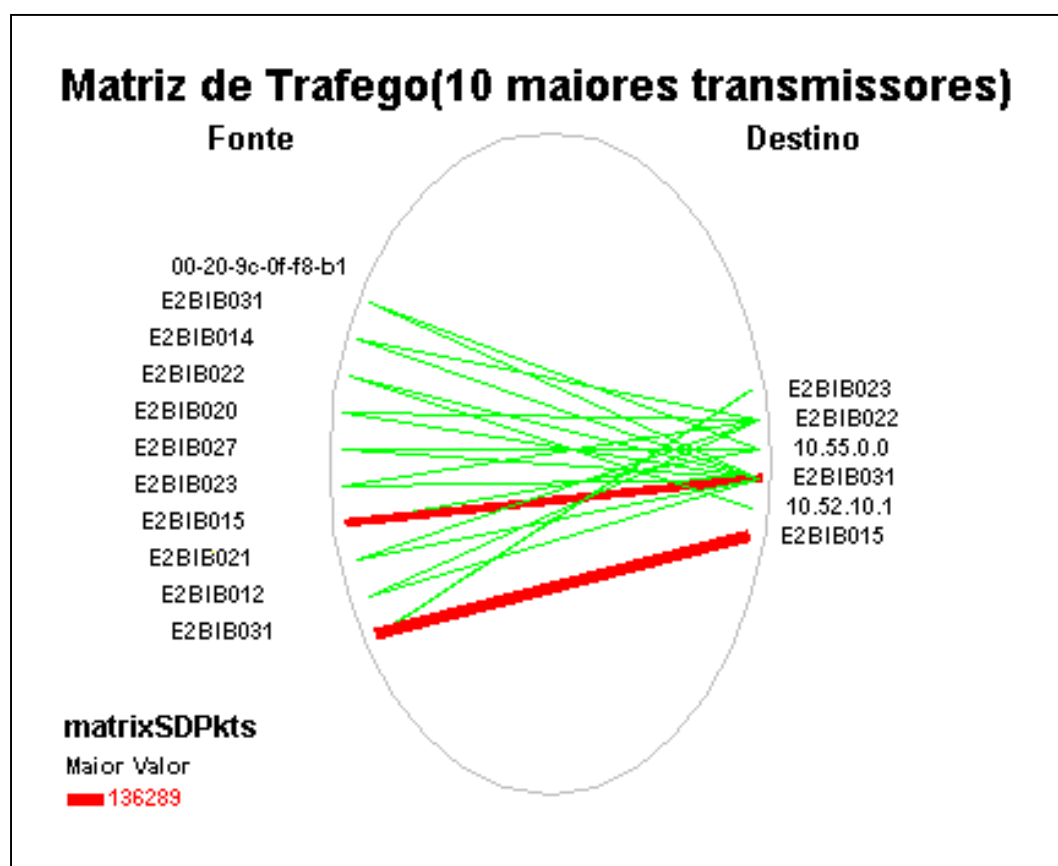
**Figura 20 – Maiores Transmissores do Bloco C**

### **5.3.3 Descobrimo os maiores transmissores e seus destinos na rede da Biblioteca**

Analisando o gráfico de Matrix do switch 3300 – S2BIB001, podemos verificar quais máquinas estão mais transmitindo informações entre si. Como pode ser visto na Figura 21, o equipamento que mais recebeu pacotes foi o E2BIB031 pois várias máquinas enviaram para

essa estação de trabalho. Inclusive dá para notar que houve uma grande troca de dados entre o E2BIB031 e o E2BIB015.

Investigando o porquê do ocorrido, notou-se que a máquina E2BIB031 é a única com CD-ROM e que o compartilha com as outras. De posse dessa informação pode-se melhorar o tráfego instalando mais CD-ROMs compartilhados em outras máquinas da Biblioteca.

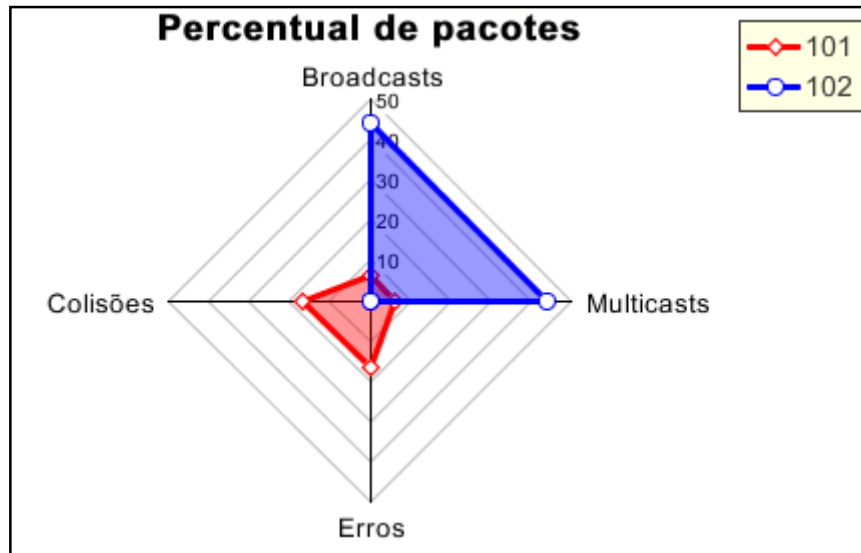


**Figura 21 – Tráfego entre equipamentos da Rede da Biblioteca**

### 5.3.4 Comparando o funcionamento de duas interfaces do switch S2BIB001

Analisando o gráfico de Radar do switch 3300 – S2BIB001, podemos verificar como está o tráfego de informações entre duas interfaces a 101 e a 102. Como pode ser visto na

Figura 22, a interface 102 apresenta altos índices de Broadcasts e Multicasts enquanto que a interface 101 apresenta mais erros e colisões. Com esse tipo de gráfico, podemos analisar portas, como duas interfaces Wan, e verificar o nível de erro que elas estão apresentando.



**Figura 22 – Comparação entre interfaces do Switch S2BIB001**

# Capítulo 6

## Conclusões

Neste trabalho, descrevemos e desenvolvemos componentes de software para facilitar a gerência de redes utilizando a tecnologia RMON no WebManager. Os componentes apresentam funções de coleta, armazenamento e processamento dos dados RMON e têm como objetivo possibilitar a construção de ferramentas de gerência com recursos RMON através da sua reutilização. Neste capítulo apresentamos os resultados obtidos e quais os requisitos levantados foram alcançados, também demonstramos as dificuldades encontradas e apresentamos os possíveis trabalhos futuros.

### 6.1 Avaliação dos Componentes

Verificamos que a tecnologia RMON apresenta uma grande quantidade de dados úteis para a gerência de redes. No entanto, esses dados são de muito baixo nível, o que torna sua análise bastante difícil e ineficiente sem a utilização de ferramentas capazes de tratá-los. Após levantarmos os requisitos necessários a uma aplicação desse tipo, especificamos um conjunto de componentes de software para tratamento dos dados RMON. Esses componentes ainda foram incorporados à ferramenta de gerência de rede WebManager como forma de avaliar sua funcionalidade.

Depois de terminada a fase de implementação e avaliação dos componentes, apresentamos uma análise de como os requisitos propostos foram alcançados. Na Tabela 4, temos uma lista dos requisitos e das soluções apresentadas para solucioná-los.

Requisito	Solução
R1 – Funcionamento em Arquiteturas Abertas	Utilização da API para acesso SNMP fornecida pela AdventNet [AdventNet, 2000].
R2 – Coleta de Dados	Pacote de Monitoramento, rmon.monitor
R3 – Histórico de Dados	Pacote de Armazenamento, rmon.logger
R4 – Acesso e manutenção das tabelas de controle dos grupos	Pacotes rmon.config e rmon.monitor
R5 – Armazenamento dos Dados	Pacote rmon.logger
R6 – Acesso aos dados coletados	Pacote rmon.loader
R7 – Geração de Gráficos	Pacote rmon.graph e rmon.addressT

**Tabela 4 – Requisitos alcançados pelos componentes**

Como pode ser visto na tabela 4, todos requisitos foram alcançados pelos componentes e sua utilização em conjunto com WebManager provou ser uma ferramenta útil para tratamento dos dados RMON.

Podemos destacar ainda as seguintes características:

#### **Solução desenvolvida seguindo as características da Arquitetura do WebManager**

Executamos o desenvolvimento dos componentes seguindo as características apresentadas pelo WebManager para facilitar a integração dos mesmos à aplicação. Devido a isso, várias características louváveis foram herdadas, entre elas podemos citar os seguintes pontos:

- Desenvolvimento utilizando a linguagem JAVA: característica que torna a aplicação independente de plataforma, possibilitando a portabilidade para vários sistemas operacionais, e ainda fornece mais facilidade para utilização dos recursos da rede e acesso via Web;
- Utilização de Threads: forma de se implementar processos em paralelos, possibilitando um melhor desempenho para execução das tarefas;
- Configuração dos componentes através de arquivos XML: possibilita a utilização de ferramentas para a edição dos arquivos de configuração, facilitando a visualização dos componentes e a definição de seus parâmetros de funcionamento.

### **Apresentação de um modulo de Mapeamento de endereços**

O modulo de mapeamento de endereços é uma ferramenta muito útil, pois possibilita a tradução e coleta dos endereços MACs encontrados na rede, e pode ser utilizado para aumentar a funcionalidade de aplicações de gerência mesmo que elas não incorporem a tecnologia RMON.

### **Facilidade para incorporar novos gráficos**

Pela maneira que foram desenvolvidos os componentes, em módulos separados de coleta, armazenamento e processamento, fica fácil acrescentar novos recursos de processamento, isso significa que novos gráficos podem ser criados e incorporados à arquitetura no modulo de processamento de informações.

## **6.2 Dificuldades encontradas**

Durante o desenvolvimento dos componentes algumas dificuldades e entre todas podemos destacar algumas que mais influenciaram no andamento do trabalho:

### **Documentação do RMON**

Foi muito difícil encontrar documentação sobre a Tecnologia RMON e de como esta pode ser utilizada. O que existe em bastante quantidade na Internet e em livros são descrições do RFC do RMON, mas não conseguimos encontrar documentação explicando como utilizar os dados disponíveis na tecnologia para aproveitá-los na gerência da rede.

### **Entender o funcionamento da Arquitetura WebManager**



Tivemos que entender o funcionamento de toda a arquitetura do WebManager para desenvolvermos o trabalho. Analisando os arquivos de configuração e a forma que a aplicação trabalha.

### **6.3 Trabalhos Futuros**

Apesar dos componentes atualmente possibilitarem a utilização da tecnologia RMON como fonte de informação e ferramenta de gerência, podemos destacar melhorias para serem desenvolvidas com intuito de anexar novas funcionalidades e características. Os seguintes pontos podem ser destacados como requisitos para uma nova versão:

- N1. Módulo de configuração dos probes: não existe uma forma de configurar o funcionamento do probe através da ferramenta, e essa funcionalidade poderá ajudar a facilitar a manutenção, que hoje tem que ser executada de forma manual, dos mesmos. Com isso poderíamos incorporar métodos para tratamento de alarmes, eventos e filtros na aplicação.
- N2. Suporte a versão 2 do RMON: modificar os componentes para acessar e tirar proveito da versão mais nova do RMON, que incorpora dados coletados da camada de rede.
- N3. Correlacionar dados: possibilitar criar grupos de equipamentos, todos de um campus de uma universidade, por exemplo, e depois gerar gráficos correlacionando os dados do grupo.

## Referências Bibliográficas

[3Com, 1999] 3Com, *On-line*: <http://www.3com.com/>

[AdventNet, 1999] AdventNet Inc., *On-line*: <http://www.adventnet.com/>

[Deitel, 1999] Deitel, H M. Java : How To Program. Prentice Hall. Nova Jersey, 1999

[Harnedy, 1997]Harnedy, Sean J. Total Snmp : Exploring the Simple Network Management Protocol. Segunda Edição. Prentice Hall. Nova Jersey, 1997.

[Miller, 1993]Miller, Mark A. Managing Internetworks with SNMP. Primeira edição. M&T Books. Nova York, 1993.

[OpenView, 2002] OpenView, *On-line*:  
<http://www.openview.hp.com/products/nnm/index.asp>

[Perkins, 1999]Perkins, David T.Remote Monitoring of SNMP-Managed Lans. Segunda Edição. Prentice Hall. Nova Jersey, 1999.

[Rose, 2000] Rose, Marshall T. How To Manage Your Network Using Snmp : The Networking Management Practicum. Nova Jersey : Prentice Hall, 2000

[Sauvé, 1999] Sauvé, J. P., *WebManager: A Web-Based Network Management Application*. LANOMS – Latin American Network Operations and Management Symposium, Rio de Janeiro, Brasil, Dezembro 1999.

[Szyperski, 1999] Szyperski, Clemens. Component Software - Beyond Object-Oriented Programming. Great Britain: Addison-Wesley, 1999.

[Singleton, 2000] Singleton Labs, *On-line*: <http://www.singleton-labs.com/>

[Tanenbaum, 1996] Tanenbaum, A. S. Computer Networks. Terceira edição. Prentice Hall PTR. Nova Jersey, 1996.

[Transced, 2002] Transced, *On-line*:  
[http://www.3com.com/products/en\\_US/detail.jsp?tab=support&pathtype=support&sku=3C81400B](http://www.3com.com/products/en_US/detail.jsp?tab=support&pathtype=support&sku=3C81400B)

# Anexo A

## Desenvolvimento de Software com Componentes

Cada vez mais, recursos e serviços são criados e oferecidos pelas empresas em busca de novos clientes e de conseguir manter a satisfação dos que já possuem. Com o advento da Internet, a competição e a velocidade de surgimento de novidades ficaram muito grandes, exigindo um alto grau de dinamismo para não se tornar obsoleto. Esses fatos repercutem na utilização da informática pela empresa e principalmente no desenvolvimento de seus softwares, os sistemas utilizados tem que ser capazes de possibilitar acesso via Internet, mantendo um alto grau de desempenho e ainda facilidade de extensão e manutenção. Fazem-se necessários recursos para criar mais programas e em menos tempo.

Além das necessidades de velocidade no desenvolvimento, os novos softwares devem apresentar recursos como:

Funcionamento em arquitetura distribuída: conseguindo acoplar escalabilidade, acesso via Internet, trabalhar com plataformas diferentes e comunicação com sistemas legados;

Capacidade de extensão: oferecendo a possibilidade de aumentar os recursos de um sistema apenas acoplando novas funcionalidades

Facilidade de reaproveitamento: disponibilizando alta reutilização de software

Para tentar suprir essas necessidades, novos paradigmas de desenvolvimento estão sendo utilizados e um desses paradigmas é o desenvolvimento de software baseado em componentes.

## A.1 O que são Componentes?

O desenvolvimento de software baseado em componentes funciona utilizando a idéia de dividir os problemas e resolvê-los com componentes específicos, onde o importante é não iniciar do zero sempre que um novo sistema será desenvolvido. As empresas podem comprar e desenvolver componentes e depois disso montar sistemas juntando esses componentes.

Segundo D'Souza, um componente de implementação é: “Um pacote coerente de implementação de software que (a) pode ser desenvolvido independentemente e entregue como unidade; (b) tem interfaces explícitas e bem definidas para os serviços que oferece; (c) tem interfaces explícitas e bem definidas para os serviços que requer; e (d) pode ser composto com outros componentes, talvez após a customização de algumas propriedades mas sem modificar os componentes em si.”

A análise desta e de outras definições, pode destacar três características básicas presentes nos componentes [Sauvé, 1999]:

### 1. Construção de aplicações por montagem

Deve possibilitar o desenvolvimento de aplicações através de montagem e junção de componentes. A programação desenvolve-se através da modificação de atributos, geralmente utilizando uma ferramenta gráfica, dos componentes.

O componente deve funcionar e ser acoplado apenas com essas alterações de atributos, sem a necessidade de alteração no código dos mesmos.

### 2. Um componente explicita suas interfaces

Para ser adaptável e funcionar em diversos contextos, os componentes necessitam utilizar interfaces, que poderiam ser considerados como um tipo de contrato, definindo os serviços que ira oferecer e dos serviços que necessita para o seu funcionamento adequado.

### 3. Um componente é uma unidade de empacotamento(packaging), entrega(delivery), implatação(deployment) e carga (loading)

Unidade de empacotamento: inclui as suas especificações e outros recursos necessários para seu funcionamento, como figuras ou sons;

Unidade de entrega: quando um componente é desenvolvido e entregue, todo o seu software é entregue completo, independente do ambiente que será acoplado;

Unidade de implantação: um componente pode ser configurado durante sua implantação, mas não pode ser implantado parcialmente;

Unidade de carga: quando utilizado em um sistema, o componente é executado pode inteiro.

## **A.2 Desenvolvimento de Componentes.**

Com a utilização de componentes, o desenvolvimento de software agora visa a criação de elementos altamente reutilizáveis e configuráveis e de sistemas construídos baseados nesses elementos. Com isso, novas etapas surgem para o desenvolvimento de aplicações:

Definição dos componentes que serão utilizados pelas aplicações: devemos identificar que componentes serão necessários e se eles serão comprados a terceiros ou desenvolvidos pela empresa;

Montagem da aplicação (“*Assembly*”): etapa de junção e configuração dos componentes para a criação da aplicação;

Implantação dos componentes: etapa de ajustes da aplicação, montada com componentes, para seu funcionamento no ambiente de produção;

## **A.3 JavaBeans**

Utilizando a linguagem de programação java, os componentes são considerados JavaBeans e foram criados para permitir a sua reutilização utilizando ferramentas gráficas para o desenvolvimento. Eles foram desenvolvidos com a intenção de apresentar a possibilidade de um duplo tipo de funcionamento, onde primeiramente são utilizados em ferramentas para a montagem de aplicações (*design time*) e após essa fase são configurados para funcionar em tempo de execução(*runtime*), uma das idéias era de tirar a parte de “desenvolvimento” do componente quando a aplicação fosse vendida [Szyperki,1999].

Entre as características dos componentes podemos destacar:

Utilização de eventos: são capazes de anunciar geradores ou consumidores de eventos específicos e utilizam esses eventos como forma de comunicação. Quando um JavaBean tem eventos a enviar, busca em uma lista cadastrada previamente os componentes interessados e executa o envio;

Propriedades: os JavaBeans apresenta várias propriedades, que através de métodos de busca(*getter*) e alteração(*setter*) possibilitam sua customização;

Persistência: um JavaBean é capaz de salvar seu estado de execução e possibilitar a recuperação deste estado posteriormente

Introspecção: um JavaBean pode ser analisado por uma ferramenta de desenvolvimento em busca de métodos, eventos e propriedades disponíveis para utilização na montagem de aplicações.