

UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CLAUDIO BARBOSA SCHMICHTEMBERG

ESPECIFICAÇÃO DE UMA PLATAFORMA DE GERENCIAMENTO
DE REDES BASEADA NO PROTOCOLO SNMP

CRICIÚMA, JUNHO DE 2009.

CLAUDIO BARBOSA SCHMICHTEMBERG

**ESPECIFICAÇÃO DE UMA PLATAFORMA DE GERENCIAMENTO
DE REDES BASEADA NO PROTOCOLO SNMP**

Trabalho de Conclusão de Curso apresentado
para obtenção do Grau de Bacharel em Ciência
da Computação pela Universidade do Extremo
Sul Catarinense.

Orientador: Prof. MSc. Paulo João Martins

CRICIÚMA, JUNHO DE 2009.

CLAUDIO BARBOSA SCHMICHTEMBERG

**Especificação de uma Plataforma de Gerenciamento de Redes Baseada no
Protocolo SNMP**

Submetido ao corpo docente do Curso de Ciência da Computação da
Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau
de Bacharel em Ciência da Computação.

Prof. MSc. Rogério Antônio Casagrande
Coordenador Adjunto do Curso de Ciência da Computação

Banca Examinadora:



Prof. MSc. Paulo João Martins
Orientador



Prof. MSc. Rogério Antônio Casagrande



MSc. Ricardo Portes

*Dedicado à minha mãe Eliane, minha irmã
Thaís e minha avó Salete.*

AGRADECIMENTO

Agradeço a todos que estiveram presente e que contribuíram para esta conquista.

Agradeço a todos os familiares que sempre estiveram juntos me dando apoio, em especial a minha mãe que com orgulho me apóia a cada dia. A minha prima Bruna onde passei muitas noites acordado no desenvolver do trabalho.

Agradeço a todos os amigos e colegas do curso, em especial a Marcelo Santos e Mário Luís pela amizade.

Agradeço também a todos os educadores e envolvidos no curso que colaboraram para aquisição de conhecimento no passar destes anos.

*“A inteligência é o único meio que possuímos
para dominar os nossos instintos”*

Sigmund Freud

RESUMO

O crescimento acelerado da tecnologia de informação, juntamente com a concorrência existente nos mais diversos setores em que as organizações atuam, faz com que as mesmas busquem continuamente soluções que as permitam estarem atualizadas, para que possam acompanhar o crescimento do setor. O presente trabalho apresenta a especificação de uma plataforma para auxiliar no gerenciamento de redes, que traz dentre seus requisitos desejáveis o fato de criar um sistema dinâmico, onde possibilite que novas técnicas sejam adicionadas, tornando o sistema renovável e atualizado. As plataformas de gerenciamento são softwares que manipulam os dados recuperados na rede por meio dos protocolos de gerência e os disponibilizam a um conjunto de ferramentas que acopladas a plataforma criam uma solução de gerenciamento ampla, que atenda a várias das áreas funcionais do gerenciamento de redes. A especificação presente no trabalho descreve características de algumas plataformas conhecidas, destaca as desejáveis para uma implementação futura e define as diretrizes a serem seguidas no processo de desenvolvimento. O processo de especificação utiliza recursos da linguagem UML para descrever de maneira abrangente a plataforma. Por fim um protótipo desenvolvido em linguagem Java apresenta alguns dos recursos do protocolo SNMP.

Palavras Chave: Gerência de Redes; Plataformas de Gerenciamento; Protocolo SNMP; Modelagem de Software.

ABSTRACT

The rapid growth of information technology, together with competition in several sectors in which the organizations operate, makes them to continually seek solutions that are updated to, so they can monitor the growth of the sector. This paper presents the specification of a platform to assist in the management of networks, which brings out the fact its requirements desirable to create a dynamic system, where possible that new techniques are added, making the system renewed and updated. The platform management software that is handling the data retrieved by the network management protocols and offer a set of tools that create a platform coupled to solution of comprehensive management, which responds to several of the functional areas of management of networks. The specification in this paper describes some characteristics of known platforms, highlights the desirable for future implementation and define policies to be followed in the development process. The process of using the specification language UML to describe the way a comprehensive platform. Finally a prototype developed in Java language presents some of the features of SNMP.

Keywords: Network Management, Platform Management, SNMP Protocol.

LISTA DE FIGURAS

Figura 1. Representação das camadas do modelo OSI	21
Figura 2. Arquitetura do modelo Internet (TCP/IP).....	25
Figura 3. Multiplexação e Demultiplexação de Portas	27
Figura 4. Componentes do modelo SNMP.....	33
Figura 5. Árvore MIB com o OID dos objetos.....	34
Figura 6. Convenção textual de um tipo de objeto	36
Figura 7. Formatos SNMP.....	37
Figura 8. Seqüência das PDUs SNMP.....	38
Figura 9. Estrutura da Arquitetura Centralizada.....	40
Figura 10. Estrutura da Arquitetura Hierárquica.....	41
Figura 11. Estrutura da Arquitetura Distribuída.....	42
Figura 12. Arquitetura geral de um sistema de gerência.....	45
Figura 13. Blocos de construção UML.....	49
Figura 14. Diagramas UML.....	49
Figura 15. Interface visual da plataforma OpenNms.....	53
Figura 16. Interface Gráfica do Nagios.....	55
Figura 17. Interface gráfica da plataforma HP OpenView.....	57
Figura 18. Região de monitoramento de distribuição.....	59
Figura 19. Arquitetura do Sistema ZenOSS.....	61
Figura 20. Diagrama de casos de uso.....	63
Figura 21. Arquitetura da Plataforma de Gerência	64
Figura 22. Recuperação de dados na plataforma.....	67
Figura 23. Hierarquia da MIB II.....	70

Figura 24. Árvore do Grupo Interfaces.....	71
Figura 25. Árvore do Grupo IP.....	73
Figura 26. Árvore do Grupo ICMP.....	74
Figura 27. Objetos do grupo SNMP.	78
Figura 28. Tela de manipulação do protocolo de gerência SNMP.....	82
Figura 29. Tela de configuração do protocolo SNMP.....	84
Figura 30. Objetos Utilizados pela API.	85
Figura 31. Métodos do SnmpTarget.	85
Figura 32. Métodos do SnmpTarget.	85
Figura 33. Carregar Arquivo MIB.....	86
Figura 34. Métodos Para Manipulação do SNMP.....	86

LISTA DE TABELAS

Tabela 1. Tipos de Dados Básicos Definidos na SMI.	35
Tabela 2. Plugins do Nagios.	55
Tabela 3. Tabela de Descrição dos Objetos do Grupo System.....	70
Tabela 4. Descrição dos Objetos do Grupo Interfaces.....	72
Tabela 5. Descrição dos Objetos do Grupo IP.....	74
Tabela 6. Descrição dos Objetos do Grupo TCP.....	75
Tabela 7. Descrição dos Objetos do Grupo UDP.	76
Tabela 8. Descrição dos Objetos do Grupo EGP.....	76
Tabela 9. Descrição dos Objetos do Grupo Transmission.	77

LISTA DE SIGLAS

API	<i>Application Programming Interfaces</i>
ASCII	<i>American Standard Code for Information Interchange</i>
ASN.1	<i>Abstract Syntax Notation One</i>
AT	<i>Address Translation</i>
ATM	<i>Asynchronous Transfer Mode</i>
CLI	<i>Command Line Interface</i>
CMIP	<i>Common Management Information Protocol</i>
DNS	<i>Domain Name System</i>
EGP	<i>External Gateway Protocol</i>
GUI	<i>Graphical User Interface</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
ICMP	<i>Internet Control Message Protocol</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
JDK	<i>Java Development Kit</i>
MAC	<i>Media Access Control</i>
MIB	<i>Manager Information Base</i>
OID	<i>Object Identifier</i>
OVW	<i>Open View Windows</i>
PDF	<i>Portable Document Format</i>
PDU	<i>Protocol Data Unit</i>

QoS	<i>Quality of Service</i>
ReMAV	Redes Metropolitanas de Alta Velocidade
RFC	<i>Request for Comments</i>
RM-OSI	<i>Open System Interconnection Reference Model</i>
SGRI	Sistema de Gerenciamento de Redes Integrados
SLA	<i>Service Level Agreement</i>
SLM	<i>Service Level Management</i>
SMI	<i>Structure of Management Information</i>
SNMP	<i>Simple Network Management Protocol</i>
TCP	<i>Transmission Control Protocol</i>
TMR	<i>Tivoli Management Region</i>
UDP	<i>User Datagram Protocol</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVO GERAL	15
1.2	OBJETIVOS ESPECÍFICOS	16
1.3	JUSTIFICATIVA	16
1.4	ESTRUTURA DO TRABALHO	17
2	GERENCIAMENTO DE REDES	19
2.1	MODELOS DE REFERÊNCIA	20
2.1.1	Modelo OSI.....	20
2.1.2	Modelo Internet.....	25
2.2	PROTOCOLOS DE GERENCIAMENTO DE REDE.....	28
2.2.1	Gerenciamento Segundo o Modelo OSI.....	29
2.2.2	Gerenciamento Segundo o Modelo Internet (TCP/IP).....	31
2.3	PARADIGMAS DE GERENCIAMENTO.....	39
2.3.1	Definição de Leinwand.....	39
2.3.2	Definição de Schönwälder	42
2.4	SOFTWARES DE GERENCIAMENTO	43
2.4.1	Plataformas de Gerenciamento.....	45
3	MODELAGEM DE SOFTWARE	47
3.1	UNIFIED MODELING LANGUAGE (UML).....	48
4	TRABALHOS CORRELATOS.....	50
4.1	OPENNMS.....	50
4.2	FREENMS	50

4.3	NAGIOS.....	51
4.4	ADMINISTRAÇÃO E GERÊNCIA DE REDES DE COMPUTADORES.....	51
5	TRABALHO DESENVOLVIDO	52
5.1	CARACTERÍSTICAS DAS PLATAFORMAS DE GERÊNCIA ANALISADAS	52
5.1.1	OpenNms	52
5.1.2	Nagios.....	54
5.1.3	HP OpenView.....	55
5.1.4	IBM Tivoli	57
5.1.5	ZenOSS.....	59
5.2	CARACTERÍSTICAS DA PLATAFORMA DE GERÊNCIA	61
5.3	MODELAGEM	62
5.3.1	Modelagem dos Dados.....	69
5.3.2	Análise de Funcionalidades que Podem Ser Adicionas ao Modelo.....	80
5.4	PROTÓTIPO DE MANIPULAÇÃO DO PROTOCOLO SNMP.....	81
5.5	RESULTADOS OBTIDOS.....	86

1 INTRODUÇÃO

A crescente troca de dados e informações entre as organizações, e a busca contínua por um melhor aproveitamento de recursos, torna necessária a constante disponibilidade e segurança dos dados. Nas organizações, os recursos oferecidos pelas redes de computadores são de tal importância que simplesmente não podem falhar (RNP, 1997). Pensando em obter um melhor aproveitamento destes recursos e em oferecer qualidade de serviço a seus usuários, foi criada a área de gerência.

Esta área conta com o auxílio de diversas ferramentas computacionais de gerenciamento, que monitoram o uso dos recursos, com o intuito de maximizar a produtividade e evitar ou minimizar falhas ocasionadas pelos mais diversos motivos, como por exemplo, problemas de segurança e equipamentos sobrecarregados.

É imprescindível o uso de instrumentação de apoio adequado para um gerenciamento eficaz. O gerenciamento de uma rede com ferramentas inadequadas, que não apresentem claramente suas características, torna-o improdutivo e deixa a rede vulnerável.

Estas ferramentas seguem um protocolo de gerenciamento, que define as diretrizes dos objetos gerenciados. Entre os protocolos existentes destaca-se o *Simple Network Management Protocol* (SNMP), que atua na camada de aplicação das redes com arquitetura *Transmission Control Protocol/Internet Protocol* (TCP/IP), definindo o formato e a ordem que deve ser seguida no intercâmbio das informações de gerenciamento (SOARES; LEMOS; COLCHER, 1995).

1.1 OBJETIVO GERAL

Desenvolver a especificação de um software de gerência de redes baseado no

protocolo SNMP, com o uso de licença livre.

1.2 OBJETIVOS ESPECÍFICOS

Esta pesquisa tem como objetivos específicos os itens relacionados abaixo:

- a) compreender as áreas funcionais de gerenciamento de redes;
- b) entender o funcionamento do protocolo de gerência SNMP;
- c) identificar as características básicas de algumas ferramentas de gerência de redes;
- d) analisar o funcionamento das plataformas de gerenciamento de redes;
- e) definir alguns dos requisitos desejáveis em uma plataforma de gerenciamento de redes;
- f) desenvolver um modelo para um desenvolvimento futuro de uma plataforma de gerência.

1.3 JUSTIFICATIVA

O crescimento acelerado da tecnologia de informação, juntamente com a concorrência existente nos mais diversos setores em que as organizações atuam, faz com que as mesmas busquem continuamente soluções que as permitam estarem atualizadas, para que possam acompanhar o crescimento do setor.

É imprescindível que elas tenham controle sobre as soluções utilizadas, mantendo a segurança das informações e a disponibilidade dos dados sempre que necessário.

As pesquisas na área de gerência de redes estão em constante desenvolvimento, justamente com o intuito de oferecer soluções que atendam as necessidades das organizações.

As alternativas de gerência de redes disponíveis apresentam estruturas definitivas, com foco de análise específico, tornando necessário o uso de várias ferramentas para um gerenciamento completo da rede, sem permitir que novas técnicas sejam adicionadas, além de representarem um alto custo para as organizações, quando da necessidade do uso de ferramentas comerciais (CARVALHO et al, 2003).

A especificação da plataforma para auxiliar no gerenciamento de redes proposta, traz dentre seus requisitos desejáveis o fato de criar um sistema dinâmico, onde possibilite que novas técnicas sejam adicionadas, tornando o sistema renovável e atualizado.

O uso de ferramentas com licença livre e código aberto, possibilita aprimorar um software por meio de comunidades de desenvolvimento, e aparece como alternativa viável à abordagem de uma solução que pretende estar em constante atualização. Além de utilizar o protocolo SNMP, que tem como principal característica sua popularidade, por ser simples e concentrar a maior parte do processamento na máquina do gerenciador (FREITAS; MONTEIRO, 2004).

A confecção da especificação proposta justifica-se ainda pela aquisição de conhecimento na área, ficando a disposição da universidade para posterior uso em disciplinas como redes de computadores e gerência de redes, possibilitando aos acadêmicos a aquisição de conhecimento por meio da análise das características da mesma. Além de permitir a produção de novos TCCs, pelo vasto conteúdo que abrangerá, e pela construção de diversos módulos de gerenciamento.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está dividido em seis capítulos relacionados as tecnologias envolvidas na elaboração da especificação proposta.

Inicialmente é apresentada a estrutura geral do trabalho, com os objetivos gerais e específicos, a justificativa da pesquisa e a organização do mesmo.

Algumas classificações relacionadas aos diferentes paradigmas de gerenciamento são descritas no Capítulo 2.

Os conceitos relacionados aos modelos de referência das arquiteturas de redes de computadores, tal como suas respectivas estruturas e protocolos envolvidos são abordados no Capítulo 3, que trata também dos protocolos específicos para o gerenciamento de redes e da base de informações gerenciais, além de apresentar as definições relacionadas aos *softwares* de gerenciamento, suas características e aplicabilidades.

Em seguida é apresentada uma breve introdução aos conceitos de engenharia de software e os padrões adotados para o desenvolvimento.

O Capítulo 5 descreve alguns trabalhos correlatos com o desenvolvido.

Finaliza-se com o trabalho desenvolvido que aborda a especificação da plataforma, descrevendo as características de alguns softwares existentes e o modelo a ser seguido em um desenvolvimento futuro.

2 GERENCIAMENTO DE REDES

O crescimento acelerado das redes de computadores, motivado pelas facilidades oferecidas pela comunicação dos sistemas, torna crucial um bom gerenciamento.

Gerenciar de forma correta além de poupar tempo, evita possíveis problemas que podem surgir sem o gerenciamento (ZACKER; DOYLE, 2000).

Ele trata do controle de atividades e do monitoramento de recursos, obtendo informações para identificar e diagnosticar possíveis falhas, auxiliando no planejamento dos ambientes computacionais.

Essa necessidade advém dos objetivos requeridos da rede em questão, muitas vezes redes pequenas precisam de uma grande quantidade de informações, como por exemplo, para análise das aplicações utilizadas ou para relatórios de auditoria.

Em ambientes que não dependam de informações sobre os dados que trafegam entre seus componentes ou que tenham poucos equipamentos, o gerenciamento não aparece com tanta expressividade, pela facilidade de acesso as máquinas e pela fácil visão geral da rede, o que simplifica a detecção de problemas e possibilita uma ação rápida dos administradores. Porém os sistemas de comunicação vem crescendo rapidamente, de forma que nós são adicionados constantemente, tornando as redes maiores e mais complexas, o que faz com que a identificação de um simples problema, em muitos casos, seja obtida somente após a análise de inúmeros equipamentos, muitas vezes geograficamente dispersos, fazendo com que um bom gerenciamento dependa de uma estrutura bem definida e baseada em protocolos.

2.1 MODELOS DE REFERÊNCIA

Vários projetos de uma estrutura para comunicação de dados foram elaborados, aperfeiçoando os anteriores e criando métodos que possibilitassem a comunicação da maneira mais eficiente possível. Dentre os projetos criados, destacou-se a estrutura de rede baseada em camadas hierárquicas, onde as superiores utilizam os serviços oferecidos pelas inferiores (SOARES; LEMOS; COLCHER, 1995).

Os dados são transferidos de uma camada a outra de mesmo nível de forma transparente, passando pelas camadas inferiores. A transferência dos dados entre as estações é feita somente pelo nível mais baixo, que acumula as informações passadas pelas camadas superiores.

A estação que recebe os dados transmitidos faz o caminho inverso, retirando as informações que dizem respeito à camada atual e passando o restante para superior até que somente os dados propriamente ditos possam ser passados para a aplicação.

Cada fabricante definia quantos níveis sua estrutura de rede possuiria, permitindo que seus equipamentos se comunicassem. Porém, com a necessidade da comunicação entre equipamentos de diferentes fabricantes ficou inviável utilizar arquiteturas de rede proprietárias. No início dos anos 70, a *International Organization for Standardization* (ISO) juntou esforços e criou um modelo de referência público, denominado *Open System Interconnection Reference Model* (RM-OSI), conhecido somente como OSI.

2.1.1 Modelo OSI

O modelo OSI define as funções necessárias para que haja a interconexão entre sistemas distribuídos (TEIXEIRA JÚNIOR et al, 1999).

O modelo OSI é definido por sete camadas, como mostra a Figura 1.

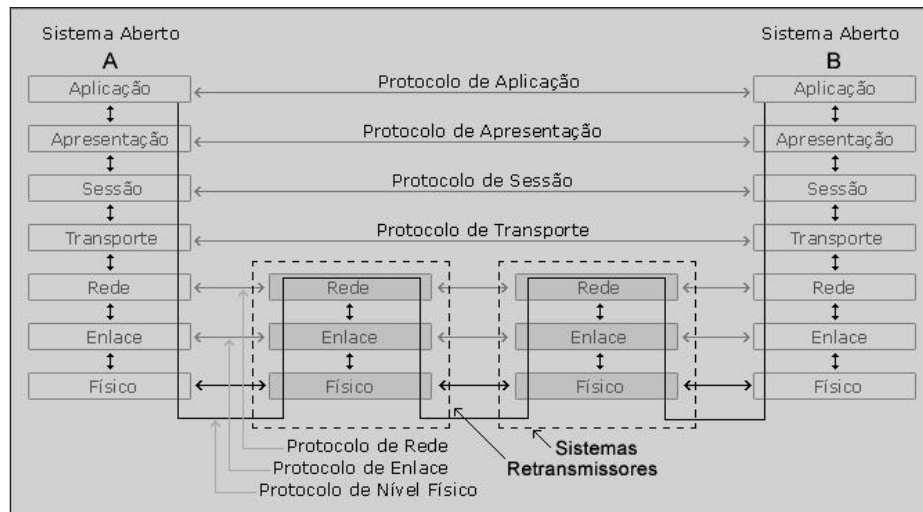


Figura 1. Representação das camadas do modelo OSI.

Fonte: SOARES, L. F. G; LEMOS, G; COLCHER, S. (1995).

2.1.1.1 Camada Física

É responsável pela transmissão dos *bits*, definindo as características elétricas, mecânicas e funcionais envolvidas nesta transmissão, além de tratar os procedimentos que definem o início e o fim da transmissão (SOARES; LEMOS; COLCHER, 1995).

Os protocolos que trabalham nesta camada preocupam-se em definir basicamente, o tempo de transferência entre os *bits*, se a comunicação acontece nos dois sentidos simultaneamente (*full-duplex*) ou em um sentido de cada vez (*half-duplex*) e ainda na função de cada pino utilizado pelos conectores de rede (TANENBAUM, 2003).

2.1.1.2 Camada de Enlace de Dados

Tem como principal objetivo oferecer a camada superior uma linha de transmissão que pareça livre de erros (TANENBAUM, 2003), verificando os *bits* que chegam

pelo meio físico.

As principais funções desta camada são: reconhecer e montar quadros (*frames*) de dados com os *bits* que chegam da camada física passando-os para camada de rede; tratar do controle de fluxo dos dados transmitidos, evitando que a estação emissora sobrecarregue a estação receptora; e detectar e se possível corrigir erros ocasionados por problemas na camada física ou na própria camada de enlace (EMBRATEL, 1997).

2.1.1.3 Camada de Rede

Tem como principal objetivo efetuar a transferência dos dados entre as estações de forma transparente em relação as sub-redes e aos meios de comunicação utilizados.

Destaca-se a função de roteamento, que permite definir as rotas para transferência dos pacotes, podendo cada pacote do conjunto de dados seguir uma rota distinta (roteamento dinâmico). Outra função importante determina o controle de congestionamento de pacotes, evitando gargalos na rede, por meio de parâmetros que determinam a taxa de erros, vazão (*throughput*), retardo na comunicação, entre outros.

2.1.1.4 Camada de Transporte

Tem como principais objetivos garantir uma comunicação entre as unidades sem preocupar-se com aspectos de roteamento e dependendo do tipo de serviço utilizado, garantir que os pacotes sejam entregues sem falhas, duplicidade ou fora da ordem que foram enviados (EMBRATEL, 1997).

A partir da camada de transporte, a comunicação entre a origem e o destino acontece definitivamente fim a fim, de modo que a comunicação do protocolo acontece

diretamente entre a origem e o destino, nos níveis inferiores a comunicação acontece de maneira encadeada entre cada máquina existente na rota utilizada pela conexão (Figura 1).

2.1.1.5 Camada de Sessão

Oferece serviços que permitem organizar e sincronizar a comunicação entre as estações. Os principais serviços disponibilizados pela camada de sessão são: o controle de *token*¹, o controle de diálogo e o gerenciamento de atividades (SOARES; LEMOS; COLCHER, 1995).

Mesmo que uma comunicação aconteça em um circuito *full-duplex*, pode haver a necessidade de se determinar que uma estação só possa transmitir depois que a outra tenha transmitido, nesse sentido a camada de sessão oferece o controle de *token*, conceito utilizado em comunicações *half-duplex*, para determinar quem irá transmitir em determinado momento.

O controle de diálogo define pontos de sincronização durante um diálogo, que são lançados pela estação emissora e confirmados pela estação receptora. Caso o diálogo seja interrompido, ele pode recomeçar do último ponto de sincronização confirmado (EMBRATEL, 1997).

Uma conexão de sessão permite que somente uma atividade seja executada por vez. O gerenciamento de atividades trata da possibilidade de uma tarefa ser interrompida e retomada em uma mesma conexão de sessão, permitindo que várias atividades possam ser executadas ao mesmo tempo, cada uma utilizando um espaço de tempo da conexão (SOARES; LEMOS; COLCHER, 1995).

¹ Ficha ou Sinal. Conceito utilizado em redes de computadores do tipo *token ring* para definir qual equipamento tem direito a enviar dados em determinado momento, evitando o congestionamento ocasionado por transmissões simultâneas (SANTOS JUNIOR, 2002).

2.1.1.6 Camada de Apresentação

Tem como objetivo prover mecanismos que tratem das diferentes formas possíveis de representar as informações geradas pelos diferentes sistemas (EMBRATEL, 1997).

A principal função trata da transformação da sintaxe dos dados. A camada de aplicação passa os dados em uma sintaxe abstrata definida pelo sistema local, por meio da sintaxe de transferência utilizada, os dados são transformados e transmitidos para camada de sessão, quando dados chegam é novamente utilizada a sintaxe de transferência, que transforma eles na sintaxe abstrata e entrega-os a camada de aplicação.

A camada de apresentação não se preocupa com o conteúdo dos dados, e sim com a maneira com que eles são representados. Como exemplo uma aplicação pode encaminhar um dado em sintaxe abstrata de tipo *character* com valor 'A' e a sintaxe de transferência definir então este dado em codificação *American Standard Code for Information Interchange* (ASCII) (EMBRATEL, 1997).

2.1.1.7 Camada de Aplicação

Oferece meios para que as aplicações possam acessar o ambiente de comunicação OSI (SOARES; LEMOS; COLCHER, 1995).

Os aspectos que determinam a comunicação com o ambiente OSI são chamados de entidades de aplicação (EMBRATEL, 1997), cada entidade possui vários elementos que são responsáveis por funções comuns utilizadas.

2.1.2 Modelo Internet

O modelo OSI foi projetado principalmente para redes de longa distância. Mais tarde com a interconexão entre redes locais e de longa distância surgiu o modelo Internet, voltado para ambientes heterogêneos.

A arquitetura Internet, também conhecida como arquitetura *Transport Control Protocol/Internet Protocol* (TCP/IP) não possui uma estrutura formal como a do modelo OSI, ela é definida basicamente utilizando um protocolo para cada camada. O nome TCP/IP vem dos protocolos mais utilizados pela arquitetura.

A organização é dividida em quatro camadas principais, que atuam sobre uma quinta, que não faz parte do modelo (EMBRATEL, 1997). O modelo é composto também por vários componentes, responsáveis pela interface de comunicação entre as camadas (Figura 2).

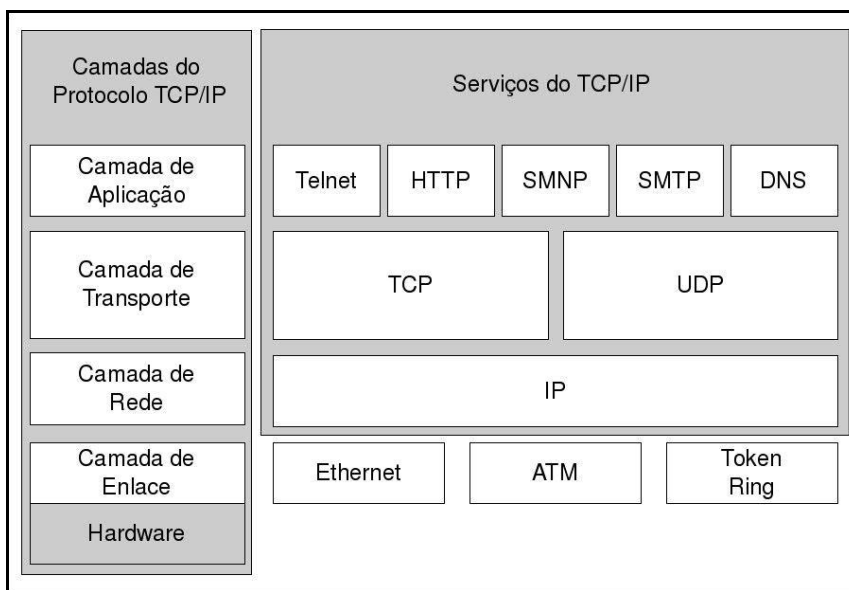


Figura 2. Arquitetura do modelo Internet (TCP/IP).
Adaptado: EMBRATEL. (1997).

2.1.2.1 Protocolos da Camada de Enlace

A arquitetura TCP/IP não define um padrão específico para a camada de enlace, permitindo que qualquer padrão de sub-rede seja usado. Os protocolos que atuam na camada de enlace tem como objetivo transmitir os *bits* de uma máquina a outra diretamente, desde que haja um meio de comunicação. O uso de uma interface de comunicação que implemente a comunicação entre o IP, presente na camada de rede, e a sub-rede utilizada, faz com que o modelo seja compatível com as mais diversas tecnologias de comunicação de redes, como por exemplo, *Ethernet*, *Token Ring*, *Asynchronous Transfer Mode (ATM)*, entre outras.

2.1.2.2 Protocolo da Camada de Rede

O IP, é responsável pela comunicação entre sub-redes, faz com que um pacote possa ser encaminhado de uma estação a outra, mesmo que estas estejam em redes distintas.

Este protocolo encaminha os pacotes contendo todas as informações necessárias para que ele possa chegar ao seu destino, por ser um protocolo não orientado a conexão, ele não garante que os pacotes serão entregues ou que chegarão na mesma ordem em que foram enviados, várias regras são utilizadas, definindo como as máquinas devem gerenciar os pacotes para geração de erros ou descarte. Permite que os pacotes se reconfigurem durante o trajeto pela rede, assumindo diferentes rotas de comunicação.

2.1.2.3 Protocolos da Camada de Transporte

Tem como objetivo a comunicação entre programas e não mais entre estações como nos protocolos anteriores.

Vários aplicativos podem utilizar ao mesmo tempo a rede, a definição do conceito de portas, permite que os pacotes sejam entregues para a aplicação correta. Cada aplicação faz a sua conexão associada a uma porta, que é definida por um número inteiro, as mensagens encaminhadas pela aplicação são então multiplexadas pelos protocolos da camada de transporte que encaminham o pacote a camada de rede. Na estação receptora, os pacotes que chegam a camada de transporte sofrem a demultiplexação e são então encaminhados a aplicação correta (Figura 3).

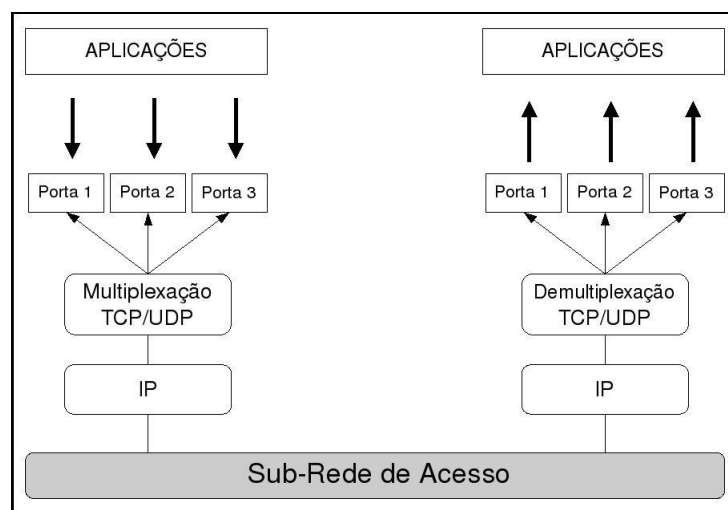


Figura 3. Multiplexação e Demultiplexação de Portas.
Fonte: EMBRATEL. (1997).

Para realizar a comunicação entre os programas a camada de transporte utiliza basicamente dois protocolos, o TCP e o *User Datagram Protocol* (UDP):

- a) **UDP:** é o protocolo da camada de transporte não orientado a conexão, o UDP é descrito no Request for Comments (RFC) 768. Ele permite a uma aplicação enviar datagramas a outra, sem a garantia de que este chegará, o UDP não tem recursos que ofereçam uma transferência confiável dos dados. Os datagramas transferidos que chegam com erro são ignorados e devem ser recuperados pela aplicação. O UDP é muito utilizado em situações onde não há necessidade da

confirmação dos dados enviados, como em uma requisição *Domain Name System* (DNS) por exemplo, onde o cliente envia uma mensagem ao servidor, e este encaminha a resposta, sem a necessidade de verificação da transferência dos dados, caso a resposta seja perdida, o cliente gera um timeout² e reenvia a mensagem. Outras situações onde o protocolo é largamente utilizado, são em aplicações de tempo real, como áudio ou vídeo conferência;

- b) **TCP:** é o protocolo da camada de transporte orientado a conexão, ele é descrito pelo RFC 793. Ele garante a confiabilidade na transferência dos dados, independente da qualidade de serviços oferecidos pela rede, oferecendo mecanismos de controle de erros, controle de fluxo e segmentação dos dados.

2.1.2.4 Protocolos da Camada de Aplicação

A camada de aplicação possui inúmeros protocolos, cada um relacionado ao objetivo da aplicação. Portanto cada aplicação pode ter seu próprio padrão, passando parâmetros para os protocolos das camadas inferiores.

Como exemplo, tem-se o *HyperText Transfer Protocol* (HTTP) utilizado na comunicação entre os navegadores e os servidores *web*.

2.2 PROTOCOLOS DE GERENCIAMENTO DE REDE

Os protocolos de gerência definem a estrutura que o sistema de gerenciamento deverá seguir. A necessidade de padrões advém da tendência nos ambientes empresariais, que

² Erro ocasionado quando uma transmissão ou programa não responde dentro de um determinado período de tempo (SANTOS JUNIOR, 2002).

buscam redes para os mais diversos tipos de usuários e aplicações, tornando-as cada dia maiores e mais complexas (TEIXEIRA JÚNIOR; et al, 1999).

Duas arquiteturas de gerenciamento são disponibilizadas, uma definida pela ISO para o modelo OSI e outra pelo modelo Internet, por meio dos protocolos *Common Management Information Protocol* (CMIP) e *Simple Network Management Protocol* (SNMP) respectivamente. As arquiteturas de gerenciamento utilizam uma base de informações de gerenciamento chamada *Manager Information Base* (MIB) para obter os dados gerenciáveis (TEIXEIRA JÚNIOR; et al, 1999).

2.2.1 Gerenciamento Segundo o Modelo OSI

Dado o horizonte de equipamentos interligados por meio do modelo OSI, surgiu a necessidade de uma arquitetura de gerenciamento para este modelo. Em 1989 a ISO apresentou um esquema básico de arquitetura de gerenciamento e definiu diversas funções de gerenciamento, dividindo-as em cinco áreas funcionais (EMBRATEL, 1997):

- a) **gerência de desempenho:** oferece funções que disponibilizam dados referentes ao nível de desempenho da rede (TEIXEIRA JÚNIOR; et al, 1999). Por meio do monitoramento e da análise estatística dos dados coletados, a gerência de desempenho auxilia para que os administradores possam identificar propriedades da rede, como a necessidade de aumento das linhas de conexão ou os horários de pico da rede;
- b) **gerência de falhas:** define as funções responsáveis por detectar, isolar e corrigir as anormalidades encontradas nos componentes do modelo OSI (SOARES; LEMOS; COLCHER, 1995). Uma falha na rede é definida por uma condição anormal de algum componente que a compõe, como um roteador que

encaminha os pacotes ao destino errado ou um cabo com quebras que dificultam a passagem dos sinais. Por meio de um sistema de gerência de falhas os administradores de rede podem identificar o local que está causando a falha, isolando-o, de modo que a rede continue a operar, mesmo com dependências, e corrigindo ou substituindo o equipamento com falha o mais breve possível;

- c) **gerência de configuração:** define funções que possibilitam a manutenção dos modelos de configuração da estrutura física e lógica da rede, permitindo a reconfiguração da mesma (EMBRATEL, 1997). Uma das funções mais importantes da gerência de configuração é a distribuição das atualizações de softwares, permitindo que todos os terminais possam estar com seus sistemas atualizados sem que os técnicos precisem realizar a atualização individualmente (TEIXEIRA JÚNIOR; et al, 1999);
- d) **gerência de segurança:** é definida pelas funções que respondem aos recursos de acesso de usuários a rede, distribuição de chaves, expiração de senhas, armazenamento de informações relevantes em arquivos de *log*³, e funções de auditoria e de segurança das informações. Esta área funcional pode ser dividida em três categorias de gerenciamento, responsáveis pela segurança do sistema, dos serviços e dos mecanismos, garantindo a política de segurança da rede (EMBRATEL, 1997);
- e) **gerência de contabilização:** provê mecanismos que determinam os custos gastos com a utilização dos recursos da rede (SOARES; LEMOS; COLCHER, 1995). Esta área da gerência faz análise dos recursos de rede utilizados pelos

³ Trata-se de um arquivo que armazena de forma automática uma lista detalhada das ações praticadas por determinado aplicativo. Os arquivos de *log* são utilizados para que o usuário possa futuramente recuperar dados, caso necessário, (SANTOS JUNIOR, 2002) ou para análise de funcionamento e monitoramento do aplicativo.

usuários, podendo definir parâmetros para cobrança pelos serviços utilizados, além de oferecer ao administrador uma facilidade no planejamento da rede.

O CMIP protocolo usado no padrão OSI, acabou caindo em desuso nas redes de computadores devido a sua complexidade de implementação, seu uso é mais intenso em telecomunicações.

2.2.2 Gerenciamento Segundo o Modelo Internet (TCP/IP)

O modelo de referência TCP/IP não descreve áreas funcionais de gerenciamento como no modelo OSI, o gerenciamento é baseado no protocolo SNMP que define os objetivos a serem alcançados com o gerenciamento.

2.2.2.1 SNMP

É o protocolo utilizado pelo modelo TCP/IP para o gerenciamento de redes, definido pelo RFC 1157. Ele atua na camada de aplicação de forma a obter informações relevantes sobre os equipamentos gerenciáveis remotamente, de modo a ser independente da arquitetura e mecanismo de funcionamento dos dispositivos gerenciados, permitir expansões futuras de maneira flexível e minimizar a complexidade das funções de gerenciamento. As informações podem ser obtidas de qualquer equipamento gerenciável, desde que este tenha suporte ao processo agente SNMP, a comunicação realizada entre as estações cliente e servidor é feita geralmente sobre o protocolo UDP da camada de transporte.

Dentre os principais objetivos do SNMP estão o de reduzir o tráfego na rede com mensagens de gerenciamento, diminuir a complexidade e aumentar a flexibilidade das ferramentas de gerência, utilizar operações que sejam implementadas facilmente e permitir a

extensão do protocolo com a adição de novas características. Em contrapartida o protocolo tem algumas limitações, como a falta de segurança, limitações quanto ao método *SetRequest*, falta de comunicação entre os gerentes, seus comandos transportam poucos dados por vez e por utilizar o UDP para comunicação eventualmente perde muitos dados (BLACK, 2008).

A estrutura de gerenciamento utilizada pelo modelo SNMP é baseada em quatro componentes principais: agente, gerente, base de informações de gerenciamento e protocolo de gerenciamento (SNMP) (Figura 4).

O agente é um software disponível junto aos dispositivos que serão gerenciados, responsável por enviar os dados do dispositivo ao gerente (PEREIRA, 2001). O agente faz o envio de dados relevantes de forma automática ou quando solicitado pelo gerente. Ele não tem a função de fazer análise dos dados obtidos, todo o processamento é realizado no gerente, para que não ocorram alterações significativas nos equipamentos gerenciados. Os dados podem ser encaminhados a estação gerente por meio dos *pollings* que são solicitações ao agente ou pelas *traps* que definem limites para alguns recursos do dispositivo, quando um desses limites for ultrapassado uma mensagem é encaminhada, este recurso visa diminuir o tráfego na rede.

O gerente por sua vez é responsável por coletar os dados referentes aos dispositivos. Os dados encaminhados pelos elementos da rede por meio do agente são centralizados no gerente, que tem a capacidade de manipulá-los e apresentar ao usuário de forma transparente informações relevantes.

O conjunto dos objetos que representam os recursos da rede é chamado de base de informações de gerenciamento (MIB). Os objetos controlados pelo protocolo SNMP são obtidos desta base, que é composta por uma árvore de objetos que armazenam o estado interno dos componentes da rede. A semântica e o conjunto dos objetos em que o SNMP atuará são definidos pela MIB, que utiliza a linguagem *Abstract Syntax Notation One*

(ASN.1) para descrever a síntese das mensagens e dos objetos que gerencia (SZTAJNBERG, 1996).



Figura 4. Componentes do modelo SNMP.
Fonte: SILVA, J.C.B. (2004).

2.2.2.2 MIB

É o conjunto dos objetos que podem ser acessados pelos protocolos de gerência, como o SNMP (COMER, 2001). Cada recurso que se deseja gerenciar é modelado, a estrutura resultante formada por esses modelos são os objetos gerenciáveis. Estes elementos contidos na MIB representam recursos reais da rede.

A base utilizada para o gerenciamento de redes baseadas na pilha de protocolos TCP/IP foi apresentada inicialmente pela RFC 1066, a MIB I. Mais tarde surgiu a MIB II, apresentada pela RFC 1158, que expandiu a base de informações, e foi aceita e formalizada como padrão pela RFC 1213.

A estrutura organizacional de uma MIB pode ser vista na forma de uma árvore, onde os objetos recebem identificadores únicos chamados *Objects Identifiers* (OIDs), que são definidos de forma hierárquica. O OID de um objeto gerenciado é construído pela seqüência de inteiros que atravessa a árvore MIB (FARREL, 2005). O OID completo do objeto SNMP por exemplo é definido pela notação ASN.1 como 1.3.6.1.2.1.11 (Figura 5).

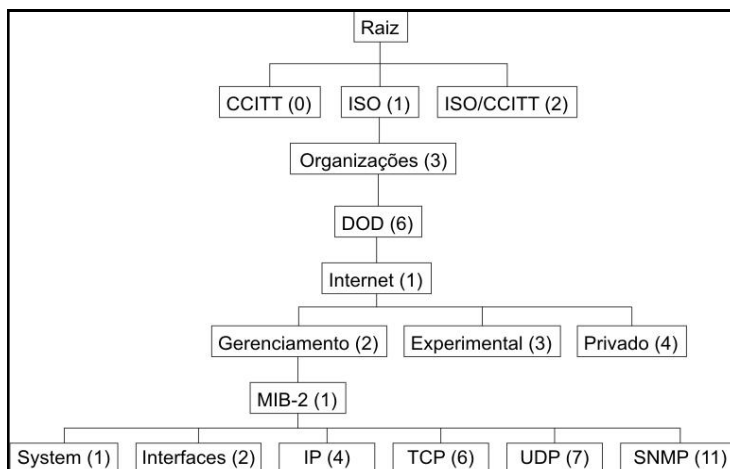


Figura 5. Árvore MIB com o OID dos objetos.
Fonte: FARREL, A. (2005).

Os módulos MIB responsáveis pela Internet, que contém os objetos que podem ser gerenciados por cada camada do modelo, podem ser divididos em três conjuntos principais: Experimental, Privada e Gerenciamento. A primeira constituída dos objetos que estão em fase de desenvolvimento; a segunda representada pelo conjunto de objetos que fornecem informações dos equipamentos privados da rede, sendo desenvolvida pelo próprio fabricante do equipamento; e por último a Gerenciamento da qual deriva a MIB II, formada pelos objetos que fornecem informações gerais sobre os equipamentos tais, como o número de mensagens transmitidas, o estado da interface, entre outras (SILVA, 2004).

A estrutura das informações de gerenciamento contidas na MIB é definida pela *Structure of Management Information* (SMI) especificada por meio da RFC 2578. A SMI define os tipos de dados que formarão a base gerenciável e como eles serão representados e nomeados. A intenção da estrutura é oferecer simplicidade e extensibilidade, fazendo com que a MIB utilize tipos de dados simples, não permitindo o uso de estruturas de dados complexas (STALLINGS, 2005). Por meio de um subconjunto do ASN.1 a SMI pode codificar os objetos que serão transmitidos nas requisições de gerenciamento, transformando a estrutura dos dados de maneira inteligível a humanos e legível pela máquina (FARREL, 2005).

Para dar suporte aos módulos MIB a SMI oferece um conjunto restrito de tipos de

dados da ASN.1, este conjunto de tipos de dados é razoavelmente simples, porém atende a maioria das necessidades do gerenciamento de redes. O conjunto de tipos de dados permitido pela SMI pode ser observado na Tabela 1.

Tabela 1. Tipos de Dados Básicos Definidos na SMI.

Tipo de Dado	Descrição
INTEGER	Inteiros na faixa de -2^{31} a $2^{31}-1$.
UInteger32	Inteiros na faixa de 0 a $2^{32}-1$.
Counter32	Um inteiro não negativo que pode ser incrementado em módulo 2^{32} .
Counter64	Um inteiro não negativo que pode ser incrementado em módulo 2^{64} .
Gauge32	Um inteiro não negativo que pode aumentar ou diminuir, mas não deve exceder um valor máximo. O valor máximo não pode ser maior que $2^{32}-1$.
TimeTicks	Um inteiro não negativo que representa o tempo, módulo 2^{32} , em centésimos de segundo.
OCTET STRING	String de octeto para dados binários ou textuais arbitrários; pode-se limitar a 255 octetos.
IpAddress	Um endereço de inter-rede de 32 bits.
Opaque	Um campo de bit arbitrário.
BIT STRING	Uma enumeração de bits nomeados.
OBJECT IDENTIFIER	Nome administrativamente atribuído a objeto ou outro elemento padronizado. O valor é uma seqüência de até 128 inteiros não negativos.

Fonte: STALLINGS, W. (2005).

Conforme McCloghrie (1999), um tipo de objeto pode ser definido por meio de uma convenção textual, elas servem para apresentar as definições de um tipo de objeto. Como exemplo, um objeto do grupo Internet possui cinco campos (Figura 6):

- a) **objeto**: representa o nome textual do tipo de objeto (OBJECT-TYPE);
- b) **sintaxe**: sintaxe abstrata para um tipo de objeto, define o tipo de dado contido pelo objeto (SYNTAX);
- c) **acesso**: define o tipo de acesso ao objeto, que pode ser de leitura, leitura e escrita, escrita ou sem acesso (MAX-ACCESS);
- d) **descrição**: apresenta a semântica do tipo de objeto definido, a descrição é

importante para que os objetos definidos tenham significado consistente independente da máquina que estiver. As regras apresentadas na descrição devem ser cumpridas para manter a consistência do tipo de objeto definido (DESCRIPTION);

- e) **status:** indica se a definição é atual ou histórica. Assume os valores, *current* para uma definição atual e válida, *obsolete* para uma definição obsoleta que não deve ser implementada e *deprecated* também para uma definição obsoleta, mas que continua em execução para atender a implementações mais antigas (STATUS).

```

sysDescr OBJECT-TYPE
SYNTAX      OCTET STRING
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "Este valor deve incluir todo o nome
    e identificação da versão do tipo de
    hardware do sistema, software de
    sistema operacional e software de
    rede. É obrigatório que contenha
    apenas caracteres ASCII imprimíveis."
 ::= { system 1 }

```

Figura 6. Convenção textual de um tipo de objeto.
Fonte: SZTAJNBERG, A. (2008).

2.2.2.3 Operações do Protocolo SNMP

O mecanismo de comunicação entre a estação gerente e agente acontece de maneira simplificada pelo SNMP, por meio da troca de mensagens, onde a estação gerente emite requisições e a estação agente retorna respostas (FARREL, 2005). As mensagens SNMP são formadas basicamente por um cabeçalho simples que contém a versão do protocolo, o nome da comunidade usada para a transferência e por uma *Protocol Data Unit* (PDU) que contém os dados e a requisição solicitada (Figura 7) (STALLINGS, 1999).

version	community	SNMP PDU				
Mensagem SNMP						
PDU type	request-id	0	0	variable-bindings		
GetRequest PDU, GetNextRequest PDU e SetRequest PDU						
PDU type	request-id	error-status	error-index	variable-bindings		
GetResponse PDU						
PDU type	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variable-bindings
Trap PDU						
name 1	value 1	name 2	value 2	...	name n	value n
variable-bindings						

Figura 7. Formatos SNMP.

Fonte: STALLINGS, W. (1999).

Conforme Stallings (1999), existem cinco PDUs principais para o protocolo, com as seguintes funções (Figura 8):

- a) **GetRequest:** utilizada pela estação gerente para realizar a leitura de um ou uma lista de objetos na estação agente. O agente SNMP retorna uma GetResponse informando se a consulta foi bem sucedida ou retornando um erro;
- b) **GetNextRequest:** funciona da mesma maneira que o GetRequest, porém retorna o próximo objeto da árvore MIB. Pode ser utilizado quando o gerente precisa percorrer uma tabela MIB sem saber quais objetos estão preenchidos ou sem saber o tamanho da tabela;
- c) **SetRequest:** utilizado pelo gerente SNMP para alterar os valores de um ou uma lista de objetos. O agente SNMP retorna uma GetResponse informando se a consulta foi bem sucedida ou retorna um erro;
- d) **GetResponse:** utilizada para dar retorno as requisições GetRequest, GetNextRequest e SetRequest.
- e) **Trap:** utilizada pela estação agente para prover informações de forma assíncrona ao gerente, relatando eventos significantes que ocorram.

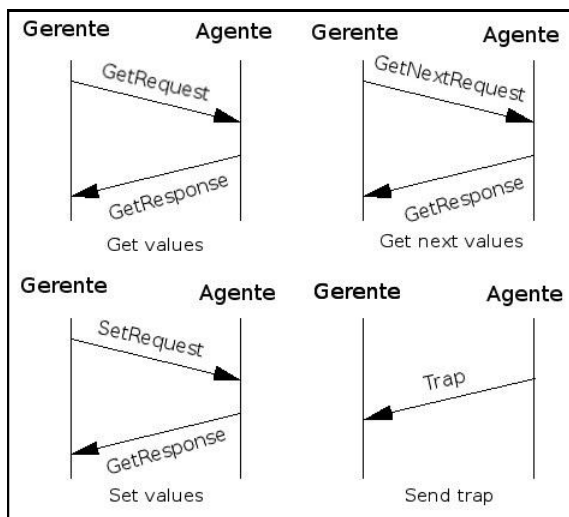


Figura 8. Seqüência das PDU's SNMP.

Fonte: STALLINGS, W. (1999).

2.2.2.4 Versões do Protocolo SNMP

A *Internet Engineering Task Force* (IETF), comunidade internacional responsável pelas padronizações e protocolos envolvidos na arquitetura Internet, definiu o protocolo SNMP. Inicialmente o protocolo foi desenvolvido como uma alternativa temporária de gerenciamento, que seria substituída por um protocolo que atendesse ao modelo OSI. A primeira versão do protocolo apresentava recursos simples, que muitas vezes não atendiam as necessidades de gerenciamento. Devido a complexidade na estruturação de um protocolo para o modelo OSI a sua implementação demorou demasiadamente, de modo que vários fabricantes já estavam incorporando módulos do SNMP em seus equipamentos. Logo, com a intenção de prover recursos mais avançados ao protocolo surgiu o SNMPv2 (SPECIALSKI, 2002).

Com o SNMPv2 várias melhorias foram adicionadas ao protocolo como a troca de mensagens entre gerentes, permitindo uma arquitetura de gerenciamento distribuída, suporte a multi-protocolos na camada de transporte e a definição de uma nova estrutura das informações de gerenciamento, apresentada nas RFCs 1902 a 1904 como SMIV2, além da

criação de duas novas PDUs. Uma das grandes melhorias esperadas para o SNMPv2 era em relação a segurança, que até então permitia que qualquer usuário conectado a rede que pudesse trocar pacotes UDP com os equipamentos gerenciados poderia ler ou alterar os objetos da MIB. Porém, as soluções apresentadas para segurança do protocolo tornaram-se complexas de mais e os esforços da IETF para simplificá-las não surtiram efeito, fazendo com que a solução para segurança fosse deixada de lado.

O SNMPv3 surgiu justamente para resolver o problema relacionado a segurança, definindo recomendações que ofereçam autenticação criptográfica em nível de aplicação para os usuários e definam mecanismos que possam dar privacidade aos objetos da MIB. A estrutura do SNMPv3 permaneceu a mesma da versão 2, utilizando as mesmas PDUs, diferenciando-se apenas pelo cabeçalho das mensagens (FARREL, 2005).

2.3 PARADIGMAS DE GERENCIAMENTO

Os paradigmas organizam a disposição dos elementos pela rede quanto ao gerenciamento, eles definem quais componentes serão responsáveis por coletar os dados, processá-los, armazená-los e disponibilizá-los. A arquitetura dos sistemas de gerenciamento é dividida pelos autores de diferentes formas. Algumas dessas arquiteturas são descritas a seguir.

2.3.1 Definição de Leiwand

Esta classificação divide as arquiteturas de gerenciamento em três modelos denominados Centralizado, Hierárquico e Distribuído (BLACK, apud Leiwand, 2008).

A arquitetura centralizada caracteriza-se por utilizar um gerente de rede único que

recebe, processa e disponibiliza todos os dados referentes ao gerenciamento (Figura 9). O modelo tem como pontos fortes a facilidade nas ações de gerenciamento, por ter todos os recursos centralizados, permite também a expansão facilitada por utilizar uma base de dados única. Em contrapartida gera uma sobrecarga no gerente e torna o sistema vulnerável, caso a estação central para toda a gerência fica comprometida.

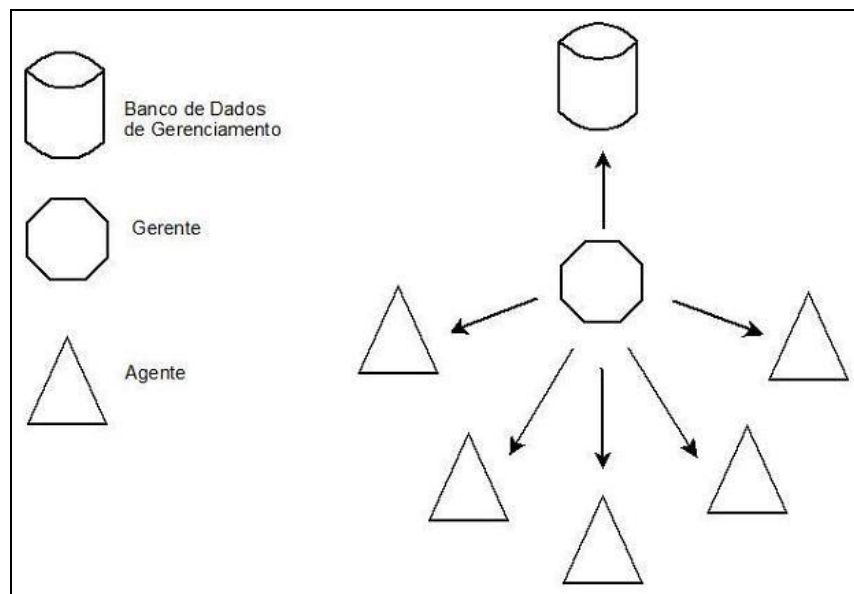


Figura 9. Estrutura da Arquitetura Centralizada.
Fonte: BLACK, T. L. (2008).

Na arquitetura hierárquica a estrutura caracteriza-se por ter várias estações gerenciadoras (Figura 10). Cada estação trabalha individualmente como na estrutura centralizada e repassa suas informações para uma central que armazena os dados vindos de todos os pontos. As principais vantagens estão em torno da distribuição de tarefas o que reduz os gargalos na rede, em contrapartida aumenta os riscos em relação à segurança por ter um número maior de dispositivos envolvidos.

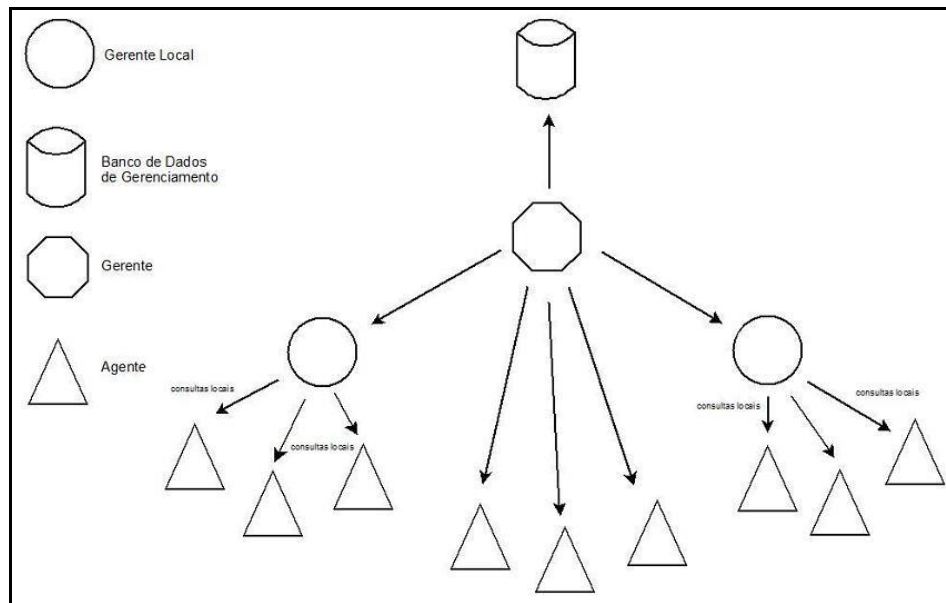


Figura 10. Estrutura da Arquitetura Hierárquica.
Fonte: BLACK, T. L. (2008).

A arquitetura distribuída é um modelo que contempla uma junção dos dois anteriores, criando vários ambientes de gerenciamento, cada um com sua respectiva base de dados (Figura 11). Esses dados são replicados ficando disponíveis a qualquer estação de gerenciamento central, porém gera um maior tráfego na rede.

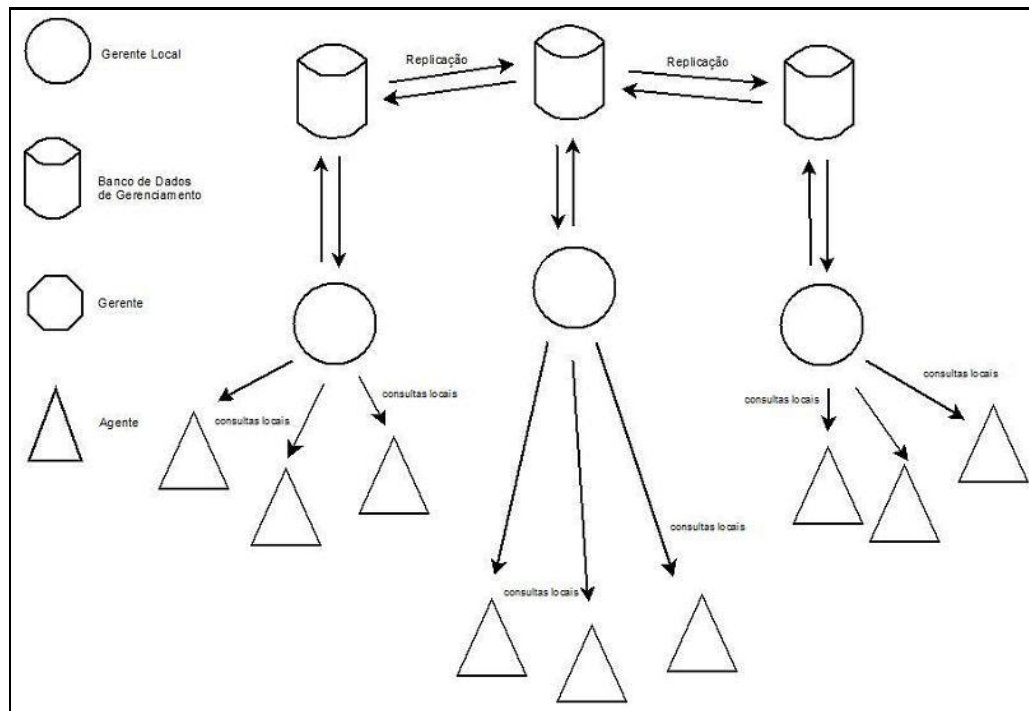


Figura 11. Estrutura da Arquitetura Distribuída.
 Fonte: BLACK, T. L. (2008).

2.3.2 Definição de Schönwälder

Esta classificação divide as arquiteturas de gerenciamento de acordo com o número de componentes que estão presentes na rede. Ela é dividida em: gerenciamento centralizado, quando somente um gerente faz parte do sistema, gerenciamento fracamente distribuído, para estruturas que possuem poucos gerentes em relação aos componentes totais da rede, gerenciamento fortemente distribuído, quando o número de gerentes é grande em relação aos componentes e gerenciamento cooperativo, quando o número de gerentes praticamente se iguala ao número de agentes (BLACK, apud Schönwälder, 2008).

Os sistemas centralizados são mais simples de serem utilizados, porém sua escalabilidade e flexibilidade é menor que nos sistemas fortemente distribuído e cooperativo, onde ocorre também uma melhor distribuição do uso de recursos dos gerentes e da rede.

2.4 SOFTWARES DE GERENCIAMENTO

Inúmeros *softwares* de gerência são utilizados com o intuito de auxiliar os administradores na identificação de problemas e na tomada de decisões, as ferramentas de suporte ao gerenciamento são extensamente utilizadas, independente do tamanho da rede gerenciada.

Os *softwares* utilizados para configurar e monitorar os dispositivos que fazem parte de uma rede, são estruturados pelos fabricantes muitas vezes sem seguirem critérios padronizados de gerenciamento, o que torna demasiadamente complexa a tarefa de gerenciamento aos administradores, dado o número de equipamentos de diferentes fabricantes que podem fazer parte da mesma. Equipamentos que seguem padrões de gerenciamento, como os definidos pelo SNMP, tendem a oferecer maiores facilidades aos administradores, que podem utilizar a mesma ferramenta para o gerenciamento de diversos dispositivos semelhantes.

Os aplicativos de gerenciamento podem ser disponibilizados na forma de *Command Line Interface* (CLI). Ela permite que o usuário opere o aplicativo por meio de um conjunto de comandos textuais definido, que são escritos em um terminal. A sintaxe definida por esses aplicativos apresenta-se em muitos casos de forma complexa, e oferece aos usuários respostas totalmente textuais e nem sempre formatadas de forma legível, o que pode dificultar a interpretação dos resultados. As vantagens na utilização de aplicativos na forma CLI, estão em relação a uma posterior execução de uma tarefa já realizada, como toda a interação entre o usuário e o aplicativo acontece por meio de comandos textuais, arquivos de configuração podem ser armazenados em texto simples, facilitando a modificação, e permitindo ao aplicativo interpretá-los de forma transparente, como se fosse o usuário quem estivesse digitando os comandos (FARREL, 2005).

Outra forma de apresentação disponível para softwares de gerenciamento, são as *Graphical User Interfaces* (GUIs). As interfaces gráficas apresentam-se de forma mais amigável para o usuário, e não requerem que ele saiba a sintaxe dos comandos utilizados pelos dispositivos. Por meio de um conjunto de telas intuitivas ou com um sistema de ajuda contextual o usuário manipula e monitora a rede. As GUIs oferecem ao usuário grande vantagem na representação dos dados obtidos pelo aplicativo, como por exemplo pela navegação por uma tabela, que pode mostrar apenas resultados gerais e ao clicar em um resultado aparece a discriminação dos dados específicos, ou ainda com animações gráficas, que facilitam a comparação de resultados. Uma GUI pode ser criada “em cima” de uma aplicação CLI, mapeando os comandos utilizados e processando os resultados obtidos a fim de apresentá-lo ao usuário de forma mais interativa (FARREL, 2005).

Muitas vezes para o gerenciamento da rede são utilizadas diversas ferramentas em conjunto, dado as limitações de cada uma e o seu foco de aplicação.

Elas estão basicamente divididas em quatro tipos (PEREIRA, 2001):

- a) **ferramentas a nível físico:** responsáveis por identificar problemas relacionados à camada física do modelo OSI, como cabos ou *hardwares* de rede defeituosos;
- b) **monitores de rede:** analisam o tráfego que passa pela rede, como o número de pacotes de cada estação ou os que contém erro;
- c) **analísadores de rede:** ajudam a corrigir problemas na rede por meio da captura e decodificação de pacotes;
- d) **sistemas de gerenciamento de redes integrados (SGRI):** gerenciam a rede completamente de um ponto central.

2.4.1 Plataformas de Gerenciamento

As plataformas de gerenciamento são softwares que podem acoplar várias ferramentas, de modo a criar uma solução o mais completa possível. Um dos objetivos das plataformas é formar um ambiente que atenda várias das áreas funcionais do gerenciamento de redes por meio da comunicação entre as diversas ferramentas. A comunicação entre as ferramentas e a padronização nos métodos de aquisição de informações da rede e de apresentação dessas informações aos usuários, aumenta o horizonte gerenciável, minimizando esforços na identificação de problemas e oferecendo recursos de gerenciamento robustos. A plataforma de gerência em um sistema de gerenciamento pode ser vista pela Figura 12.

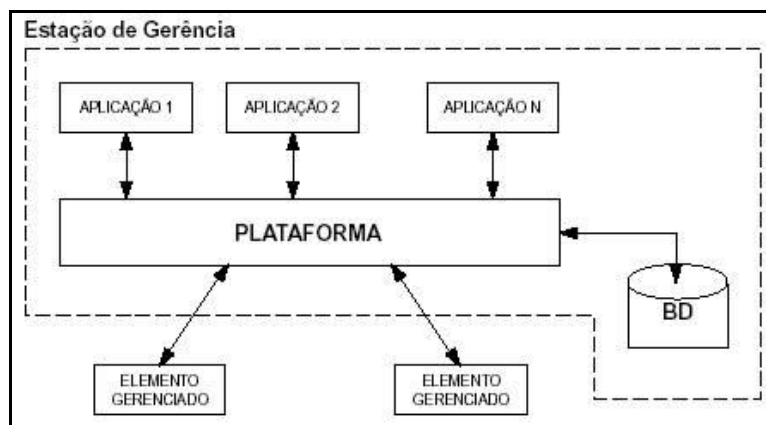


Figura 12. Arquitetura geral de um sistema de gerência.
Fonte: MALIMA CONSULTORIA. (2008).

As plataformas de gerenciamento são relatadas por Pereira (2001), como softwares especiais que possuem basicamente as seguintes características:

- a) funções básicas das aplicações de gerência, como descoberta automática de topologia da rede, exibição gráfica da topologia, rotinas de acesso a MIB e armazenamento das informações coletadas em uma base de dados;
- b) exibição da evolução dos valores das variáveis coletadas dos agentes;

- c) uma *Application Programming Interface* (API) possibilitando que ferramentas possam ser adicionadas a plataforma, beneficiando-se dos recursos oferecidos pela mesma;
- d) uma interface de comunicação objetiva e simples para o usuário final, preferencialmente que ofereça acesso *web*, e que seja multiplataforma;
- e) gráficos que apresentem o monitoramento de tráfego dos agentes.

3 MODELAGEM DE SOFTWARE

A definição de engenharia de software que apresenta a modelagem estrutura o processo de criação de software organizando as etapas de criação, entrega e manutenção dos sistemas (LARMAN, 2000).

Esses processos tem como objetivo permitir que os sistemas desenvolvidos possam ser facilmente modificados, adaptando-se as necessidades específicas, desconhecidas no tempo de criação dos mesmos, além de possibilitar a reutilização das técnicas desenvolvidas e evitar possíveis frustrações com escolhas baseadas em situações isoladas.

O desenvolvimento de software pode ser dividido em um nível macro por três partes gerais, descritas por Larman (2000) como planejar e elaborar, construir e instalar.

Gustafson (2003) ao apresentar os tipos de atividades do ciclo de vida de um sistema especializa as divisões apresentadas a cima, acrescentando de forma geral os passos de testes, treinamento aos usuários e manutenção do sistema.

O ciclo de vida de um software representa as etapas passadas e as atividades executadas durante todo o processo de existência do mesmo.

Dentre os modelos existentes Gustafson (2003) cita quatro, apresentando-os como os mais comuns.

- a) **Modelo Linear Seqüencial (Cascata):** estrutura as fases de desenvolvimento de maneira linear, executando uma etapa após a outra.
- b) **Modelo de Prototipagem:** é baseado no conceito de construir protótipos que visão conceitos e requisitos de partes do sistema, uma vez que o protótipo é analisado e aprovado, começa então o desenvolvimento do sistema final segundo os critérios utilizados nos protótipos e as fases do modelo linear.
- c) **Modelo Incremental:** tem como princípio apresentar um sistema que seja

usável, contendo as características básicas do software desenvolvido, mantendo ao longo do tempo atualizações que adicionem pequenos incrementos ao mesmo.

- d) **Modelo Espiral de Boehm:** é descrito por meio da imagem de um espiral, que continuamente repassa todas as tarefas de desenvolvimento do software.

Os processos de desenvolvimento podem tornar-se pouco flexíveis quando utilizados de maneira única e rigorosa, isto leva o desenvolvimento de software a atuar sobre a perspectiva de utilizar as melhores metodologias disponibilizadas por cada um e criar um processo adaptado as necessidades da aplicação, Larman (2000) chega a citar que a habilidade se criar um bom projeto não depende necessariamente do seguimento de um processo ou método oficial.

3.1 UNIFIED MODELING LANGUAGE (UML)

É uma linguagem padrão, utilizada para o desenvolvimento da modelagem de sistemas. Ela tem como objetivo visualizar, especificar, construir e documentar sistemas, independente do processo de desenvolvimento utilizado (RUMBAUGH; JACOBSON; BOOCH, 2000).

A modelagem UML apresenta de maneira abrangente a arquitetura estrutural de um *software*, definido por meio de seus elementos que baseiam-se principalmente no uso de blocos de construção, regras para interligar esses blocos e alguns mecanismos adicionais. Os blocos de construção da linguagem (Figura 13) são divididos em três conjuntos, definidos como itens, relacionamentos e diagramas.

Os itens são utilizados para descrever as operações do sistema definindo as ações, funcionalidades, componentes, características físicas entre outras.

Os relacionamentos são utilizados para interligar os itens, dando a eles as qualificações de dependência, associação, generalização e realização. Eles definem as diretrizes que o sistema deverá seguir e a comunicação entre suas partes.



Figura 13. Blocos de construção UML.

Os diagramas por sua vez apresentam os itens e suas interligações por meio dos relacionamentos, dando uma visão superficial do sistema modelado (Figura 14). Esses diagramas podem apresentar o sistema sobre diferentes pontos de vista, muitas vezes fazendo referência aos mesmos processos.

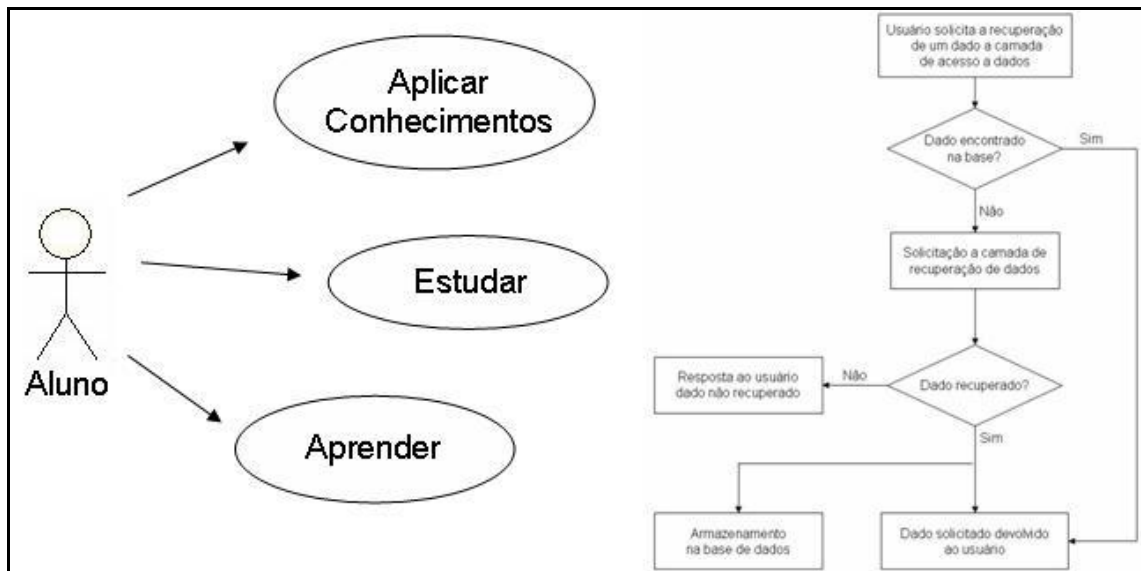


Figura 14. Diagramas UML.

4 TRABALHOS CORRELATOS

A área de gerência de redes vem aprimorando-se a cada dia, devido ao crescimento exponencial das redes, que hoje interligam os mais diversos tipos de dispositivos, tornando-as cada vez mais abrangentes e complexas. A produção de soluções para a administração dos equipamentos e da qualidade dos recursos oferecidos vem se popularizando, e novas alternativas de gerenciamento surgem a cada dia. Abaixo descreve-se algumas soluções de gerenciamento que condizem com o trabalho proposto, pela utilização de softwares de distribuição livre e do padrão de gerenciamento SNMP.

4.1 OPENNMS

É um projeto dedicado ao desenvolvimento de uma plataforma de gerência de redes escrito em linguagem Java e que usa como protocolo padrão de gerenciamento o SNMP. Por ser um projeto de código aberto (*open-source*), possui uma comunidade que desenvolve cada módulo da ferramenta de acordo com padrões pré-definidos. A ferramenta está disponível para *download* no site do projeto (www.opennms.org).

4.2 FREENMS

É um projeto que está sendo desenvolvido por um grupo de pesquisadores do laboratório “Redes Metropolitanas de Alta Velocidade” (ReMAV) da PUCRS. Esse projeto tem como objetivo a construção de uma plataforma de gerenciamento baseada em software livre e que usa o SNMP como protocolo padrão de gerência de redes. Como diferencial, a plataforma possui além das características tradicionais de gerência de redes, a gerência de

nível de serviço, *Service Level Management* (SLM), por meio da funcionalidade de acompanhamento de *Quality of Service* (QoS) contratados, denominados *Service Level Agreement* (SLA).

4.3 NAGIOS

É uma aplicação desenvolvida sobre licença livre e código aberto, responsável pela monitoração da rede. O Nagios monitora *hosts* e serviços que estão presentes na rede, alertando sobre possíveis falhas e relatando problemas resolvidos, apresentando as informações por uma interface *web*.

4.4 ADMINISTRAÇÃO E GERÊNCIA DE REDES DE COMPUTADORES

Trabalho desenvolvido pelo Departamento de Informática e Estatística (INE) do Curso de Pós-Graduação em Ciência da Computação (CPGCC) da Universidade Federal de Santa Catarina (UFSC), que mostra os principais conceitos e terminologias de gerenciamento de redes, além de apresentar uma proposta de gerenciamento que utiliza acordos de nível de serviço, SLA. Outro aspecto importante do trabalho é o esforço na utilização de sistemas de domínio público, que vem de encontro com a tendência e o esforço mundial de utilização de *software*.

5 TRABALHO DESENVOLVIDO

A especificação estrutural da plataforma de gerência proposta compõe as definições para o desenvolvimento da mesma, indicando as principais características presentes em algumas plataformas conhecidas, apresentando as consideradas desejáveis para o sistema e organizando o processo de desenvolvimento, esta dividida metodologicamente nas seguintes etapas: Características das plataformas de gerência analisadas; Características fundamentais da plataforma de gerenciamento; modelagem; Protótipo de manipulação do protocolo SNMP e resultados.

5.1 CARACTERÍSTICAS DAS PLATAFORMAS DE GERÊNCIA ANALISADAS

As plataformas de gerenciamento de redes podem apresentar ao usuário administrador, diversas formas de representação do conteúdo. A presença de gráficos e tabelas comparativas é bastante significativa nas ferramentas, ampliando a possibilidade de interpretação das informações apresentadas, elas possuem diversas funcionalidades em comum, diferenciando-se principalmente pela apresentação do conteúdo ao usuário final.

5.1.1 OpenNms

Possui interface *web* (Figura 15) e apresenta a maior parte dos conteúdos através de tabelas ou de relatórios que podem ser gerados em formatos como *Portable Document Format* (PDF) e *HyperText Markup Language* (HTML), possui gráficos que mostram ao usuário um comparativo de informações. O acesso as funcionalidades é realizado através de menus gerais que estão presentes em todas as páginas na parte superior, e um submenu abaixo

que dá acesso às funcionalidades específicas de cada módulo.

The screenshot displays the OpenNMS web interface for a specific node. The top navigation bar includes links for Node List, Search, Outages, Path Outages, Dashboard, Events, Alarms, Notifications, Assets, Reports, Charts, Surveillance, Distributed Status, and Map. The user is logged in as 'demo' on 23/04/2009 at 19:32 EDT.

The main content area is divided into several sections:

- General (Status: Active):** Shows the node name 'barbrady.internal.opennms.com' and provides links for View Events, View Alarms, Asset Info, and Resource Graphs.
- Availability:** Displays an overall availability of 100.000% for the last 24 hours. A detailed table shows the status of various services:

Service	Status
Overall	100.000%
HTTP-8980	Not Monitored
ICMP	100.000%
SNMP	Not Monitored
SSH	100.000%
StrafePing	100.000%
- Asset Information:** Lists the description as 'Webserver' and includes a comments field.
- SNMP Attributes:** Provides details such as Name (barbrady.internal.opennms.com), Object ID (.1.3.6.1.4.1.8072.3.2.10), Location (Unknown), Contact (root <root@localhost>), and Description (Linux barbrady.internal.opennms.com).
- Notification:** Shows 'You: Outstanding: (Check)' and 'You: Acknowledged: (Check)'.
- Recent Events:** A table listing recent events:

ID	Time	Severity	Description
7650762	23/04/09 17:33:34	Normal	A services scan has been completed on this node.
7650738	23/04/09 17:32:02	Minor	SNMP data collection on interface 172.20.1.11 failed.
7633616	22/04/09 17:32:07	Normal	A services scan has been completed on this node.
7618075	21/04/09 17:30:07	Normal	A services scan has been completed on this node.
7603572	20/04/09 17:28:20	Normal	A services scan has been completed on this node.
- Recent Outages:** States 'There have been no outages on this node in the last 24 hours.'

Figura 15. Interface visual da plataforma OpenNms.

Dentre as funcionalidades foram identificados recursos que mostram o estado dos serviços e das interfaces de rede que rodam nos agentes. Os serviços e dispositivos presentes na rede são identificados automaticamente por meio de varreduras efetuadas pela plataforma, que são realizadas por um intervalo de tempo configurável. O monitoramento dos serviços é realizado através de testes de comunicação entre a plataforma e os agentes, o *software* possui ainda um mecanismo que permite criar calendários de manutenção, onde as informações coletadas nestes períodos não são usadas nos cálculos estatísticos. Possui uma tela de monitoramento com controle de alarmes, que avisam o administrador sobre a ocorrência de anormalidades (BARCELAR, 2008).

5.1.2 Nagios

O Nagios é um aplicativo de gerenciamento distribuído sobre licença livre, e disponibilizado nativamente para plataforma Linux, alguns pacotes inclusive são preparados especificamente para distribuições como Debian, Ubuntu, OpenSuse, entre outras. Uma de suas grandes vantagens está no mecanismo de plugins, que permite a comunidade de desenvolvimento ou aos próprios usuários criarem novos mecanismos para o gerenciamento.

Possui mecanismos para o monitoramento de serviços disponibilizados na rede, como servidores *web* e de e-mail, por meio do módulo SNMP permite monitorar recursos dos sistemas presentes nos clientes. Os problemas detectados nos objetos da rede podem ser notificados aos administradores via e-mail, *pager* ou outro método escolhido pelo usuário. A interface do sistema (Figura 16) pode ser acessada via *web* para visualização de dados da rede ou dos arquivos de *log*, que são gerados automaticamente. A configuração das funcionalidades do sistema acontece por meio de um arquivo chamado *nagios.cfg* que é gerado na instalação do software, trazendo opções padrões (BLACK, 2008).

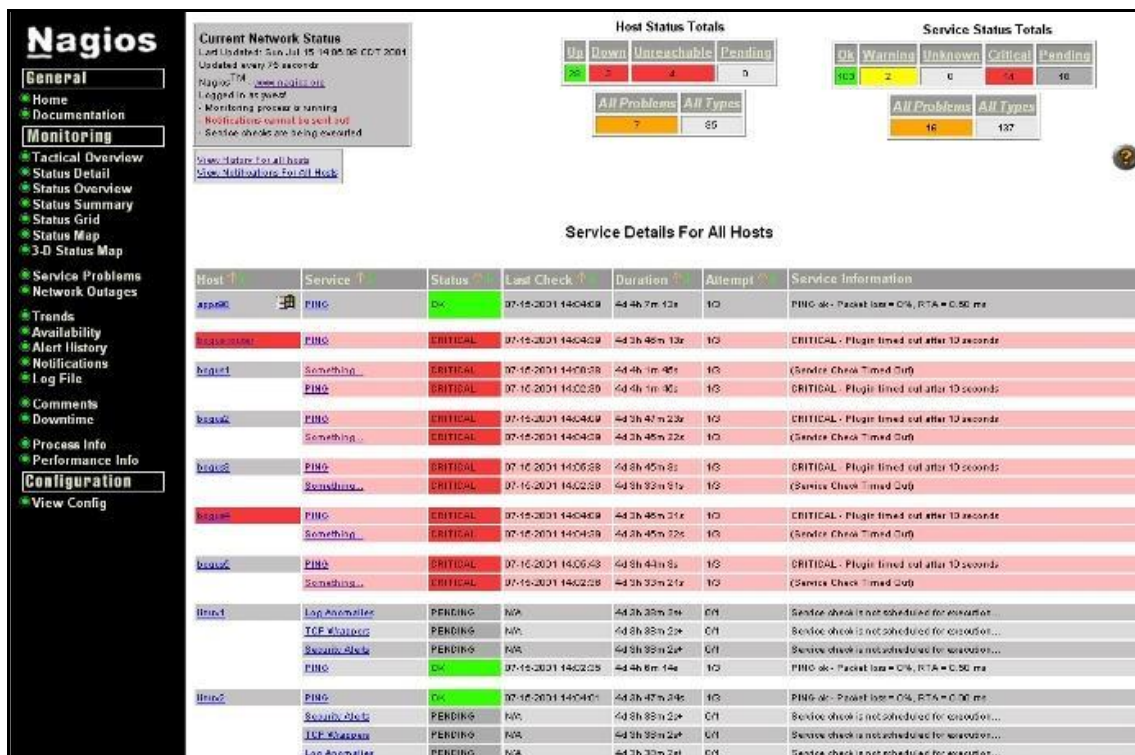


Figura 16. Interface Gráfica do Nagios.

Alguns plugins conhecidos disponibilizados para o Nagios são apresentados na

Tabela 2.

Tabela 2. Plugins do Nagios.

Plugin	Onde Encontrar?	Descrição
Centreon	www.centreon.com	FrontEnd em linguagem PHP para o banco de dados MySQL.
NagCon	vanheusden.com/nagcon	Console monitor para UNIX.
Opsview	opsview.sourceforge.net	Configurador web para alertas de monitoramento.
phpNagios	www.phpnagios.com	Configurador web para o Nagios.
PerfParse	perfpase.sf.net	Analisador de dados sobre performance.

Fonte: Adaptado de BLACK, T. L. (2008).

5.1.3 HP OpenView

É constituído por um conjunto de *softwares* que juntos oferecem recursos para um gerenciamento amplo e extensível, dando suporte ao uso dos protocolos SNMP para redes TCP/IP e para o padrão OSI os protocolos CMIP e CMOT. Possui um *kit* de desenvolvimento, onde interessados podem criar aplicativos que interajam com o protocolo

SNMP, por meio do uso de duas APIs de desenvolvimento, uma que dá acesso aos recursos do SNMP e outra que permite a manipulação de aplicações interagindo com o mapa de equipamentos da rede, permite acesso a MIB pelo usuário final e a manipulação de *traps*. O padrão de desenvolvimento visual, dá suporte ao *Open View Windows* (OVW), baseado no protocolo X11 e Motif⁴. O mapa da rede (Figura 17) é montado com descoberta automática de equipamentos e serviços, possui recursos que permitem a interação entre o mapa e outros aplicativos, de forma que o usuário pode clicar sobre o nó desejado e este executar uma tarefa ou abrir um aplicativo. A representação do estado dos serviços disponíveis é apresentada por diferentes ícones que facilitam a identificação do usuário. No conjunto de *softwares* que compõem o HP OpenView encontram-se também *softwares* agentes, que possibilitam a gerência pelos protocolos TCP/IP, UDP/IP e SNMP, atendendo não só a soluções da HP OpenView, mas a qualquer outro gerente com as características necessárias, o *software* agente possui mecanismos para geração de *traps* e autenticação, controlando acesso não autorizado (TAROUCO, 2008).

⁴ Motif é uma GUI padrão para o desenvolvimento de ferramentas para o X Window System, descrita pela especificação IEEE 1295 (TECHTARGET, 2008).

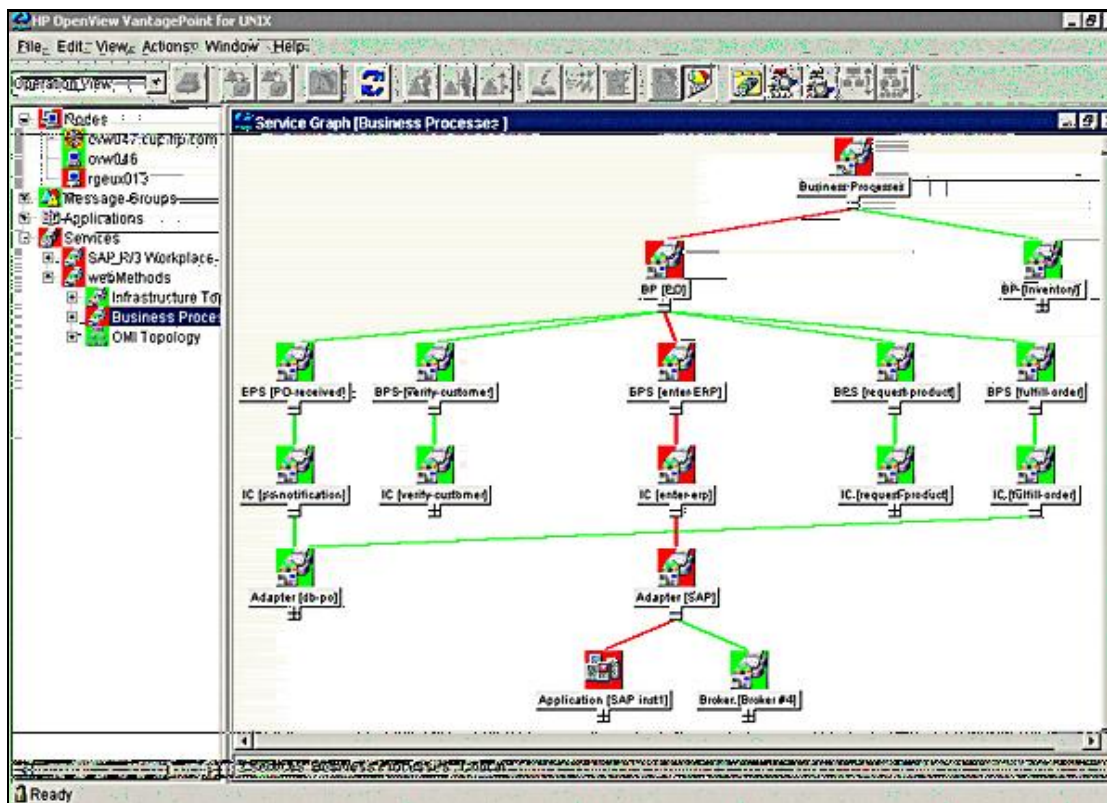


Figura 17. Interface gráfica da plataforma HP OpenView.

5.1.4 IBM Tivoli

O IBM Tivoli disponibiliza um ambiente de gerenciamento de redes baseado na política de regiões de policiamento, *Tivoli Management Region* (TMR), definindo para cada região uma função e uma política de acesso independente, composta pelos seguintes elementos (MELO; VIEIRA; SARI, 2008):

- a) **TMR Server:** Responsável pelo controle de uso e de gerenciamento da região, além de possuir uma base de dados e controlar a comunicação com os *Managed Nodes*;
- b) **Managed Nodes:** Atua como um *Endpoint*, porém com algumas funções extras, como a possibilidade de manipular ou intermediar a comunicação de um conjunto de nós finais na rede;

- c) **Endpoint Gateway:** Atua como um intermediário entre o TMR Server e os *Endpoints*, interpretando as requisições passadas pelo TMR Server e encaminhando aos *Endpoints* as instruções para realizarem o gerenciamento solicitado;
- d) **Endpoint:** Responsável por realizar a operação requisitada no componente final.

Para o gerenciamento da rede a solução Tivoli é dividida em diferentes sistemas, alguns desses sistemas são brevemente apresentados a seguir:

O NetView em geral apresenta a possibilidade de gerência de falhas, configuração e desempenho. Apresenta o estado dos serviços que estão disponíveis e permite a execução de tarefas previamente definidas de acordo com os eventos gerados pelos agentes SNMP. Possui um navegador de MIB onde o usuário pode definir um conjunto de objetos da MIB para monitorar e armazenar a variação de seus valores em uma base de dados, para posterior análise. Disponibiliza também um mapa com os nodos da rede, possibilitando um agrupamento personalizado dos mesmos, de acordo com a necessidade do usuário.

O sistema de monitoramento distribuído (Figura 18) encaminha as informações coletadas dos nodos da rede por meio de gráficos disponibilizados via *web* ou por notificações internas. Esse sistema permite a recuperação de dados referentes aos recursos dos dispositivos, como espaço em disco, carga média do sistema, dados do processador entre outros.

O sistema de inventário armazena em uma base de dados informações sobre o *hardware* e o *software* que estão presentes nos dispositivos, essas informações são coletadas por um processo chamado de *scanning*.

O sistema de *software* é responsável pela distribuição e atualização dos aplicativos que serão instalados nos componentes finais, definido os arquivos que devem ser

manipulados e os procedimentos a serem executados em caso de falhas.

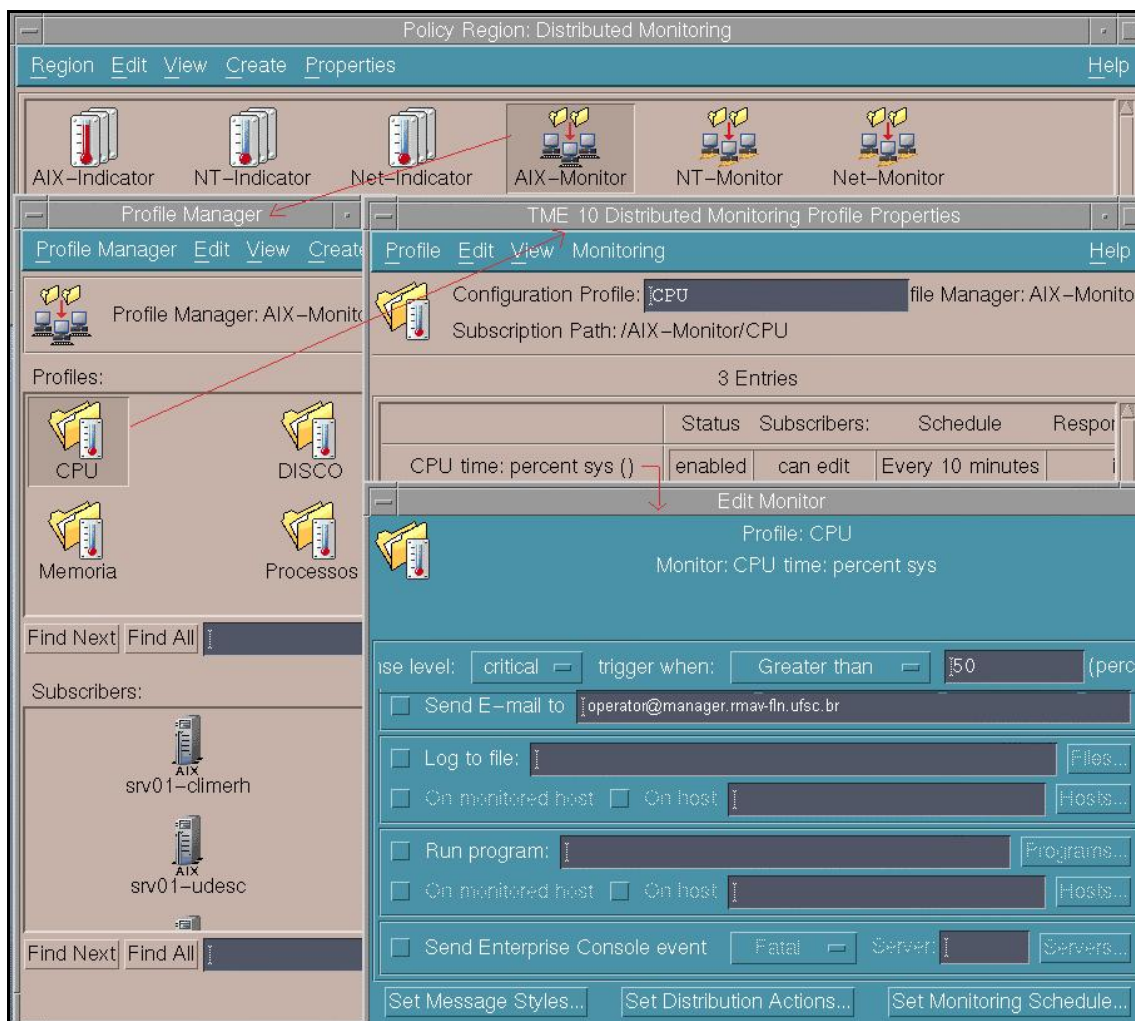


Figura 18. Região de monitoramento de distribuição.
Fonte: MELO, E. T. L; VIEIRA, E. M; SARI, S. T. (2008).

5.1.5 ZenOSS

Construído para atuar como uma solução completa de gerenciamento, possui uma estrutura bem abrangente provendo recursos para comunicação entre seus componentes e definindo regras para o processamento. Segundo Black (2008) a arquitetura estrutural do sistema é dividida basicamente em três grandes camadas denominadas, camada do usuário; camada de dados e camada de coleção e controle de serviços (Figura 19).

A primeira camada é responsável pelos recursos apresentados ao usuário. Possui interface *web* com autenticação dos usuários e traz a descrição dos serviços, redes e dispositivos, apresenta relatórios em HTML e utiliza recursos da linguagem Ajax para melhorar a usabilidade, os relatórios são customizáveis e podem gerar gráficos.

A camada intermediária, responsável pelos dados, os divide em três tipos, um responsável pelos dados do gerenciamento de configuração, outro utilizado para armazenar o histórico das informações e gerar relatórios e por fim uma base que armazena dados provenientes de eventos ocorridos.

Na camada de coleção e controle de serviços são executados processos responsáveis pelo controle do monitoramento, dos eventos e da geração de respostas. Nela é oferecido suporte ao sistema para ser executado de forma distribuída, além de oferecer um mecanismo por meio de linha de comando que permite a interação do usuário sem a necessidade de uma tela gráfica.

Uma característica interessante do sistema é a estrutura da base de dados, que é organizada hierarquicamente, facilitando a localização dos dados. Os dados são armazenados respeitando sua posição na hierarquia, iniciada por *Devices*, *Networks*, *Services*, *Manufacturers*, *Events*, *Locations*, *Systems* e *Groups*. O aplicativo ainda oferece mecanismos de busca automática por dispositivos e serviços na rede, integração de aplicações e o uso de plugins.

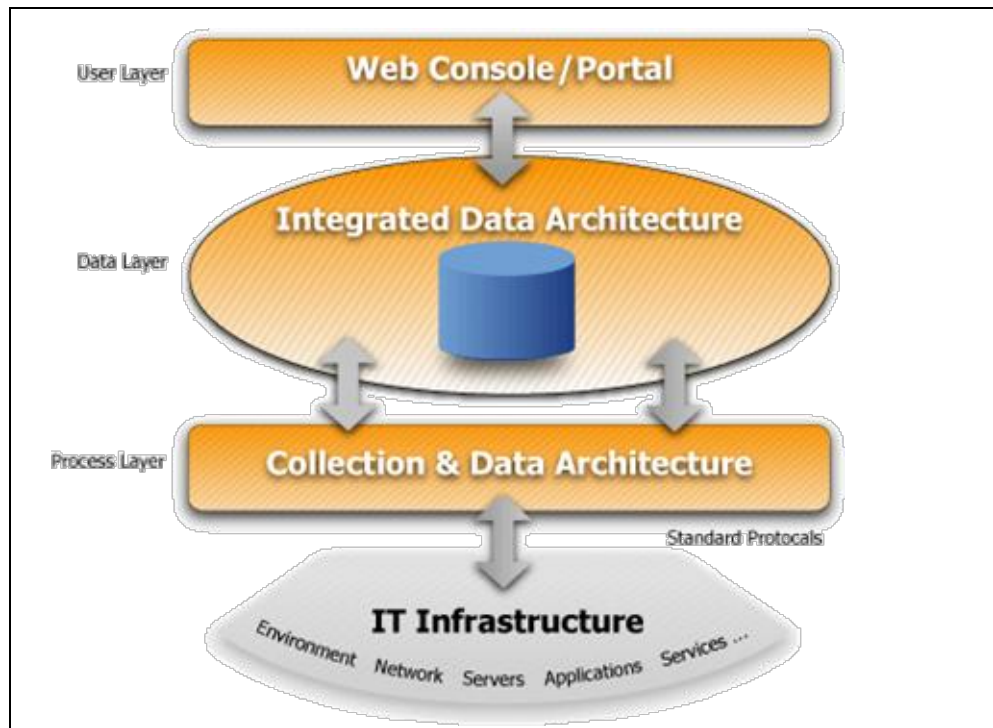


Figura 19. Arquitetura do Sistema ZenOSS.
Fonte: ZENOSS, (2009).

5.2 CARACTERÍSTICAS DA PLATAFORMA DE GERÊNCIA

Dentre as funções que o sistema como um todo deve conter, destacam-se algumas, consideradas básicas ao processo de gerenciamento:

- permitir a identificação automática de equipamentos que estejam na rede e apresentar ao usuário a disposição dos mesmos por meio de mapas;
- identificar automaticamente os serviços disponíveis nos componentes e definir o seu estado atual;
- capturar as variáveis pertencentes a MIB por meio dos agentes SNMP e deixá-las disponíveis ao usuário;
- exibir de maneira comparativa a evolução dos dados coletados da rede disponibilizando-os por meio de gráficos e tabelas;
- ter uma interface visual dinâmica e amigável com o usuário;

- f) alertar sobre eventos ocorridos (*traps*), identificando por meio de ícones e/ou cores os eventos;
- g) controlar o acesso ao sistema por meio da autenticação de usuários previamente cadastrados.

Os aplicativos de interação com o usuário, sejam eles CLIs ou GUIs, tem total independência sobre a manipulação das informações obtidas através dos recursos da plataforma, porém é importante manter um padrão de design visual entre os aplicativos, facilitando a interatividade com o usuário. Inúmeros recursos podem ser utilizados, como interfaces que dividem as tarefas por abas, ou menus gerais de conteúdo.

5.3 MODELAGEM

A finalidade geral do sistema é atender as necessidades de gerenciamento de redes dos administradores, de forma que possam concentrar as funções de gerenciamento em uma única plataforma, simplificando assim o processo. Além de gerentes e gestores tem como públicos alvo, acadêmicos e afins da área que desejam aprimorar seus conhecimentos por meio da análise de recursos incorporados ao sistema ou do desenvolvimento de novos módulos funcionais.

A interatividade do administrador de redes com o sistema de gerenciamento é apresentada através do diagrama de casos de uso (Figura 20), que mostra algumas das possíveis tarefas que podem ser executadas pelos usuários do sistema.

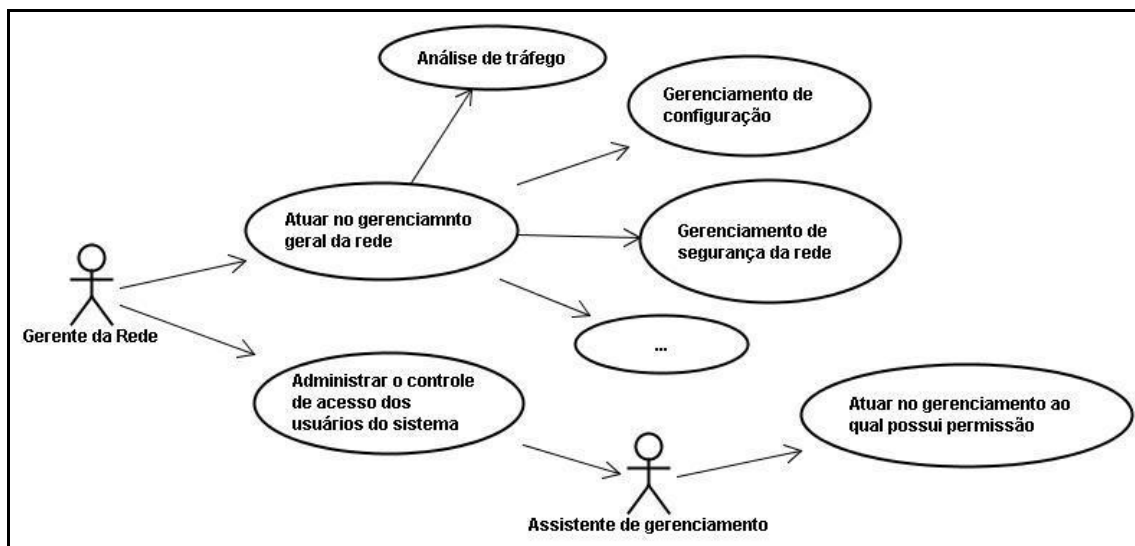


Figura 20. Diagrama de casos de uso.

O diagrama de casos de uso apresentado a cima possui dois atores, que representam o gerente da rede e o assistente de gerenciamento. O gerente da rede é responsável pelas tarefas administrativas da rede, tendo acesso a todos os recursos disponíveis na plataforma. É também responsável por controlar o nível de acesso dos demais usuários da plataforma, podendo assim definir qual usuário terá acesso a determinado recurso da plataforma. Por sua vez o assistente de gerenciamento terá funções definidas pelo gerente e poderá utilizar somente os recursos que forem disponibilizados a ele.

Para assegurar um bom funcionamento da estrutura de gerenciamento, a plataforma de gerência deve estar preparada com mecanismos que permitam eficiência na recuperação de dados sem sobrecarregar a rede, e eficiência na identificação e apresentação de possíveis problemas ao administrador. A estrutura funcional da plataforma, apresentada pela Figura 21 demonstra como são realizadas as tarefas de recuperação e armazenamento de dados, além da integração de módulos distintos com a ferramenta.

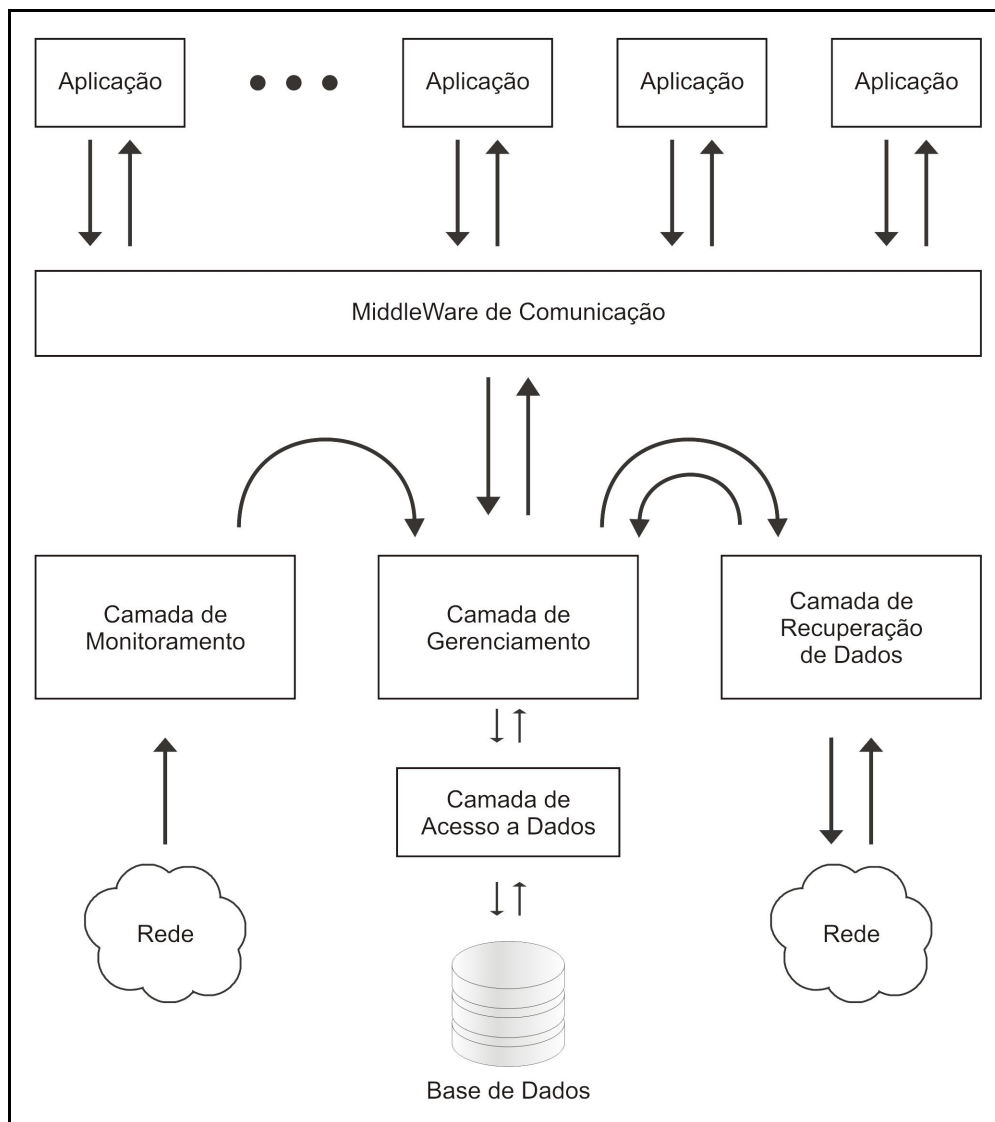


Figura 21. Arquitetura da Plataforma de Gerência.

Cada módulo da plataforma tem uma função definida, o que permite a alteração dos métodos utilizados para executar esta função sem que seja necessário alterar toda a estrutura funcional da plataforma. As funcionalidades de cada módulo são apresentadas a seguir:

- a) **base de dados:** é responsável por armazenar de forma centralizada todos os dados referentes a rede. A importância do armazenamento centralizado e unificado de dados está na grande capacidade de recuperação de informações, dado que a apresentação de informações aos administradores deve ser rápida e

com a maior precisão possível, é mais conveniente pesquisar sobre o histórico da rede em um ambiente único e padronizado do que em diversos recursos, como arquivos de configuração e de *logs*;

- b) **camada de acesso a dados:** é responsável por tratar a comunicação entre a aplicação e a base de dados de forma transparente. Tem por objetivo permitir a independência no armazenamento dos dados, de modo que o banco de dados possa ter sua estrutura modificada sem alterações na aplicação, permitindo o uso de bancos replicados e distribuídos. A camada de acesso aos dados deve conter as informações referentes ao caminho de conexão e autenticação com a base de dados, além de oferecer mecanismos que permitam salvar, atualizar, recuperar e excluir dados;
- c) **camada de monitoramento:** deve implementar recursos que sejam exclusivamente responsáveis por identificar eventos ocorridos na rede, como a geração de *traps* enviadas pelos agentes SNMP. Esta camada deve estar preparada para trabalhar com o protocolo de gerência da rede (neste caso o SNMP), porém não deve permitir que este encaminhe solicitações a rede;
- d) **camada de recuperação de dados:** deve implementar recursos que permitam a recuperação de dados da rede. Esta camada deve estar preparada para trabalhar com o protocolo de gerenciamento da rede (neste caso o SNMP), permitindo que sejam utilizados todos os recursos do protocolo, para recuperação de objetos da MIB e configuração de objetos gerenciáveis. Esta camada sempre deve solicitar o dado requisitado primeiro à base de dados, e somente deverá buscá-lo na rede caso a base não possua;
- e) **camada de gerenciamento:** tem como objetivo atender as requisições das demais camadas pertencentes a plataforma. Por ter acesso a todas as camadas

modulares da plataforma, esta é responsável por interpretar e repassar as solicitações recebidas. Solicitações recebidas pela camada de monitoramento são encaminhadas à base de dados e repassadas ao *middleware* para que sejam apresentadas ao usuário por aplicações de monitoramento. Solicitações de dados ou objetos da MIB vindas das aplicações por meio do *middleware* de comunicação são encaminhadas à camada de recuperação de dados, após a recuperação dos dados, a resposta é então enviada ao *middleware* novamente que a repassa para camada de aplicação;

- f) **middleware de comunicação:** responsável por manter a independência da aplicação, possibilitando que ela possa ser distribuída. Esta camada tem como objetivo centralizar os módulos responsáveis pela parte funcional da plataforma, que tratam da recuperação e armazenamento de dados e da geração de alertas, permitindo assim que vários usuários da plataforma possam acessar estes recursos ao mesmo tempo, cada um com a sua respectiva estrutura de aplicativos;
- g) **camada de aplicativos:** responsável por permitir a interação do usuário que administra a rede com a plataforma de gerenciamento. Deve garantir que novas funcionalidades possam ser adicionadas à ferramenta, os aplicativos agregados seguirão um padrão de desenvolvimento para que possam interagir com o *middleware* de comunicação.

Os aplicativos desenvolvidos para trabalharem em conjunto com a plataforma terão funções diversificadas, utilizando os recursos necessários da plataforma para que esta atenda as suas necessidades. Um requisito indispensável da plataforma de gerencia é a possibilidade de recuperar dados da rede e retornar estes dados à aplicação que solicitou. A Figura 22 demonstra como deve ser o processo de recuperação de dados, para garantir o

menor uso de tráfego na rede e a maior eficiência e precisão na resposta a solicitação.

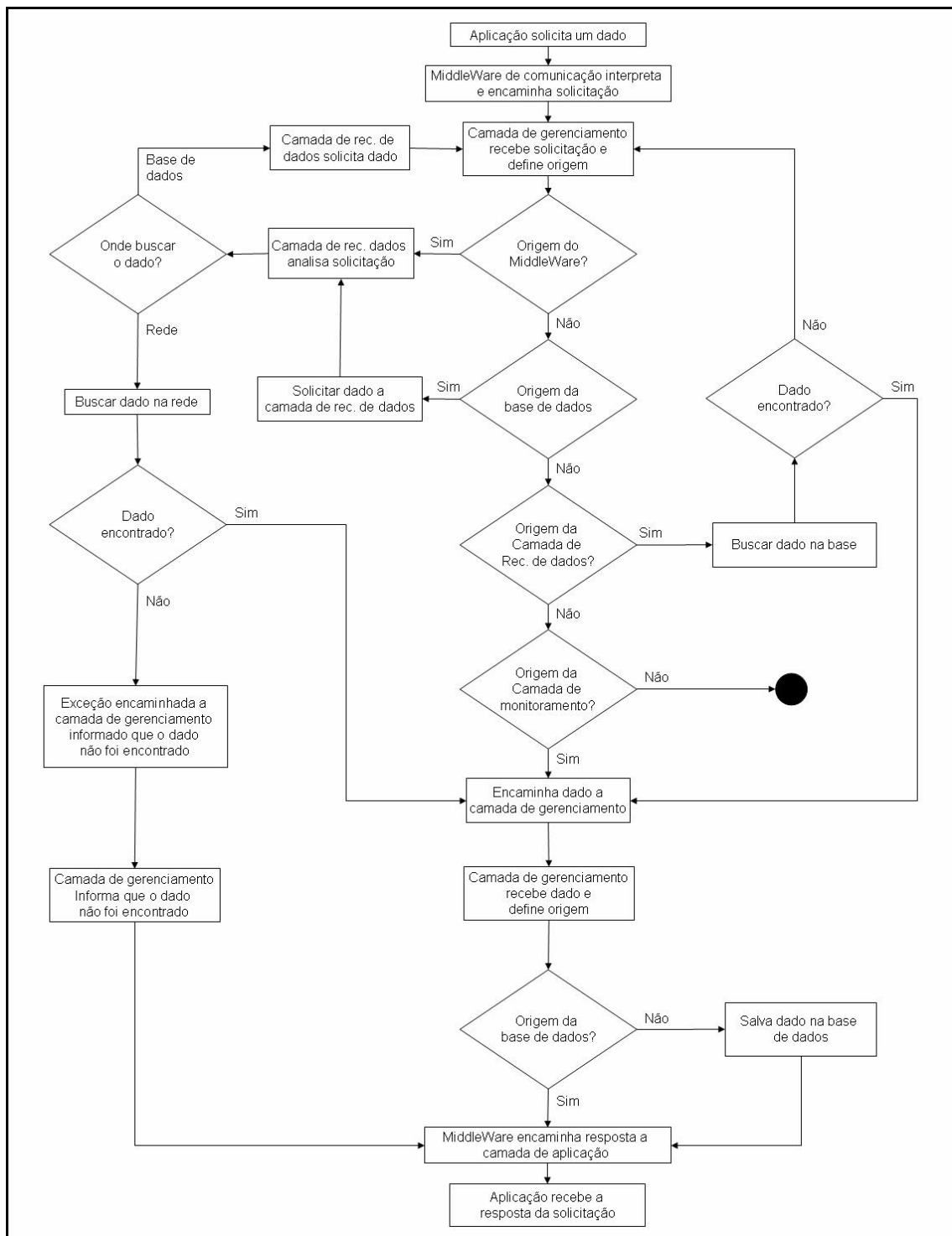


Figura 22. Recuperação de dados na plataforma.

O diagrama apresentado mostra que a camada de gerenciamento tem como função

principal centralizar as transações que ocorrem durante o processo de recuperação de dados, todas as solicitações recebidas por ela são analisadas ao ponto de identificar sua origem, para cada origem, a camada efetua uma operação distinta e repassa o resultado. O processo começa pela requisição de um dado pela camada de aplicação ao *middleware*, que traduz a solicitação e a encaminha para camada de gerenciamento, esta que por sua vez identifica a solicitação vinda do *middleware* e repassa para análise na camada de recuperação de dados. Esta análise tem por objetivo identificar se a busca deve ser realizada diretamente na rede ou se deve ser realizada primeiramente na base. Após a análise caso o dado deva ser buscado diretamente na rede a camada de recuperação de dados faz a requisição por meio do protocolo de gerenciamento e encaminha a resposta a camada de gerenciamento que repassa ao *middleware*, chegando até a aplicação. Seguindo o caminho contrário, caso a análise identifique que a busca deve ser realizada primeiramente na base de dados, a camada de gerenciamento é então solicitada novamente, desta vez tendo como origem a camada de recuperação de dados. Seguindo, a requisição é feita à base de dados que retorna o resultado encontrado até que chegue na aplicação, caso não encontre o resultado na base repassa a busca para que finalmente seja realizada na rede.

Algumas considerações importantes devem ser observadas, como o fato de que a camada de gerenciamento não deve buscar a informação diretamente na base de dados sem antes passar pela análise da camada de recuperação de dados, este procedimento deve ser seguido para evitar situações onde por exemplo, a aplicação solicita a última ocorrência de determinado evento e a base de dados já possui registros sobre este evento, porém o último registro contido na base não é necessariamente a última ocorrência do mesmo.

Observa-se ainda, que caso a origem da requisição encaminhada a camada de gerenciamento não seja definida, esta finaliza o processo. Esse mecanismo de permite novos módulos funcionais sejam adicionados a plataforma desde que se comuniquem diretamente

com a camada de gerenciamento.

Uma característica importante, demonstrada na parte inferior do fluxograma, e que deve ser incorporada a camada de gerenciamento, é a de identificar a origem dos dados que está recebendo e sempre que estes não vierem da camada de acesso a dados salvá-los na base, garantindo dessa forma que a base esteja sempre atualizada.

5.3.1 Modelagem dos Dados

A base de dados da plataforma de gerenciamento deverá ter a capacidade de armazenar as informações dos objetos da MIB coletos dos agentes por meio do protocolo SNMP. Portanto a base deve ser estruturada seguindo a hierarquia da MIB e mantendo tipos de dados compatíveis.

A MIB II, que contém objetos voltados ao gerenciamento, é uma extensão da MIB I que acrescenta novos objetos e grupos para serem utilizados pelo protocolo de SNMP. Ela está estruturada hierarquicamente contendo os seguintes grupos (Figura 23)(RODRIGUES G. C., 2008):



Figura 23. Hierarquia da MIB II
Fonte: RODRIGUES G. C., 2008.

5.3.1.1 Grupo System

Armazena informações sobre o sistema gerenciado. Alguns dos principais objetos do grupo system são descritos abaixo (Tabela 3):

Tabela 3. Tabela de Descrição dos Objetos do Grupo System.

Fonte: RODRIGUES G. C., 2008.

5.3.1.2 Grupo interfaces

Armazena informações sobre as interfaces das subredes do sistema. O grupo interfaces é dividido inicialmente em dois níveis: um que contém o número de interfaces presentes no sistema (IfNumber), e o outro uma tabela com as informações de cada uma das interfaces, dessa maneira formando para cada interface um novo objeto gerenciável (Figura 24).

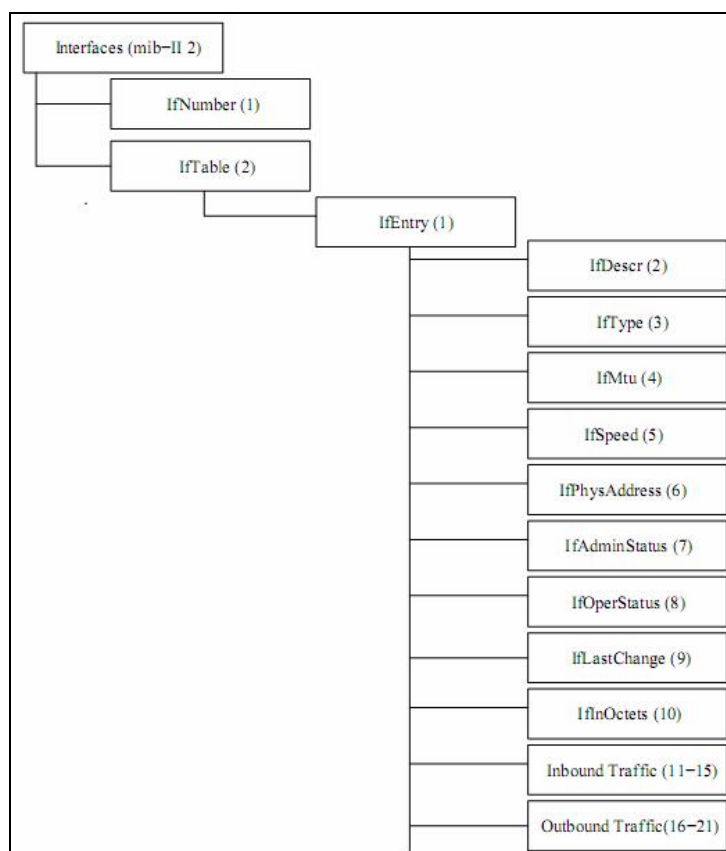


Figura 24. Árvore do Grupo Interfaces.
Fonte: RODRIGUES G. C., 2008.

Os objetos referenciados na Figura 24 como *Inbound Traffic* e *Outbound Traffic* são responsáveis pelos valores referentes aos cálculos do tráfego de entrada e saída respectivamente das interfaces do sistema. Os demais objetos tem suas características

apresentadas na Tabela 4.

Tabela 4. Descrição dos Objetos do Grupo Interfaces.

Objeto	Sintaxe	Acesso	Descrição
ifDescr	DisplayString	RO	Informações gerais sobre o objeto.
ifType	Integer	RO	Tipo da interface.
ifMtu	Integer	RO	Área de dados do protocolo, em octetos.
ifSpeed	Gauge	RO	Total de dados transportados pela interface.
ifPhysAddress	PhysAddress	RO	Endereço da interface.
ifAdminStatus	Integer	RW	Alterar status da interface.
ifOperStatus	Integer	RO	Atual status da interface.
ifLastChange	Time ticks	RO	Valor do “sysUpTime” da interface.
ifInOctets	Counter	RO	Total de octetos recebidos pela interface.

Fonte: RODRIGUES G. C., 2008.

5.3.1.3 Grupo Address Translation (AT)

O grupo AT é composto por um conjunto de fileiras que representam as interfaces físicas presentes ao sistema. Este grupo está atualmente obsoleto.

5.3.1.4 Grupo Internet Protocol (IP)

O grupo IP é considerado um dos grupos mais complexos da estrutura MIB II. Ele contém informações que tratam da operação e implementação do protocolo IP nos dispositivos que o utilizam como *routers* e *hosts*. Alguns dos objetos presentes neste grupo podem ser observados pela Figura 25.

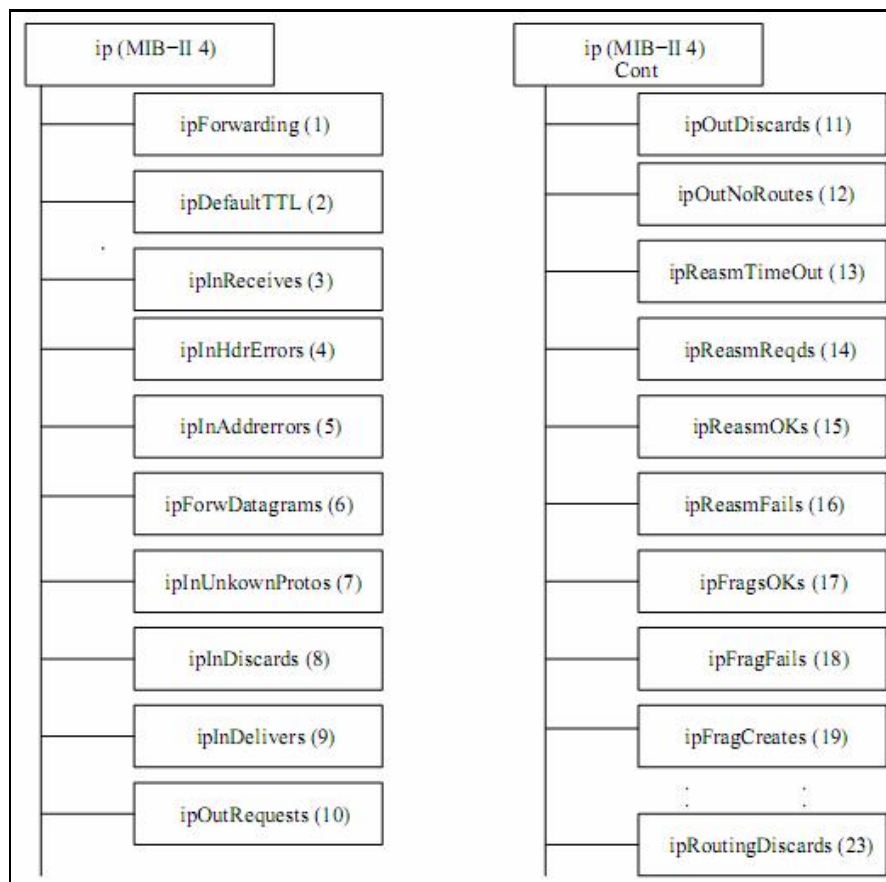


Figura 25. Árvore do Grupo IP.
Fonte: RODRIGUES G. C., 2008.

Dentre os objetos que pertencem ao grupo IP quatro se destacam, o IpAddrTable(20), que possui informações sobre o endereço IP do sistema; o IpRouteTable(21), que possui informações sobre o roteamento; o IpNetToMediaTable(22), que armazena o mapeamento entre os endereços *Media Access Control* (MAC) e IP; e o IpForward(24), que traz uma nova implementação para o IpRouteTable com mais objetos.

As funcionalidades de alguns outros objetos do grupo IP são apresentadas pela Tabela 5.

Tabela 5. Descrição dos Objetos do Grupo IP.

Objeto	Sintaxe	Acesso	Descrição
ipForwarding	Integer	RW	Pode atuar como gateway(1) ou não (2).
ipDefaultTTL	Integer	RW	Valor <i>default</i> do campo <i>Time-To-Live</i> .
ipInReceives	Counter	RO	Total de datagramas recebidos.
ipInHdrErrors	Counter	RO	Dat. descartados devido a erros de cabeçalho.
ipInAddrErrors	Counter	RO	Dat. descartados devido a erro de endereçamento.
ipForwDatagrams	Counter	RO	Dat. encaminhados ao gateway.
ipInUnknwnProtos	Counter	RO	Dat. recebidos corretamente, descartados.
ipInDiscard	Counter	RO	Dat. recebidos corretamente, descartados.
ipInDelivers	Counter	RO	Dat. recebidos e entregues corretamente.
ipOutRequests	Counter	RO	Dat. fornecidos para transmissão de IP não local.
ipOutDiscards	Counter	RO	Dat. enviados corretamente, descartados.
ipOutNoRoutes	Counter	RO	Dat. descartados devido a não possuir rota correta.
ipReasmTimeout	Integer	RO	Tempo limite para recebimento de datagramas.
ipReasmReqds	Counter	RO	Dat. recebidos que necessitam ser remontados.
ipReasmOks	Counter	RO	Dat. corretamente remontados.
ipReasmFails	Counter	RO	Dat. que não foram remontados.
ipFragOks	Counter	RO	Dat. corretamente fragmentados.
ipFragFails	Counter	RO	Dat. que não foram fragmentados.
ipFragCreates	Counter	RO	Dat. fragmentados pelo objeto.
ipRoutingDiscards	Counter	RO	Número de rotas inválidas, descartadas.

Fonte: RODRIGUES G. C., 2008.

5.3.1.5 Grupo Internet Control Message Protocol (ICMP)

O grupo ICMP possui objetos que contém informações sobre o protocolo ICMP, contendo basicamente contadores de mensagens de entrada e saída do protocolo (Figura 26).

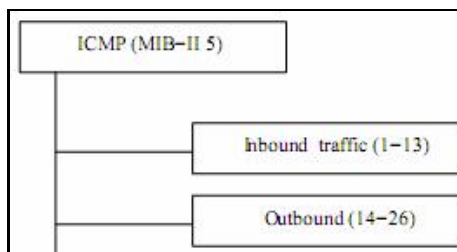


Figura 26. Árvore do Grupo ICMP.

Fonte: RODRIGUES G. C., 2008.

5.3.1.6 Grupo Transmission Control Protocol (TCP)

O grupo TCP contém objetos que possuem informações sobre a implementação e operação do protocolo TCP. A Tabela 6 descreve a os principais objetos do grupo.

Tabela 6. Descrição dos Objetos do Grupo TCP.

Objeto	Sintaxe	Acesso	Descrição
tcpRtoAlgorithm	Integer	RO	Tempo de retransmissão(Valor).
tcpRtoMin	Integer	RO	Tempo mínimo de retransmissão.
tcpRtoMax	Integer	RO	Tempo máximo de retransmissão.
tcpMaxConn	Integer	RO	Limite de conexões suportadas.
tcpActiveOpens	Counter	RO	Número de conexões ativas suportadas.
tcpPassivesOpens	Counter	RO	Número de conexões passivas suportadas.
tcpAttemptFails	Counter	RO	Número de tentativas de conexões que falharam.
tcpEstabResets	Counter	RO	Número de reinicializações.
tcpCurrEstab	Gauge	RO	Número de conexões <i>Established</i> ou <i>Close-Wait</i> .
tcpInSegs	Counter	RO	Segmentos recebidos, incluso com erros.
tcpOutSegs	Counter	RO	Segmentos enviados, excluindo os retransmitidos.
tcpRetranSegs	Counter	RO	Segmentos retransmitidos .
tcpConnTable	Sequence	NA	Tabela de conexões TCP.
tcpInErrors	Counter	RO	Segmentos recebidos com erros .
tcpOutRsts	Counter	RO	Segmentos enviados contendo a <i>RST Flag</i> .

Fonte: RODRIGUES G. C., 2008.

5.3.1.7 Grupo User Datagram Protocol (UDP)

Possui informações sobre a implementação e operação do protocolo UDP. Um objeto que merece destaque neste grupo é o `udpTable`, que representa uma tabela que contém informações sobre os pontos finais, como uma aplicação que está recebendo o conteúdo de um datagrama. As funcionalidades dos principais objetos do grupo são apresentadas pela Tabela 7.

Tabela 7. Descrição dos Objetos do Grupo UDP.

Objeto	Sintaxe	Acesso	Descrição
udpInDatagrams	Counter	RO	Dat. UDP entregues.
udpNoPorts	Counter	RO	Dat. UDP recebidos pela aplicação errada.
udpInErrors	Counter	RO	Dat. UDP recebidos, porém com erros.
udpOutDatagrams	Counter	RO	Dat. UDP enviados.
udpTable	Sequence	NA	Tabela UDP.

Fonte: RODRIGUES G. C., 2008.

5.3.1.8 Grupo External Gateway Protocol (EGP)

Contém informações sobre a implementação e operação do protocolo EGP. As funcionalidades dos principais objetos deste grupo são observadas por meio da Tabela 8.

Tabela 8. Descrição dos Objetos do Grupo EGP.

Objeto	Sintaxe	Acesso	Descrição
egpInMsgs	Counter	RO	Mensagens EGP recebidas sem erros.
egpInErrors	Counter	RO	Mensagens EGP recebidas com erro.
egpOutMsgs	Counter	RO	Mensagens EGP enviadas.
egpOutErrors	Counter	RO	Mensagens EGP não enviadas, falha de protocolo.
egpNeightTable	Sequence	NA	Tabelas EGP.
egpAs	Integer	RO	Número de sistemas.

Fonte: RODRIGUES G. C., 2008.

5.3.1.9 Grupo Transmission

Contém informações referentes aos esquemas de transmissão e acesso dos protocolos que atuam sobre as interfaces. O grupo *transmission* é utilizado para definir um prefixo aos nomes dos objetos que representam uma definição do padrão Internet. Alguns dos objetos do grupo *transmission* são descritos na Tabela 9.

Tabela 9. Descrição dos Objetos do Grupo Transmission.

Objeto	Descrição
dot3StatsIndex	Um valor índice que identifica unicamente a interface para um meio ethernet-like. A interface identificada pelo valor particular deste índice é a mesma interface como identificada pelo mesmo fator do ifIndex.
dot3StatsSingleCollisionFrames	Um contador de quadros transmitidos com sucesso em uma interface particular pelo qual a transmissão é inibida por exatamente uma colisão. Um quadro que é contado por uma instancia desse objeto é também contado pela instancia correspondente do ifOutUcastPkts, ifOutMulticastPkts, ou ifOutBroadcastPkts, e não é contado pela instancia correspondente do objeto dot3StatsMultipleCollisionFrames.
dot3StatsMultipleCollisionFrames	Uma contagem dos frames que tiveram erro de recepção em uma interface, devido a um recebimento de um erro interno pela subcamada MAC. O frame somente é contado pela instância deste objeto se ele não é contado pela instância correspondente tanto do objeto dot3StatsFrameTooLongs, quanto do objeto dot3StatsAlignmentErrors, ou do objeto dot3StatsFCSErrors. O significado preciso representada pela instancia deste objeto é a aplicação específica. Em particular, uma instancia desse objeto possivelmente representa uma contagem de erros recebidos em uma interface particular que não é contada de outra forma.
dot3StatsDeferredTransmissions	Um contador de quadros no qual a primeira tentativa de transmissão em uma interface particular é atrasada porque o meio está ocupado. O contador representado por uma instancia desse objeto não inclui quadros envolvidos em colisões.

Fonte: Adaptado de KOGA, I. (2005).

5.3.1.10 Grupo SNMP

Possui informações sobre os recursos de utilização do protocolo SNMP. Os objetos pertencentes a este grupo podem ser observados pela Figura 27.

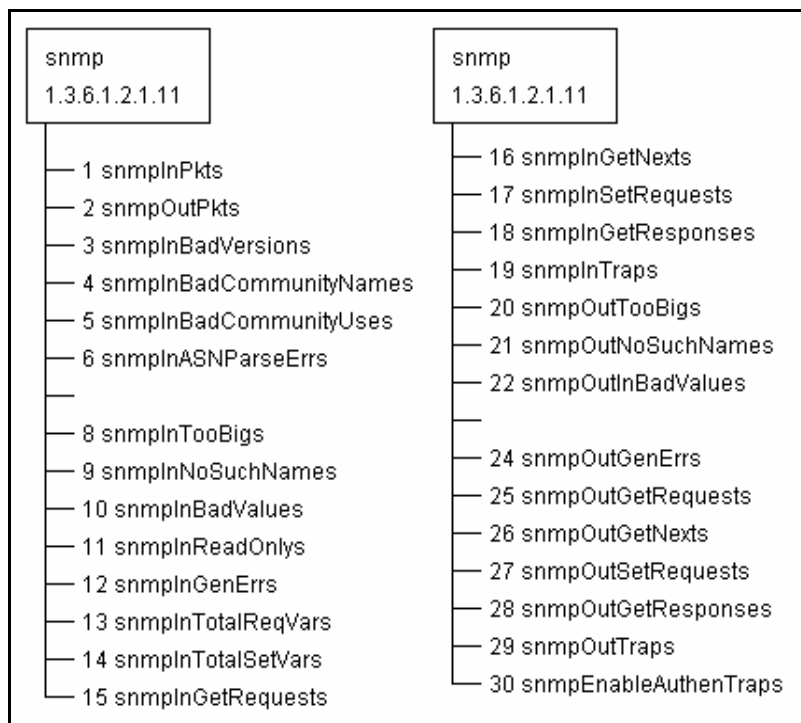


Figura 27. Objetos do grupo SNMP.

Os objetos do grupo SNMP tem suas funções descritas resumidamente abaixo (MENDONÇA, 2009):

- a) **snmpInPkts**: Número mensagens entregues ao SNMP pelo serviço de transporte.
- b) **snmpOutPkts**: Número de mensagens passadas do SNMP para o serviço de transporte.
- c) **snmpInBadVersions**: Número de mensagens entregues não suportadas pela versão do SNMP.
- d) **snmpInBadCommunityNames**: Número de mensagens entregues com o nome de comunidade desconhecida.
- e) **snmpInBadCommunityUses**: Número de mensagens entregues bloqueadas pela comunidade.
- f) **snmpInASNParseErrs**: Número de erros ao decodificar mensagens SNMP.
- g) **snmpInBadTypes**: Número de PDUs desconhecidas ou não suportadas.

- h) **snmpInTooBigs**: Número de PDUs entregues onde o campo error-status foi 'too big'.
- i) **snmpInNoSuchNames**: Número de PDUs entregues onde o campo error-status foi 'noSuchName'.
- j) **snmpInBadValues**: Número de PDUs entregues onde o campo error-status foi 'BadValue'.
- k) **snmpInReadOnlys**: Número de PDUs entregues onde o campo error-status foi 'readOnly'.
- l) **snmpInGenErrs**: Número de PDUs entregues onde o campo error-status foi 'genErr'.
- m) **snmpInTotalReqVars**: Número de objetos recuperados com sucesso de PDUs do tipo Get-Request ou Get-Next-Request.
- n) **snmpInTotalSetVarss**: Número de objetos alterados por PDUs do tipo Set-Request.
- o) **snmpInGetRequests**: Número de PDUs Get-Request aceitas e precessadas.
- p) **snmpInGetNexts**: Número de PDUs Get-Next-Request aceitas e precessadas.
- q) **snmpInSetRequests**: Número de PDUs Set-Request aceitas e precessadas.
- r) **snmpInGetResponses**: Número de PDUs Get-Responses aceitas e precessadas.
- s) **snmpInTraps**: Número de PDUs Trap aceitas e precessadas.
- t) **snmpOutTooBigs**: Número de PDUs geradas onde o campo error-status foi 'tooBig'.
- u) **snmpOutNoSuchNames**: Número de PDUs geradas onde o campo error-status foi 'noSuchName'.
- v) **snmpOutBadValues**: Número de PDUs geradas onde o campo error-status foi

'badValue'.

- w) **snmpOutReadOnlys**: Número de PDUs geradas definidas como read-only.
- x) **snmpOutGenErrs**: Número de PDUs geradas onde o campo error-status foi 'genErr'.
- y) **snmpOutGetRequests**: Número PDUs Get-Request geradas pela entidade SNMP.
- z) **snmpOutGetNexts**: Números de PDUs Get-Next-Request geradas pela entidade SNMP.
- aa) **snmpOutSetRequests**: Número de PDUs Set-Request geradas pela entidade SNMP.
- bb) **snmpOutGetResponses**: Números de PDUs Get-Response geradas pela entidade SNMP.
- cc) **snmpOutTraps**: Números de PDUs Trap que geradas pela entidade SNMP.
- dd) **snmpEnableAuthenTraps**: Define se o agente pode gerar traps referentes a falhas de autenticação.

5.3.2 Análise de Funcionalidades que Podem Ser Adicionas ao Modelo

Todo o conjunto de informações úteis para as cinco áreas de gerenciamento podem ser adicionadas ao sistema. Algumas tecnologias e aplicabilidades que podem ser adicionadas a plataforma são discutidas a seguir:

- a) acompanhamento de qualidade de serviço (QoS). Responsável por garantir que os SLAs firmados pela mantenedora da rede sejam cumpridos;
- b) integração de ferramentas funcionais existentes. Permitindo que softwares simples que coletam dados da rede como NMAP e TraceRoute sejam

- utilizados pela plataforma;
- c) integrar ferramentas de apresentação de conteúdo, como a MRTG, para que colem os dados da plataforma;
 - d) técnicas de inteligência artificial para recuperar conhecimento da base de dados gerando relatório mais elaborados e indicando caminhos a serem seguidos pelos administradores, ou minimizando falhas baseadas em casos já ocorridos;
 - e) módulo de inventário, que armazene informações sobre os equipamentos e os softwares disponíveis na rede. Utilizado também para o planejamento do crescimento da rede;
 - f) soluções de ativos de TI, como as descritas pela bibliotecade práticas ITIL;
 - g) módulo que permita o gerenciamento por meio de dispositivos móveis, como celulares, expandindo o ambiente utilizado pelos administradores;
 - h) gerenciamento de dispositivos sem fio, avaliando inclusive a qualidade do sinal dos componentes.

5.4 PROTÓTIPO DE MANIPULAÇÃO DO PROTOCOLO SNMP

Um protótipo foi desenvolvido para exemplificar a manipulação do protocolo SNMP e sua interatividade com o usuário final, o desenvolvimento foi realizado utilizando-se a linguagem de programação Java por meio do Kit de Desenvolvimento Java (JDK) fornecido livremente pela Sun Microsystems Inc., além do uso de uma API para o acesso e manipulação do protocolo de gerência, disponibilizada pela Advent Net, o aplicativo de exemplo é multiplataforma e tem sua GUI implementada com uso dos recursos da biblioteca *swing* do JDK.

Este protótipo não segue os critérios de desenvolvimento sugeridos pela

modelagem da plataforma, a implementação foi realizada unicamente com o intuito de apresentar uma das muitas maneiras possíveis de se manipular o protocolo de gerência, a arquitetura simplificada utilizada no protótipo justifica-se pelo pouco tempo hábil para o desenvolvimento. Por tratar-se de um protótipo as funcionalidades foram implementadas para situações específicas, caso estas sofram variações podem ocorrer erros inesperados.

Tomando como ponto inicial a interface visual do aplicativo, tem-se a tela que realiza efetivamente a interação com o protocolo SNMP (Figura 28). Ela divide-se basicamente em três setores, responsáveis respectivamente por, navegar pela árvore da MIB carregada (setor 1), exibir o retorno dos métodos invocados que manipulam o SNMP (setor 2) e exibir a descrição das características do objeto selecionado na árvore da MIB carregada (setor 3).

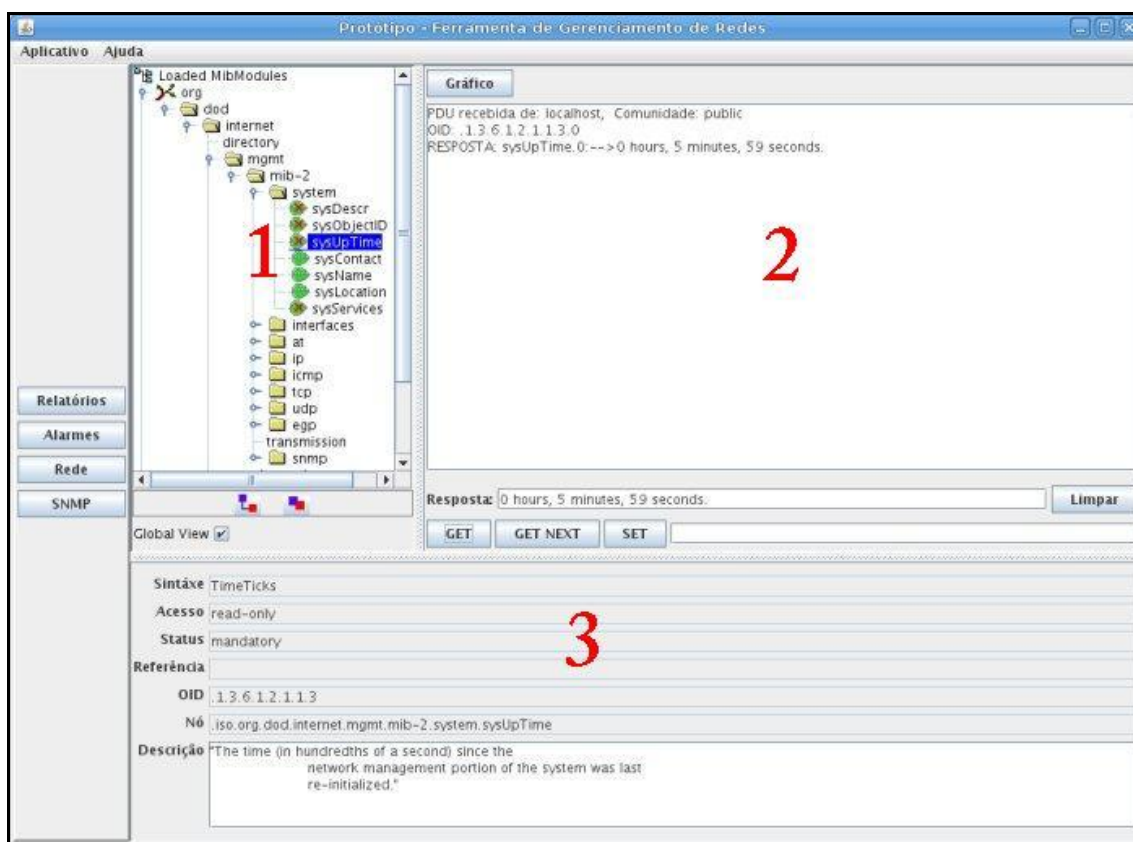


Figura 28. Tela de manipulação do protocolo de gerência SNMP.

A funcionalidade da árvore (setor 1) esta associada ao arquivo carregado por meio do menu *Aplicativo* item *Carregar MIB*. O arquivo carregado deve ser do tipo MIB, esses arquivos geralmente são disponibilizados pelo fabricante do equipamento que possui suporte ao SNMP ou pelo fabricante do *software* agente que roda no mesmo. Ao expandir-se a árvore e clicar-se em um nó que represente um objeto, o aplicativo imediatamente apresenta a descrição de suas características (setor 3).

Os botões denominados *GET*, *GET NEXT* e *SET* (setor 2), são responsáveis pela execução dos métodos definidos no SNMP respectivamente com os nomes de *GetRequest*, *GetNextRequest* e *SetRequest*, exibindo o retorno pela implementação do método *GetResponse* na caixa de texto *Resposta* e detalhando o mesmo na área de texto localizada acima. O botão *SET* só deve ser utilizado para objetos que permitam o acesso a escrita (*read-write*), respeitando-se o tipo de dado do objeto.

Um gráfico pode ser acessado por meio do botão *Gráfico* (setor 2), ele apresenta a variação dos valores do objeto selecionado na árvore em relação ao tempo, os valores referentes ao objeto são recuperados no intervalo de *polling* definido na tela de configuração.

A configuração do protocolo SNMP para acessar o objeto gerenciável é realizada por meio da tela *Settings Dialog* (Figura 29), localizada no menu *Aplicativo* item *Configurar SNMP*. Assim como a árvore e o gráfico, essa tela é disponibilizada pela própria API que implementa o protocolo, nela pode-se configurar as diretivas que definem o host a ser gerenciado, a porta ao qual esta ativo o agente SNMP, o intervalo de *polling* e as informações para autenticação dos usuários, caso utilize-se a versão 3 do protocolo.

Select User	User Name	Security Level	Auth Protocol	Priv Protocol	Target Host	Target Port
<input type="radio"/>						
<input type="radio"/>						
<input type="radio"/>						
<input type="radio"/>						
<input type="radio"/>						
<input type="radio"/>						

UserName **Security Level**

AuthPassword **PrivPassword**

AuthProtocol

HostName **Timeout**

Snmp Port **Snmp Version**

Retries

Poll Interval fo...

Figura 29. Tela de configuração do protocolo SNMP.

O protótipo possui no menu de gerenciamento localizado a esquerda, botões que representam algumas das diversas funcionalidades que um sistema de gerenciamento deve possuir, porém, com exceção a manipulação do protocolo SNMP as outras funcionalidades são meramente ilustrativas e não possuem funcionalidade alguma.

O uso da API para manipulação do protocolo é bastante simples, possuindo métodos com nomes intuitivos e uma documentação sobre a função de cada um. O trecho de código mostrado pela Figura 30 apresenta os objetos presentes na API que são utilizados pelo protótipo.

```

1 - private SnmpPoller poller = new SnmpPoller();
2 - private SnmpTarget target = new SnmpTarget();
3 - private MibTree mibTree = new MibTree();
4 - private LineGraphBean grafico = new LineGraphBean();

```

Figura 30. Objetos Utilizados pela API.

Na linha 1, o objeto *poller* é utilizado para definir o espaço de tempo entre uma requisição e outra ao SNMP para recuperação automática de determinado objeto da MIB. O objeto instanciado na linha 2 é utilizado para recuperar e alterar valores na MIB por meio do SNMP. A instância de *MibTree* criada na linha 3 é o objeto visual que permite a navegação pela árvore de objeto da MIB. Por fim na linha 4 a instância criada monta o gráfico.

O objeto *SnmpTarget* representa um servidor SNMP, portanto os parâmetros de configuração definidos na tela *Settings Dialog* podem ser passados aos métodos desse objeto, como mostra a Figura 31.

```

1 - target.setTargetPort(631);
2 - target.setTargetHost("localhost");
3 - target.setTimeout(1000);
4 - target.setSnmpversion(2);

```

Figura 31. Métodos do SnmpTarget.

A configuração a respeito do intervalo de *polling* pode ser passada ao objeto *poller* que é uma extensão de *SnmpTarget*, para tratar especificamente desta função. A Figura 32 mostra alguns de seus métodos.

```

1 - poller.setPollInterval(2000);
2 - poller.stopPolling();
3 - poller.restartPolling();
4 - poller.setObjectID("1.1.1.0.1.0.1.0");

```

Figura 32. Métodos do SnmpTarget.

No sub-menu *Carregar MIB*, o arquivo carregado pelo navegador de arquivos é adicionado a árvore de objetos por meio do método *addMib* (Figura 33).

```

1 - JFileChooser navegador = new JFileChooser();
2 - navegador.showDialog(mibTree, "open");
3 - mibTree.addMib(navegador.getSelectedFile().toString());

```

Figura 33. Carregar Arquivo MIB.

A Figura 34 apresenta outros métodos relevantes, utilizados para manipulação do protocolo de gerência.

```

1 - private void btnGetActionPerformed(java.awt.event.ActionEvent evt) {
2 -     target.setObjectID((mibTree.getSelectedMibNode().getNumberedOIDString().toString()+".0"));
3 -     txtResposta.setText(target.snmpGet());
4 - }

5 - private void btnGetNextActionPerformed(java.awt.event.ActionEvent evt) {
6 -     txtResposta.setText(target.snmpGetNext());
7 - }

8 - private void btnSetActionPerformed(java.awt.event.ActionEvent evt) {
9 -     target.setObjectID((mibTree.getSelectedMibNode().getNumberedOIDString().toString() + ".0"));
10 -     try {
11 -         txtResposta.setText(target.snmpSet(txtSetValue.getText());
12 -     } catch (Exception ex) {
13 -         txaRetornoMib.insert("Erro! Set SNMP: " + ex.getMessage(), txaRetornoMib.getText().length());
14 -     }
15 - }

```

Figura 34. Métodos Para Manipulação do SNMP.

O método *btnGetActionPerformed* é responsável pela ação executada pelo botão *GET*, na linha 2 o OID do objeto selecionado na árvore é recuperado e passado ao *target*, na linha 3 o método *snmpGet* implementa o *GetRequest* definido pelo protocolo. A implementação do *btnGetNextActionPerformed* na linha 6 traz o resultado da função *GetNextRequest* definida pelo protocolo. Por fim *btnSetActionPerformed* recupera o OID do objeto na árvore e altera o valor por meio da implementação do método *SetRequest* definido pelo SNMP.

5.5 RESULTADOS OBTIDOS

Com o processo de análise das plataformas, evidenciou-se a dificuldade no acesso a informações sobre esses aplicativos, foram encontradas poucas fontes que detalhassem os aspectos e as tecnologias utilizadas no desenvolvimento desses. Os aplicativos com licença

livre por sua vez, possuem documentação que trata principalmente da usabilidade das plataformas, o que justifica a descrição principalmente dos aspectos visuais e funcionalidades básicas.

A especificação apresentada foi estruturada para permitir a construção de uma plataforma robusta, que pode ser desenvolvida aos poucos, dada a independência de seus módulos. Referente ao processo de especificação, destaca-se a dificuldade em encontrar softwares que seguem uma modelagem para o processo de desenvolvimento, em muitos casos ela é feita erroneamente após a construção do sistema.

O estudo sobre o protocolo SNMP e o uso do mesmo no protótipo apresentado, deixaram claro os fatores que o caracterizam como o mais popular para o gerenciamento de redes; a simplicidade na configuração e manipulação de seus métodos, a imensa quantidade de recursos existentes que permitem interagir com inúmeras tecnologias, a descrição detalhada em documentos específicos sobre o protocolo, além de sua aplicação em todas as áreas funcionais de gerenciamento.

CONCLUSÃO

Os ambientes computacionais apresentam-se estruturalmente de diversas formas, e possuem um conjunto de variáveis que devem estar sintonizadas para um serviço de rede de qualidade. A manutenção, o monitoramento e a constante avaliação dessas variáveis são imprescindíveis para o sucesso dos objetivos finais.

As ferramentas de gerenciamento auxiliam os gestores a manter o bom funcionamento da rede, apresentando a oscilação das variáveis presentes e indicando os pontos que geram ou podem gerar problemas na mesma.

As plataformas tem como objetivo centralizar os dados que são oferecidos as ferramentas, mantendo a integridade dos resultados e permitindo a comunicação entre elas. Essas plataformas em sua grande maioria são *softwares* robustos que apresentam inúmeras funcionalidades, possuindo poucas soluções gratuitas e apresentando dificuldades na configuração.

As soluções para manutenção dos ambientes computacionais são geralmente baseadas em protocolos de gerenciamento, utilizados para auxiliar na recuperação e atualização de variáveis da rede.

O protocolo SNMP é atualmente o mais utilizado para este tipo de gerenciamento, sua estrutura é ampla, e sua manipulação para recuperação e atualização dos objetos da MIB acontece de maneira simples, como observado na implementação do protótipo.

A descrição das plataformas e a especificação desenvolvida, oferece a base para construção de um *software* robusto, que atenda as mais variadas necessidades de gerenciamento.

Desta forma trabalhos futuros podem ser realizados seguindo as diretrizes desta especificação, adicionando recursos funcionais à plataforma, como a aplicação de técnicas de

inteligência artificial, gerenciamento pró-ativo, utilização de protocolos de gerenciamento variados entre outros.

REFERÊNCIAS

BARCELAR, R. R.; MERIZIO, E. L.. **Ferramenta de Gerência de Redes OpenNms**. União de Escolas Superiores de Rondonópolis, UNIR, Rondonópolis, 2008.

BLACK T. L. **Comparação de Ferramentas de Gerenciamento de Redes**. Dissertação (Especialização) – Curso de Especialização em Tecnologias, Gerência e Segurança de Redes de Computadores, Instituto de Informática, UFRGS, 2008.

CARVALHO, F; et al. **FreeNMS: Uma Plataforma Software Livre para Gerência e Administração de Nível de Serviço (SLM)**. In: WORKSHOP RNP2, 4. 2003, Natal. **Anais....** Natal: 2003.

COMER, D. E.. **Redes de Computadores e Internet**: abrange transmissão de dados, ligação inter-redes e web. 2. ed. Porto Alegre: Bookman, 2001.

EMBRATEL (Brasil). **Arquitetura de Redes de Computadores OSI e TCP/IP**. 2. ed. Rio de Janeiro: Makron Books, 1997.

FARREL, A.. **A internet e seus protocolos**: uma análise comparativa. Rio de Janeiro: Elsevier, 2005.

FREITAS, C. A; MONTEIRO, J. W. A.. **As Vantagens do SNMP**. In: _____. Análise de protocolos na área de gerência de redes (SNMP/RMON). Goiânia: UFG, 2004.

GUSTAFSON, D. A.. **Teoria e Problemas de Engenharia de Software**. Porto Alegre: Bookman, 2003. 207 p.

PEREIRA, M. C.. **Administração e Gerência de Redes de Computadores**. 2001, 106 f. Área de concentração: Ciência e Tecnologia. Pós-graduação em Ciência da Computação - UFSC, Florianópolis - SC.

KOGA, I; et al. **MIB 2**: Mapeamento de objetos dos grupos IP e Transmission-Etherlike em critérios de gerência FCAPS. 2005. Dissertação (Mestrado) - Curso de Redes de Computadores, Departamento de Núcleo de Pesquisas Interdepartamental em Redes de Computadores (NUPERC), UNIFACS, Salvador, 2005.

LARMAN, G.. **Utilizando UML e Padrões**: uma introdução à análise e ao projeto orientados

a objetos. Porto Alegre: Bookman, 2000. 492 p.

MALIMA CONSULTORIA. **Ferramentas de Gerenciamento de Redes**. Disponível em: <http://www.malima.com.br/article_read.asp?id=49>. Acesso em: 05 fev. 2008.

MELO, E. T. L.; VIEIRA, E. M.; SARI, S. T.. **Implantação de um Sistema de Gerenciamento Distribuído**: Utilizando Tivoli. In: WORKSHOP RNP2, 2. 2000, Belo Horizonte - MG. Anais... Disponível em: <<http://www.rnp.br/wrnp2/2000/aceitos.html>>. Acesso em: 05 maio 2008.

MENDONÇA, F. D.. **SNMP**: Estrutura, Protocolo e Aplicações. Avaliado por Otto M. B. Duarte - Centro de Tecnologia - Departamento de Engenharia Eletrônica - UFRJ. Disponível em: <<http://www.gta.ufrj.br/grad/anteriores98/snmp-duboc/teleinfo.html>>. Acesso em: 01 fev. 2009.

McCLOGHRIE, K; et al. **Structure of Management Information Version 2 (SMIV2)**. STD 58, RFC 2578. 1999.

RNP (Brasil). **Introdução ao Gerenciamento de Redes TCP/IP**, v. 1, n. 3, ago. 1997. Disponível em: <<http://www.rnp.br/news/gen/9708/n3-2.html#ng-introducao>>. Acesso em: 05 maio 2008.

RODRIGUES, G C.. **vMIB**: Uma MIB genérica para gerenciamento de recursos virtuais. 2008. 104 f. Dissertação (Mestrado) - Curso de Ciência da Computação, PUCRS, Porto Alegre, 2008.

RUMBAUGH, J; JACOBSON, I; BOOCH G.. **UML**: guia do usuário. Rio de Janeiro: Ed. Campus, 2000. 472 p.

SANTOS JUNIOR, M. J. F. dos. **Novo Dicionário de Informática**. 2. ed. Goiânia: Terra, 2002.

SILVA, J. C. B.. **Monitoramento Remoto de Servidores com Plataforma Linux Utilizando o Protocolo de Gerenciamento SNMP**. 2004, 118f. Área de concentração: Ciência e Tecnologia. Graduação em Ciência da computação - UNESC, Criciúma – SC.

SOARES, L. F. G; LEMOS, G; COLCHER, S.. **Redes de Computadores**: das LANs MANs e WANs às Redes ATM. 2. ed. Rio de Janeiro: Campos, 1995.

SPECIALSKI, E.. **Gerência de Redes de Computadores e de Telecomunicações**. Florianópolis: 2002. Disponível em: <<http://notes.ufsc.br/aplic/beth.nsf>>. Acesso em: 28 ago. 2008.

STALLINGS, W.. **Redes e Sistemas de Comunicação de Dados**. Rio de Janeiro: Elsevier, 2005.

STALLINGS, W.. **SNMP, SNMPv2, SNMPv3 and RMON 1 AND 2**. 3. ed. Reading: Addison-Wesley, 1999.

SZTAJNBERG, A.. **Gerenciamento de Redes: Conceitos Básicos sobre os Protocolos SNMP e CMIP**. Disponível em: <<http://www.gta.ufrj.br/~alexsz/ger/snmpcmip.html>>. Acesso em: 10 set. 2008.

TANENBAUM, A. S.. **Redes de Computadores**. 4. ed. Rio de Janeiro: Campus, 2003.

TAROUCO, L. M. R.. **Redes de Computadores e suas Aplicações na Educação**. Disponível em: <<http://penta.ufrgs.br/>>. Acesso em: 01 ago. 2008.

TECHTARGET. **Search Bitpipe**. Disponível em: <<http://www.bitpipe.com/>>. Acesso em: 02 maio 2008.

TEIXEIRA JÚNIOR, J. H. et al. **Redes de Computadores: Serviços, Administração e Segurança**. São Paulo: Makron Books, 1999.

ZACKER, C; DOYLE, P.. **Redes de Computadores: Configuração, Manutenção e Expansão**. São Paulo: Makron Books, 2000.

ZENOSS. **Commercial Open Source Application, Systems & Network Monitoring**. Disponível em: <<http://www.zenoss.com/product/architecture>> Acesso em: 01 de jun. 2009.