Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Elétrica
Estágio Supervisionado 2

Proposta de Arquitetura de Gerência para a RedUnB usando a Plataforma Tivoli TME 10

Autores:
Alvaro Magalhães Neto – 93/04495
Trajano Passos Ferraz Moreira – 94/13481

Orientadores:
Ricardo Staciarini Puttini
Rafael Timóteo de Sousa Júnior

1º Semestre – 2000

Agradecimentos:

Agradeço aos meus pais por sempre me darem apoio incondicional em todas as horas, pelo incentivo e pelo amor dedicado a mim e a meus irmãos. Agradeço aos amigos que compartilharam comigo as alegrias e sofrimentos durante estes anos na Universidade; e agradeço a Deus, por tudo.

Alvaro

Agradeço a Deus pela vida, a meus pais pela força e amor, a meus irmãos pela amizade, a meus sobrinhos pela alegria.

Trajano

Resumo:

Neste projeto são estudados os componentes da arquitetura de gerência SNMP e é concebida e implementada uma estação de gerência para a Rede de Alta Velocidade da UnB (RedUnB) baseada na ferramenta Tivoli TME 10 e em agentes SNMP, RMOM e RMOM-2, MIB e MIB-II distribuídos pelos elementos da rede (*switches, hubs, routers e hosts*).

Inicialmente são apresentadas as teorias envolvidas para a realização deste projeto; é apresentado também o software Tivoli TME 10, assim como suas características e em que ele será útil ao projeto, em especial são tratados o Tivoli TME 10 *Framework* sobre o qual rodam os módulos, Tivoli NetView e Tivoli Remote Control.

Em seguida é apresentada uma proposta de gerência para a RedUnB tendo como base o software utilizado e as experiências realizadas durante o transcurso deste projeto.

Como o *Backbone* do Laboratório de Redes se assemelha ao *Backbone* da RedUnB, em uma escala menor, foi possível ter uma idéia bastante clara das necessidades de uma rede com o porte e a diversidade de ambientes existentes como o da RedUnB.

Finalmente são sugeridos alguns pacotes de gerência do Tivoli, cada um com várias disciplinas que realizam diferentes funções e que seriam de grande utilidade para que a gerência da RedUnB possa ser realizada de maneira eficiente.

Índice:

Índice de Figuras:

Índice de Tabelas:

Lista de Acrônimos:

- **AMO** Asset Management Option
- **CLI** Command Line Interface
- **GUI** Graphical User Interface
- **IP** Internet Protocol
- **ISO** International Organization for Standardization
- **LAN** Local Area Network
- **MIB** Management Information Base
- **OSI** Open Systems Interconnection
- **PDU** Protocol Data Unit
- **RCO** Remote Control Option
- **RFC** Request for Comment
- **RMO** Remote Monitoring Option
- **RMON** Remote Network Monitoring
- **RPC** Remote Procedure Call
- **SGMP** Simple Gateway Monitoring Protocol
- **SNMP** Simple Network Management Protocol
- **TCP** Transmision Control Protocol
- **TME** Tivoli Management Environment
- **TMR-S** Tivoli Management Regions – Server
- **UDP** User Datagram Protocol
- **WAN** Wide Area Network

1 - Introdução

A necessidade de gerência das redes de comunicação torna-se cada vez mais evidente, devido ao volume e à complexidade das informações que trafegam, agregados à constante expansão e utilização das redes, ao compartilhamento de dispositivos e à utilização de aplicações distribuídas.

Este trabalho tem por objetivo desenvolver um sistema de gerência descentralizado, primeiramente no âmbito do Laboratório de Redes da UnB para futuramente ser expandido à RedUnB, tendo em vista as necessidades da mesma. As principais motivações para a realização deste projeto são a necessidade de implementação de um ambiente de rede confiável, proporcionando um nível maior de segurança dos dados que trafegam pela rede, a diversidade física da rede, diversidade de sistemas operacionais existentes em tal ambiente, a necessidade de controle de acesso à rede e controle de tráfego de pacotes, visando tornar a rede mais eficiente, evitando gargalos.

Este trabalho dá uma visão geral em gerência de redes, e para isso são apresentados conceitos de gerência e noções dos vários protocolos utilizados para a sua implementação.

Além disso, também será apresentado o software de gerência TIVOLI em especial os módulos NetView e Tivoli Framework que são utilizados na parte prática deste trabalho.

Primeiramente serão apresentados alguns conceitos básicos sobre gerência de redes onde serão incluídas as modalidades de gerência, tais como gerência de falhas, gerência de configuração, gerência de segurança, gerência de contabilização e gerência de performance.

Mais especificamente são tratadas as disciplinas de gerência pertinentes ao Tivoli que são utilizadas nos experimentos constantes neste relatório, são elas: disciplina de monitoramento (*Distributed Monitoring*), disciplina de distribuição de software (*Software Distribution*) e disciplina de controle remoto (*Remote Control*).

Serão detalhados os agentes SNMP e RMON que são fundamentais para a realização dos experimentos realizados.

Neste relatório são mostrados também os experimentos que foram implementados com a utilização do software Tivoli e suas disciplinas.

No Capítulo 2 tem-se as definições e conceitos gerais necessários a um melhor entendimento do trabalho.

No Capítulo 3 são colocados conceitos dos protocolos utilizados (SNMP (v.1, v.2, v.3), MIB e MIB-II, RMON e RMON2), também é colocado um conceito de gerência sobre TCP/IP e em especial é apresentado como cada um dos grupos de gerência (Gerência de configuração, Gerência de desempenho, Gerência de falhas, Gerência de contabilidade e Gerência de segurança) utiliza a MIB-II para realizar suas tarefas.

No Capítulo 4 é mostrado o software Tivoli e suas disciplinas.

No Capítulo 5 são tratados a metodologia utilizada para a realização deste projeto, os resultados obtidos experimentalmente pelo grupo e as análises destes resultados.

No Capítulo 6 apresentamos a proposta de gerência da RedUnB, que é a base desse projeto, tal proposta é fundamentada nos estudos realizados durante a verificação deste projeto.

2 – Definições e Conceitos:

2.1 - Conceito de gerência de redes

Pode-se dizer que a gerência de redes consiste no controle e administração de forma racional dos recursos de hardware e software em um ambiente distribuído, buscando melhor performance e eficiência do sistema, assim como um controle de uma rede de computadores e seus serviços.

Essa é uma definição genérica, independente de padronizações usado para descrever o gerenciamento de redes em sistemas distribuídos.

O gerenciamento de redes provê mecanismos para monitoração, controle e coordenação de recursos em um ambiente OSI e define padrões de protocolos OSI para a troca de informações entre estes recursos. Tem por objetivo maximizar o controle organizacional das redes de computadores, de maneira eficiente e confiável, ou seja, planejar, supervisionar, monitorar e controlar qualquer atividade da rede. Algumas funções do gerenciamento de redes são apresentadas e detalhadas mais à frente neste relatório.

Existem também alguns níveis de gerência que podem ser implementados dependendo do que se deseja, por exemplo, Objetos gerenciados, Sistemas de gerência de elementos, Gerenciador de elementos, Interface como usuário.

2.1.1 - Agentes

Os agentes são entidades que fazem a interface com os dispositivos a serem gerenciados. Eles incluem sistemas finais que suportam aplicações de usuários bem como os nós que oferecem um serviço de comunicação, tais como processadores de *front-end*, controladores de *clusters, bridges* e roteadores.

2.1.2 - Gerentes

Os gerentes são agentes que possuem uma NMA (*network-managment application*). O NMA pode ser entendido como uma aplicação que inclui uma interface de operador para permitir a um usuário autorizado gerenciar a rede. Usa-se o termo gerente para o software que monitora e controla a rede ou estação.

2.1.3 - MIB

A base de informação gerencial (MIB - *Management Information Base*) é o nome conceitual para a informação de gerenciamento, incluindo os objetos gerenciados e seus atributos, operações e notificações. Pode-se também considerar as informações para a configuração do sistema como também pertencentes à MIB.

2.1.4 - *Polling* e Comunicação de Eventos

A informação que é útil para o monitoramento da rede é coletada e armazenada pelos agentes, e disponibilizada para um ou muitos sistemas de gerenciamento. Duas técnicas são usadas para disponibilizar a informação do agente, que servirá para o gerenciamento: *polling* e comunicação de eventos. *Polling* é uma interação de solicitações/respostas entre gerente e agente. O gerente pode questionar qualquer agente (para o qual ele tem autorização) e requisitar os valores de vários elementos de informação; os agentes respondem com informações de sua MIB.

2.1.5 - Objeto Gerenciado

Um objeto gerenciado representa um recurso sujeito ao gerenciamento, isto é, que pode ser gerenciado. Ele é definido em termos de seus atributos, das operações a que pode ser submetido, das notificações que pode emitir e de seus relacionamentos com outros objetos gerenciados. O conjunto de objetos gerenciados, juntamente com seus atributos, operações e notificações, constituem a MIB.

2.1.6 - Monitores Remotos

Monitores remotos são utilizados para a execução continua de diagnósticos e manutenção de arquivos de *log* em relação ao desempenho da rede, através da inspeção dos objetos gerenciáveis da base de informações gerenciais e conseqüente composição de estatísticas sobre os dados coletados. A monitoração remota ajuda, ainda, a minimizar o tráfego de gerenciamento, colaborando com a atividade de gerência à medida que não gera sobrecarga ao sistema.

2.2 - Modelos de Gerenciamento

Os modelos de gerenciamento diferenciam-se nos aspectos organizacionais no que se refere à disposição dos gerentes na rede, bem como no grau da distribuição das funções de gerência. Na prática existem dois modelos adotados para gerência de redes: o **Modelo Internet** e o **Modelo OSI**.

2.2.1 - Modelo Internet

O modelo de gerenciamento Internet adota uma abordagem gerente/agente onde os agentes mantêm informações sobre recursos e os gerentes requisitam essas informações aos agentes.

O padrão Internet SMI (*Structure of Management Information*) especifica uma metodologia para definição da informação de gerenciamento contida na MIB. O SMI usa um subconjunto de tipos de dados ASN.1. A MIB define os elementos de gerenciamento de informação como variáveis e tabelas de variáveis.

2.2.2 - Modelo OSI

O gerenciamento no modelo OSI - *Open Systems Interconnectíons* da ISO - *Internatíonal Organízatíon for Standardízatíon* baseia-se na teoria da orientação a objetos. Com isso, o sistema representa os recursos gerenciados através de entidades lógicas, as quais recebem a denominação de objetos gerenciados.

O modelo de gerência OSI permite a delegação das funções de monitoração aos agentes. Contudo, as funções de controle ainda ficam relegadas ao gerente, pois o conhecimento relativo à tomada de decisões gerenciais não se adapta para ser codificado em classes de objeto, ao contrário do conhecimento referente à monitoração, que é mais simples, geralmente estático e periódico.

Existem cinco áreas funcionais no gerenciamento num ambiente OSI:

- **Gerência de configuração**

A gerência de configuração (estado da rede) é composta, basicamente, pelas ferramentas que exercem controle, identificam, coletam dados, e provêem dados a objetos gerenciados, com o propósito de auxiliar a contínua operação dos serviços de comunicação.

Cabe ao gerenciamento de configuração (ou de nomes) informar quando houver um evento, e prover condições para que as pessoas dependentes daqueles recursos tenham informações suficientes para que os re-configurem com segurança.

- **Gerência de desempenho**

O objetivo da gerência de desempenho (performance ou vazão e taxa de erros) é ter uma gama de facilidades que permitam avaliar o comportamento de objetos gerenciados, bem como a efetividade e eficiência das atividades de interconexões.

As informações obtidas através das estatísticas de desempenho são de suma importância para os gerentes de rede, pois é através delas que se pode planejar, gerenciar e manter médias e grandes redes. Podendo-se, por exemplo, identificar com antecedência potenciais gargalos na rede e tomar ações preventivas para diminuir as chances de uma interrupção parcial ou total do serviço.

- **Gerência de falhas**

Pode-se definir o gerenciamento de falhas (comportamento anormal) como sendo o conjunto de facilidades que permitem a detecção, isolamento e correção de situações anormais em um ambiente de rede.

Devemos desatacar a diferença entre falha e erro; enquanto erro é apenas um evento, falhas são condições que requerem monitoramento e/ou ação por parte da gerência, a fim de sanar o problema.

Quando se dá uma falha, é necessário que, no menor período de tempo possível, sejam tomadas as seguintes atitudes:

Determinar exatamente onde a falha ocorreu;

Isolar a falha do restante da rede de forma que esta possa operar sem interferências;

Re-configurar ou modificar a rede, de forma a minimizar o impacto de operação do conjunto sem o componente em falha; e

Reparar ou trocar o componente defeituoso para que a rede volte a seu estado normal de operação.

- **Gerência de contabilidade**

A gerência de contabilidade (contabilização ou consumo de recursos) é constituída por ferramentas que permitem o estabelecimento de taxas para o uso de objetos gerenciados; assim como auxiliam a identificação de custos de uso dos mesmos.

Como razões para se efetivar tal acompanhamento pode-se incluir:

Usuários podem estar abusando de seus privilégios de acesso e, portanto, sobrecarregando a rede prejudicando o trabalho de outros;

Usuários podem estar utilizando a rede de forma ineficiente, assim o gerente desta pode auxiliar indicando formas e procedimentos para melhorar a performance e

O gerente de rede estará em melhor posição para planejar crescimentos na rede, se as informações de uso desta proverem detalhamento suficiente.

- **Gerência de segurança**

Características relacionadas com os aspectos de segurança de rede, essenciais para seu correto funcionamento e proteção dos objetos gerenciados. Aqui estão inclusos não só os aspectos de acesso a arquivos, mas também de políticas de senhas, monitoração de setores críticos e auditoria de registros (*logs*).

Em gerência de segurança (acesso) a manutenção de registros (*logs*) compõe uma importante ferramenta de trabalho. Pois estes são coleções de dados tomados ao longo de um intervalo de tempo, permitindo a análise de falhas e prevenção de incidentes.

Um dos aspectos a serem considerados no gerenciamento OSI é o fato de que tal modelo gera agentes mais complexos de serem desenvolvidos, consumindo mais recursos dos elementos de rede, enquanto economiza o uso da rede, devido a minimização dos pedidos de informações (*pollings*) necessários para obter dados sobre objetos gerenciados, livrando o gerente para tarefas mais inteligentes.

2.3 - Protocolos e Padrões de Gerenciamento

Os protocolos de gerenciamento de rede têm sido tradicionalmente implementados como protocolos do nível de aplicação. E até recentemente, cada vendedor costumava ter um método proprietário pelo qual seus agentes podiam se comunicar, o que levava a existência de incompatibilidades entre os diversos padrões.

A necessidade de uma representação padronizada foi sentida tanto pelo IAB (*Internet Activities Board*) quanto pela ISO. Enquanto a ISO trabalhou lentamente na especificação do seu padrão, o IAB saiu na frente com a proposta do SNMP em 1989, como uma solução temporária para gerenciamento de redes TCP/IP. A ISO só lançou seu padrão, chamado CMIP (*Common Management Information Protocol*), muito tempo depois. Devido à sua aceitação, o SNMP tornou-se um padrão de "facto" na indústria. Como conseqüência desse sucesso, o SNMPv2 (SNMP versão 2) foi proposto em 1993. Em 1996, foi proposto o SNMPv3 recentemente aprovado pela rfc2274.

2.3.1 - *Proxies*

O uso de SNMP requer que todos os agentes, bem como as estações de gerência, suportem UDP e IP, o que limita o gerenciamento direto de dispositivos e exclui outros, tais como *bridges* e modems, que não suportam nenhuma parte da pilha de protocolos do TCP/IP. Além disso, existem inúmeros pequenos sistemas (PCs, *workstations*, controladores programáveis) que implementam TCP/IP para suportar suas aplicações, mas para os quais não é desejável adicionar o SNMP.

Para acomodar dispositivos que não implementam SNMP, foi desenvolvido um conceito de *proxy*. Neste esquema, um agente SNMP atua como um *proxy* para um ou mais dispositivos, isto é o agente SNMP atua a favor dos dispositivos sob o *proxy*.

2.3.2 - CMIP

O CMIP é o protocolo para gerenciamento de redes definido pelo modelo OSI. O CMIP especifica os elementos de protocolo que são usados para prover os serviços de operação e notificação definidos pelo CMIS. É implementado num modelo orientado a objetos e baseado em eventos. Destina-se ao gerenciamento de diferentes níveis do modelo OSI, inclusive o de aplicações. Devido à sua complexidade, tem uso restrito.

2.4 - Comparação: SNMP versus CMIP

No gerenciamento OSI, objetos gerenciados são vistos como entidades sofisticadas com atributos, procedimentos associados e capacidades de notificação, e outras características complexas associadas com a tecnologia orientada a objetos. Para manter o SNMP simples, ele não foi projetado para trabalhar com tais conceitos sofisticados. Na verdade, os objetos no SNMP não são objetos propriamente ditos do ponto de vista da orientação a objetos; ao invés disso, objetos no SNMP são simplesmente variáveis com poucas características, tais como tipo de dados e permissões de leitura e/ou escrita.

Em relação à MIB, as duas arquiteturas adotaram a abordagem orientada a objetos para descrever e especificar as informações nela armazenadas. No caso Internet, são definidos os objetos a serem armazenados na MIB. A ISO, por sua vez, especifica algumas classes de objetos a serem empregadas pelos sistemas de gerenciamento e fornece um guia de definição dos objetos gerenciados.

A partir dos diversos aspectos apresentados sobre as arquiteturas de gerenciamento OSI e Internet, pode-se concluir que:

- As duas arquiteturas apresentam modelos de gerenciamento similares envolvendo elementos agentes e gerentes da rede, uma MIB e um protocolo de

aplicação responsável pelo transporte de operações e informações de gerenciamento entre tais elementos agentes e gerente.

- No caso da arquitetura Internet, o elemento agente é muito mais simples. A sua função básica é responder às operações de gerenciamento emitidas pelo gerente. Deve-se lembrar também dos *trap's* na arquitetura SNMP onde os agentes enviam mensagens aos gerentes. No caso OSI, o agente tanto responde às operações como também emite notificações quaisquer de gerenciamento. Tais operações são mais complexas do que as definidas para sistemas Internet. Vale salientar também que, tanto no OSI como no SNMPv2 um elemento de rede pode exercer os papéis de agente e de gerente simultaneamente, o que não acontece no SNMP.

- No que diz respeito aos protocolos de gerenciamento, o SNMP é um protocolo não orientado à conexão que normalmente utiliza os serviços prestados pelo UDP. O CMIP é um protocolo orientado à conexão executado sobre toda pilha de protocolos OSI de gerenciamento. Dentro deste contexto, pode-se afirmar que o sistema de gerenciamento OSI apresenta um nível de confiabilidade maior em relação ao da Internet. Contudo, em determinadas situações de falhas, a simplicidade do SNMP pode representar uma eficiência maior na solução do problema ocorrido.

3 - Conceitos de Protocolos Utilizados

A partir deste capítulo são expostos mais detalhadamente os conceitos básicos dos protocolos e softwares utilizados para a realização das experiências que serão descritas também com mais detalhes em capítulos posteriores.

3.1 - Gerência de Redes TCP/IP

A necessidade de gerenciamento de redes vem fazendo com que, cada vez mais, sejam pesquisadas e desenvolvidas novas técnicas e abordagens para este tipo de problema.

Aqui, são apresentados conceitos básicos sobre gerenciamento de redes TCP/IP os protocolos utilizados, tais como, o SNMP e o SNMPv2, além da sua nova versão o SNMPv3 e também a mais importante extensão da MIB-II, o RMON. E, finalmente, são mostradas as novas abordagens e tecnologias que estão surgindo para Gerenciamento de Redes TCP/IP, procurando sempre fazer um levantamento dos pontos que estão em aberto ou que ainda não estão completamente definidos.

A área de gerência de redes foi inicialmente impulsionada pela necessidade de monitoração e controle do universo de dispositivos que compõem as redes de comunicação.

Atualmente as redes de computadores e os seus recursos associados, além das aplicações distribuídas, têm se tornado fundamental e de tal importância para uma organização, que elas basicamente "não podem falhar". Isto significa que o nível de falhas e de degradação de desempenho considerados aceitáveis está cada vez mais diminuindo, sendo este nível igual até a zero, dependendo da importância da rede para uma instituição.

Com esta crescente necessidade de gerenciamento, fez-se necessário que padrões para ferramentas fossem estabelecidos. Em resposta a esta necessidade surgiram dois padrões:

- **Família de Protocolos SNMP:** o protocolo SNMP (*Simple Network Management Protocol*) refere-se a um conjunto de padrões para gerenciamento que inclui um protocolo, uma especificação de estrutura de dados, e um conjunto de objetos de dados. Este protocolo hoje já está na sua segunda versão oficial, chamada de SNMPv2. E já está em desenvolvimento o SNMPv3. Este é o protocolo de gerência adotado como padrão para redes TCP/IP, e que aqui será tratado.

- **Sistemas de gerenciamento OSI:** este termo refere-se a um grande conjunto de padrões de grande complexidade, que definem aplicações de propósito geral para gerência de redes, um serviço de gerenciamento e protocolo, uma especificação de estrutura de dados, e um conjunto de objetos de dados. Este conjunto de protocolos é conhecido como CMIP [ISO 1991] [Stallings 1993]. Pela sua complexidade, e pela lentidão do processo de padronização, este sistema de gerenciamento não é muito popular.

3.1.1 - Arquitetura de Gerenciamento

O modelo utilizado para gerenciamento de redes TCP/IP é composto pelos seguintes elementos:

- Estações de Gerenciamento;
- Agente de Gerenciamento;
- Base de Informações Gerenciais (MIB);
- Protocolo de Gerenciamento de Redes.

A estação de gerenciamento serve como interface para o gerente humano num sistema de gerenciamento de rede.

O agente de gerenciamento responde às solicitações de informações e de ações da estação de gerenciamento e deve também prover assincronamente informações importantes que não foram solicitadas por esta estação.

Os recursos a serem gerenciados são representados como objetos, e a coleção destes objetos é referenciada como Base de Informações Gerenciais (MIB).

A forma de comunicação entre a estação de gerenciamento e o agente é definido pelo protocolo de gerenciamento de rede, o SNMP.

3.2 - Simple Network Management Protocol (SNMP)

O conjunto de padrões SNMP [Stallings 1993] [Stevens 1994] é formado por:

- Um conjunto de especificações de estrutura e identificação para as informações gerenciais. Este padrão é chamado SMI (*Structure of Management Information*). Estas especificações são encontradas no RFC 1155 [Rose and McGloghie 1990].

- Um protocolo de comunicação entre o gerente e o agente, chamado simplesmente de SNMP (*Simple Network Management Protocol*). A especificação deste protocolo e apresentada no RFC 1157 [Case et al. 1990].

- Uma base de informações gerenciais que especifica quais variáveis são mantidas pelos elementos de rede. Esta base de informações é denominada MIB (*Management Information Base*), e a sua segunda versão, MIB-II, está especificada pelo RFC 1213 [McGloghie and Rose 1991].

Este conjunto de padrões pode ser utilizado para outros tipos de aplicações, além daquelas convencionais de gerência. Uma utilização possível, é o uso do SNMP para fazer o Balanceamento de carga em ambientes distribuídos [Uchoa 1995].

3.2.1 - O Protocolo SNMP

O protocolo SNMP (descrito nos RFCs 1155, 1157, 1212, 1213) foi projetado, em meados dos anos 80, como uma resposta aos problemas de comunicação entre diversos tipos de redes. A idéia básica por trás do SNMP era oferecer uma maneira

facilmente implementável e com baixo *overhead* para o gerenciamento de roteadores, servidores, *workstation* e outros recursos de redes heterogêneas. No momento de sua concepção, a meta era que ele fosse apenas uma solução provisória até que surgisse um melhor projeto de protocolo para gerência de redes. Entretanto, nenhuma solução melhor tornou-se disponível.

O SNMP é um protocolo do nível de aplicação da Arquitetura TCP/IP, operando tipicamente sobre o UDP (*User Datagram Protocol*), uma vez que quatro, de cinco mensagens SNMP, são do tipo pergunta-resposta. Ele é considerado "simples" porque os agentes requerem um software mínimo. Muito do poder de processamento de armazenamento de dados reside no sistema de gerenciamento, enquanto um subconjunto complementar dessas funções reside no sistema gerenciado.

Como conseqüência da exigência de simplicidade adotada no seu desenvolvimento, o SNMP acabou deixando de tratar algumas características, o que fez com que ele tivesse algumas deficiências (algumas destas deficiências são sanadas pelas versões 2 e 3 deste protocolo apresentadas a seguir neste relatório). Dentre essas características, destacam-se:

- Suporte para a transferência eficiente de grandes blocos de dados
- Estratégias de gerenciamento de rede centralizado
- Segurança

O protocolo SNMP define mensagens, unidades de dados chamadas PDU (*Protocol Data Unit*), para serem trocadas durante uma comunicação entre o gerente e o agente.

Os cinco tipos de mensagens SNMP são:

- *get-request-PDU*: mensagem enviada pelo gerente ao agente solicitando o valor de uma variável;

- *get-next-request-PDU*: mensagem utilizada pelo gerente para solicitar o valor da próxima variável depois de uma ou mais variáveis que foram especificadas;
- *set-request-PDU*: mensagem enviada pelo gerente ao agente para solicitar que seja alterado o valor de uma variável;
- *get-response-PDU*: mensagem enviada pelo agente ao gerente, informando o valor de uma variável que lhe foi solicitado;
- *trap-PDU*: mensagem enviada pelo agente ao gerente, informando um evento ocorrido.

Além de ter sido projetado para operar sob UDP, um protocolo não orientado à conexão, o próprio SNMP também é um protocolo não orientado à conexão, sendo cada troca de mensagens uma transação diferente entre o agente e a estação de gerenciamento.

Cada estação de gerenciamento, como também o agente, devem implementar os protocolos SNMP e, por conseqüência, UDP e IP para poderem se comunicar. Tal imposição exclui do processo de gerenciamento dispositivos que não suportam parte dos protocolos TCP/IP, ou que, apesar de implementarem o TCP/IP para suportar suas aplicações, não desejam adicionar mais carga ao seu sistema com o suporte ao protocolo SNMP.

Para que tais dispositivos também possam ser gerenciados, criou-se o conceito de agente proxy. Desta forma, este agente, que sabe se comunicar usando SNMP, responderia em nome do dispositivo que ele representa, e passaria o resultado da comunicação para o dispositivo de acordo com o protocolo que ele entende. Para tanto, este agente possui uma função de mapeamento que recebe as informações do dispositivo que ele representa e transforma tais informações em mensagens SNMP e vice-versa.

3.2.2 - Simple Network Management Protocol - SNMPv2

Apesar do alto índice de aceitação, a implementação de protocolos e aplicações SNMP apresentaram deficiências, principalmente, com relação à segurança e à

transferência eficiente de um grande número de informações do agente para o gerente. Além disso, o SNMP não se adequa ao gerenciamento de grandes redes de computadores, devido ao fato de apresentar limitações de desempenho para obtenção de requisições explícitas, e não dar suporte à comunicação gerente-gerente.

Apenas durante o ano de 1993, foram publicadas 11 RFCs definindo revisões para o SNMP e dando início ao padrão SNMPv2, sendo o primeiro, o RFC 1441 [Case et al 1993].

Esta série de revisões trouxe consigo grandes avanços que foram incorporados ao protocolo original. Tais avanços podem ser classificados de acordo com as seguintes categorias:

- Estrutura de informação;
- Primitivas de comunicação (PDUs);
- Comunicação gerente-gerente e gerenciamento hierárquico;
- Segurança.

A estrutura de informação de gerenciamento (SMIv2) para o SNMPv2 é mais elaborada, e eliminou ambigüidades nas definições dos objetos encontrados nas especificações anteriores.

Em relação às primitivas foram acrescentados dois novos PDUs:

- *get-bulk-request-PDU*, que permite que uma grande quantidade de informações possa ser transferida do agente para o gerente eficientemente;
- *inform-request-PDU*, que permite a um gerente enviar ou eventualmente solicitar informações a outro gerente.

A comunicação gerente-gerente, como também o gerenciamento hierárquico, foram incorporados ao protocolo com a introdução do novo tipo de mensagem, *inform-request;* e com a SNMPv2-M2M MIB, que é constituída por dois grupos: um grupo de alerta e um grupo de eventos.

No tocante a segurança, o SNMPv2 acrescentou ao protocolo novos conceitos e serviços que trouxeram mais segurança ao protocolo. Os conceitos incluídos foram: o

conceito de visão de MIB definido em termos de sub-árvores, restringindo o acesso a porções pré-definidas da MIB; e o conceito de contexto, que é uma coleção de objetos e seus respectivos agentes, e a especificação dos privilégios envolvidos. Os serviços incluídos são de integridade, autenticação e confiabilidade dos dados.

3.2.3 - Simple Network Management Protocol – SNMPv3

É uma versão do SNMP que apresenta uma proposta de solução para o problema de segurança encontrado nas versões anteriores do protocolo. As propriedades de segurança abordadas são:

- Autenticação

  Permite a um agente verificar se uma solicitação está vindo de um gerente autorizado e a integridade do seu conteúdo.

- Criptografia

  Permite gerentes e agentes a criptografarem mensagens para evitar invasão de terceiros

- Controle de Acesso

  Torna possível configurar agentes para oferecerem diferentes níveis de acesso a diferentes gerentes.

3.3 - Estrutura e Identificadores das Informações Gerenciais

A estrutura da MIB e a identificação dos objetos gerenciais são definidos no padrão chamado SMI (*Structure of Management Information*), encontrados no RFC 1155 [*Rose and McGloghie* 1990].

As definições são feitas utilizado-se um pequeno conjunto de características e elementos ASN.1.

3.3.1 - Tipos de Dados

SNMP utiliza apenas um pequeno conjunto de diferentes tipos de dados, classificados em tipos universais e tipos de aplicação.

3.3.1.1 - Tipos Universais

Estes são os tipos permitidos para definirem objetos da MIB. Eles são baseados na classe Universal.

- Integer (Universal 2);
- Octet String (Universal 4);
- Null (Universal 5);
- Object Identifier (Universal 6);
- Sequence, Sequence of (Universal 16);

3.3.1.2 - Tipos de Aplicação

- *DisplayString*: um string de 0 ou mais octetos. Cada variável deste tipo na MIB-II, não deve possuir mais de 255 caracteres;
- *IpAddres*s: este tipo é um *Octet String* de tamanho 4, um para cada octeto do número IP;
- *PhysAdress*: um *Octet String* que especifica o endereço físico;

- *Counter*: um inteiro não negativo, que só pode ser incrementado e não decrementado;

- *Gauge*: um inteiro não negativo, que pode ser tanto incrementado como decrementado;

- *TimeTicks*: um inteiro não negativo, que armazena o tempo em centenas de segundos, a partir de alguma época.

3.4 - Identificadores de Objetos

Um identificador de um objeto é um identificador único, que consiste numa seqüência de inteiros conhecidos como sub-identificadores.

A seqüência, lida da esquerda para a direita, define a localização deste objeto na estrutura de árvore da MIB.

3.5 - Estrutura da MIB

Associado a cada um dos objetos da MIB, está o seu identificador, que nomeia este objeto. Esta identificação segue uma estrutura hierárquica, cuja convenção também serve para identificar os tipos dos objetos.

O documento SMI, define quatro nodos abaixo do nodo Internet:

- *directory*: esta sub-árvore é reservada para o uso futuro do X.500l;
- *mgmt*: esta sub-árvore é utilizada para os objetos gerenciados definidos em documentos IAB aprovados;
- *experimental*: esta sub-árvore é utilizada para identificadores de objetos gerenciados usados em experiências na Internet;
- *private*: esta sub-árvore está reservada para a utilização de identificadores de objetos gerenciados definidos unilateralmente, por exemplo, por fabricantes que desejem ter seus próprios objetos.

3.6 - Base de Informações Gerenciais

A MIB é uma base de dados, cuja estrutura é especificada pelo padrão SMI, como já descrito anteriormente.

Ela pode ser caracterizada como um banco de dados ativo, o que possibilita que os valores das suas variáveis sejam, não só recuperados, como também alterados.

Cada agente deve manter sua própria instancia da MIB, relacionada com os objetos que estão sendo gerenciados sob o seu domínio. O RFC 1213, define um conjunto de variáveis utilizadas para a monitoração e o controle de redes TCP/IP.

Para cada novo dispositivo a ser gerenciado, que não tenha sido previsto o seu gerenciamento, é necessário que seja definido um conjunto de novas variáveis, estendendo assim, a MIB-II original.

3.7 - Remote Network Monitoring MIB (RMON)

O protocolo SNMP não é adequado para ambientes de redes corporativas e constituídas de diversas redes locais conectadas através de outra de longa distância. Esses enlaces de rede de longa distância, por operarem a taxas de transmissão inferiores às LANs que a interconectam, passam a ter grande parte da sua banda de transmissão ocupada para informações de gerenciamento. Uma solução encontrada para dirimir este problema foi o *Remote MONitoring* (RMON).

RMON é uma capacidade de gerenciamento remoto do SNMP. A especificação RMON é uma definição de uma MIB. Seu objetivo, contudo, é definir padrões de monitoração e interfaces para a comunicação entre agentes/gerentes SNMP.

RMON dá ao gerente da rede a habilidade para monitorar sub-redes como um todo, ao invés de apenas dispositivos individuais na sub-rede.

O protocolo RMON oferece suporte à implementação de um sistema de gerenciamento distribuído. Nele fica atribuída aos diferentes elementos, tais como

estações de trabalho, *hubs*, *switches* ou roteadores, das redes locais remotas a função de monitorar remotamente. Cada elemento RMON tem, então, como tarefas, coletar, analisar, tratar e filtrar informações de gerenciamento da rede e apenas notificar à estação gerente os eventos significativos e situações de erro.

No caso de existirem múltiplos gerentes, cada elemento RMON deve determinar quais informações de gerenciamento devem ser encaminhados para cada gerente.

Sendo assim, os objetivos do protocolo RMON são:

- Reduzir a quantidade de informações trocadas entre a rede local gerenciada e a estação gerente conectada a uma rede local remota.
- Possibilitar o gerenciamento contínuo de segmentos de redes locais, mesmo quando a comunicação entre o elemento RMON e a estação gerente estiver, temporariamente, interrompida.
- Permitir o gerenciamento pró-ativo da rede, diagnosticando e registrando eventos que possibilitem detectar o mau funcionamento e prever falhas que interrompam sua operação.
- Detectar, registrar e informar à estação gerente condições de erro e eventos significativos da rede.
- Enviar informações de gerenciamento para múltiplas estações gerentes, permitindo, no caso de situações críticas de operação da rede gerenciada, que a causa da falha ou mau funcionamento da rede possa ser diagnosticada a partir de mais de uma estação gerente.

Dois padrões básicos de protocolo RMON são especificados: RMON1 e RMON2, funcionalmente complementares.

3.7.1 - RMON1

O RMON1 opera somente na camada *Media Access Control* (MAC) oferecendo recursos ao administrador da rede para monitorar o tráfego e coletar informações e estatísticas da operação de um segmento de rede local, além de realizar o diagnóstico

remoto de falhas e erros ocorridos no segmento de rede a partir de funcionalidades de um analisador de protocolo suportadas pelo correspondente elemento RMON.

Porém, o fato do RMON1 só trabalhar na camada MAC, significa que este somente apresenta estatísticas para tráfego agregado – porém não apresenta estatísticas para camadas diferentes de várias pilhas de protocolos (ex. IP, FTP, IPX). Isto também significa que, por não ser capazes de monitorar a camada de rede, os dispositivos RMON1 não distinguem o tráfego originado através de um roteador, o que é uma grande deficiência.

3.7.2 - RMON2

O RMON2, por sua vez, opera no nível da camada de rede e camadas superiores, complementando portanto o RMON1, possibilitando coletar informações estatísticas e monitorar a comunicação fim-a-fim e o tráfego gerado por diferentes tipos de aplicação. A Figura 3.1 a seguir ilustra esta diferença.

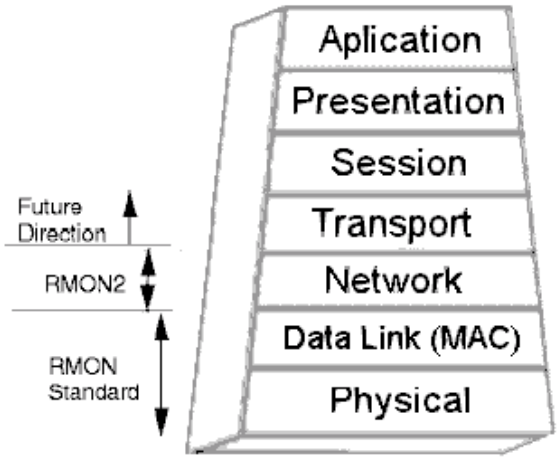


Figura 3.1 – Diferença entre RMON e RMON2

Com a utilização do padrão RMON original, um monitor RMON pode monitorar o tráfego de rede ao qual está conectado, mas não pode saber de onde está provindo originalmente este tráfego, nem tão pouco o destino final. Para tentar solucionar esta deficiência, foi criado um grupo de trabalho para desenvolver o padrão RMON2, gerando dois internet drafts: *Remote Network Monitoring MIB Version 2* [Waldbusser 1996] e *Remote Network Monitoring MIB Protocol Identifiers* [Bierman e Iddon 1996].

Um monitor RMON2 não está limitado a monitorar e decodificar o tráfego da camada de rede [Stallings 1996]. Ele também pode ver os protocolos de alto nível rodando acima da camada de rede, determinando, assim, que protocolos da camada de aplicação estão gerando este tráfego. Um gerente necessita, periodicamente, consultar os monitores para obter informações. Seria interessante, para efeitos de eficiência, que apenas os dados que foram alterados desde a última consulta fossem retornados. Para possibilitar tal facilidade, o RMON2 criou o conceito de filtragem de tempo *(time filtering*), introduzindo um *time stamp* em cada linha, que armazena a última vez em que esta foi alterada.

3.7.3 - Controle de Monitoramento Remoto

Com o objetivo de efetivamente gerenciar um monitor remoto, a RMON MIB contém características que possibilitam o controle efetivo da estação de gerenciamento. Estas características podem ser agrupadas em duas categorias:

- ***Configuração*:** tabelas de controle e tabelas de dados;
- ***Invocação de uma ação*:** um objeto utilizado para representar um comando, e uma invocação de uma ação pode ser solicitada modificando o valor de um objeto.

3.7.4 - Múltiplos Gerentes

Na arquitetura RMON, agentes podem estar sujeitos ao gerenciamento de muitas estações gerentes. No caso de um agente RMON ser compartilhado alguns problemas podem surgir, são eles:

- Requisições concorrentes;
- Captura de um recurso por um determinado gerente por um longo período de tempo;
- Recursos podem ser atribuídos para um gerente que deu pane antes de liberar o recurso.

Para tentar solucionar tais problemas uma combinação de requisitos devem ser utilizados, são eles:

- Inclusão de um campo de dono para cada linha da tabela de controle;
- Reconhecimento da não necessidade de utilização de um recurso;
- Possibilidade de negociação de um recurso entre gerentes;
- A possibilidade de um operador cancelar reservas de recursos;
- Durante uma reinicialização, um gerente pode liberar recursos.

3.7.5 - Gerenciamento de Tabelas

Um outro grande avanço introduzido com o padrão RMON, foi o gerenciamento de tabelas, atividade que não era muito clara no padrão SNMP original. A especificação RMON inclui convenções textuais e regras mais claras e disciplinadas para a adição e remoção de linhas.

3.8 - MIB

Neste novo capítulo entraremos em mais detalhes sobre as *Management Information Base* nas versões I e II, MIB I e MIB II.

O conhecimento das MIB's (Base de Informações Gerenciáveis), e principalmente, o conhecimento de como utilizar estas informações, são de fundamental importância na Gerência de Redes.

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

3.8.1 - Objeto gerenciado

Um objeto gerenciado é a visão abstrata de um recurso real do sistema. Assim, todos os recursos da rede que devem ser gerenciados são modelados, e as estruturas de dados resultantes são os objetos gerenciados. Os objetos gerenciados podem ter

permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso.

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

A árvore MIB II:

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

- um identificador de objetos (OID): seqüência de números separados por pontos.
- uma pequena descrição textual: descrição do nó rotulado

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

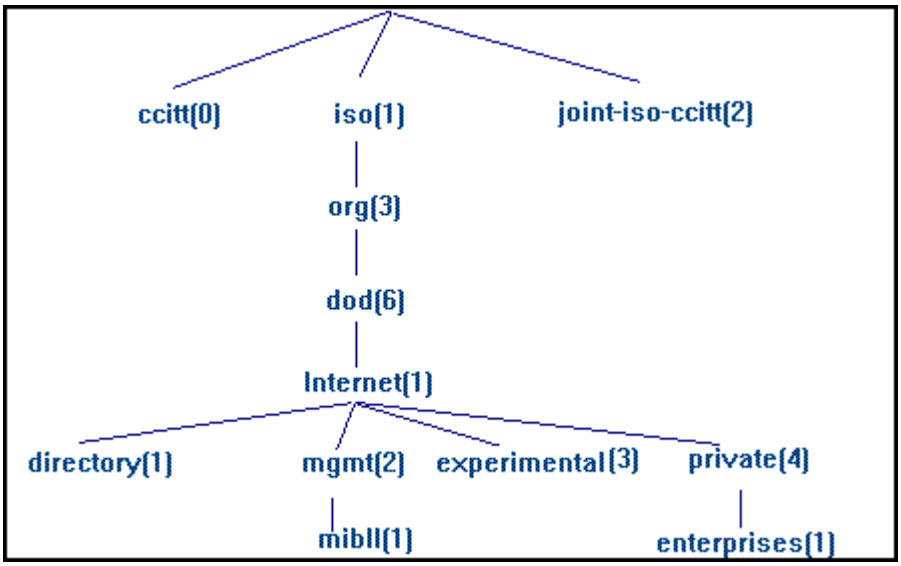


Figura 3.2 – Árvore MIB II

O nó raiz da árvore MIB não tem nome ou número, mas tem três sub-árvores:

1. **ccitt(0)**, administrada pelo CCITT
2. **iso(1)**, administrada pela ISO
3. **joint-iso-ccitt(2**), administrada pela ISO juntamente com o CCITT.

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

- **directory(1)**: contém informações sobre o serviço de diretórios OSI (X.500)
- **mgmt(2)**: contém informações de gerenciamento, é sob esta sub-árvore que está o nó da MIBII, com o identificador de objeto 1.3.6.1.2.1 ou { mgmt 1 }.
- **experimental(3)**: contém os objetos que ainda estão sendo pesquisados pela IAB.
- **private(4)**: contém objetos definidos por outras organizações.

Abaixo da sub-árvore MIBII estão os objetos usados para obter informações específicas dos dispositivos da rede. Esses objetos são divididos em 11 grupos, que são apresentados na tabela abaixo.

Grupos	Informações
system(1)	sistema de operação dos dispositivos da rede
interfaces(2)	interface da rede com o meio físico
Address translation(3)	mapeamento de endereços IP em endereços físicos
ip(4)	protocolo IP
icmp(5)	protocolo ICMP
tcp(6)	protocolo TCP
udp(7)	protocolo UDP
egp(8)	protocolo EGP
cmot(9)	protocolo CMOT
transmission(10)	meios de transmissão
snmp(11)	protocolo SNMP

Tabela 3.1 – Grupos de objetos usados pela MIB II

Cada objeto contido nos grupos apresentados na tabela acima é descrito no RFC1213. A descrição dos objetos é dividida em cinco partes: o nome do objeto, a sintaxe abstrata do objeto, a descrição textual do significado do objeto, o tipo de acesso

permitido ao objeto (*read-only*, *read-write*, *write-only* ou não acessível), e o estado do objeto (obrigatório, opcional, obsoleto).

3.8.2 - O Grupo SYSTEM

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

Objetos do Grupo *System* para Gerenciamento de Configuração

Objeto	Informação usada no gerenciamento de configuração
sysDescr	descrição do sistema
sysLocation	localização física do sistema
sysContact	pessoa responsável pelo sistema
sysName	nome do sistema

Tabela 3.2 - Objetos para Gerenciamento de Configuração (*System*)

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

Os objetos **sysLocation**, **sysContact** e **sysName** informam respectivamente , a localização física do sistema, a pessoa de contato em caso de problemas, e o nome do dispositivo da rede. Estas informações são úteis quando há necessidade de contactar com alguém para um acesso físico a um dispositivo remoto.

Objetos do Grupo *System* para Gerenciamento de Falhas

Objeto	Informação usada no gerenciamento de falhas
sysObjectID	fabricante do sistema
sysServices	qual camada de protocolo o sistema serve
sysUpTime	quanto tempo o sistema está operacional

Tabela 3.3 - Objetos para Gerenciamento de Falhas (*System*)

O identificador de objeto encontrado em **sysObjectID** informa o fabricante do sistema. Este dado é importante para resolver um problema de um dispositivo, quando necessita conhecer o seu fabricante.

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

O **sysUptime** informa a quanto tempo um sistema está funcionando. Através deste objeto, a aplicação de gerenciamento de falhas pode determinar se o sistema foi restaurado desde a última vez que o objeto foi consultado.

3.8.3 - O Grupo Interfaces

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

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

Objetos para Gerenciamento de Falhas

Objeto	Informação usada no gerenciamento de falhas
if AdminStatus	indica se a interface está administrativamente *up/down/test*
ifOper Status	indica o status operacional da interface (*up/down/test*)
if Last Change	indica quando a interface mudou seu estado operacional

Tabela 3.4 – Objetos para Gerenciamento de Falhas (Interfaces)

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

- _____
- **IfAdminStatus**
- ***Up(1)*** ***Down(2)*** ***Testing(3)***
- **ifOperStatus**
- ***Up(1)*** Operacional N/A N/A
- ***Down(2)*** Falha Down N/A
- ***Testing(3)*** N/A N/A em teste
- _____
- N/A - não aplicável.

O objeto **ifLastChange** informa quando a interface entrou no seu estado operacional atual.

Objetos para Gerência de Configuração

Objeto	Informação usada no gerenciamento de configuração
if Descr	nome da interface
IfType	tipo de interface
IfMtu	tamanho máximo do datagrama suportado pela interface
IfSpeed	largura de banda da interface
if AdminStatus	indica se a interface esta administrativamente *up/down/test*

Tabela 3.5 – Objetos para Gerência de Configuração (Interfaces)

- o objeto **ifDescr** nomeia a interface
- o objeto **ifType** atribui um tipo a interface
- o objeto **ifSpeed** é um medidor da velocidade da interface em bits por segundo. Útil quando deseja-se saber a velocidade atual de uma interface que aloca banda passante de acordo com a demanda de tráfego.

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

Objetos para Gerência de Performance

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

Tabela 3.6 – Objetos para Gerência de Performance (Interfaces)

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

porcentagem de erro de entrada = ifInErrors / (ifInUcastPkts + ifInNUcastPkts)

porcentagem de erro de saída = ifOutErrors / (ifOutUcastPkts + ifOutNUcastPkts)

- da mesma forma, com os objetos **ifInUcastPkts, ifOutUcastPkts, ifInNUcastPkts, ifOutNucastPkts, ifInDiscards, ifOutDiscards**, podemos calcular as porcentagens de descartes de entrada/saída.
- o objeto **ifInUnknownProtos** informa o número de descartes realizados devido ao recebimento de pacotes de protocolo desconhecido. Portanto não há a

detecção de nenhum problema, caso o valor dos objetos **ifInUnknownProtos** e **ifInDiscards** estiverem crescendo proporcionalmente.

- com os objetos **ifInOctets e ifOut Octets** pode-se calcular a taxa de utilização de uma interface.

  primeiro calcula-se o total de bytes recebidos e enviados em um intervalo de tempo entre *x* e *y*

  total de bytes = (ifInOctets_y - ifInOctets_x) + (ifOutOctets_y - ifOutOctets_x)

  depois calcula-se o total de bytes por segundo

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

  calcula-se o total de bits por segundo

  total de bits por segundo = total de bytes por segundo * 8

  finalmente tem-se a taxa de utilização

  taxa de utilização = (total de bits por segundo) / ifSpeed
- o objeto **ifOutQLen** indica se o dispositivo está tendo problemas em enviar dados para fora . Seu valor aumenta de acordo com o aumento do número de pacotes esperando para deixar a interface.
- os objetos **ifOutOctets e ifOutDiscards** juntos podem sinalizar um congestionamento na rede. Isto ocorre no caso de houver um aumento no valor do **ifOutDiscards** devido ao descarte de muitos pacotes que tentam deixar a interface, e uma diminuição do número total de bytes de saída, indicado pelo objeto **ifOutOctets**

  Objetos para Gerência de Contabilização

Objeto	Informação usada no gerenciamento de Contabilização
ifInOctets	taxa de bytes recebidos
ifOut Octets	taxa de bytes enviados
ifInUcastPkts	taxa de pacotes *unicast* recebidos
ifOutUcastPkts	taxa de pacotes *unicast* enviados
ifInNUcastPkts	taxa de pacotes *no-unicast* recebidos
ifOutNucastPkts	taxa de pacotes *no-unicast* enviados

Tabela 3.7 – Objetos para Gerência de Contabilização (Interfaces)

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

3.8.4 - O Grupo *Address Translation*

O grupo *Address Translation* não constitui mais um grupo separado. Seus objetos foram incorporados aos grupos de protocolos. Portanto o uso de informações de conversão de endereços no gerenciamento de redes serão analisados em cada um desses grupos.

3.8.5 - O Grupo IP

O IP é um protocolo de rede que utiliza um modo de serviço sem conexão para entregar datagramas. O grupo IP provê informações sobre o protocolo IP na entidade. Estas informações são subdivididas em quatro grupos:

1. objetos que informam erros e tipos dos pacotes IP vistos
2. tabela de informação sobre os endereços IP das entidades
3. tabela de roteamento IP da entidade
4. mapeamento de endereços IP para outros protocolos (substituindo o grupo Address Translation)

Os objetos do grupo IP podem ser aplicados ao gerenciamento de falhas, de configuração, de performance, e de contabilização.

Objetos para Gerenciamento de Falhas

Objeto	Informação usada no gerenciamento de Falhas
ipRoutTable	tabela de roteamento IP
ipNetToMediaTable	tabela de conversão de endereços IP

Tabela 3.8 – Objetos para Gerenciamento de Falhas (Translation)

Os objetos apresentados na tabela 3.8 são utilizados no gerenciamento de falhas. O ipRoutTable é uma tabela que possui as seguintes colunas:

- ipRouteDest: endereço IP do destino
- ipRouteIfIndex: número da interface
- ipRouteMetric1: métrica de roteamento #1
- ipRouteMetric2: métrica de roteamento #2
- ipRouteMetric3: métrica de roteamento #3
- ipRouteMetric4: métrica de roteamento #4
- ipRouteMetric5: métrica de roteamento #5
- ipRouteNextHop: próxima escala (endereço IP do roteador, usado para roteamento indireto)
- ipRouteType: tipo(direto, indireto, válido, inválido)
- ipRouteProto: mecanismo usado para determinar a rota
- ipRouteAge: idade da rota em segundos
- ipRouteMask: máscara da subrede para roteamento
- ipRouteInfo: ponteiro MIB para um protocolo de roteamento específico

Todos os objetos contidos no objeto **ipRouteTable** pode ser usado no gerenciamento de falhas. Por exemplo, eles podem ser usados para rastrear problemas de roteamento e de dispositivos que sinalizam informações de roteamento incorreto. Esses objetos permitem que a aplicação de gerenciamento de falhas produza a tabela de roteamento IP para um dispositivo e descubra rotas através da rede. Além disso os objetos **ipRouteType** e **IpRouteProto** informam como a informação de roteamento foi aprendida.

Assim como o **ipRouteTable**, o objeto **IpNetToMediaTable** é uma tabela com as seguintes entradas:

- ipNetToMediaIfIndex: número da interface
- ipNetToMediaPhysAddress: endereço do meio do mapeamento
- ipNetToMediaNetAddress: endereço IP do mapeamento
- ipNetToMediaType: como o mapeamento foi determinado (outro, inválido, dinâmico, estático)

Os objetos contidos no objeto **IpNetToMediaTable** informam o mapeamento de endereços IP para endereços em outros protocolos.

Objetos para Gerenciamento de Configuração

Objeto	Informação usada no gerenciamento de Configuração
IpFowarding	se o dispositivo é setado para IP
IpAddrTable	endereços IP do dispositivo
IpRouteTable	tabela de roteamento IP

Tabela 3.9 – Objetos para Gerenciamento de Configuração (Translation)

Os objetos da tabela 3.9 são utilizados no gerenciamento de configuração.

- o objeto **ipForwarding** indica se a entidade está configurada para enviar (forward) datagramas IP.
- o objeto **IpAddrTable** é composto dos seguintes objetos:
  - ipAdEntAddr: o endereço IP desta entrada
  - ipAdEntIfIndex: número da interface
  - ipAdEntNetMask: máscara de sub-rede para endereços IP
  - ipAdEntBcastAddr: o bit menos significativo do endereço IP de broadcast

  Estes dados são definidos na MIB somente para leitura (*read-only*), portanto a aplicação de gerenciamento de configuração poderá apenas consultá-los e não alterá-los.
- já o objeto **ipRouteTable** possui vários objetos definidos com permissão de leitura e escrita (*read-write*). Dessa forma, uma aplicação de gerenciamento de configuração pode entrar com novas rotas através do objeto **ipRouteDest** e mudar o tipo de uma rota com o objeto **ipRouteType**. Além disso é possível configurar-se as métricas de roteamento através dos objetos **ipRouteMetric1**, **ipRouteMetric2**, **ipRouteMetric3**, **ipRouteMetric4** e **ipRouteMetric5**.

Objetos para Gerenciamento de Performance

Objeto	Informação usada no gerenciamento de Performance
IpInReceives	taxa de datagramas de recebidos
IpInHdrErrors	taxa de erros de cabeçalho de entrada
IpInAddrErrors	taxa de erros de endereço de entrada
IpForwDatagrams	taxa de datagramas repassados
IpInUnknownProtos	taxa de datagramas de entrada para um protocolo desconhecido
IpInDiscards	taxa de datagramas de entrada descartados
IpInDelivers	taxa de datagramas de entrada entregues com sucesso
IpOutRequests	taxa de datagramas de saída (não inclui os datagramas repassados)
IpOutDiscards	taxa de datagramas de saída descartados
ipOutNoRoutes	taxa de descartes ocorridos por falta de informação de roteamento
ipRoutingDiscards	taxa de entradas de roteamento descartadas
ipReasmReqds	taxa de datagramas recebidos necessitando de remontagem
ipReasmOKs	taxa de datagramas com sucesso na remontagem
ipReasmFails	taxa de datagramas com falhas na remontagem
ipFragOKs	taxa de datagramas com sucesso na fragmentação
ipFragFails	taxa de datagramas com insucesso na fragmentação
ipFragCreates	taxa de fragmentos gerados

Tabela 3.10 – Objetos para Gerenciamento de Performance (Translation)

Os objetos listados na tabela 3.10 podem ser usados no gerenciamento de performance. A seguir serão apresentadas algumas das formas que estes objetos podem ser utilizados para auxiliar no gerenciamento de performance.

- os objetos **ipInReceives e ipOutRequest** juntamente com alguns objetos do grupo Interface permitem o cálculo da taxa de tráfego IP de entrada e saída de uma entidade:
  porcentagem de tráfego ip de entrada: (ifInUcastPkts + ifInNUcastPkts) / ipInReceives

porcentagem de tráfego ip de saída: (ifOutUcastPkkts + ifOutNUcastPkts) / ipOutRequest

- pode-se calcular a porcentagem de erros de datagramas IP:
  porcentagem de erros de entrada:
  (ipInDiscards+ipInHdrErrors+ipInAddrErrors)/ipInReceives
  porcentagem de erros de saída:
  (ipOutDiscards+ipOutHdrErrors+ipOutAddrErrors)/ipOutRequests
- o aumento do valor do objeto **ipRoutingDiscards** pode significar que um dispositivo está descartando entradas válidas devido à falta de recursos.

- um freqüente aumento no valor do objeto **ipInUnknownProtos** pode causar problemas de performance, pois os recursos estarão sendo desperdiçados em checagens de erros e determinação do destino, sendo que por fim o datagrama será descartado por ser destinado a um protocolo da camada superior desconhecido.

- pode-se calcular a taxa de *forwarding* e de recepção de datagramas IP por um dispositivo em um intervalo tempo entre *x* e *y* em segundos.
  taxa de forwarding = (ipForwDatagrams_y - ipForwDatagrams_x) / (y-x)
  taxa de entrada = (ipInReceives_y - ipInReceives_x) / (y - x)

Tendo estas duas taxas pode-se determinar se o sistema está repassando os datagramas IP rápido o bastante para satisfazer os requerimentos da rede. Se os pacotes recebidos estão sendo repassados a outros sistemas, a taxa de entrada e taxa de forwarding devem ser iguais. No entanto para que o cálculo seja mais exato deve-se subtrair da taxa de entrada a taxa de erros e pacotes IP destinados ao próprio sistema.

Objetos para Gerenciamento de Contabilização

Objeto	Informação usada no gerenciamento de Performance
ipOutRequests	numero de pacotes IP enviados
ipInReceives	numero de pacotes IP recebidos
ipInDelivers	numero de pacotes IP de entrada entregues com sucesso

Tabela 3.11 – Objetos para Gerenciamento de Contabilização (Translation)

A tabela 3.11 contém os objetos do grupo IP usados para auxiliar no gerenciamento de contabilização

- os objetos **ipOutRequests** e **ipInReceives** informam o número total de pacotes IP enviados e recebidos por uma entidade.
- o objeto **ipInDelivers** informa o número de pacotes IP entregues com sucesso a protocolos de camada superior e aplicações.

3.8.6 - O Grupo ICMP

O ICMP é um protocolo que carrega mensagens de erro e controle para dispositivos IP. O grupo ICMP contém objetos que fornecem informações sobre o protocolo ICMP na entidade em questão. Todos os seus objetos são aplicados ao gerenciamento de performance.

Os objetos do grupo ICMP são apresentados na tabela abaixo:

Objeto	Informação usada para gerenciamento de performance
icmpInMsgs	taxa de recebimento de mensagens
icmpInErrors	taxa de erros de entrada
icmpInDestUnreachs	taxa de mensagens de Destino não Alcançado recebidas
icmpInTimeExcds	taxa de mensagens de Tempo Excedido recebidas
icmpInParmProbs	taxa de mensagens de Problemas com Parâmetros recebidas
icmpInSrcQuenchs	taxa de mensagens de Fonte Apagada recebidas
icmpInRedirects	taxa de mensagens de Redirecionamento recebidas
icmpInEchos	taxa de mensagens de Ecos (requisição)recebidas
icmpInEchoReps	taxa de mensagens de Respostas a Ecos recebidas
icmpInTimestamps	taxa de mensagens de icmpInTimestampReps
icmpInAddrMasks	taxa de mensagens de Requisição de Máscaras de Endereços recebidas
icmpInAddrMaskReps	taxa de mensagens de Resposta a Máscaras de Endereços recebidas
icmpOutMsgs	taxa de saída de mensagens
icmpOutErrors	taxa de erros de saída
icmpOutDestUnreachs	taxa de mensagens de Destino não Alcançado enviadas
icmpOutTimeExcds	taxa de mensagens de Tempo Excedido enviadas
icmpOutParmProbs	taxa de mensagens de Problema com Parâmetros enviadas
icmpOutSrcQuenchs	taxa de mensagens de Origem Apagada enviadas
icmpOutRedirects	taxa de mensagens de Redirecionamento enviadas
icmpOutEchos	taxa de mensagens de Eco(requisição)enviadas
icmpOutEchoReps	taxa de mensagens Respostas de Eco enviadas
icmpOutTimestamps	taxa de mensagens de requisição de Timestamp enviadas
icmpOutTimestampReps	taxa de mensagens de respostas ao pedido de Timestamp enviadas
icmpOutAddrMasks	taxa de mensagens de Requisição de Máscara de Endereços enviadas
icmpOutAddrMaskReps	taxa de mensagens de Resposta de Máscara de Endereços enviadas

Tabela 3.12 – Objetos do Grupo ICMP

3.8.7 - O Grupo TCP

O TCP é um protocolo de transporte que provê conexões confiáveis entre aplicações. Muitas implementações do TCP incluem recursos adicionais para lidar com controle de fluxo, congestionamento da rede, e a retransmissão de segmentos perdidos. O grupo TCP pode ajudar no gerenciamento de configuração, performance, de contabilização, e de segurança.

Este grupo é subdividido em dois grupos:

1. objetos gerais sobre o TCP no sistema
2. uma tabela de valores para cada conexão TCP corrente, a qual é alterada a cada começo e fim de uma conexão TCP.

Objetos para Gerenciamento de Configuração

Objeto	Informação usada para gerenciamento de Configuração
tcpRtoAlgorithm	algoritmo utilizado para determinar o "*time out*" de retransmissão de octetos TCP não confirmados
tcpRtoMin	valor mínimo permitido para o "*time-out*" de retransmissão TCP, em milissegundos
tcpRtoMax	valor máximo permitido para o "*time-out*" de retransmissão TCP, em milissegundos
tcpMaxConn	limite de conexões que podem ser abertas pela entidade de transporte do dispositivo
tcpCurrEstab	número de conexões de transporte corretamente abertas

Tabela 3.13 – Objetos para Gerenciamento de Configuração (TCP)

- o objeto **tcpRtoAlgorithm** permite a configuração do algoritmo de retransmissão de octetos TCP não confirmados. A configuração inadequada pode resultar em congestionamento na rede ou distribuição injusta de banda passante. Consultando-se freqüentemente os objetos **tcpRtoMin, tcpRtoMax** e

tcpRtoAlgorithm pode-se verificar se a configuração está adequada ao ambiente de seu sistema de rede.

- o objeto **tcpMaxConn** ajuda a configurar a rede para suportar o número de conexões TCP remotas necessárias. Este número pode ser calculado observando-se o objeto **tcpCurrEstab** que informa o número de conexões TCP estabelecidas no momento.

Objetos para Gerenciamento de Performance

Objeto	Informação usada para gerenciamento de Performance
tcpAttempt Fails	numero de tentativas de conexão falhadas
tcp EstsabResets	numero de reinicializacões de conexões estabelecidas
tcpRetransSegs	numero de segmentos retransmitidos
tcpInErrs	numero de pacotes recebidos com erro
tcpOutRsts	numero de vezes que a entidade tentou reinicializar uma conexão
tcpInSegs	taxa de segmentos TCP recebidos
tcpOutSegs	taxa de segmentos TCP enviados

Tabela 3.14 – Objetos para Gerenciamento de Performance (TCP)

- pela observação do objeto **tcpAttemptFails** pode-se medir a confiabilidade da rede, onde um número menor de falhas indicam uma rede mais confiável.
- pela observação do objeto **tcpEstabResets** também pode-se medir a confiabilidade da rede, sendo que, quanto maior o número de conexões estabelecidas reinicializadas, menos confiável é a rede.
- o objeto **tcpRetransSegs** informa o número de segmentos TCP que o sistema está retransmitindo, esta informação pode indicar se uma entidade está tendo que fazer várias retransmissões para garantir a confiabilidade.
- o objeto **tcpInErrs** indica o número de segmentos recebidos com erro. O aumento deste objeto pode ser causado pelo encapsulamento incorreto dos segmentos pelo sistema de origem, alguma rede repassando os segmentos com erro, ou outras razões.

- o objeto **tcpOutRsts** informa o número de vezes que a entidade tentou reinicializar uma conexão..
- os objetos **tcpInSegs** e **tcpOutSegs** permitem a checagem da taxa de segmentos TCP que entram e saem da entidade.

Objetos para Gerenciamento de Contabilização

Objeto	Informação usada para gerenciamento de Contabilização
tcpActiveOpens	numero de vezes que o sistema abriu uma conexão
tcpPassiveOpens	numero de vezes que o sistema recebeu um pedido de abertura de conexão
tcpInSegs	numero total de segmentos TCP recebidos
tcpOutSegs	numero total de segmentos TCP emitidos
tcpConnTable	tabela das conexões TCP correntes

Tabela 3.15 – Objetos para Gerenciamento de Contabilização (TCP)

- os objetos **tcpActiveOpens** e **tcpPassiveOpens** informa o número total de vezes em que uma conexão foi feita de, ou para um sistema, respectivamente.
- os objetos **tcpInSegs** e **tcpOutSegs** juntos contam os segmentos TCP que entram e saem da entidade, respectivamente.
- o objeto **tcpConnTable** é uma tabela com as atuais conexões TCP, e contém os seguintes campos:
    - tcpConnState: estado da conexão
    - tcpConnLocalAddress: endereço TCP local
    - tcpConnLocalPort: endereço IP local
    - tcpConnRemAddress: endereço TCP remoto
    - tcpConnRemPort: endereço IP remoto

O campo **tcpConRemAddress** determina o endereço do sistema remoto que está conectado à entidade. A consulta freqüente a este campo permite o conhecimento de quais sistemas usam os recursos da rede e durante quanto tempo.

Muitas aplicações TCP usam portas bem definidas, tornando possível determinar-se quais aplicações estão fazendo ou recebendo conexões TCP.

Objetos para Gerenciamento de Segurança

As informações da tabela **tcpConnTable** também podem ser usados para gerenciamento de segurança, pois permite o conhecimento dos sistemas que acessam recursos via TCP. O tempo de *polling* influenciará grandemente na eficiência do gerenciamento, pois um intruso pode levar apenas alguns segundos para pegar as informações que deseja e fechar a conexão. Se nenhum *poll* for feito neste intervalo, o intruso não será detectado.

3.8.8 - O Grupo UDP

O UDP é um protocolo de transporte que, ao contrário do TCP, não garante segurança e nem estabelece conexões, ao invés disso ele usa um fluxo de datagramas para transportar as informações. O grupo UDP possui um número limitado de objetos. O grupo TCP pode ajudar no gerenciamento de performance, de contabilização, e de configuração.

Este grupo é subdividido em dois grupos:

1. objetos gerais sobre o UDP nesta entidade
2. entradas sobre as aplicações UDP, que estão recebendo datagramas correntemente, na entidade em questão

Objetos para Gerenciamento de Performance

Objeto	Informação usada para gerenciamento de Performance
udpInDatagrams	taxa de datagramas recebidos
udpOutDatagrams	taxa de datagramas enviados
UdpNoPorts	taxa de datagramas que não foram enviados para uma porta valida
UdpInErrors	taxa de datagramas UDP recebidos com erro

Tabela 3.16 – Objetos para Gerenciamento de Performance (UDP)

- a consulta periódica aos objetos **udpInDatagrams** e **udpOutDatagrams** pode determinar a taxa de entrada e saída de datagramas.

- o objeto **udpNoPorts** informa quando a entidade está recebendo datagramas de uma aplicação inválida. Uma taxa alta desses datagramas pode resultar em problemas de performance.

Objetos para Gerenciamento de Contabilização

Objeto	Informação usada para gerenciamento de Performance
udpInDatagrams	numero total de datagramas UDP recebidos
udpOutDatagrams	numero total de datagramas UDP enviados
UdpTable	portas UDP recebendo datagramas correntemente

Tabela 3.17 – Objetos para Gerenciamento de Contabilização (UDP)

- os objetos **udpInDatagrams** e **udpOutDatagrams** determinam quantos datagramas UDP foram recebidos e enviados pela entidade.
- o objeto **udpTable** é composto pelos seguintes campos:
    - udpLocalAddress: endereço IP local
    - udpLocalPort: porta UDP local

Como o UDP não é um protocolo baseado em conexão, as entradas da tabela acima são válidas somente para o período em que a aplicação escuta uma porta.

Objetos para Gerenciamento de Configuração:

Checando o objeto **udpTable**, pode-se determinar se as aplicações da entidade estão setadas corretamente. Por exemplo, se é conhecido que uma entidade tem uma aplicação que oferece impressão remota em uma determinada porta, esta configuração pode ser facilmente verificada usando o objeto **udpTable**.

Objetos para Gerenciamento de Segurança

O objeto **udpTable** também pode ser usado para gerenciamento de segurança. Pode-se checar este objeto para assegurar que uma entidade não executou uma determinada aplicação. Por exemplo, se determinarmos que uma determinada aplicação escuta requisições por uma porta UDP específica. A ferramenta de gerenciamento pode

checar o objeto **udpTable** de todos os sistemas para verificar se esta porta UDP local está sendo escutada.

3.8.9 - O Grupo EGP

O EGP é um protocolo que informa a um dispositivo de rede IP como alcançar outras redes IP. Ele não informa a rota completa para a outra rede, mas ela permite que um dispositivo saiba em que direção que a rede existe. Redes IP podem ser agrupadas em áreas lógicas chamadas *sistemas autônomos*. Um *sistema autônomo* geralmente é uma rede e suas subredes associadas ou uma coleção de redes e subredes sob uma mesma administração. Dois dispositivos de rede em dois *sistemas autônomos* distintos podem compartilhar informações de alcançabilidade via EGP.

Os dispositivos de rede que comunicam com o EGP entre *sistemas autônomos* são chamados *vizinhos EGP*. Cada processo EGP tem uma relação um-para-um com cada vizinho. Cada vizinho EGP conversa um protocolo *hello* que periodicamente informa outros vizinhos que ele ainda está ativo. Quando o sistema consulta a informação de alcançabilidade do vizinho, ele está fazendo um *EGP poll*.

Os **objetos do grupo EGP** são subdivididos em dois grupos:

1. informações sobre o EGP nesta entidade.
2. uma tabela de entradas contendo informações sobre cada vizinho EGP.

Os objetos do grupo EGP podem ser aplicados ao gerenciamento de falhas, de configurações, de performance e de contabilização.

Objetos para Gerenciamento de Falhas

Objeto	Informação usada para gerenciamento de Falhas
EgpNeighState	estado de cada vizinho EGP
egpNeighStateUps	número de vezes que um vizinho EGP entrou no estado UP
egpNeighStateDows	número de vezes que um vizinho EGP entrou no estado DOWN

Tabela 3.18 – Objetos para Gerenciamento de Falhas (EGP)

Os objetos listados acima estão contidos na tabela de vizinhos EGP (objeto **egpNeighTable**).

O estado de um vizinho EGP pode prover informações de como a informação de roteamento é injetada no sistema autônomo. A aplicação de gerenciamento de falhas pode usar o objeto **egpNeighState** para conhecer o estado atual de um vizinho EGP. Se o vizinho EGP está no estado **up**, então ele deverá estar enviando informações sobre alcançabilidade de redes ao processo EGP local.

Sabendo-se quando um vizinho entra no estado up pode sinalizar sobre novas informações de roteamento que devem entrar no sistema autônomo. Já saber quando o vizinho pára a comunicação, e entra no estado **down** pode ser útil na resolução de problemas de roteamento.

Objetos para Gerenciamento de Configuração

Objeto	Informação usada para gerenciamento de Configuração
egpNeighState	estado de cada vizinho EGP
egpNeighAddr	endereço IP do vizinho EGP
EgpNeighAs	sistema autônomo do vizinho EGP
egpNeighIntervalHello	intervalo entre retransmissões de comandos *Hello*
egpNeighIntervalPoll	intervalo entre retransmissões de comandos *Poll*
egpNighMode	modo de *polling* desta entidade EGP
egpNeighEventTrigger	permite iniciar ou finalizar uma comunicação
EgpAs	sistema autônomo local

Tabela 3.19 – Objetos para Gerenciamento de Configuração (EGP)

- o objeto **egpAs** informa o número do sistema autônomo da entidade EGP local.
- os demais objetos estão contidos no objeto **egpNeighTable** e dizem respeito à configuração de um vizinho específico .
- o objeto **egpNeighEventTrigger** pode ser usado para iniciar e encerrar uma comunicação com um vizinho EGP (já existente). Este objeto permite o controle

do processo EGP do sistema. Este é o único objeto do grupo EGP que pode ser setado pelo engenheiro de rede através de um comando SNMP *Set-Request*.

Objetos para Gerenciamento de Performance

Objeto	Informação usada para gerenciamento de Performance
EgpInMsgs	taxa de mensagens recebidas
EgpInErrors	taxa de mensagens recebidas com erro
EgpOutMsgs	taxa de mensagens enviadas
egpOutErrors	taxa de mensagens não enviadas devido a ocorrência de erros
egpNeighInMsgs	taxa de mensagens recebidas deste vizinho EGP
egpNeighInErrs	taxa de mensagens recebidas com erro deste vizinho EGP
egpNeighOutMsgs	taxa de mensagens enviadas a este vizinho EGP
egpNeighOutErrs	taxa de mensagens não enviadas a este vizinho EGP devido a erros
egpNeighInErrMsgs	taxa de mensagens de erro recebidas deste vizinho EGP
egpNeighOutErrMsg	taxa de mensagens de erro enviadas a este vizinho EGP

Tabela 3.20 – Objetos para Gerenciamento de Performance (EGP)

- os objetos **egpInMessages** e **egpOutMessages** permitem calcular a taxa de mensagens EGP que entram e saem da entidade. Geralmente esta taxa será insignificante, mas em alguns momentos de instabilidade da rede entre os vizinhos EGP, essa taxa pode aumentar e influenciar na performance da entidade.
- o aumento do valor dos objetos **egpInErrors** e **egpOutErrors** geralmente coincide com o aumento do número de mensagens recebidas e enviadas pela entidade. Se uma mensagem é recebida com erro e uma resposta válida não é enviada, o vizinho EGP originador deverá retransmitir a mensagem. Quando a entidade não pode enviar mensagens EGP válidas devido a limitações de recursos, o valor do objeto **egpOutErros** irá aumentar. Consequentemente, quando a taxa de **egpInMessages** aproxima-se da taxa de **egpOutErrors**, a

entidade provavelmente estará tendo dificuldades em construir e enviar mensagens EGP.

- da mesma forma usando os objetos **egpNeighInMsgs, egpNeighInErrs, egpNeighOutMsgs e egpNeighOutErrs** permite o cálculo da taxa de entrada e saída de mensagens e erros de cada vizinho.

3.8.10 - O Grupo CMOT

O Grupo CMOT existe somente por razões históricas. O CMOT é um protocolo que ajuda na transição do SNMP para o CMIS/CMIP. No entanto, embora a definição para o CMOT exista, nenhum trabalho significativo tem sido feito em cima deste protocolo desde algum tempo, e logo não há nenhum objeto neste grupo.

3.8.11 - O Grupo *Transmission*

O Grupo *transmission* provê informações sobre o meio específico que forma a base das interfaces no sistema. Quando os padrões Internet para gerenciar vários tipos de meios forem definidos, este grupo será o prefixo para estas informações.

3.8.12 - O Grupo SNMP

Os objetos do grupo SNMP podem ser aplicados em todas as cinco áreas de gerenciamento. Aplicações de gerenciamento de falhas observando só problemas SNMP podem achar útil conhecer o número de erros SNMP e sua freqüência, enquanto aplicações de gerenciamento de performance podem calcular a taxa de pacotes SNMP entrando e deixando a entidade. Já aplicações de gerenciamento de contabilização podem usar os objetos SNMP para encontrar o número de pacotes SNMP enviados ou recebidos pela entidade. E por fim, alguns objetos do grupo SNMP podem ajudar no gerenciamento de configuração e segurança.

Objetos para Gerenciamento de Falhas

Objeto	Informação usada para gerenciamento de Falhas
snmpInASNParseErrs	total de mensagens recebidas com erros ASN
snmpInTooBigs	total de mensagens recebidas com erro "too big"
snmpInNoSuchNames	total de mensagens recebidas com erro "SuchName"
snmpInBadValues	total de mensagens recebidas com erro "badValue"
snmpInReadOnlys	total de mensagens recebidas com erro "readOnly"
snmpInGenErrs	total de mensagens recebidas com erro "genErr"
snmpOutTooBigs	total de mensagens enviadas com erro "too big"
snmpOutNoSuchNames	total de mensagens enviadas com erro "noSuchName"
snmpOutBadValues	total de mensagens enviadas com erro "badValue"
snmpOutGenErrs	total de mensagens enviadas com erro "genErr"

Tabela 3.21 – Objetos para Gerenciamento de Falhas (SNMP)

Os objetos listados na tabela 3.21 informam erros referentes a mensagens SNMP. Esses erros não indicam erros na rede em si, mas pode informar que a entidade não está manipulando os pacotes SNMP apropriadamente. O número e tipos de erros também podem indicar que a entidade está recebendo pacotes SNMP com erros dos dispositivos da rede. A solução para esses erros geralmente está na configuração do gerente ou agente SNMP. Se a reconfiguração não diminuir o número de erros, o problema provavelmente residirá na implementação do gerente ou agente SNMP.

Objetos para Gerenciamento de Performance

Objeto	Informação usada para gerenciamento de Falhas
SnmpInPkts	taxa de pacotes SNMP recebidos
SnmpOutPkts	taxa de pacotes SNMP enviados
snmpInTotalReqVars	taxa de Get/Get-Next-Requests recebidas
snmpInTotalSetVars	taxa de Set-Requests recebidas
snmpInGetRequests	taxa de Get-Requests recebidas
SnmpInGetNexts	taxa de Get-Next-Requests recebidas
snmpInSetRequests	taxa de Set-Requests recebidas
snmpInGetResponses	taxa de Get-Responses recebidas
SnmpInTraps	taxa de Traps recebidas
snmpOutGetRequests	taxa de Get-Requests enviadas
snmpOutGetNexts	taxa de Get-Next-Requests enviadas
snmpOutSetRequests	taxa de Set-Requests enviadas
snmpOutGetResponses	taxa de Get-Responses enviadas
SnmpOutTraps	taxa de Traps enviadas

Tabela 3.22 – Objetos para Gerenciamento de Performance (SNMP)

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

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

Objetos para Gerenciamento de Contabilização

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

Tabela 3.23 – Objetos para Gerenciamento de Contabilização (SNMP)

Os objetos **snmpInPkts, snmpOutPkts, snmpInTraps,** e **snmpOutTraps** permitem calcular o total de pacotes e traps recebidos e enviados. Estas informações podem ser úteis no gerenciamento de contabilização.

Objetos para Gerenciamento de Segurança

Objeto	Informação usada para gerenciamento de Segurança
snmpInBadCommunityNames	total de pacotes com uma *community string* incorreta
snmpInBadCommunityUses	total de pacotes com *community string* que não permite a operação requisitada

Tabela 3.24 – Objetos para Gerenciamento de Segurança (SNMP)

- o objeto **snmpInBadCommunityNames** conta o número de vezes que um usuário ou aplicação, na tentativa de comunicar-se com o SNMP de uma entidade, não informou a *community string* correta.
- o objeto **snmpInBadCommunityUses** conta o número de vezes que um pacote SNMP é recebido contendo uma *community string* que não permite a operação requisitada.

Objetos para Gerenciamento de Configuração

Objeto	Informação usada para gerenciamento de Configuração
snmpEnableAuthenTraps	indica se o agente SNMP pode enviar traps

Tabela 3.25 – Objetos para Gerenciamento de Configuração (SNMP)

3.9 - Comparação entre a MIB da OSI e a MIB da Internet

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

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

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

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

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

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

3.10 - Remote Procedure Call – RPC

RPC é uma chamada remota de procedimentos (*Remote Procedure Call*), que fornece um mecanismo para um *host* fazer uma chamada de procedimento que aparentemente faz parte do processo local mas é realmente executada em outra máquina

da rede. O *host* no qual é executada a chamada do procedimento tem recursos que não estão disponíveis na máquina que a chamou.

3.10.1 - Modelo Cliente / Servidor

A distribuição de serviços computacionais impõe uma relação cliente/servidor entre os dois *hosts*, o *host* que possui um recurso é o servidor para aquele recurso, o *host* que chama um recurso se torna cliente do servidor que o possui.

3.10.2 - Mecanismo

Muitos mecanismos cliente/servidor podem ser definidos para cada máquina na rede. Um servidor para um recurso é seguidamente um cliente de muitos outros na mesma rede.

3.10.3 - Funcionamento do RPC

3.10.3.1 - Argumentos do procedimento

O sistema RPC envia os argumentos passados ao procedimento através de um datagrama de rede.

3.10.3.2 - Comunicação

O cliente RPC cria uma sessão ao localizar o servidor apropriado, e envia um datagrama ao processo no servidor que pode executar o RPC. No servidor, os argumentos são desempacotados; é executado o procedimento, o resultado é empacotado (se existir), e enviado de volta ao cliente. De volta ao cliente, a resposta é convertida em um valor de retorno para a chamada de procedimento, e a aplicação de usuário é recomeçada como se um procedimento local tivesse terminado. Esse é o fim de “sessão” como definido no modelo ISO.

3.10.3.3 - Transporte

O transporte é efetuado pelos protocolos TCP e UDP. Serviços RPC podem ser construídos tanto em transportes TCP ou UDP. A maioria dos serviços são orientados a UDP porque eles estão baseados em requerimentos de curta-duração. Usando UDP força que a chamada de RPC contenha suficiente informação de contexto para sua execução independente de qualquer outro requerimento de RPC, já que os pacotes UDP podem chegar em qualquer ordem.

3.10.3.4 - *Timeout*

Quando uma chamada de RPC é realizada, o cliente deve especificar um período de *timeout* no qual a chamada deve se realizar. Se o servidor estiver sobrecarregado ou desligado, a chamada remota não será executada antes do fim do *timeout*. A ação tomada no *timeout* do RPC varia de aplicação para aplicação. Algumas mandam de volta a chamada de RPC, enquanto outras procuram outro servidor.

3.10.4 - Identificando Serviços RPC

Serviços disponíveis através de RPC são identificados através de quatro valores:

- Número de Programa
- Número da Versão
- Número do Procedimento
- Protocolo (UDP ou TCP)

O número de programa unicamente identifica o serviço RPC

O serviço em si (identificado com um único número de programa) pode ser composto por vários procedimentos. O número de procedimento é passado num requerimento RPC como um opcode ao servidor RPC. Números de procedimento

começam em 1 (0 é a função *null*). Enquanto os números de programa são bem divulgados, os números de procedimento são particulares ao serviço. Seguidamente, os números de procedimento estão contidos num arquivo de cabeçalho (*header*) que é compilado dentro do programa cliente. Por exemplo, os números de procedimento de NFS (*Network Filesystem*) estão definidos no arquivo de cabeçalho /usr/include/nfs/nfs.h

3.10.4.1 - RPC *Portmapper*

O serviço *portmap* existe para registrar serviços RPC e fornecer seus números de porta IP quando dado um certo número de programa RPC. O próprio *portmapper* é ele mesmo um serviço RPC, com uma porta IP bem conhecida (111) para poder ser contatada diretamente pelos *hosts* remotos. O *portmapper* é também utilizado para tratar o RPC Broadcast, um requerimento que é enviado para todos os servidores para um serviço RPC particular.

3.10.4.2 - Números de versão RPC

Cada nova implementação de um servidor RPC tem seu próprio número de versão. Números diferentes são usados para coordenar múltiplas implementações do mesmo serviço, cada um dos quais podem ter interfaces diferentes. À medida que um serviço RPC amadurece, o autor do serviço pode achar necessário adicionar novos procedimentos ou novos argumentos para procedimentos existentes. Modificando a interface implica em incrementar o número de versão. A primeira versão de um programa RPC é 1. Quando contactando o *portmapper* num *host* remoto, tanto o lado local como o lado remoto devem se entender quanto ao número de versão para o serviço RPC que será usado. A regra básica é a de usar o número de versão mais alto que os dois lados conseguem entender.

4 - Software Tivoli

4.1 - Tivoli Management Environment 10 (TME 10)

O Tivoli Management Environment 10 (TME 10) é um ambiente de gerenciamento de sistemas distribuídos. Os principais módulos são apresentados nos itens seguintes:

4.1.1 - Tivoli TME 10 Estrutura do Produto

A Figura 4.1 a seguir ilustra a estrutura do Tivoli TME.

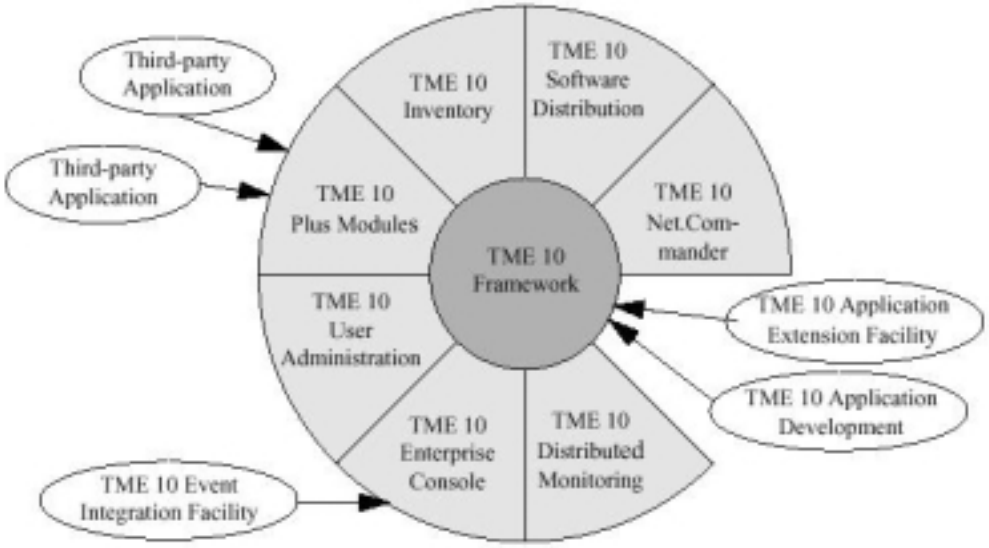


Figura 4.1 – Estrutura do Tivoli TME

Antes de maiores explicações sobre o TME é preciso lembrar que o TME *Framework* é um produto distinto das outras aplicações do TME; é possível adquiri-lo e instalá-lo separadamente para qualquer aplicação de gerencia, tanto para o Tivoli como para soluções do tipo *third party*.

No TME 10, algumas facilidades como instalação, o *desktop* de gerência do Tivoli, os subsistemas de notificação, e as regiões (*policy regions*) são partes do *Framework*, particularmente uma das aplicações do Tivoli. Quando o *Framework* é instalado pode-se ver a interface gráfica completa que é composta dos seguintes ícones;

- *Managed node icons*
- *Policy regions*
- *Notification bulletin board*

- *Administrator definitions*
- *Policy and task libraries*
- *Scheduler*

4.1.2 - Tivoli Management Environment 10 Framework

Tivoli Management Environment 10 (TME 10) Framework é o componente básico da linha de produtos TME 10, produz capacidades de administração de sistemas básicos, bem como serviços fundamentais para aplicações de gerenciamento. Estas capacidades de serviços incluem facilidades na administração do Tivoli, facilidades no policiamento de sistemas e facilidades nas notificações. Com arquitetura de gerenciamento orientada a objetos que possibilita a operação de um ambiente multiplataforma. Interface única entre os módulos de gerenciamento, adere a padrões abertos permitindo a integração com várias ferramentas de terceiros. Implementa em suas aplicações as estruturas de *Policy Region* e *Profile Manager*. A primeira consiste em estruturas hierárquicas criadas para organizar os recursos gerenciados; a segunda, em perfis de gerência, são utilizados por aplicações básicas da solução Tivoli para executar tarefas. Os perfis são compostos de informações relativas a cada aplicação. Por exemplo, perfis da aplicação *Tivoli Distributed Monitoring* vão conter informações como o que deve ser monitorado num determinado grupo de servidores, e que a ação deve ser tomada quando os limites pré-estabelecidos forem alcançados.

Gerenciadores de Perfis são grupos de *profiles*, os quais são aplicados a *endpoints*, que em última instância serão os servidores ou estações alvo das aplicações.

O TME 10 Famework permite instalar e criar vários serviços de gerência que possibilitam ao mesmo gerenciar recursos em sua rede. O usuário pode instalar qualquer serviço dependendo de suas necessidades. Somente o TMR *server* pode ser instalado. A seguir é mostrado uma lista de serviços de gerência proporcionados pelo TME 10 Framework, e a descrição de cada um.

- **TMR *server* (TMR-S)**

O TMR-S inclue as livrarias, binários, data files, e a interface gráfica do usuário (GUI) é necessário para instalar e gerenciar seu ambiente do TME 10. TMR *servers* mantém o banco de dados do TMR *server* e coordena todas as comunicações com o TME 10 *managed nodes*. O servidor também presume toda autenticação e verificação necessária para prover a segurança ao banco de dados do TME 10.

- ***Managed node***

O TME 10 *managed node* roda o mesmo software que o TMR *server*. *Managed nodes* mantém seus próprios bancos de dados, cada um pode ser acessado pelo TMR *server*. Quando um *managed node* se comunica com outro *managed node*, essa comunicação ou operação de segurança é feita da mesma forma que é feita pelo TMR *server*. A diferença primordial entre TMR *server* e *managed node* é o tamanho do banco de dados.

- ***Endpoint gateway***

Um *endpoint gateway* controla todas as comunicações e operações nos TME 10 *endpoints*. Um simples *gateway* pode suportar comunicação com milhares de *endpoints*. O *gateway* pode inicializar aplicações nos *endpoints* ou por eles.

Um *gateway* é criado em um *managed node* já existente. Esse *managed node* responsável prove acesso as aplicações nos *endpoints* e também comunicação com o TMR *server* que ocasionalmente os *endpoints* requisitam.

- ***Endpoints***

O *endpoint* é qualquer sistema que roda um serviço *endpoint* (ou daemon) tipicamente, um *endpoint* é instalado em uma máquina que não usa diariamente operações de gerência. O *endpoint* roda um pequeno número de softwares e não tem um

banco de dados específico a essa aplicação. A maioria dos sistemas nas instalações do TME 10 são *endpoints*. O TME GUI não é instalado nos *endpoints*. Se for escolhido rodar o *desktop* em um *endpoint*, o TME 10 *Desktop* terá que ser instalado.

- ***PC managed node and PC agent***

O PC *managed node* é um PC rodando o software agente. O software *agent* tem que ser instalado e tem que ser criado um PC *managed node* se tiver aplicações do TME 10 que não estão ainda habilitadas para o uso com *endpoints*.

4.1.2.1 - Componentes do TME 10 *Framework*

Temos a seguir os componentes da linha de produtos TME 10 e a relação destes produtos entre si de forma hierárquica:

- O TME 10 *Framework*, baseado em um sistema industrial padrão, constituído de capacidades básicas de sistemas de administração assim como um grupo de serviços fundamentais para aplicações de gerencia. Essas capacidades e serviços incluem instrumentos de administração, instrumentos que fornecem privilégios administrativos, além de sistemas responsáveis por definir a política e instrumentos de notificação.
- O TME *Desktop* proporciona uma interface gráfica com o usuário (GUI) assim como uma interface do tipo linha de comando (CLI).
- O TME 10 possui também ferramentas para realizar gerenciamento em tipos específicos de sistemas e serviços, incluindo, entre outros, aplicações de auditoria e distribuição de softwares, monitoramento remoto, gerenciamento de usuários e gerencia de internet e intranet.

As demais facilidades do TME 10 serão apenas citadas para que o leitor possa ter uma visão mais clara do esquema apresentado abaixo:

- TME 10 AEF (*Application Extension Facility*) que permite que o TME 10 customize dinâmicamente as aplicações suportadas pelo TME.
- TME 10 EIF (*Event Integration Facility*) permite que construção de adaptadores de eventos para mapear eventos de qualquer aplicação, recurso ou componente em um formato compatível com o TME 10 *Enterprise Console*.
- TME 10 ADE (*Application Development Environment*) onde estão incluídas as ferramentas de programação para a criação de novos aplicativos de gerenciamento que funcionarão a partir do *Framework*.

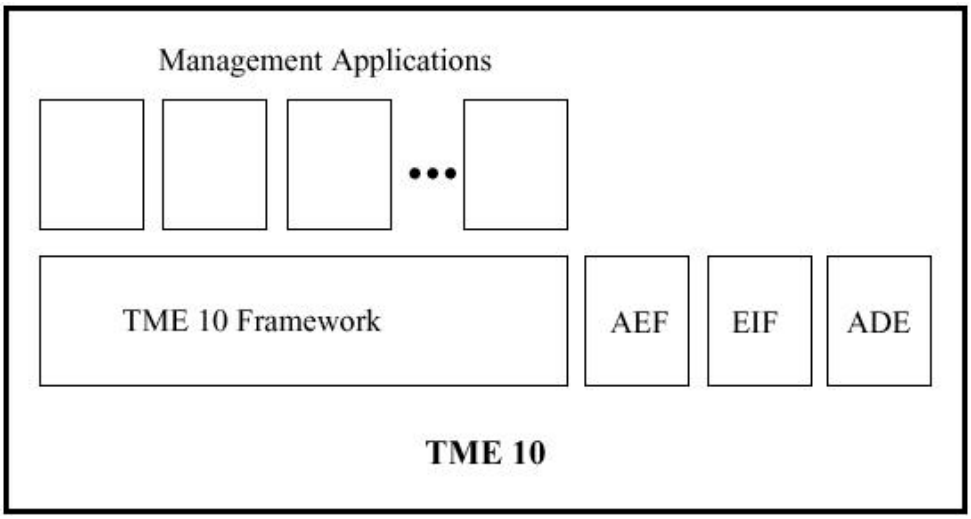


Figura 4.2 - Relação dos produtos da linha TME 10 entre si de forma hierárquica

4.2 - O *Gateway* e o Cliente *Endpoint*

O *Gateway* e o Cliente *Endpoint* permitem a instalação do TME 10 como sendo uma solução do tipo *third party* como mostra a figura 4.3.

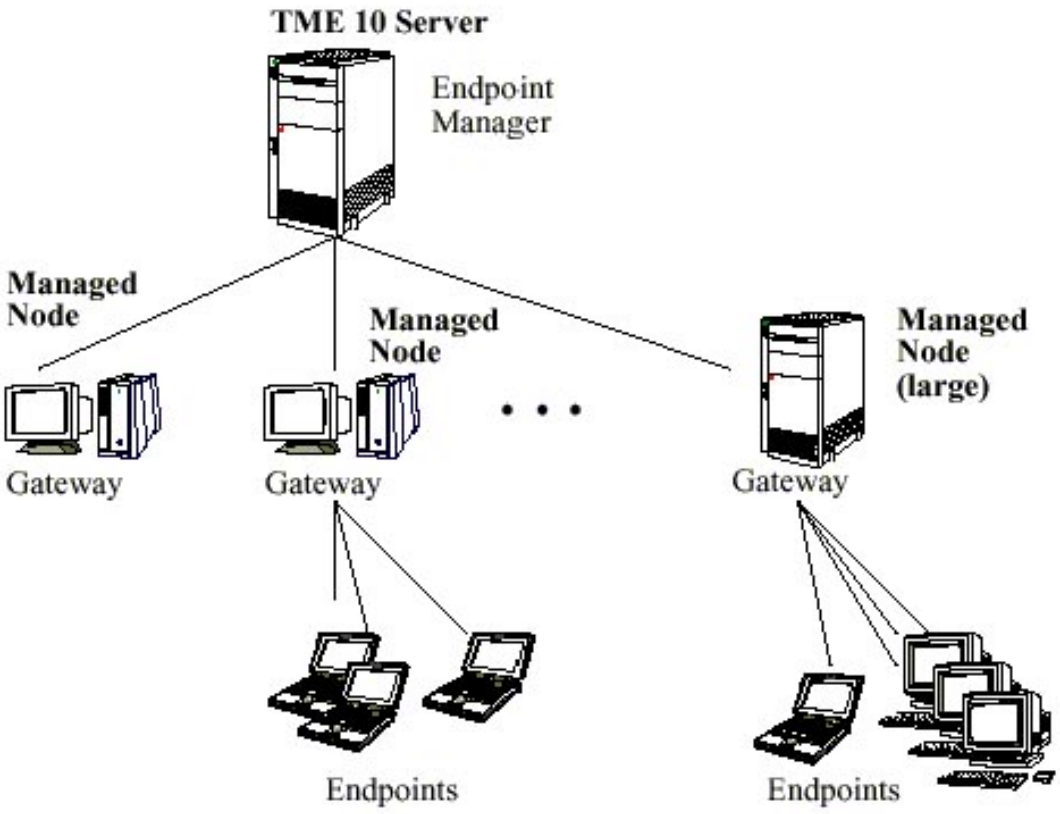


Figura 4.3 – Gateway e o Cliente Endpoint

Com um número reduzido de nós a serem gerenciados a quantidade de comunicação entre os nós e o TMR-S é diminuída de forma significativa. Os *endpoints* não se comunicam com o TMR-S a não ser durante o processo inicial de *login*. Todas as comunicações dos *endpoints* são passadas ao *Gateway*. Na maioria dos casos o *Gateway* proporciona todo o suporte que um *endpoint* necessita sem que seja necessário haver comunicação com o TMR-S.

Em redes pequenas é possível que o *Gateway* seja criado no próprio TMR-S. O TMR-S é capaz de lidar com toda a comunicação requerida quando se trata de uma rede com poucas máquinas.

Mas isso não é aceitável quando se trata de redes de grande porte; o TMR-S, nesse caso, ficaria sobrecarregado.

4.3 - Disciplinas do Tivoli

São softwares que têm funções distintas entre si , que são instalados e rodam a partir do TME 10.

As disciplinas encontradas no Tivoli TME são apresentadas a seguir para que se possa ter uma visão melhor do propósito de cada uma delas.

4.3.1 - Disciplinas de Monitoramento da Rede

Englobam o acompanhamento da performance da rede sobre os protocolos mais usuais, levantamento dos equipamentos ligados à rede *(auto-discovery)*, elaboração da topologia e gerência de eventos com notificações de problemas. Podendo ser classificadas como parte das modalidades de gerência de falhas e performance (Netview).

4.3.2 - Disciplinas de Segurança

São os aplicativos que permitem a aplicação de políticas de segurança, como: proteção de vírus com atualização automatizada, autenticação e controle de acesso à rede e recursos unificados e realização de cópias de segurança em resposta a eventos que indiquem comprometimento dos serviços em execução, confundindo-se com a modalidade de gerência de segurança (*Tivoli Security Management*).

4.3.3 - Disciplina de Inventário

Compreende a manutenção de base de dados contendo os dispositivos identificados na rede. No caso de computadores, esta disciplina prevê a catalogação dos recursos disponíveis (memória, disco, periféricos, etc.), bem como da lista de aplicativos instalados (*Tivoli Inventory*).

4.3.4 - Disciplina de Distribuição de Software

Funciona em conjunto com a disciplina de inventário, ao distribuir a todas ou a um grupo de máquinas da rede (que obedecem a um critério de classificação, como as que têm sistema operacional Unix, por exemplo) pacotes de aplicativos. Esta distribuição pode tomar duas formas: a primeira sendo a execução de instalação não acompanhada do software, em que um agente apropriado responde às indagações do programa instalador; e a segunda sendo a cópia de um pacote já instalado, com seus arquivos acessórios e alterações no sistema operacional (*Tivoli Software Distribution*).

4.3.5 - Disciplina de Servidores e Clientes

Esta disciplina prevê a monitoração e controle de servidores com funções específicas, como autenticadores de contas, correios eletrônicos, bancos de dados e outros. Em alguns casos é referenciada como disciplina de *groupware* (conjunto de aplicativos para uso em grupos de colaboração).

Além disso a disciplina de servidores e clientes engloba a gerência de estações clientes, monitorando protocolos de rede instalados, serviços em execução e outros parâmetros de interesse; está sob a modalidade de configuração.

4.3.6 - Disciplina de *Help Desk*

Destacam-se, nesta disciplina, os aplicativos voltados ao controle remoto de estações, acompanhamento de ordens de serviço, e a manutenção de bases de dados com os problemas e soluções mais comuns para um dado ambiente. Estes dados podem ser, a título de ilustração, disponibilizados em uma interface internet para consulta dos usuários, buscando a melhoria dos conhecimentos da massa usuária de serviços de informática, e por conseqüência deixando mais tempo aos técnicos para planejamento de ações preventivas.

4.3.7 - Disciplina de Internet;

A priori, está disciplina está contida na disciplina de servidores e clientes, entretanto, como o acesso e aplicações voltadas à internet estão cada vez mais presentes nas corporações, já se faz necessário um acompanhamento diferenciado, tal como: monitorar a performance do servidor *WEB* e implementar eficiente segurança contra ataques externos.

4.4 – Definições Específicas

4.4.1 - TME 10 *desktop*

É uma interface de usuário que provê acesso rápido á apresentação e componentes do TME 10.

4.4.2 - Remote Control

É uma disciplina do TME 10 que permite ao administrador do sistema controlar o teclado e o mouse, além da saída do monitor de um sistema remoto rodando diversas plataformas, tais como, Windows NT, Windows 95/98, Windows 3.x ou OS/2. O administrador pode monitorar o PC e permitir ao usuário manter o controle de seu sistema, mas também permite ao gerente assumir o controle total da máquina a ser gerenciada com ou sem a autorização do usuário, podendo modificar configurações, instalar e remover programas, além de apagar arquivos e diretórios. Os procedimentos de instalação do agente serão apresentados com detalhes mais adiante neste relatório.*TME 10 Remote Control* pode ser usado até mesmo para reiniciar um PC remotamente dando grande flexibilidade de gerenciamento.

Essas funcionalidades permitem: uma melhora no suporte de primeiro nível, diminuição da necessidade de envio de técnicos para dar suporte local, assegura diagnósticos mais rápidos e precisos e redução no tempo de duração das chamadas do *help-desk.*

4.4.3 - NetView

O NetView é uma ferramenta de gerenciamento de sistema e de rede que proporciona funções de configuração, falha e desempenho. Possui uma plataforma de gerenciamento de rede aberto que habilita a integração de aplicações SNMP (*Simple Network Management Protocol*) e CMIP(*Common Management Information Protocol*). Permite ao usuário descobrir redes TCP/IP, mapear topologia, receber eventos e *traps* SNMP e monitorar performance. Além disto, é possível inserir produtos dos mais variados fabricantes de equipamentos de infra-estrutura de redes do mercado, visualizando graficamente estes equipamentos e sendo possível configurá-los através do Tivoli NetView.

O módulo NetView executa o *discovery* da rede através de pings sucessivos enviados para redes pré-definidas através da configuração de comunidades, e tem como principais características:

- Gerenciamento do ambiente TCP/IP;
- Gerenciamento de sistemas UNIX;
- Gerenciamento de problemas e inventário;
- Gerenciamento de clientes e servidores IPX e NetBIOS;
- Gerenciamento de redes SNA;
- Gerenciamento de *hubs* IBM;
- Gerenciamento de redes distribuídas e heterogêneas utilizando protocolos de gerenciamento baseados em TCP/IP;
- Distribuição de tarefas de gerenciamento pela rede, reduzindo o tempo de processamento, o tráfego broadcast da rede;
- Gerencia todos os dispositivos IP que possuam agente SNMP;
- Possui interface gráfica padrão OSF/Motif e Xwindow System;

- Customização da inteface gráfica de acordo com a necessidade do usuário;
- Software possui um conjunto de objetos MIB já instalados. Outros podem ser acrescidos;
- Possui APIs (Application User Interface) que proporciona aplicações sem necessidade de conhecer os detalhes de implementação:
    - End-User Interface (EUI) API
    - X/Open Management Protocols (XMP) API
    - SNMP API
    - Event Filtering API
    - Process Management API
    - Network Logging and Tracing API
    - General Topology (GTM) API

Por definição, o NetView armazena os dados em arquivos internos. Contudo, ele pode ser configurado para trabalhar com banco de dados relacionais comerciais. O NetView recebe os dados da estação de trabalho que está gerenciando. Os dados são armazenados no *site* do servidor de banco de dados, o qual é executado nas diferentes estações onde o NetView se encontra. Estes dados podem ser acessados e compartilhados por qualquer estação que tenha o *run-time* e a configuração correta do banco de dados.

4.4.4 - *Distributed Monitoring*

Outra disciplina do software Tivoli usada neste trabalho é o *Distributed Monitoring* onde o status de uma variedade de recursos da rede, tais como, sistemas, aplicações e processos são verificados, tal aplicação é usada para monitorar recursos locais ou remotos e apresentar eventos e alarmes da rede. Disponibiliza uma monitoração distribuída dos recursos de sistemas remotos, proporcionando a integração

destes ao ambiente corporativo, dando uma visão centralizada de toda a estrutura. Este produto implementa um grande número de monitores que verificam a "saúde" dos vários tipos de servidores de um ambiente distribuído. É possível monitorar desde a quantidade de espaço livre em disco, porcentagem de utilização de CPU, até quantidade de *swap* de páginas em memória.

Os componentes utilizados pelo *Distributed Monitoring* são o TME 10 DM *profiles*, o TME 10 DM *engine* e *Indicator collections*.

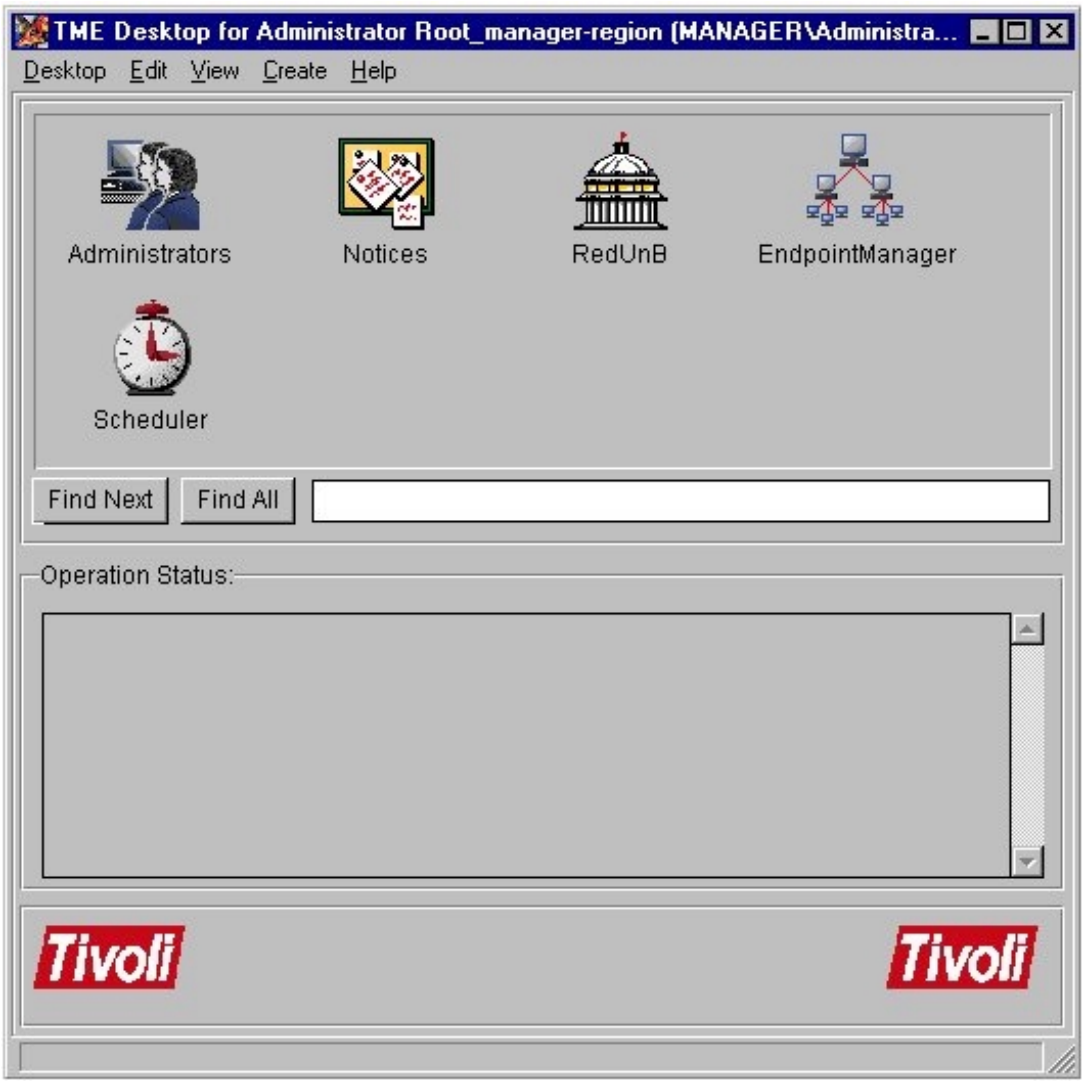
4.4.5 - *Software Distribuiton*

Finalmente será apresentada a disciplina do software *Tivoli Software Distribuiton* que permite a distribuição e instalação automática de softwares em máquinas de uma rede heterogênea de forma eficiente. O software é distribuído a partir de um repositório central para os servidores intermediários *(fanout servery)*, e destes para as estações alvo, que estarão sempre sincronizadas umas com as outras. O produto ainda oferece características como a elaboração automática de pacotes, determinação da utilização de banda e agenda para o processo de distribuição. Tem total integração com o módulo de inventário, sendo possível pesquisar os pré-requisitos de hardware e software das máquinas alvo da distribuição.

5- Metodologia, Resultados e Análise

5.1 – TME 10 Framework

A descrição deste experimento é iniciada pela base do Tivoli TME 10, o TME Framework, como já foi descrito anteriormente na parte que trata do Tivoli TME 10 Estrutura do Produto (item 4.1.1 deste relatório).



Na figura 5.1 é apresentada a interface gráfica do software:

Figura 5.1 – Interface gráfica do Software Tivoli

A instalação do *Tivoli Framework* é bastante simplificada e pode ser realizada acompanhando as instruções do capítulo 6 do Manual do *Framework* (TME 10 *Framework Planning and Installation Guide*, localizado no CD do *Framework* no arquivo MPG2.PDF), algumas informações consideradas relevantes ao processo de instalação serão apresentadas a seguir.

O TMR-S e os seus clientes devem possuir seus próprios *daemons* **oserv** (objetos de envio que permanecem sempre ativos). O *daemon* **oserv** do TMR-S coordena as comunicações entre ele e os clientes. Cada **oserv** dentro dos clientes deve utilizar a mesma porta que o TMR-S; é necessário instalar o *Framework* no TMR-S antes de se instalar qualquer cliente do TME 10 ou qualquer uma das suas aplicações.

Antes da instalação do TME 10 são necessários alguns procedimentos, destacando-se os seguintes: os nomes dos nós gerenciados pelo TME 10 devem ser incluídos no DNS.

Se estiver sendo utilizado o DHCP nos clientes utilizando Windows NT deve-se assegurar que o TMR-S também deverá ser uma máquina rodando Windows NT;

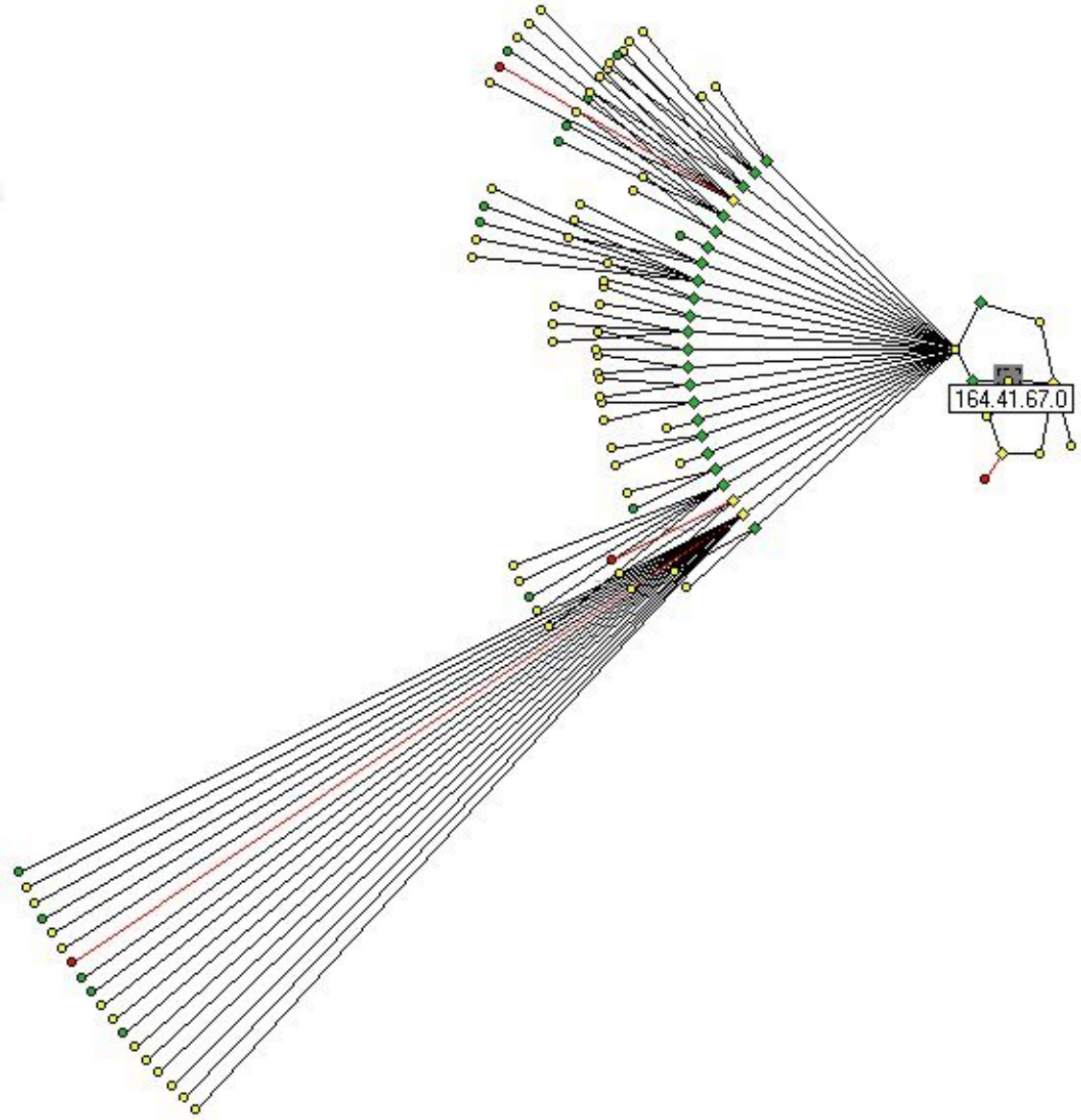
Se na sua rede houve clientes Windows NT é necessário que seja instalado o *Tivoli Remote Execution System* em pelo menos um dos sistemas NT, se o TMR-S for Windows NT este serviço é instalado automaticamente;

Se estiver no planejamento compartilhar os binários na rede é preciso, para o Windows NT, que se crie uma conta de acesso remoto.

O *Tivoli Remote Execution Service* (TRIP) instala clientes do Windows NT remotamente, o TRIP monitora serviços do tipo exec na porta 512; ele deve ser instalado localmente em cada cliente NT. O primeiro sistema NT localizado pelo *Framework* que contém o TRIP instalado será designado como o repetidor NT.

5.2 – NetView

No que se refere ao experimento de gerência de falhas, desempenho e configuração da rede foi utilizado o software Tivoli NetView. A instalação desta disciplina foi feita na máquina gerente. O Netview, como foi explicado anteriormente,



mapeou a RedUnB e a rede do ENE. A figura 5.2 apresenta a topologia lógica mapeada pelo software, que será explicada mais a frente:

Figura 5.2 -Mapeamento da RedUnB e rede ENE pelo software NetView

Após ter se instalado e configurado o Framework pode-se instalar o Tivoli NetView para que se inicie o descobrimento (Discovery) e o monitoramento (Monitoring) da rede; novamente apenas as informações consideradas relevantes serão apresentadas. O método completo de instalação pode ser acompanhado a partir do CD de instalação do NetView (arquivo readme.html).

Antes de se instalar o NetView alguns requisitos de software devem ser observados, são eles: sistema operacional Windows NT versão 4, protocolo TCP/IP instalado e configurado, e serviço SNMP instalado e configurado; se for necessário instalar o kit sem que o SNMP esteja rodando é preciso usar o flag –**ss** com o *setup*. Apesar de existir esta possibilidade é altamente recomendado que a instalação seja realizada com o serviço SNMP rodando, senão é preciso ter um arquivo de origem para realizar o *autodiscovery* da rede. Opcionalmente pode-se instalar o Microsoft SQL versão 6.5.

Partiu-se para a instalação do NetView seguindo as instruções contidas no manual, após a instalação e configuração, o NetView funcionou com perfeição começando a realizar o autodiscovery da rede.

A parte mais importante da instalação do NetView é a configuração das *communities*, deliberadas pelos gerentes das redes locais, que definem níveis de autoridade dos mesmos, permitindo que os equipamentos possam ser gerenciados. No caso de não terem sido configuradas todas as *communities* da rede a ser gerenciada nesta primeira configuração, é possível acrescentar outras a partir da interface gráfica do NetView.

É importante preservar tais *communities* pois com elas um invasor pode facilmente modificar configurações em equipamentos fundamentais para o funcionamento da rede podendo modificar inclusive o endereço IP de roteadores e *switches*.

Com o *autodiscovery* acionado e com as *communities* configuradas em pouco tempo é possível se ter uma idéia bastante clara da topologia lógica da rede que está sendo gerenciada.

A interface gráfica do NetView permite visualmente que o gerente da rede fique a par das condições de funcionamento da rede através do código de cores utilizado pelo software, onde vermelho significa estado critico, ou seja, máquina desligada ou ainda uma rede fora do ar; amarelo, que significa que algum nó está com problemas naquela rede, podendo ser apenas um computador pessoal desligado. Existe a possibilidade de se escolher que nós gerenciar e que nós ignorar, uma política a esse respeito é fundamental para que, ao ser gerado um alarme de falha, moderada ou crítica, o gerente saiba

exatamente onde tal falha ocorreu e qual a sua relevância para o funcionamento geral da rede; e finalmente a cor verde que significa funcionamento normal do equipamento. Tal conceito de ferramenta facilita muito o gerente da rede no que se refere à gerência de falhas.

Outra grande utilidade do NetView é a ferramenta *Event Browser*, de onde se pode selecionar que tipos de evento se deseja observar, a partir desta ferramenta pode-se ter uma visão mais específica dos problemas ocorridos na rede.

Finalmente citamos a ferramenta mais importante do NetView, segundo a concepção dos realizadores deste projeto, que é a ferramenta MIB *Browser* a partir da qual pode-se realizar as tarefas de gerência de falhas, desempenho e configuração, cujas explicações são apresentadas pelas tabelas da parte de MIB do relatório (item 3.8).

O pacote Netview pode fazer a coleta e modificar MIB´s de todos os equipamentos mapeados e que tem um agente SNMP instalado e configurado. A modificação das MIB´s é uma ferramenta poderosa, tendo em vista que a MIB, como foi explicado anteriormente compreende as informações e as configurações da máquina.

Para melhorar o nível de gerência e para visualizar mais a fundo o funcionamento dos softwares do Tivoli, foi instalado o software Sniffer Pro (item 5.4). Como já era esperado, foi observado que o NetView usa o protocolo SNMP para realizar as aplicações como captura e modificação de MIB´s, bem como outras aplicações (ver figura 5.3).

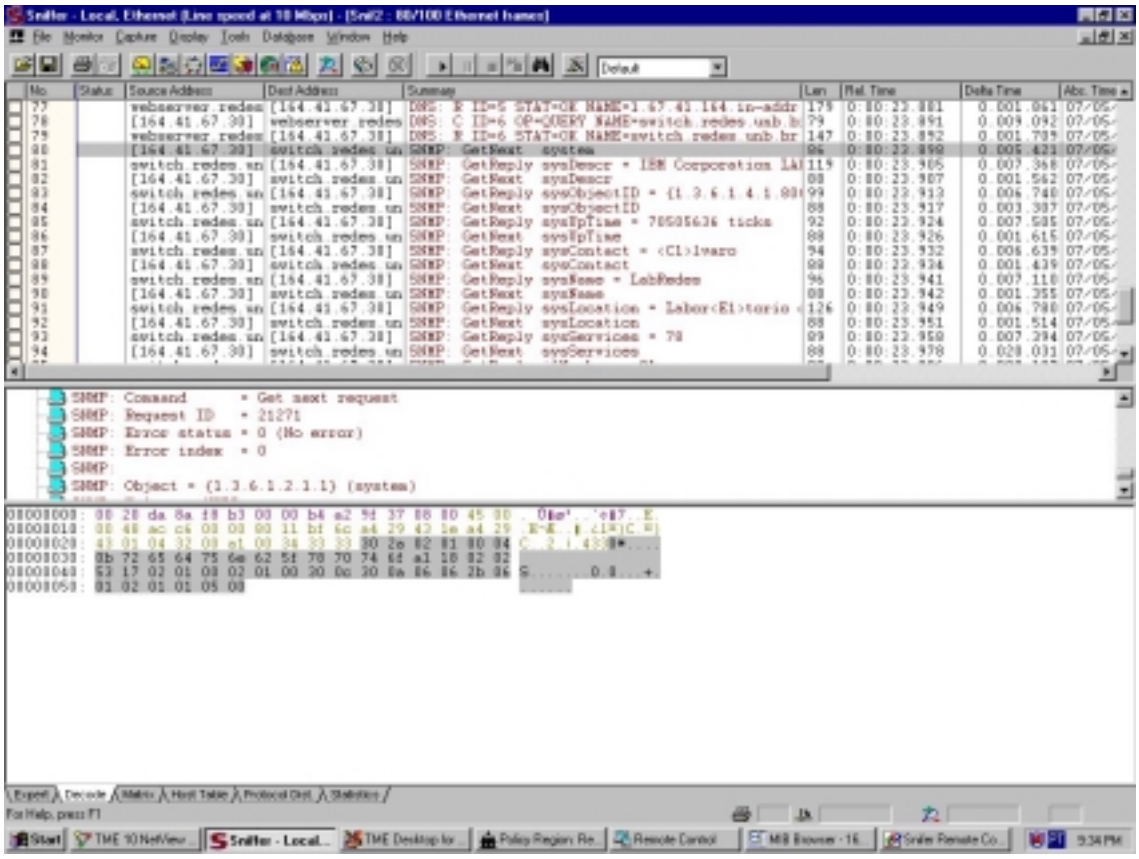


Figura 5.3 – Visualização com sniffer da utilização do protocolo SNMP pelo Netview

O experimento do módulo NetView mapeou as redes do backbone da RedUnB e do backbone do ENE indicando uma quantidade de nós superior a 1700 entre HUBS, SWITCHES e placas de rede chegando ao nível de computadores pessoais. Testes realizados, como coleta de MIB's, troca de valores de MIB's e monitoração da rede foram executados com êxito, tornando este e outros softwares de gerência modelo de demonstrações a pessoas interessadas no assunto.

5.3 - *Remote Control*

A experiência com a disciplina *Remote Control* foi feita instalando-se agentes Tivoli em algumas máquinas de fácil acesso, localizadas no Laboratório de Redes da UnB, e que não estavam sendo utilizadas. Para que esta disciplina do Tivoli seja disponibilizada é necessário que se crie um *PcManage Node* (ver figura 5.4 a seguir) no *Tivoli desktop* do TMR-S, em seguida deve-se criar um ícone de *Remote Control*.

Ao se clicar no ícone *Remote Control*, se escolhe a máquina em que se quer realizar esta operação (ver figura 5.5). O agente do *Remote Control* precisa ser instalado na máquina cliente pelo gerente por meio do software Tivoli *Desktop*, na opção *Install Products*. Os testes foram realizados de todas as formas disponibilizadas pelo software (ver figura 5.6), como monitor/monitor que permite a monitoração da máquina pelo gerente, *control/active* que permite controle total da máquina pelo gerente, *control/monitor*, que permite ao usuário escolher se deseja ser controlado ou só monitorado e o *Reboot*, que, como o próprio nome diz, reinicializa a máquina (ver figura 5.7). Existem algumas opções de tempo de espera pela resposta do cliente (ver figura 5.8); as mesmas foram testadas, é bom ser citado que a escolha do tempo de espera de 0 segundos possibilita ao gerente monitorar, assumir o controle ou reinicializar a máquina sem a necessidade da permissão do usuário. O fato do usuário não estar presente para dar permissão ao gerente não impede que o gerente não execute a tarefa desejada, pelo fato de existir uma opção *Proceed if time out*, que responde ao comando se o tempo de resposta estipulado pelo gerente espirar. Outras opções são apresentadas na tela.

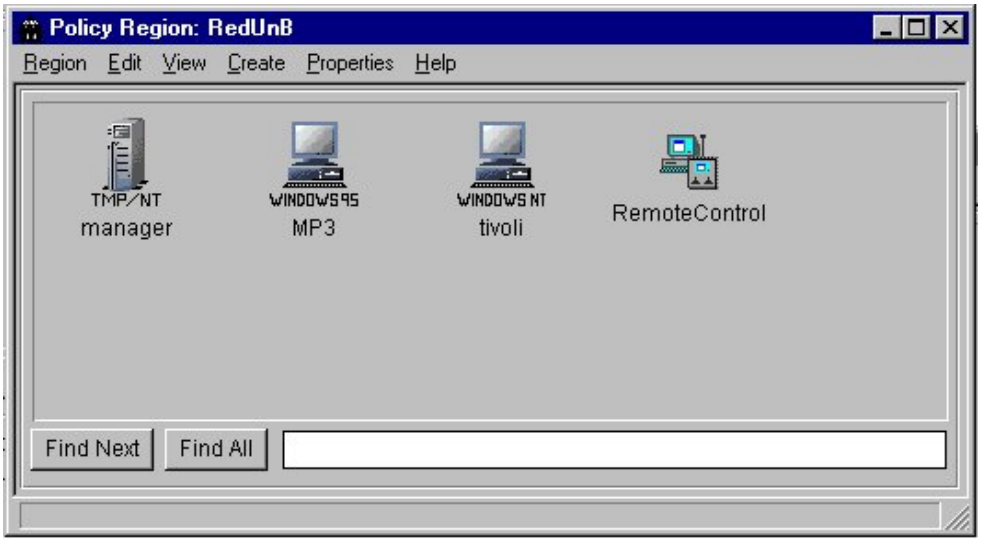


Figura 5.4 – Policy Region: RedUnB

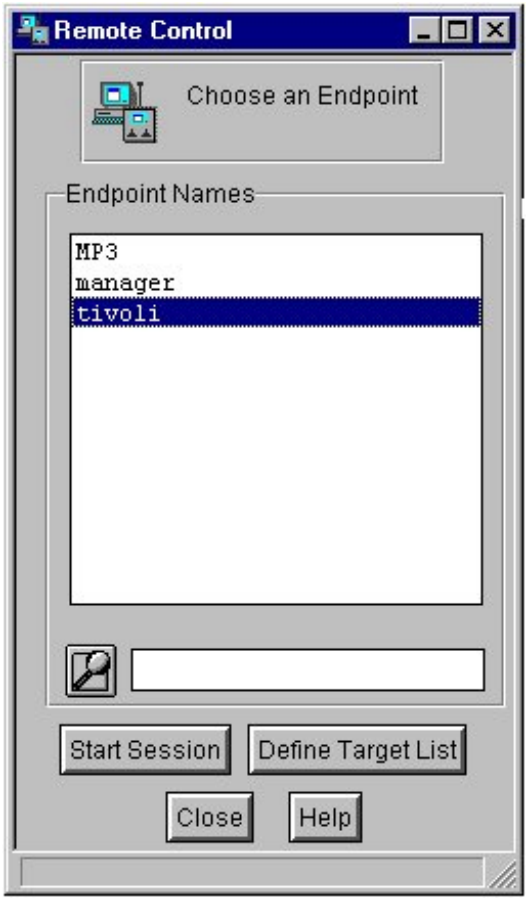


Figura 5.5 – Remote Control

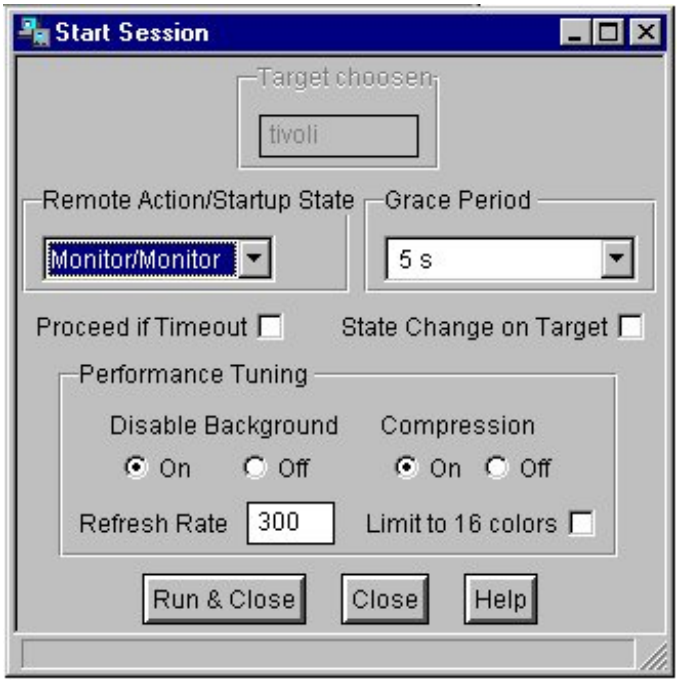


Figura 5.6 – Remote Control Options

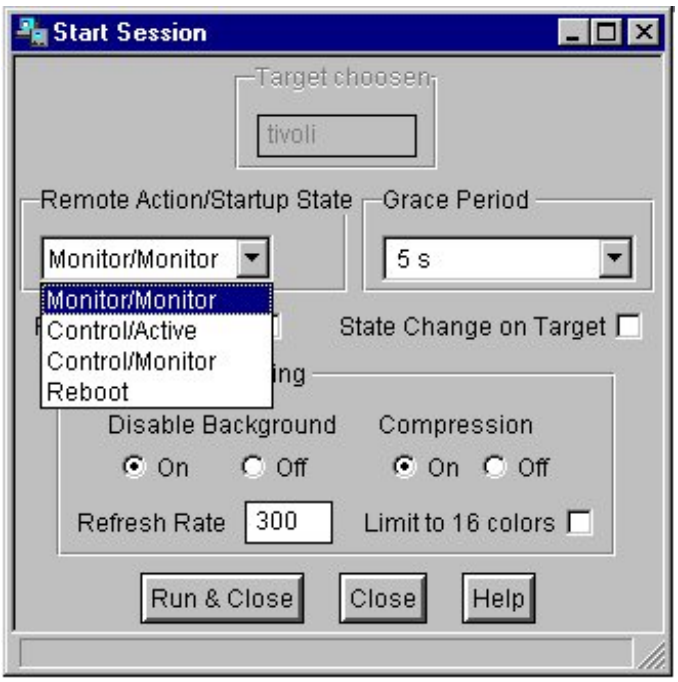


Figura 5.7 – Remote Control action options

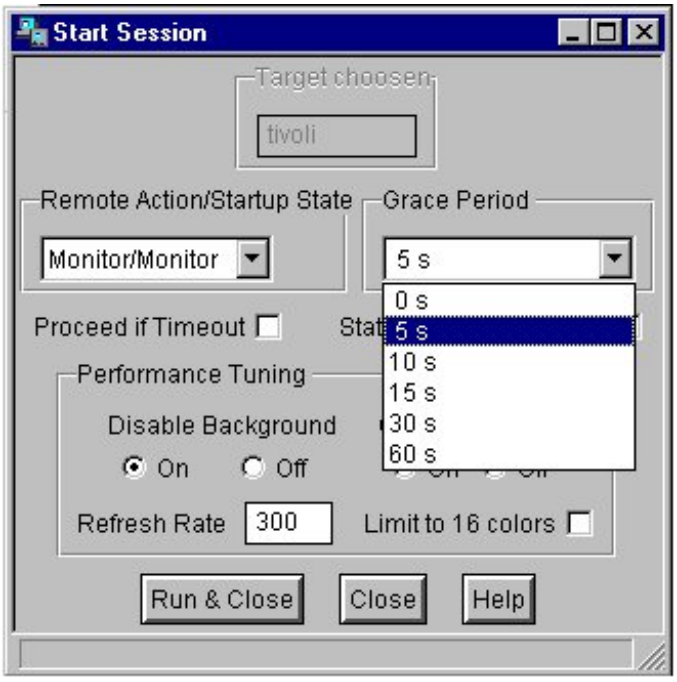


Figura 5.8 – Remote Control time options

Nas primeiras experiências com o *Remote Control* não obtivemos êxito. Muitas tentativas foram realizadas e muitos erros aconteciam, principalmente erros em arquivos binários na máquina cliente. Tentou-se muitas formas de instalação, formatação de

máquinas e procura por vírus, mas os problemas continuaram. Foram realizados testes em várias máquinas, e a máquina em que estava instalada o gerente foi trocada. Reinstalou-se o Tivoli *Framework* e a disciplina *Remote control*, mas os erros continuavam. Depois de muitas tentativas, descobriu-se que o erro estava na forma de instalação. A maneira correta de se instalar o software e seus agentes está descrita no item 5.3.1 à frente.

Na aplicação do *Remote Control* foi realizado a experiência com o SNIFER também, a fim de descobrir quais protocolos eram usados na mesma. Descobrimos que esta aplicação usa o protocolo TCP, como já era esperado para estabelecer uma conexão entre o gerente e o cliente, e, estabelecida esta conexão a aplicação usa o protocolo RPC, que é apresentado na seção 3.10, para executar o *Remote Control* (ver figura 5.9).

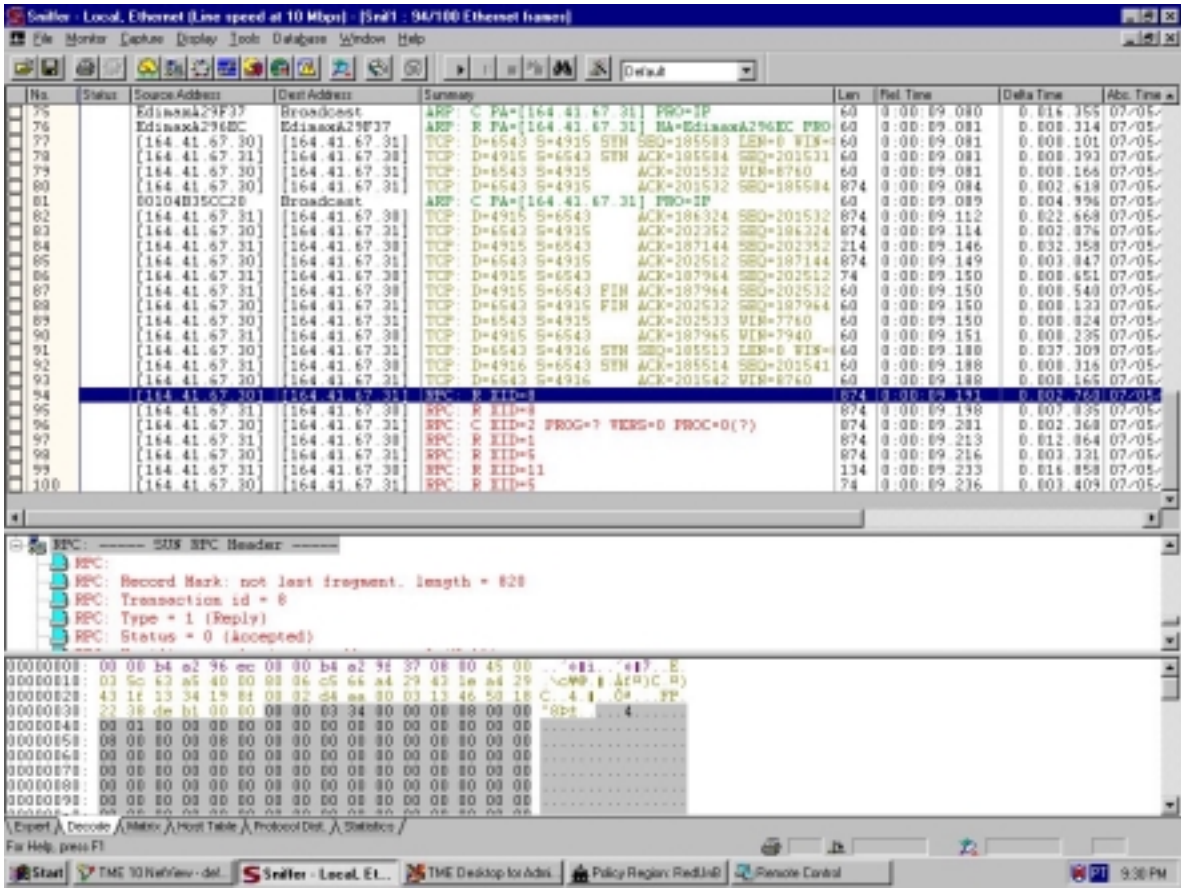


Figura 5.9 – Descoberta de qual protocolo é usado na aplicação Remote Control

O experimento Remote Control, depois da resolução dos problemas mencionados anteriormente, funcionou em três diferentes máquinas, com diferentes configurações e diferentes sistemas operacionais, provando que está aplicação está pronta para ser colocada em prática em uma rede maior. Demonstrações foram realizadas também com esta aplicação.

5.3.1 – Resolução de Problemas do Remote Control

Para que a instalação e o funcionamento da aplicação Remote Control funcione perfeitamente, alguns passos têm que ser seguidos, conforme descritos abaixo:

- Instale no cliente o agente do Remote control que se encontra no CD TME 3.6 Products no diretório especificado abaixo:
  RC36/LCF4WIN/FLOPPY/ENGLISH/WINNT40/DISK1/SETUP.EXE
- Instale no cliente o agente Tivoli, que se encontra no CD TME 10 Framework no diretório especificado abaixo:
  PC/TCPAGENT/DISK1/SETUP.EXE
- Rode no cliente os arquivos .BAT que estão no CD TME Products no diretório especificado abaixo:
  RC36/SDPACK/WINNT/CTL OU RC36/SDPACK/WINNT/TGT

Entre no Tivoli Framework instalado na estação de gerência. Clique em Install Products, configure o path para o diretório RC36 do CD TME Products, e instale os agentes do Remote Control pelo Tivoli Framework da estação de gerência isto instalara os arquivos binários necessário para que a aplicação Remote Control funcione.

5.4 – Sniffer

Um *sniffer* é um software que recolhe informações que trafegam por uma rede. Esta rede pode estar rodando qualquer protocolo: Ethernet, TCP/IP, IPX, ou outros (ou qualquer combinação destes). O propósito de um *sniffer* é colocar a interface de rede neste caso em modo promíscuo e, fazendo isso, capturar todo tráfego da rede que passa pelo barramento em que o mesmo está conectado.

O software *Sniffer Pro version* 2.50.07 foi instalado na máquina gerente, afim de se obter um conhecimento maior da rede que se deseja gerenciar, bem como conhecer melhor os softwares que executam esta gerência. Com a coleta de pacotes que trafegam naquele barramento especifico, conseguimos descobrir quais protocolos eram usados em cada aplicação, e se essas aplicações aumentavam muito o trafego de pacotes pela rede. O *sniffer* também proporciona que seja visto, com quais máquinas o gerente está se comunicando, observando assim o *polling* realizado pela máquina gerente na rede. Este software também pode, e é usado, na segurança, tendo em vista que pacotes não podem chegar, ou sair da estação gerente sem serem capturados pelo *sniffer*, logicamente quando o mesmo está ativo.

Esta versão de *sniffer* usada nesta experiência é capaz de capturar e armazenar os 100 últimos pacotes transmitidos na rede, o que para descobrir protocolos, ou descobrir com quem a máquina gerente está se comunicando, é o necessário. Porém para que se crie uma gerência segura e mais completa em todos os sentidos, seria necessário a aquisição de um software de *sniffer* mais completo, que armazenasse mais pacotes e com mais recursos que o atualmente utilizado.

Como já foi mencionado em outras experiências descritas anteriormente, esta experiência nos ajudou muito no aprendizado de outros softwares, e mesmo em uma melhora significativa no nível de gerência que gostaríamos de ter em nossa rede. O sniffer se mostrou uma ferramenta indispensável a uma gerência de qualidade.

6 - Proposta de gerência

De acordo com os estudos realizados pelos integrantes deste grupo com a orientação dos professores e ao analisarmos a rede lógica apresentada pelo software Tivoli NetView através da sua interface gráfica e realizado a operação de Discovery da rede, chegou-se a seguinte proposta para implementação de uma rede gerenciada de forma eficiente e descentralizada, utilizando para isso, o software Tivoli TME.

Pode-se ter uma visão geral da proposta observando-se a figura 6.1.

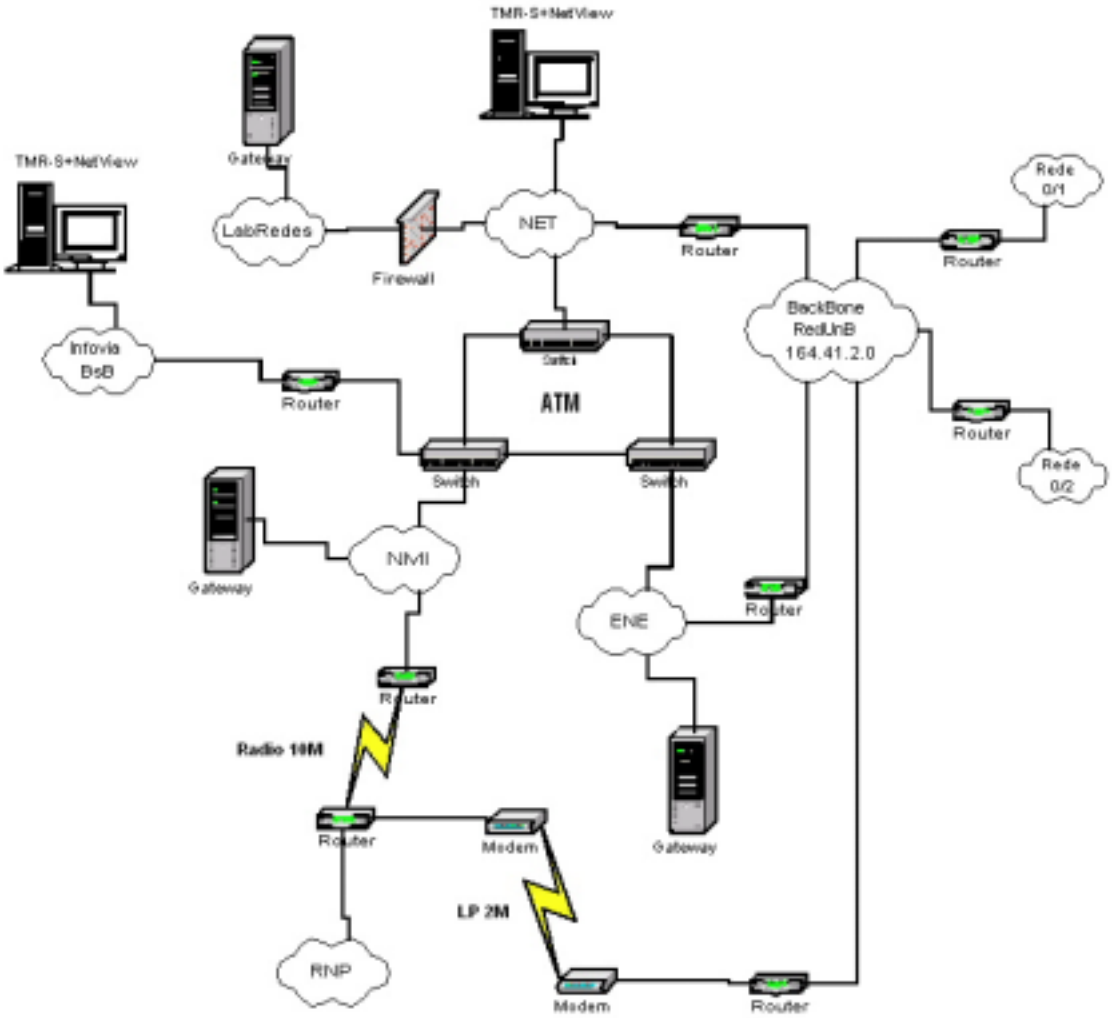


Figura 6.1 – Proposta de Gerência

A estação central de gerência TMR-S (*Tivoli Management Regions – Server*) está localizada no Departamento de Engenharia Elétrica no Laboratório de Redes, nesta máquina estão instalados os softwares de gerência da IBM, a base destes programas é o Tivoli *FrameWork*, sobre o qual está instalado o software Tivoli e seus pacotes em especial o NetView que realiza o mapeamento da rede de forma gráfica.

Foi decidido manter o TMR-S gerenciando o *backbone* da RedUnB até as bordas, ou seja, até os roteadores que fazem fronteira com as redes locais dos departamentos localizados no ICC na UnB, além do *backbone* ATM que é constituído de três *switches* ATM que formam um anel, como pode ser visualizado na figura acima apresentada, também localizado no Laboratório de Redes.

Optou-se por instalar um *Gateway* Tivoli no laboratório ao lado do que está instalado o TMR-S, isso foi feito para que se testasse a eficiência do Tivoli gerenciando uma rede localizada atrás de um *Firewall*. Também se pretendia realizar o experimento da distribuição de software de forma remota para as máquinas localizadas naquele laboratório.

Outro local onde deverá ser instalado outro *Gateway* é no *backbone* do Departamento de Engenharia Elétrica – ENE para descentralizar a gerência evitando que o TMR-S fique sobrecarregado com, por exemplo, o tráfego de pacotes de *polling* realizado constantemente pelo NetView. Além de se tratar de uma rede externa ao ambiente de trabalho em que o TMR-S está inserido.

Finalmente propõe-se a instalação de um *Gateway* no ambiente do NMI por também ser um ambiente externo ao Laboratório de Redes e por possuir características especiais como um link de rádio direto com a RNP.

As instalações dos *Gateways* não foram realizadas pois era necessária uma licença de instalação de cliente Tivoli que não estava disponível no período da realização deste projeto.

Esta proposta de gerência distribuída foi elaborada desta forma para que fosse possível obter informações de maneira segura de várias redes com as mais diversas configurações e ambientes, os *gateways* proporcionariam uma visão descentralizada da rede como um todo, facilitando a localização de possíveis problemas, além de

minimizar o tráfego de informações entre os *endpoints* e o TMR-S ampliando significativamente o número de máquinas a serem gerenciadas.

Esta decisão foi tomada tendo por base as informações contidas no Manual do TME 10 *Framework* (***TME 10 Framework Planning and Installation Guide***), tais informações estão apresentadas a seguir.

6.1 - Sugestões de Pacotes do Tivoli TME 10 a Serem Adquiridos

Para encerrar esta proposta de gerência serão apresentadas sugestões de Pacotes do Tivoli TME 10 a serem adquiridos e que, segundo a concepção deste grupo juntamente com a orientação dos professores, seriam de grande utilidade para que a gerência da RedUnB possa ser realizada de maneira eficiente.

A plataforma Tivoli apresenta vários pacotes, cada um com várias disciplinas que realizam diferentes funções. Os pacotes apresentados pelo Tivoli em seu site serão apresentados abaixo. Os pacotes apresentados são:

- ***Asset Management***
- ***Availability Management***
- ***Change Management***
- ***Network Management***
- ***Operations Management***
- ***Security Management***
- ***Service Management***
- ***Storage Management***

Availability Management:

Este pacote inclui as seguintes disciplinas:

- *Tivoli Application Performance Management:*
- *Tivoli Distributed Monitoring:*
- *Tivoli Global Enterprise Manager*

- *Tivoli Management Framework*
- *Tivoli Netview Performance Monitor*
- *Tivoli Remote Control*
- *Wired for Management*

Change Management:

Este pacote inclui as seguintes disciplinas:

- *Tivoli Asset Management*
- *Tivoli Decision Support*
- *Tivoli Inventory*
- *Tivoli Management Framework*
- *Tivoli Manager for Network Hardware*
- *Tivoli NetView Distribution Manager*
- *Tivoli NetView File Transfer Program*
- *Tivoli Problem Management*
- *Tivoli Remote Control*
- *Tivoli Security Management*
- *Tivoli Service Desk*
- *Tivoli Software Distribution*
- *Wired for Management*

Network Management:

Este pacote inclui as seguintes disciplinas:

- *Distributed Solutions-TCP/IP Network Management*
- *Tivoli NetView 6.0*
- *Tivoli Manager for Network Hardware*
- *Tivoli Comprehensive Network Address Translator*
- *Tivoli Network Connectivity and Performance Manager*
- *Tivoli NetView Performance Monitor for TCP/IP*

7 - Conclusões

Essa proposta de gerência contribuiu para que futuras melhoras nos níveis de serviço de rede da RedUnB sejam realizados, servindo de base para que uma posterior política de gerência seja implantado na mesma, de forma que o desempenho, a eficiência, a segurança e a confiabilidade da rede desta instituição tão conceituada dê aos alunos, professores e outros usuários desta rede condições de trabalho e pesquisa melhores.

Durante este projeto alguns problemas surgiram e a maioria deles foi superada com sucesso, não foi possível realizar todos os experimentos que haviam sido planejados no princípio deste projeto, tais como o *Software Distribution*, e configuração de *Endpoints* e *Gateways*, tais problemas surgiram principalmente pela falta de licenças de instalação dos produtos que realizariam tais operações.

O grupo considera que o trabalho realizado foi de grande valia pois espera-se que ele vá servir de base à trabalhos futuros que visem a implantação de uma política de gerência envolvendo toda a RedUnB.

Tendo realizado os testes em ambiente semelhante ao da RedUnB, apenas em uma escala menor, pôde-se perceber a força do software utilizado e sua versatilidade ao gerenciar um ambiente multiplataforma; além dos estudos realizados visando ter um maior conhecimento das necessidades de gerência da RedUnB e as soluções que o Tivoli oferece.

Bibliografia

[1] **Stallings, Willian. "SNMP, SNMPv2, SNMPv3 e RMON1 e 2" terceira edição. Publicado por Addison Wesley, Dezembro, 1998.**

[2] **Douglas E. Comer. "Internetworking with TCP/IP – Vol I – Principles, Protocols and Arquiteture" terceira edição. Publicado por Prentice Hall, 1995.**

[3] **Site da Tivoli IBM – http://www.tivoli.com/products/solutions**

[4] **Repositório de Documentos Técnicos do Grupo de Redes – UFRGS.**

[5] **Sites:** http://www.di.ufpe.br/~flash/ais98/gerrede/gerrede.html

http://www.penta.ufrgs.br/gr952/trab1/2mibII.html

http://penta.ufrgs.br/homegere.htm

http://penta.ufrgs.br/gr952/trab1/rmon.html

http://penta.ufrgs.br/home_red.htm

http://penta.ufrgs.br/homeosi.html

http://www.rnp.br/newsgen/ascii/n3.txt

http://www.rnp.br/newsgen/9712/9712index.html