

Aplicações de Gerenciamento de Redes Inteligentes

Claudia Marcia Pereira Moutinho
Av. Miruna, 327/92 - Moema
04084-001 São Paulo - SP
e-mail: claudiam@vnet.ibm.com

Michael Anthony Stanton
Departamento de Informática
Pontifícia Universidad Católica/RJ
22453 - 900 Rio de Janeiro - RJ
e-mail: michael@inf.puc-rio.br

Sumário

Apesar de padrões para gerenciamento de redes terem sido definidos há vários anos, as ferramentas de gerenciamento hoje disponíveis mostram um baixo grau de sofisticação e automação, tornando necessária a excessiva intervenção manual nos procedimentos gerenciais por parte dos administradores de rede. Neste trabalho, discutimos algumas características avançadas de ferramentas de gerenciamento, que começam a ser incorporadas em produtos ou em projetos, e que podem tornar estas ferramentas mais "inteligentes" e o trabalho do administrador de rede menos árduo.

1. Introdução

O contínuo crescimento em número e diversidade de componentes das redes de computadores tem tornado a atividade de gerenciamento de rede cada vez mais importante. Os benefícios da integração dos sistemas computacionais de uma empresa, de natureza e portes diferentes, como forma de distribuir as tarefas e compartilhar os recursos disponíveis, são hoje uma realidade. As grandes redes corporativas, que são inter-redes formadas pela interconexão de pequenas redes, assumiram um papel fundamental para os negócios das empresas que delas se utilizam. Por este motivo, estas redes requerem um sistema de gerenciamento eficiente para que as informações da corporação estejam sempre disponíveis no local e no momento onde forem requisitadas.

Desde 1986, vários grupos têm trabalhado para definir arquiteturas padronizadas ("abertas") para o gerenciamento de inter-redes heterogêneas, ou seja, inter-redes compostas por equipamentos de diferentes fabricantes. As principais arquiteturas "abertas" de gerenciamento de redes desenvolvidas até hoje são das tecnologias **TCP/IP**, também conhecida como **Internet** [ROSE91, ROSE93], e **OSI (Open Systems Interconnection, da ISO)** [CARV93, ROSE90] e estas são conhecidas mais facilmente pelos nomes dos protocolos de gerenciamento utilizados, ou seja, o Simple Network Management Protocol - **SNMP** - (versões 1 [SNMP] e 2 [SNMPv2]) de **TCP/IP**, e o Common Management Information Protocol - **CMIP** - de **OSI** [CMIP].

Muitos produtos de gerenciamento já foram desenvolvidos, que obedecem estes padrões. Por razões históricas, amplamente discutidas em [ROSE91], os primeiros produtos seguiram o padrão **SNMP**, e até hoje este é o protocolo que possui o maior número de implementações. Embora atualmente existam algumas aplicações de gerenciamento muito sofisticadas, a maioria destas aplicações possibilita apenas o monitoramento dos nós de uma rede e não possui "inteligência" para auxiliar os administradores de rede na execução de sua tarefa.

Este trabalho descreve algumas características mais avançadas que podem ser incorporadas aos aplicativos de "monitoramento" de redes existentes atualmente, transformando-os em aplicações de gerenciamento de redes "inteligentes". Algumas das características selecionadas já estão presentes em produtos comerciais ou em desenvolvimento.

O trabalho aqui apresentado faz parte de um estudo comparativo de ferramentas para o gerenciamento de redes, a ser apresentado em 1994 como dissertação de mestrado de um dos autores (CMPM) no Departamento de Informática da PUC-Rio.

2. A Arquitetura de Gerenciamento SNMP

O arquitetura de gerenciamento **SNMP**, adotada na tecnologia **TCP/IP**, supõe a existência de *estações de gerenciamento*, onde são executadas as *aplicações de gerenciamento*, e os *nós gerenciados*, que são os elementos da rede (estações, roteadores e outros equipamentos de comunicação), que desempenham funções de comunicação na operação normal da inter-rede, através dos chamados *protocolos úteis*. Estes protocolos são *instrumentados* para permitir o *monitoramento e controle* do seu funcionamento.

Este monitoramento e controle é exercido usando o paradigma de *depuração remota*, onde o nó gerenciado é concebido possuir uma coleção de *objetos*, cujos valores podem ser obtidos ou alterados através de operações remotas *get* e *set*. A coleção de objetos forma uma **Management Information Base (MIB)**, sob a guarda de um *agente* de gerenciamento. Comunicação entre este agente e um *gerente*, localizado na estação de gerenciamento, é feita utilizando o protocolo **SNMP**. Além das operações *get* e *set*, o **SNMP** inclui o *trap*, que é a maneira pela qual um agente possa tomar a iniciativa de avisar ao seu gerente da ocorrência de um evento excepcional.

A **MIB** e o protocolo **SNMP** utilizam uma restrição do padrão **ASN.1** do **OSI** [SMI, ASN.1] para a definição dos objetos e dos **Protocol Data Units (PDUs)**. Inicialmente, esta escolha foi feita para permitir uma compatibilidade com o protocolo **CMIP**, mas este objetivo não existe mais.

Como todos os padrões da tecnologia **TCP/IP**, as definições usadas no gerenciamento **SNMP** foram publicadas na série RFC (Requests For Comments). As definições originais do protocolo **SNMP**, bem como dos objetos gerenciados foram publicadas em 1989. Em 1990, foi feita uma revisão da **MIB**, que passou a se chamar de **MIB-II** [MIB-II]. Em 1993, foi

publicado um conjunto de padrões novos, chamado de **SNMPv2**, com alterações ao protocolo e extensões às definições dos objetos [SNMPv2]. O **SNMPv2** também permite estruturas hierárquicas de gerentes.

3. Aplicações de Gerenciamento Atuais

O estado de desenvolvimento da maioria das aplicações **SNMP** disponíveis atualmente é desencorajador. As aplicações ou são muito específicas, ou são genéricas, e em geral elas são muito simples.

As aplicações específicas conseguem fazer um bom gerenciamento das entidades de rede que pertencem a um único fornecedor. Elas podem exibir várias informações de gerenciamento, permitem o controle de entidades através da operação **set** e podem até auxiliar o usuário na realização de tarefas de gerenciamento. Porém, estas aplicações não interoperam bem com módulos de **MIBs** definidos por outras organizações. Se a rede possui elementos de diversos fornecedores, normalmente diferentes aplicações são necessárias para o gerenciamento de diferentes dispositivos, com a perda das vantagens de um gerenciamento de rede integrado.

Em contraste, aplicações genéricas procuram oferecer ferramentas que podem ser configuradas pelo usuário para a coleta de informações de diversos módulos de **MIBs**. Existem dois formatos deste tipo de aplicação: *browsers* e ferramentas de monitoramento configuráveis. Ambas são muito limitadas, uma vez que as informações mais importantes sobre os objetos de uma **MIB** não estão disponíveis em um formato que possa ser compreendido por uma máquina.

Um *browser* simplesmente coleta dados de entidades da rede e exibe-os para o usuário. Uma ferramenta de monitoramento configurável pode ser programada para coletar informações da **MIB** selecionadas pelo usuário, executar algumas manipulações nos dados coletados e gerar alarmes, também definidos pelo usuário, baseados nos dados coletados ou na manipulação destes dados. Estes alarmes podem ser baseados na definição de valores limites ou em asserções como um comando condicional de uma linguagem de programação.

Além disso, várias plataformas de desenvolvimento de aplicações de gerenciamento de redes que já estão disponíveis permitem ao usuário desenvolver as suas próprias aplicações ou utilizar aplicações que sejam compatíveis com esta plataforma. Estas opções resultam em sistemas similares às aplicações específicas ou genéricas, mas simples, citadas acima.

4. Geração de Alarmes

Uma parte significativa do processo de gerenciamento baseia-se na aquisição de informações sobre a rede, sendo as mais importantes aquelas relativas a erros, falhas e

outras condições excepcionais. Tais dados devem ser armazenados em forma bruta, sendo importante definir os valores aceitáveis como limiares de tolerância que, quando ultrapassados, determinam uma sinalização para pedir intervenção de um operador, ou o início de uma operação corretiva. Tais limites não são necessariamente absolutos, tais como a taxa de erros num enlace de dados, sendo necessário dispor de estatísticas de erros em função do tráfego existente. Um determinado limiar pode ser aceitável numa situação de carga leve na rede, mas intolerável numa outra situação, de carga mais intensa, no qual o número de retransmissões faria com que o tráfego total excedesse a capacidade do enlace, afetando seriamente o tempo de resposta.

Nesta seção apresentamos facilidades para a programação de alarmes incluídas nos produtos **SunNet Manager**, da Sun Microsystems [SNM], e o **NetCentral** da Cisco Systems [NETC].

4.1 O Event Report do SunNet Manager

A função **Event Report** do **SunNet Manager** possibilita que o gerente receba um alarme sempre que uma determinada condição ocorre em um nó da rede. No **SunNet Manager**, o gerente precisa definir os tipos de eventos que devem ser informados. Para criar uma requisição de **Event Report** o gerente abre a seguinte janela:

The screenshot shows the 'SunNet Manager 1.1beta Event Report: wordstwo.host11.f' window. It contains the following configuration fields and controls:

- Name:** Collisions
- Proxy System:** (empty)
- Report Frequency:** 'Interval' Until Cancelled
- Interval:** 60 (with increment/decrement arrows)
- Count:** 0 (with increment/decrement arrows)
- Options:** (empty)
- Key:** (empty)
- Restart:** False
- Send Once:** False
- Deferred Sending:** False
- Upon Completion:** Delete Request
- Attribute values:** A list box containing 'colls'.
- Attribute:** colls
- Threshold Rel1:** Greater Than
- Threshold Value1:** 100
- Threshold Rel2:** Threshold Not Set
- Threshold Value2:** (empty)
- Signal Options:** Blink Glyph & Ring Bell
- Receiver:** (empty)
- Priority:** Medium
- Buttons:** Start, Hold, Reset, Apply, Reset, Delete

At the bottom left, it says 'Interface data table (key on name)'. At the bottom right, it says 'Attribute colls . 3ded.'

Nesta janela, o gerente deve definir a frequência na qual o agente deve testar a condição que será definida. A definição de uma condição é composta por um atributo, um ou dois valores limites (*Threshold Value1* e *Threshold Value2*) e um ou dois operadores relacionais para estes valores (*Threshold Rel1* e *Threshold Rel2*) que podem ser: igual a, diferente de, menor que, menor ou igual a, maior que, maior ou igual a, alterado, incrementado por, decrementado por, incrementado por mais de, incrementado por menos de, decrementado por mais de, decrementado por menos de.

O agente que recebe uma requisição do tipo **Event Report**, testa o valor do atributo especificado na requisição. Se o valor do atributo satisfizer a condição definida pelo gerente, o agente informa o gerente da ocorrência de um evento. Senão, o agente espera um intervalo de tempo também definido pelo gerente (frequência) e testa o valor do atributo novamente. Este procedimento pode ser executado indefinidamente ou pelo número de vezes especificado na requisição.

Quando a estação de gerenciamento recebe a informação da ocorrência de um evento, o gerente é avisado através de um sinal (o ícone correspondente ao nó onde o evento ocorreu pode mudar de cor, piscar, etc.) que também é definido na requisição do evento.

Embora esta ferramenta do **SunNet Manager** seja bastante útil, ela não é muito flexível, pois o gerente só pode testar o valor de um atributo para cada requisição. Além disso, o agente reporta a ocorrência de um evento se qualquer uma das duas condições especificadas pelo gerente for satisfeita. Assim, o gerente não pode definir um evento como a ocorrência de duas condições simultaneamente.

Além destes eventos, o **SunNet Manager** também envia um alarme para o gerente quando recebe uma mensagem **SNMP** do tipo **trap**.

4.2 Alarmes do NetCentral

NetCentral é um *software* de gerenciamento de redes desenvolvido pela **Cisco Systems**, para o monitoramento, planejamento e análise de redes complexas. Para atingir este objetivo, o **NetCentral** mantém um mapa da rede criado pelo usuário e um banco de dados relacional, com informações sobre os nós e interfaces da rede, altamente integrado baseado no banco de dados relacional **Sybase**. O aplicativo **NetCentral** implementa o protocolo **SNMP**.

Ao contrário do **SunNet Manager**, o **NetCentral** checa cada interface definida no mapa da rede periodicamente e atribui uma cor aos nós e às interfaces de acordo com o seu estado operacional. A cor verde indica que um nó ou interface está operando

satisfatoriamente. A cor vermelha indica que um nó ou interface não respondeu às últimas três mensagens de *polling*. A cor azul indica que o estado operacional de um nó ou interface mudou para *up*. A cor cinza indica que uma interface está no estado de teste. A cor indigo, indica que um nó ou interface não é reconhecido, ou seja, o dispositivo respondeu às últimas mensagens com um estado operacional desconhecido pelo gerente. A cor laranja indica que um nó não respondeu à mensagem inicial do gerente. A cor amarelo indica que a configuração de um dispositivo não está consistente.

Embora o NetCentral envie alarmes para o gerente indicando a ocorrência de eventos importantes nos nós da rede, este aplicativo não permite que o usuário defina os seus próprios eventos.

5. Auto-Topologia

Auto-topologia é normalmente citada por usuários e fornecedores como uma das mais importantes ferramentas de gerenciamento de redes. A idéia da auto-topologia é que uma estação de gerenciamento possa reconhecer automaticamente a configuração de toda uma rede e exibí-la em um formato gráfico ideal para o gerenciamento. Embora muitos aspectos desta idéia já tenham sido implementados, estas ferramentas ainda não podem ser consideradas "inteligentes".

Um gráfico ou mapa, que represente logicamente uma rede, é uma ferramenta essencial para os administradores de rede. Porém, a construção destes mapas requer uma grande quantidade de informações sobre a rede em questão.

Os três aspectos que uma ferramenta de auto-topologia deve explorar são: reconhecimento da topologia, reconhecimentos dos nós e desenho do mapa.

5.1 Reconhecimento da Topologia

Nesta fase, o sistema de gerenciamento deve encontrar todos os dispositivos e sub-redes que compõem uma rede e as relações que existem entre eles. Esta topologia é formada por redes locais, redes geograficamente distribuídas, roteadores, *bridges*, *hubs*, repetidores e outros dispositivos que podem ser conectados à uma rede.

As informações das tabelas de roteamento podem ser usadas para a construção de um primeiro mapa. Infelizmente, este procedimento não reconhece dispositivos transparentes como *bridges*, *hubs* e repetidores e os administradores de rede precisam enxergar estes dispositivos para a solução de problemas da rede. A padronização e implementação de MIBs para *bridges* e repetidores possibilitará a recuperação de informações sobre estes dispositivos e sua inclusão no mapa da rede.

5.2 Reconhecimento dos Nós

Após a descoberta da topologia da rede, um sistema de gerenciamento precisa reconhecer todos os nós ligados à rede. Uma das estratégias utilizadas é a transmissão de um pacote **SNMP broadcast** que solicita à todos os nós que forneçam suas identificações. Este procedimento não deve ser utilizado em redes com um grande número de dispositivos devido ao grande número de respostas que podem congestionar a rede. Também não existe nenhuma garantia de que todos os dispositivos da rede possuam um agente **SNMP** e de que este agente responderá à solicitação.

Um outro método para o reconhecimento dos nós de uma rede é a transmissão de pacotes **echo** do **ICMP** [COME91] para todos os endereços da rede em sequência. Este é conhecido como o método de Monte Carlo. Este método consome muito tempo e recursos da rede. Em uma rede **IP** da classe *A*, com 4 milhões de possíveis endereços, a aplicação deste método é inviável, mas até mesmo em uma rede **IP** da classe *B*, este processo geraria no mínimo 65.000 pacotes que consumiriam um tempo excessivo (embora, provavelmente os 65.000 pacotes **ARP broadcast** derrubariam a rede, se a rede não estivesse dividida em subredes).

Um terceiro método consiste na utilização do potencial de um dispositivo **RMON (Remote Network Monitoring)** [RMON, WALD92c] para o reconhecimento dos nós da rede através do endereço fonte dos pacotes que estes enviam.

A **MIB RMON (Remote Network Monitoring MIB)** define os objetos necessários para gerenciar completamente um dispositivo que monitore **LANs Ethernet** via **SNMP** e está estruturada de forma que outros tipos de **LANs**, como *Token Ring*, possam ser adicionadas. Os objetos incluem os seguintes grupos: estatísticas baseadas no tipo de **LAN** que está sendo monitorada, histórico de amostras estatísticas, objetos para geração de alarmes, tabelas de estatísticas baseadas nos endereços fonte e destino dos pacotes que trafegam na **LAN**, captura de pacotes e geração de eventos.

A **MIB RMON** foi amplamente implementada desde a sua publicação. Um sistema simples de análise de redes baseado na **MIB RMON** consiste de vários coletores **RMON** e uma estação de gerenciamento **RMON**. Os coletores coletam pacotes e estatísticas através da leitura de todos os dados que trafegam na rede, como um analisador de protocolos. Estes dados são formatados de acordo com o padrão **RMON** e podem ser transmitidos pela estação de gerenciamento **RMON** para uma estação de gerenciamento **SNMP**. A estação de gerenciamento **RMON** formata os dados e os apresenta para o administrador da rede da forma mais eficiente possível, normalmente

utilizando uma interface gráfica. Como os coletores **RMON** executam todas as suas operações de entrada e saída através da rede não há necessidade de utilização de monitores e sistemas de discos sofisticados. Somente a estação de gerenciamento precisa possuir uma boa interface gráfica com um sistema de janelas. Normalmente, pode-se carregar o software de gerenciamento **RMON** na mesma estação de gerenciamento **SNMP**, para reduzir os custos.

A utilização desta solução torna o gerenciamento mais eficiente, pois o gerente da rede pode visualizar várias redes em diferentes janelas simultaneamente. Como os coletores **RMON** podem realizar várias tarefas ao mesmo tempo, o gerente da rede pode executar diferentes funções de diagnóstico simultaneamente, sem que a coleta de estatísticas seja interrompida. Se existirem mais de um administrador de rede, eles podem acessar o coletor **RMON** ao mesmo tempo sem afetarem um ao outro. Em contraste, muitos analisadores de protocolos são baseados em plataformas monotarefa e monousuário.

Uma das principais vantagens das aplicações **RMON** é que como o **RMON** é um padrão aberto, os administradores de redes podem escolher os melhores produtos para um ambiente específico, independente da plataforma dos coletores **RMON**. Isto evita que o usuário fique preso à uma única aplicação e permite que os fabricantes produzam softwares para diferentes aplicações, desde o monitoramento de falhas e performance até o gerenciamento de configuração. Outra vantagem das aplicações **RMON** é que estas podem facilmente integradas com outras ferramentas de gerenciamento de redes. O acesso aos dados **RMON** a partir dos produtos de gerenciamento **SNMP** disponíveis atualmente é trivial.

O custo de um **RMON** dedicado pode significar uma barreira, mas na medida em que roteadores, *hubs* e outros dispositivos implementarem a **MIB RMON**, este custo será amplamente reduzido.

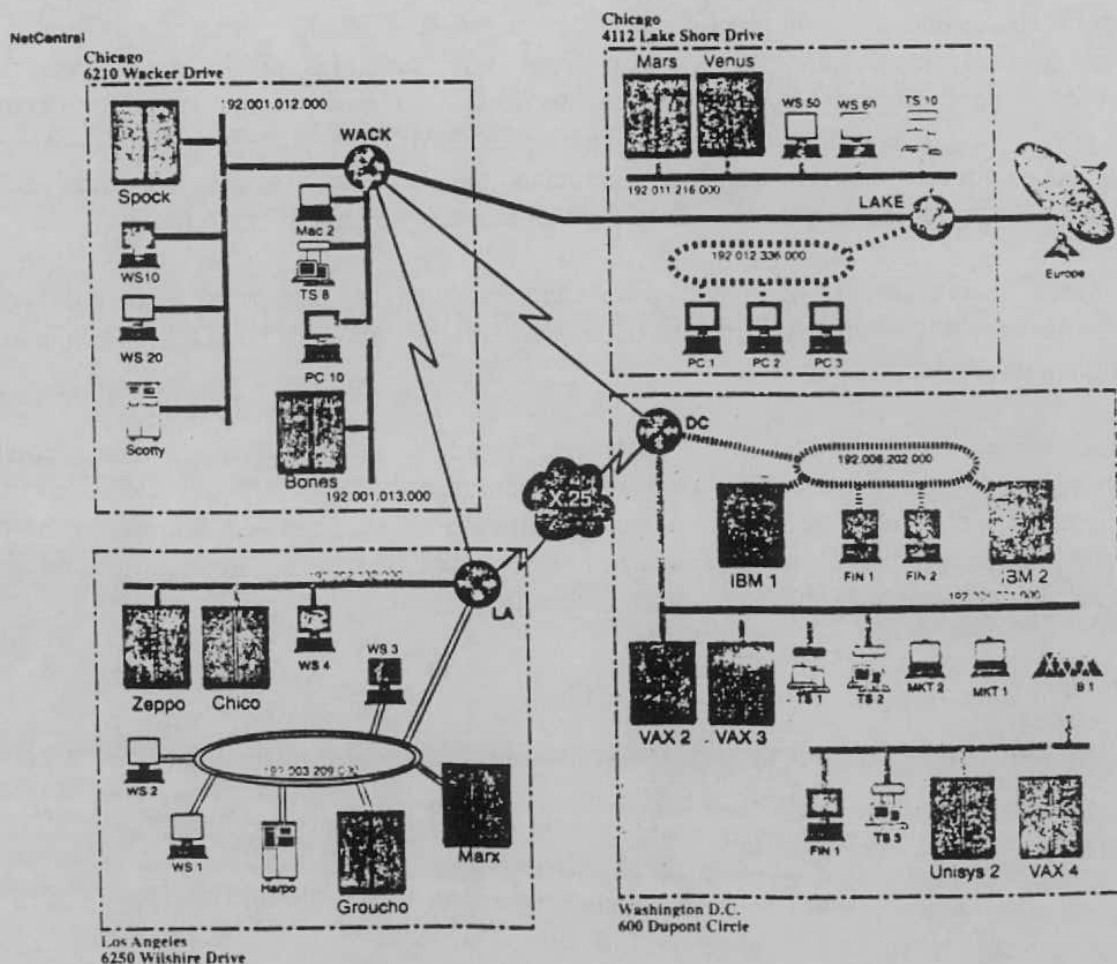
Para que estes mecanismos sejam eficientes, é necessário que todos os dados coletados sobre a topologia e os nós de uma rede sejam armazenados em um banco de dados. Informações de configuração dos nós também podem ser coletados e armazenados no banco de dados. Um bom sistema também deve possuir facilidades para a alteração automática do banco de dados quando alterações na rede forem detectadas.

5.3 Desenho do Mapa

O próximo passo é a construção automática de uma representação gráfica da rede, de

acordo com a topologia e os nós que foram encontrados. Normalmente, o resultado desta operação é um mapa contendo centenas ou milhares de ícones que não apresentam nenhuma estrutura. Isto ocorre porque as informações sobre a topologia da rede não são suficientes para que esta seja representada logicamente da mesma forma como é vista por uma pessoa. Um administrador vê a rede em termos geográficos ou administrativos, enquanto o sistema de gerenciamento só consegue descobrir as conexões da rede. O sistema não possui informações suficientes para saber que as LANs de um mesmo prédio devem ser desenhadas lado a lado. Ao contrário, elas devem aparecer espalhadas pelo mapa. É função do administrador alterar o mapa para que os ícones sejam dispostos com alguma lógica. Em um sistema de gerenciamento eficiente, esta tarefa deve ser resumida na movimentação de ícones para que estes sejam pOSicionados corretamente.

A figura abaixo ilustra o mapa de uma rede criado para o NetCentral:



O maior problema encontrado na auto-topologia, é que o sistema de gerenciamento não sabe identificar quais os dispositivos que interessam ao administrador. Todos os dispositivos encontrados são exibidos, e cabe ao administrador eliminar aqueles que não lhe interessam. Em alguns casos, o número de elementos que devem ser eliminados é tão grande que o administrador opta por fazer o desenho da rede e não utilizar a ferramenta de auto-topologia. O administrador também pode definir alguns critérios para que um elemento seja escolhido e incluído no mapa pelo sistema de gerenciamento. Alguns destes critérios poderiam ser:

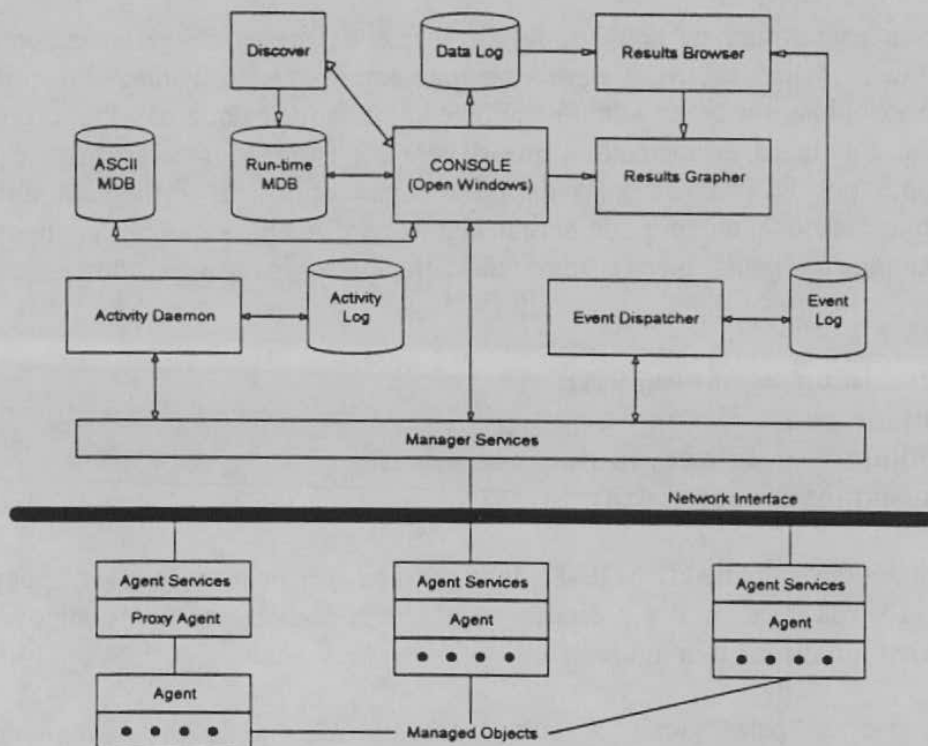
- `ipForwarding = gateway(1)`
- `ifNumber > 2`
- `sysObjectID = bridge, router, server, etc.`
- `ipOutRequests/sec > 100`

Todos estes objetos fazem parte da **MIB-II**, mas outras heurísticas podem ser aplicadas a outras **MIBs**. Na verdade, a simples presença ou ausência dos objetos que compõem uma **MIB** podem tornar um dispositivo interessante.

Existem várias razões pelas quais a auto-topologia não pode ser completamente automatizada. Sistemas automatizados não podem obter informações suficientes sobre a topologia da rede, não sabem distribuir os elementos de uma rede em uma mapa de forma lógica e não podem decidir quais são os dispositivos considerados importantes pelo administrador da rede. Alguns algoritmos e mecanismos podem ser desenvolvidos para tornar estas ferramentas mais eficientes, mas a participação do administrador é indispensável no processo de representação da rede.

5.4 A Ferramenta Discover do SunNet Manager

O **SunNet Manager (SNM)** é uma plataforma para o gerenciamento de redes distribuídas, que utilizam o modelo gerente-agente descrito no **OSI** e a família de protocolos **TCP/IP**. O gerente é um processo inicializado pelo usuário. O agente é um processo que faz acesso ao objeto gerenciado (o elemento da rede que se deseja gerenciar) e coleta dados no lugar do gerente. A figura na página seguinte apresenta o diagrama funcional do **SunNet Manager**.



A *Console* é a aplicação de gerenciamento central do **SunNet Manager**, onde o usuário inicia as tarefas de gerenciamento e onde os dados coletados são exibidos.

A *Console* do **SunNet Manager** representa graficamente os elementos de uma rede e coordena a atividade dos agentes para a coleta de informações sobre a rede. A *Console* utiliza um banco de dados (MDB) composto por uma coleção de arquivos de estruturas e arquivos de instâncias (*structure* e *instance files*) que descrevem a rede. Os arquivos de estruturas possuem a extensão **.schema* e descrevem a estrutura dos elementos encontrados em uma rede. As instâncias destes elementos e suas relações com outros elementos são armazenados nos arquivos de instâncias, que possuem a extensão **.db*.

Discover é um utilitário que busca os elementos da rede gerenciada pelo **SunNet Manager** que possuam um endereço IP e automaticamente adiciona estes elementos ao banco de dados dinâmico (*run_time database*) utilizado pela *Console*. Este utilitário pode ser inicializado através da *Console* ou diretamente na linha de comandos da estação de gerenciamento. Se inicializado através da *Console*, o utilitário *Discover* utiliza seus parâmetros *default* e opera da seguinte forma:

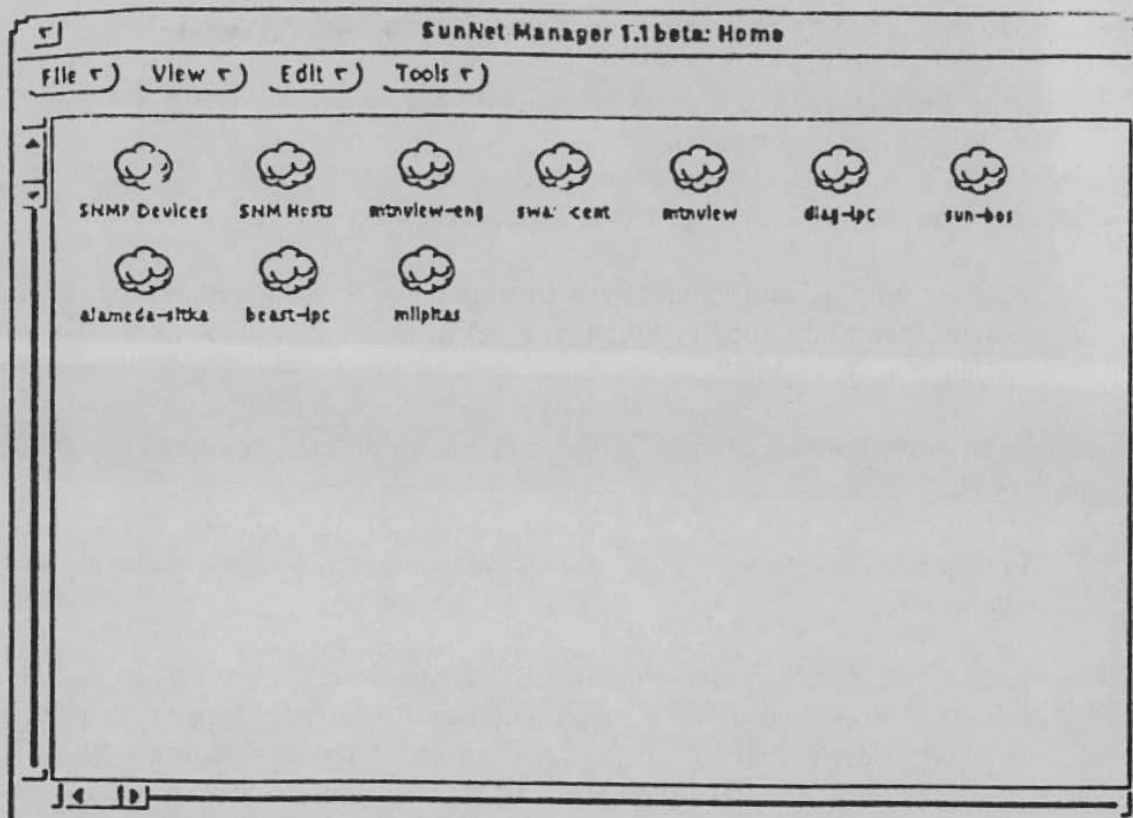
- Consulta a tabela de roteamento da estação gerenciadora em busca de informações sobre conexões com outras redes;

- Envia um "ping" para todos os endereços IP encontrados na tabela;
- Checa a presença do agente SNMP e de outros agentes do SunNet Manager nos dispositivos que responderam ao "ping";
- Inclui estes dispositivos no banco de dados dinâmico;
- Cria duas novas visões na visão Home: uma que contém todos os dispositivos que possuam agentes do SunNet Manager e outra que contém todos os dispositivos que possuam o agente SNMP.

O utilitário *Discover* também pode ser inicializado a partir da linha de comandos, em um dos seguintes modos:

- Modo *default*, envia um "ping" para cada endereço IP encontrado na tabela de roteamento.
- Modo *watch*, o utilitário monitora todo o tráfego na rede, checando os endereços IP da fonte e do destino de cada pacote que transita na rede. Este modo possibilita a descoberta de sub-redes e dispositivos em redes remotas. Neste modo não são enviados "pings". O utilitário continua a operar enquanto encontra novos elementos na rede. Somente quando nenhum elemento novo é encontrado durante um intervalo de 15 minutos, o utilitário pergunta ao usuário se deve prosseguir com a busca. Este modo de operação consome muito tempo de processamento.

Quando a *Console* é inicializada, o SunNet Manager exibe a visão *Home*. Esta visão contém o nível mais alto da hierarquia criada para representar graficamente o domínio de gerenciamento. O utilitário *Discover* adiciona visões da rede em ordem hierárquica a partir da visão *Home*, na medida em que novos elementos são encontrados. Cada rede é representada na visão *Home* através de um ícone em forma de nuvem, como ilustra a figura na página seguinte:



Cada visão pode conter outros elementos, como estações e sub-redes. Se uma sub-rede de uma rede remota for encontrada, o utilitário *Discover* pode ser executado sobre esta sub-rede para que seus elementos sejam identificados.

Além do utilitário *Discover*, o usuário também pode utilizar ferramentas de edição gráfica na construção da representação de uma rede.

6. Compiladores de MIBs

Além da descrição dos objetos gerenciados e suas relações, uma **MIB** contém informações detalhadas sobre cada objeto, como por exemplo, o tipo de acesso a um objeto, um valor *default* razoável para um objeto e um conjunto de valores que um objeto pode assumir. Estas informações possuem um valor inestimável para os fornecedores de softwares para agentes e gerentes pois elas permitem que vários fornecedores utilizem um mesmo conjunto de informações de gerenciamento, obedecendo um conjunto de características padrão. Além disso, uma **MIB** pode ser compilada por um compilador de **MIBs**, de forma que as informações presentes na **MIB** estejam disponíveis para aplicações como **MIB browsers** e *graphers*. Estas aplicações são consideradas aplicações genéricas. São aplicações simples que obtêm toda a sua capacidade de gerenciamento através da análise de uma **MIB**, sem qualquer intervenção humana [WALD92b].

Além de checar a sintaxe de uma **MIB**, um compilador de **MIBs** pode gerar automaticamente as estruturas de dados e o código necessários para que um agente implemente uma determinada **MIB**. Um compilador de **MIBs** também pode fazer com que as informações sobre os objetos gerenciados de **MIBs** proprietárias ou de novas **MIBs** que sejam padronizadas estejam disponíveis para uma aplicação de gerenciamento existente.

A entrada para um compilador de **MIBs** é uma coleção de módulos de **MIBs** escritos em um sub-conjunto da linguagem **ASN.1**. Estes módulos contêm definições de objetos gerenciados que correspondem às informações sobre os dispositivos da rede que podem ser manipulados através do protocolo **SNMP**. Os compiladores de **MIBs** podem gerar várias representações das definições dos objetos gerenciados contidos nas **MIBs** usadas como entrada. Estas representações podem ser processadas mais facilmente pelos agentes e aplicações de gerenciamento do que a representação **ASN.1**.

Algumas destas representações são declarações de estruturas de dados em linguagens de programação de alto nível, como **C**, que podem ser compiladas e ligadas em uma aplicação de gerenciamento ou agente. Outras são arquivos de dados contendo representações das definições dos objetos gerenciados que podem ser lidas para a memória por uma aplicação de gerenciamento ou agente em tempo de execução. Em alguns casos, o compilador de **MIBs** gera um código de saída que auxilia na implementação das **MIBs** de entrada. Por exemplo, um compilador de **MIBs** pode gerar esqueletos de rotinas para a recuperação ou alteração do valor de um objeto gerenciado, ou rotinas para a geração de **Trap-PDUs** específicas.

A habilidade de reconhecer as descrições presentes em uma **MIB** mecanicamente é muito atraente, principalmente para os fabricantes de aplicações genéricas, pois estas podem cobrir uma grande variedade de agentes e **MIBs**. Com o grande número de **MIBs** padronizadas e **MIBs** proprietárias disponíveis atualmente, os compiladores de **MIBs** reduzem o esforço dos fornecedores para manterem suas aplicações atualizadas.

Assim, muitos esforços estão concentrados em facilitar a forma pela qual diversas **MIBs** possam ser compiladas em cada produto de gerenciamento. Tentativas estão sendo feitas para que mais informações possam ser reconhecidas dinamicamente possibilitando que uma aplicação gerencie eficientemente um dispositivo completamente desconhecido para o fornecedor do produto de gerenciamento e para o usuário. Embora os compiladores de **MIB** já tenham provado sua utilidade, o gerenciamento de redes inteligente não poderá ser alcançado pela simples utilização desta tecnologia.

Infelizmente, a informação mais importante da **MIB**, ou seja, o texto que descreve detalhadamente um objeto, não pode ser compreendido por um compilador de **MIB** (com a tecnologia disponível atualmente). Por exemplo, um compilador pode ler a descrição de um objeto da **MIB-II** e aprender que este objeto é um inteiro que pode assumir os valores um e dois, o valor deste objeto pode ser lido e alterado e a implementação deste objeto é obrigatória. Mas somente um ser humano pode compreender a partir da descrição em linguagem natural do objeto *ipForwarding*, que se o valor deste objeto for igual a um então

o sistema descrito por este objeto está atuando como um *gateway*, senão o sistema é apenas um nó da rede. Além disso, existem informações conhecidas por um administrador de rede experiente que não são descritas em nenhuma **MIB**, como por exemplo, o fato de que em algumas circunstâncias, pode ser perigoso para um sistema atuar como um *gateway*. Para a construção de um sistema de gerenciamento de redes inteligente, as aplicações devem conter todo este conhecimento. Como este conhecimento não pode ser fornecido no formato da **MIB**, os fornecedores de aplicações de gerenciamento devem desenvolver outras formas de incluir estes conhecimentos em suas aplicações.

Sem esta inteligência, muitas aplicações genéricas ficam limitadas à coleta, formatação e exibição das informações de gerenciamento. Estas informações são apresentadas para o usuário, que aplica sua inteligência humana para analisá-las. Esta carga só poderá ser retirada das mãos do administrador de rede se as aplicações se tornarem mais inteligentes. Sem esta inteligência será muito difícil que uma aplicação possa coletar informações suficientes de uma **MIB** desconhecida para gerenciar eficientemente um dispositivo desconhecido.

7. Interface com o Usuário

A indústria de gerenciamento de redes reconhece que, embora os protocolos e **MIBs** de gerenciamento tenham progredido muito, as aplicações de gerenciamento ainda deixam muito a desejar. Alguns membros da comunidade de padronização acreditam que a solução é a inclusão de novos tipos de informação ao formato da **MIB** padrão. As informações orientadas para a aplicação deveriam ser adicionadas às definições dos objetos da **MIB**. Além das informações existentes, como tipo e descrição, as informações orientadas para a aplicação consistem de *labels* para tabelas ou gráficos, informações para formatação, valores-limiares (*thresholds*), texto de ajuda (*help*), e outras. Estas informações seriam lidas pela estação de gerenciamento, que as utilizaria para produzir uma melhor interface com o usuário para os objetos de gerenciamento **SNMP** [WALD93].

Como exemplo, o objeto *etherStatsCollisions* da MIB RMON seria representada da seguinte forma:

```
etherStatsCollisions APPLICATION-INFO
    -- suitable for column header
SHORT-LABEL      "Collisions"
LONG-LABEL       "Ethernet CSMA/CD Collisions"
PRINT-FORMAT     "decimal"

USEFUL-STAT      "per etherStatsPackets"
    -- i. e., also useful per second
USEFUL STAT      "per sysUpTime * 100"

    -- absolute value isn't interesting
THRESHOLD        ".03 per etherStatsPackets"

    -- polling is useful
VOLATILITY volatile

ICON-BIT-MAP
    "120A9847E48C92A001F28437B5900E12
    8A7712C890203D487565D6080C4E7478
    3B921C983A782C08314E78213C019C28
    3E774C182A001B2438A7565D6080437B"

HELP-TEXT        "us-english"
    "A collision is an event on an Ethernet network that is a part of
    everyday operation, but when excessive, can signal that the
    network is overloaded or is too long (especially when the average
    packet size is small). To avoid excessive collisions, use Token
    Ring."
```

7.1. A Utilização das Informações Orientadas para a Aplicação

As informações orientadas para a aplicação são lidas pela estação de gerenciamento. Algumas destas informações podem controlar como a aplicação vai exibir os dados coletados através do protocolo **SNMP**, enquanto outras informações podem ser exibidas para ajudar o usuário a entender o significado destes dados. No entanto, as aplicações que utilizam estas novas facilidades não podem ser consideradas "inteligentes", pois não são capazes de fazer recomendações para o usuário com base nos dados recebidos através do protocolo **SNMP**. Esta inteligência, tão desejada pelos gerentes de redes, só poderá ser fornecida adequadamente pelas aplicações desenvolvidas para uma MIB particular.

Como as **MIBs** proprietárias multiplicam o número de **MIBs** padrão por um fator de dez, é muito difícil para os fabricantes de aplicações de gerenciamento de rede suportar todas as **MIBs** proprietárias. Para estas **MIBs**, as informações adicionais orientadas para a aplicação são muito importantes.

7.2. Como Divulgar as Informações da Interface com o Usuário

Foi sugerido que as informações orientadas para a aplicação fossem adicionadas ao formato padrão das **MIBs**, pela extensão da macro **OBJECT-TYPE**. Por várias razões, este não é o lugar ideal para definir estas informações. Uma **MIB** é um contrato entre os projetistas de agentes e os projetistas de aplicações de gerenciamento, realizado por uma entidade de padronização ou por um fabricante. Uma **MIB** descreve os objetos de gerenciamento para garantir que a implementação dos agentes e gerentes utilizem as mesmas definições. Os autores de **MIBs** em geral e as entidades de padronização em particular não podem assumir a responsabilidade adicional de projetar a interface para os usuários. Os grupos que trabalham na definição de novas **MIBs** possuem muitos outros detalhes para se preocupar.

Além disso, os grupos de padronização são internacionais. Claramente, não seria apropriado que um padrão definisse uma interface em inglês. Por outro lado, seria igualmente inadequado que um destes grupos de trabalho gaste um tempo enorme traduzindo as informações de uma interface em 20 línguas diferentes, garantindo que todos os *labels* possam ser exibidos em uma tela de 80 colunas!

7.3. O Caminho Certo

A forma correta de adicionar estas informações nas aplicações seria a criação de uma nova macro, ligada ao objeto da **MIB** ao qual se refere. Arquivos de macros **APPLICATION-INFO** poderiam ser fornecidos pelos vendedores de agentes para definir partes da interface para o usuário das estações de gerenciamento. Esta estratégia seria mais apropriada para **MIBs** proprietárias, que, de outra forma, não poderia esperar suporte à aplicação de todas as estações de gerenciamento de redes.

É importante que as aplicações de gerenciamento se tornem mais eficientes. Porém, a adição de novas funções a estas aplicações deve se concentrar nas áreas corretas. As informações sobre a interface com o usuário não pertencem às **MIBs** padrão, mas serão muito úteis se estiverem disponíveis a partir de outras fontes.

8. Conclusão

Embora a arquitetura de gerenciamento **SNMP** tenha possibilitado o monitoramento dos nós gerenciados, ela não provocou a produção de aplicações de gerenciamento "inteligentes". A principal causa desta situação é que as informações de gerenciamento foram definidas em um nível muito baixo. Foram produzidas diversas **MIBs** contendo vários objetos gerenciados, mas não foram produzidos documentos descrevendo como estes objetos podem ser usados no gerenciamento eficiente de uma rede. O resultado disto é que a maioria das aplicações de gerenciamento é de *browsers*, que não possuem nenhuma inteligência. Por outro lado, alguns dos produtos desenvolvidos nos últimos anos possuem características mais inteligentes e portanto úteis do que estas ferramentas mais simples. Embora que algumas das ferramentas aqui descritas sejam bem mais interessantes do que a média dos produtos disponíveis, entendemos que ainda existe bastante espaço para o aprimoramento de ferramentas e, em particular, para a automação de boa parte do trabalho de gerenciamento de redes.

9. Referências Bibliográficas

- [ASN.1] *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*, ISO-8824, dezembro de 1987.
- [CARV93] Carvalho, T.C.M.B., (ed.), *Gerenciamento de Redes: Uma Abordagem de Sistemas Abertos*, Makron Books, 1993.
- [CMIP] *Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol Specification*, ISO-9596, maio de 1990.
- [COME91] Comer, D.E., *Internetworking with TCP/IP, Vol. I: Principles, Protocols and Architecture (2a ed.)*, Prentice-Hall, 1991.
- [MIB] McCloughrie, K., "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II", RFC1156, maio de 1990.
- [MIB-II] Rose, M.T., "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II", RFC1158, maio de 1990.
- [NETC] Cisco Systems Inc., *NetCentral Installation and User Guide*, Software Release 1.3, março de 1992.
- [RMON] Waldbusser, S., "Remote Network Monitoring Management Information Base", RFC 1271, novembro de 1992
- [ROSE90] Rose, M.T., *The Open Book: A Practical Perspective on OSI*, Prentice-Hall, 1990.
- [ROSE91] Rose, M.T., *The Simple Book*, Prentice-Hall, 1991.
- [ROSE93] Rose, M.T., *The Simple Book (2a. ed.)*, McGraw-Hill, 1993,
- [SMI] Rose, M.T., "Structure and Identification of Management Information for TCP/IP-based Internets", RFC1155, maio de 1990.
- [SNM] Sun Microsystems Inc., *SunNet Manager*, Release 1.1, outubro de 1990.
- [SNMP] Case, J.D., "A Simple Network Management Protocol", RFC 1157, maio de 1990.
- [SNMPv2] Case, J.D., "Introduction to version 2 of the Internet-standard Network Management Framework", RFC 1441, abril de 1993.

- [WALD92a] Waldbusser, S., "Exposing the Myths about Autotopology", *The Simple Times*, 1 (1), abril de 1992.
- [WALD92b] Waldbusser, "Today's MIB Compilers - Too Much of a Good Thing?", *The Simple Times*, 1 (2), junho de 1992.
- [WALD92c] Waldbusser, S., "How RMON Stands to Replace the Traditional Protocol Analyser", *The Simple Times*, 1 (5), dezembro de 1992.
- [WALD93] Waldbusser, S., "Should user interface information be standardized?", *The Simple Times*, 2 (1), janeiro de 1993.