Um Sistema Especialista para Gerência Pró-ativa Remota

Esmilda Sáenz Artola* Liane M.R. Tarouco**

Curso de Pós-Graduação em Ciência da Computação
Instituto de Informática, UFRGS
Caixa postal 15064
91501-970 Porto Alegre- RS[1]

RESUMO

A tendência, com o surgimento de novas aplicações nas redes atuais e com muito mais usuários fazendo uso delas, é que o desempenho dessas redes degrade se não forem gerenciadas de uma forma mais inteligente que a convencionalmente usada. Ou seja, as redes precisam serem gerenciadas de forma pró-ativa. Podendo com uma monitoração contínua, fazer uma análise de tendências, evitando dessa maneira que determinados problemas na rede aconteçam, ou que o impacto deles, se chegarem à acontecer, seja o menos prejudicial possível para o desempenho da rede.

O principal objetivo deste trabalho é propor uma solução para gerência pró-ativa de redes, auxiliando-se de ferramentas como sistemas especialistas e monitores remotos. Para atingir esse objetivo, no protótipo proposto são usados um monitor remoto que implementa a RMON MIB, a qual apresenta características ideais para fazer este tipo de gerenciamento; e um conjunto de regras que constituem o cerne do sistema especialista orientado à gerência pró-ativa que faz parte do assim denominado Olho Vivo.

1 Introdução

A gerência de redes de computadores é uma aplicação muito importante no atual contexto de redes, pois surgem constantemente novos dispositivos e aplicações distribuídas que tornam a rede complexa. Devido à diversidade desses recursos, torna-se necessária a gerência de redes.

* Mestranda do CPGCC (Curso de Pós-Graduação em Ciência da Computação) da UFRGS.
E-mal: esmilda@inf.ufrgs.br

** Professora orientadora UFRGS/CPGCC. E-mail: liane@penta.ufrgs.br

O modelo de referência ISO/OSI subdividiu a gerência de redes em cinco grandes áreas funcionais: Gerência de Falhas, de Configuração, de Desempenho, de Segurança e de Contabilização[SMA 88]. Essas funções têm sido comumente aplicadas para desenvolver uma gerência reativa de redes, ou seja, estão sendo utilizadas na detecção de problemas nas redes e na busca de uma solução quando esses problemas ocorrem.

Como foi mencionado anteriormente a tecnologia de redes está evoluindo a cada dia e o seu uso está-se expandindo em grandes proporções. Nesse sentido, a gerência reativa torna-se muito limitada. É o momento de começar adicionar inteligência nos sistemas de gerência de redes para alcançar uma ação pró-ativa nesse ambiente[JAN 93].

A gerência pró-ativa de redes significa a capacidade de antecipar problemas que provocarão determinado impacto na redes, principalmente em seu desempenho. Alem disso, a gerência pró-ativa deve ter a capacidade de evitar a ocorrência desses problemas ou que seu impacto seja o menos prejudicial possível.

Existem alguns elementos que, em conjunto com os atuais protocolos e plataformas de gerenciamento, contribuem para que a gerência pró-ativa de redes de computadores seja mais confiável. Entre tais elementos estão: os sistemas especialistas, os monitores remotos, os agentes procuradores e os programas de simulação.

Este trabalho tem como principal objetivo descrever uma solução para gerência pró-ativa de redes de computadores. Para atingir esse objetivo é apresentado um protótipo incluindo alguns dos elementos mencionados anteriormente.

No protótipo é utilizada a RMON MIB, [WAL 95] a qual foi escolhida por compreender a definição de objetos com os quais é possível ser feita uma análise de tendências, fundamentalmente para possibilitar a previsão de ocorrência de problemas numa rede. Esse protótipo também engloba um conjunto de regras que constituem o cerne do sistema especialista no agente procurador.

Na seção 2 deste trabalho, apresenta-se uma descrição da RMON MIB. Na seção 3, são descritos algumas formas de representação do conhecimento para sistemas especialistas e as suas principais características ao serem aplicados na gerência de redes de computadores. A seção 4, apresenta uma descrição de problemas que direta ou indiretamente afetam o desempenho de uma rede. Na seção 5 apresenta-se o protótipo do Olho Vivo como um todo. Finalmente na conclusão são apresentadas as principais dificuldades encontradas no desenvolvimento deste trabalho, assim como são indicados elementos que no futuro podem ser adicionados ao sistema para o seu aperfeiçoamento.

2 RMON MIB

A RMON MIB tem-se destacado pelas características de monitoração remota, característica que é muito importante para implementar a gerência pró-ativa de redes.

A gerência de redes remotas através de agentes remotos (por exemplo, agentes implementam a RMON MIB) possui cinco funções[WAL 91]:

- operações *offline*: são as operações nas quais uma estação de gerenciamento não necessita estar em contato direto com seus dispositivos de monitoração remotos. Essa função na RMON MIB permite que os agentes sejam configurados para realizar diagnósticos e coletar estatísticas continuamente, mesmo que a comunicação entre a estação de gerenciamento não seja possível ou não seja eficiente.

- monitoração pró-ativa: os recursos disponíveis nos monitores são potencialmente úteis para continuamente executar diagnósticos e manter *logs* do desempenho da rede. Essas informações são importantes para desenvolver a função ***baseline***. A função ***baseline***, refere-se ao fato de manter um histórico da operação normal de uma rede por um tempo estendido, com o objetivo dessas informações posteriormente serem analisadas para identificar problemas potenciais numa rede, garantindo assim, o melhor nível de desempenho aos usuários, o qual é um dos principais objetivos da gerência pró-ativa.

- detecção e registro de problemas: o monitor remoto pode ser configurado para o reconhecimento de determinadas condições, realizando constantes averiguações.

- valorização dos dados coletados: devido a que um monitor remoto é dedicado exclusivamente para funções de gerência e como ele está localizado diretamente na parte da rede que está sendo monitorada, o monitor remoto determina um valor significativo aos dados por ele coletados.

A RMON está definida no RFC 1757 [WAL 95], o qual é atualmente um documento *internet standard*, tendo tornado obsoleto o RFC 1271 [WAL 91]. As definições no RFC 1757 são principalmente para redes *ethernet*. A RMON MIB para redes *Token Ring* é definida no RFC 1513[WAL 93].

2.1 Grupos da RMON MIB

Grupo *Statistics*: esse grupo contém estatísticas medidas pelo monitor para cada interface monitorada no dispositivo. As estatísticas incluem número de pacotes, octetos, pacotes *broadcast*, pacotes *multicast* e quantidade de colisões no segmento local, bem como o número de ocorrências de pacotes perdidos pelo agente. Cada estatística é mantida em seu próprio contador cumulativo de 32 bits. Atualmente, esse grupo consiste só da tabela *etherStatsTable*. Os grupos *Token Ring Mac-Layer Statistics* e *Token Ring Promiscuos Statistics*, definidos no RFC 1513, mantêm as estatísticas para redes *Token Ring*. O grupo é composto por 21 elementos, todos formando parte da tabela *etherStatsTable*.

Grupo *History*: esse grupo registra amostras estatísticas periodicamente e as armazena para uma posterior recuperação. O grupo consiste de duas tabelas: *historyControlTable* e *etherHistoryTable*, com 7 e 15 elementos respectivamente. A tabela *historyContolTable* armazena entradas de configuração, onde cada uma define uma interface, um período de *polling* e outros parâmetros. Cada entrada da tabela *etherHistoryTable* mantém amostras históricas de estatísticas *ethernet* de uma interface *ethernet* em particular.

Grupo *Host*: esse grupo contém estatísticas associadas a cada *host* descoberto na rede. Os endereços fonte e destino desses *hosts* são mantidos numa lista e obtidos dos pacotes bons que foram promiscuamente recebidos pela interface. O grupo consiste de três tabelas: (a) *hostControlTable*, que é uma lista de parâmetros que estabelecem o descobrimento de hosts na interface e a coleção de estatísticas em relação a esse *host*; (b) *hostTable*, que mantém uma

Grupo *Capture*: esse grupo permite que os pacotes sejam capturados sobre a correspondência de um filtro e que o sistema de gerenciamento crie múltiplos *buffers* de captura, controlando se o *buffer* de monitoração (*trace buffer*) continua ou interrompe a captura de pacotes quando estiver cheio. Dependendo do agente, o usuário pode expandir ou contrair o tamanho do *buffer*, ajustando-o às necessidades imediatas para a captura de pacotes sem comprometer a memória, que nem sempre é necessária. A RMON inclui tamanhos de *slice* de captura configuráveis para armazenar os primeiros bytes de um pacote onde vários cabeçalhos de protocolos estão localizados, ou no limite para armazenar o pacote todo. O tamanho por *default* são os primeiros cem octetos. O grupo está formado por duas tabelas: *bufferControlTable* e *captureBufferTable*, com 13 e 7 objetos respectivamente. A tabela *bufferControlTable* mantém um conjunto de parâmetros que controlam a coleção de um fluxo de pacotes que têm filtros correspondentes. A tabela *captureBuffer* mantém uma lista de pacotes capturados de um canal.

Grupo *Event*: Esse grupo controla a geração e notificação de eventos. O grupo tem duas tabelas: *eventTable* e *logTable*, com 7 e 4 objetos respectivamente. A tabela de eventos descreve os parâmetros do evento que pode ser gerado. A tabela de *logs* mantém uma lista de eventos que têm sido armazenados. Esse *log* inclui o tempo do dia para cada evento e uma descrição do evento escrito pelo distribuidor do monitor.

No anexo A-1 estão as tabelas de objetos dos grupos da RMON MIB que foram utilizados neste trabalho, os grupos restantes podem ser encontrados em [WAL 95].

2.3 Evolução da RMON MIB

A RMON MIB foi inicialmente definida em um documento *Internet Draft*, o RFC 1271 [WAL 91]. O grupo de trabalho da IETF (IETF-WG: *Internet Engineering Task Force Working group*) para a RMON MIB utilizou esse documento como uma base de discussão para fazer extensões da RMON MIB e definir um documento padrão dessa MIB.

Foi mencionado que o RFC 1271 suporta somente redes *ethernet*. Entretanto, o IETF-WG tem trabalhado com o objetivo de estender a RMON MIB para outros tipos de redes como *Token Ring*. Em setembro de 1993, o grupo emitiu o RFC 1513 [WAL 93] que descreve as extensões da RMON MIB para *Token Ring*. Em fevereiro de 1995, foi emitido um documento *Internet Standard*, o RFC 1757 [WAL 95]. Com isso, o RFC 1271 tornou-se um documento obsoleto.

O IETF-WG para a RMON MIB também está trabalhando na criação de um documento que especifique a RMON II. O grupo está utilizando como texto de discussão para esse trabalho o documento: *draft-ietf-rmonmib-rmon2-01.txt* que está disponível via ftp nos endereços: **ftp.isi.edu:/internet-drafts** e **ftp.cisco.com:/ftp/rmonmib**. Entre os grupos propostos para fazerem parte da RMON II estão:

-*Protocol Directory*: lista de protocolos que o agente tem a capacidade de decodificar e contar;

-*Protocol Distribution*: mantém estatísticas da distribuição dos protocolos identificados;

- *Address Mapping*: mapeamento de endereços de redes a endereços físicos.

coleção de estatísticas para um *host* particular descoberto numa interface desse dispositivo; e (c) *hostTimeTable*, que também mantém estatísticas de *hosts* descobertos na rede, incluindo a ordem relativa do tempo no qual cada host foi descoberto pelo agente. O grupo *host* tem 26 objetos, divididos nas três tabelas.

Grupo *HostTopN*: esse grupo é usado para preparar relatórios que especificam os principais *hosts* de uma lista ordenada por uma de suas estatísticas. Por exemplo, os primeiros 20 *hosts* com maior número de pacotes enviados ou uma lista ordenada de todos os *hosts* de acordo com o número de erros enviados nas últimas 24 horas. Esse grupo tem duas tabelas: *hostTopNControl* e *hostTopNTable*, com 10 e 4 objetos respectivamente. A *hostTopNControlTable* é usada para iniciar a geração do relatório. A estação de gerenciamento pode selecionar os parâmetros do relatório. Entre esses parâmetros estão as interfaces, estatísticas, *hosts* e os tempos de início e fim da amostragem. Quando o relatório é preparado, as entradas são criadas na tabela *hostTopN* que está associada com uma entrada da *hostTopNControlTable*. Essas entradas são estáticas para cada relatório depois que ele foi preparado.

Grupo *Filter*: esse grupo permite que os pacotes sejam capturados com uma expressão de filtro arbitrária. Dados lógicos e um fluxo de eventos ou canal são formados pelos pacotes que combinam com a expressão filtro. O grupo filtro destaca um filtro genérico que ativa todas as funções de captura e eventos. A função enche o *buffer* de captura de pacotes com os que combinam com o filtro instalado pelo usuário. Qualquer pacote individual que combina com o filtro pode servir como um início ou fim de um *trace trigger*. Os conteúdos do *trace* são controlados por qualquer combinação dos filtros selecionados pelo usuário. As condições de um único filtro podem estar associadas com as expressões booleanas *AND* ou *NOT*. Múltiplos filtros são associados com a função booleana *OR*.. Os usuários podem capturar pacotes válidos, inválidos ou algum dos cinco tipos de pacotes de erros (pacotes curtos, pacotes longos, *jabbers*, pacotes fragmentados e pacotes com erros de CRC ou alinhamento).

O grupo filtro está formado por duas tabelas: *filterTable* e *channelTable*, com 11 e 12 objetos respectivamente. A tabela *filterTable* mantém um conjunto de parâmetros para um filtro de pacotes aplicados numa interface particular e a tabela *channelTable* mantém um conjunto de parâmetros para um canal de pacotes aplicados sobre uma interface particular.

Grupo *Alarm*: esse grupo periodicamente obtém amostras estatísticas de variáveis do monitor e as compara com os limiares previamente configurados. Se a variável que compõe o monitor supera o limite, um evento é gerado. Para limitar a geração de alarmes é implementado o mecanismo de *histerese*. Esse mecanismo define dois limites para cada objeto monitorado: o limite de subida (ou superior) e de descida (ou inferior). As condições para que um evento de subida ou descida seja gerado são semelhantes.

Esse grupo tem a tabela *alarmTable* que consiste de 12 objetos utilizados para uma verificação periódica das condições de alarmes. A implementação desse grupo requer que o grupo *Event* também seja implementado

Grupo *Matrix*: esse grupo armazena a estatística do tráfego e número de erros entre pares de *hosts*. O grupo é formado por três tabelas: *matrixControlTable*, *matrixSDTable* e *matrixDSTable*, com 6 objetos cada. A tabela *matrixControlTable* mantém informação da matriz de tráfego numa interface particular.

Segundo as minutas de trabalho do IETF-WG para a RMON MIB (obtidas através da lista eletrônica da RMON), se está trabalhando na criação de um RFC *Draft Standard* para RMON II.

A RMON I (RFC 1757) pode ser considerado um importante passo na gerência inteligente de redes, pois ela contém objetos necessários para gerenciar as camadas física e de enlace em uma rede local. Porém, todos os seus objetos ainda não são suficientes para gerenciar as camadas de rede e aplicação. A gerência dessas camadas é importante porque o desempenho de uma rede pode ser degradado mesmo que sejam corrigidos todos os problemas detectados nas camadas inferiores (física, enlace e de rede). As principais causas desses problemas podem ser a configuração inadequada de uma aplicação ou problemas nos próprios protocolos das aplicações. Por essa razão, é sumamente importante gerenciar as aplicações que são executadas em redes.

Acredita-se que com o surgimento da RMON II será possível gerenciar uma rede local desde o nível físico até o nível de aplicação. Isto viabilizará o diagnóstico mais completo permitindo implantar uma efetiva Gerência Pró-ativa.

3 Agregando Inteligência à Gerência Pró-ativa

Sistemas especialistas são programas que atuam como consultores inteligentes. Um sistema especialista permite que o conhecimento e a experiência de um ou mais especialistas sejam capturados e armazenados num computador. Esses recursos podem então ser utilizados sem a presença do(s) especialista(s).

Um dos elementos mais importantes num sistema especialista é o conhecimento por ele acumulado. Portanto, é necessário saber como representar e adquirir as informações que irão fazer parte da base de conhecimentos desse sistema especialista. Existem várias técnicas para a representação do conhecimento, uma delas é a de Regras de Produção.

A mente humana possui um amplo estoque de conhecimentos relacionados a uma incontável lista de objetos e idéias. Nossa sobrevivência depende da nossa habilidade em aplicar esses conhecimentos em qualquer situação e aprender continuamente com as novas experiências, sendo assim capazes de responder a situações similares no futuro. O que geralmente é considerado inteligência, pode ser dividido numa coleção de fatos e num meio de se utilizar esses fatos para alcançar os objetivos. Isso é feito, em parte, pela formulação de conjuntos de regras relacionadas a todos os fatos armazenados no cérebro. As regras de produção descrevem o conhecimento em termos de regras do tipo Situação-Ação ou "**SE-Então**" [TAR 90]. Segundo [HAY 85], os sistemas baseados em regras constituem um meio bastante usual para codificar o *Know-How* dos especialistas humanos para resolver problemas.

3.1 Algumas caraterísticas de sistemas especialistas aplicados à gerência de redes de Computadores

Tem-se identificado que na gerência de redes os sistemas especialistas orientados a diagnósticos são os que mais prevalecem. A maioria dos sistemas de diagnósticos atuais usam regras de produção para representar o conhecimento, um método de inferência de encadeamento para adiante (*forward-chaining*) e um padrão de comparação como a forma de localizar o conhecimento relevante.

Segundo [LIE 88], há diversas áreas na gerência de redes nas quais os sistemas especialistas têm aplicação, algumas delas são descritas a seguir.

Interpretação e Diagnóstico. Pode ser classificada dentro da interpretação e correção de falhas bem como em funções de antecipação de falhas. Essa experiência é útil na avaliação do desempenho e isolamento de falhas na rede, observando informações de sensoriamento e alarmes. O sistema identifica padrões de falhas crônicas de equipamentos antes de sua ocorrência e recomenda soluções, evitando assim maiores problemas.

Monitoração. Envolve a comparação de dados observáveis para prever e fixar hipóteses do estado interno usando monitores de dados e dados interpretados. A monitoração *online* é usada na avaliação de desempenho da rede e em ações corretivas de planejamento. Por exemplo, no re-roteamento do tráfego devido a um *crash* no sistema ou a congestionamento na rede.

Predição. É a antecipação de prováveis conseqüências em determinadas situações. A predição pode ser modelada na base de dados históricos e tendências de desempenho. A manutenção pró-ativa e ações preventivas podem ser possíveis usando capacidades de predição para prevenir as falhas futuras.

Controle. Envolve a execução de determinadas ações para trazer o sistema ao nível desejado. As ações de controle podem ser iniciadas como um resultado da interpretação dos dados monitorados para manter o desempenho da rede dentro dos limiares aceitáveis. Essas ações podem incluir imposição de mudanças de rotas, reconfiguração da rede e outras sugestões corretivas.

4 Problemas mais comuns que afetam o desempenho de uma rede

Nesta seção são descritos alguns dos problemas mais comuns que afetam o desempenho de uma rede.

a) Problemas no nível físico

No nível físico, conectores de redes e placas em mau estado são alguns dos principais problemas que afetam o desempenho da rede. Isso provoca uma alta taxa de colisões, ainda num nível de utilização baixo, que logicamente pode provocar congestionamento e portanto a degradação do desempenho.

Nesse nível, o interessante para monitorar e poder determinar se há problemas de hardware ou mesmo de software, são as taxas de erros de transmissão. Segundo [NEM 92], esses erros estão classificados como:

- erro CRC (*Cyclic Redundancy Check*): erro de *checksum* no pacote *ethernet*, usualmente percebido também quando ocorrem os outros tipos de erros;

- erro de alinhamento (*alignment errors)*: ocorre quando o pacote não é múltiplo de 8 bits usualmente indicativo de um erro de *framing*, ou seja, que não têm o comprimer especificado pelas normas;

- pacotes com o tamanho em bytes abaixo do limite inferior permitido (*Undersize Packet* conforme [ISO 92], o limite inferior permitido é 64 bytes;

- pacotes com o tamanho em bytes acima do limite superior permitido (*Oversize Packet:* conforme [ISO 92], o limite superior permitido é de 1518 bytes.

As taxas significativas desses tipos de erro indicam mau funcionamento de *hardware*. P exemplo, erros em pacotes com o tamanho excedido resultam quando os *transceivers* pegam pacotes com o número de bytes superior a 1518, indicando a possibilidade do *transceiver* est defeituoso. Também esse tipo de erro pode indicar problemas com a placa controladora da rede.

b) Congestionamento

O congestionamento poderá ocorrer por duas razões. A primeira seria quando ur computador de alta velocidade gera tráfego mais rápido do que a rede pode transportá-lo. Po exemplo, um supercomputador que gera tráfego através de redes interconectadas. Eventualment os datagramas gerados podem precisar atravessar uma outra rede de baixa velocidade, embora próprio supercomputador esteja em uma rede de alta velocidade. Nesse exemplo, congestionamento acontecerá no roteador da rede de menor velocidade, pois os datagrama chegam até ele em uma velocidade maior do que sua capacidade de encaminhá-los ao destino. A segunda situação seria quando muitos computadores simultaneamente necessitam envia datagramas através de um único roteador. Com essa sobrecarga de processamento, o roteador poderá entrar em congestionamento.

Quando os datagramas chegam ao *host* ou roteador mais rápido que a sua capacidade de processamento, cria-se uma fila em memória temporária. Se os datagramas são parte de um pequeno conjunto, esse armazenamento temporário resolve o problema. Porém, se o tráfego é contínuo, o roteador eventualmente esgotará sua memória e descartará aqueles datagramas que não podem ser armazenados. Um roteador pode usar uma mensagem ICMP (*Internet Control Message Protocol*) *source quench* para reduzir o congestionamento. Uma mensagem *source quench* é uma requisição para que o *host* fonte reduza sua taxa atual de transmissão de datagramas. Usualmente, roteadores congestionados enviam uma mensagem *source quench* para cada datagrama que eles descartam. Não há uma mensagem ICMP em resposta a uma mensagem *source quench*. Para isso, um *host* que recebe uma mensagem *source quench* de uma máquina M, diminui a taxa de transmissão dos datagramas dirigidos a essa máquina até que não sejam mais recebidos mensagens *source quench*. Após um determinado tempo, o *host* incrementa gradualmente a sua taxa de transmissão até atingir o nível normal, desde que ele não mais receba mensagens *source quench*.

O problema de congestionamento na rede afeta fortemente o seu desempenho, pois o tempo de resposta resultante é muito alto. Além da incapacidade da linha em transportar todo o tráfego requerido, o congestionamento e o roteamento estão também frequentemente relacionados, pois uma das causas mais importantes que ocasionam o congestionamento são as decisões deficientes de roteamento [COM 91].

c) Configuração

Uma rede com uma topologia mal configurada pode causar problemas como altas taxas de colisões e de utilização. Por exemplo, um segmento com muitos *hosts* ou muitos *hosts* em diferentes segmentos ligados a um mesmo servidor (essa topologia pode saturar o elemento de ligação com o segmento da rede onde o servidor está localizado) .

Uma má configuração da rede pode provocar uma degradação no seu desempenho. Em redes pequenas, uma causa possivel de problemas de desempenho é a não obediência a normas e padrões no que tange a comprimento de cabos, por exemplo. Em [NEM 92], são citados alguns exemplos dessa tipo de problema:

- quando o segmento da rede excede os 185 metros de comprimento em redes 10Base 2 (Ethernet cabo fino) ou 500 metros de comprimento em redes 10Base 5 (Ethernet cabo grosso) ou ainda 100 metros de comprimento em segmentos de rede Ethernet 10BaseT (par trançado);

- quando é excedido o número máximo de estações permitidas em um segmento (esse número, segundo [ISO 92], é de 100 estações em redes *ethernet* 10Base 5 e 30 em redes 10Base2;

- quando os cabos dos *transceivers* (AUI, *Attachment Unit Interface)* excedem 50 metros de comprimento máximo, segundo [ISO 92], esse comprimento deve ser menor ou igual a 50 metros.

d) Roteamento

Dois dos principais problemas de roteamento em redes TCP/IP, são descritos a seguir[BOS 88].

Muitos dos algoritmos de roteamento comumente usados estão sujeitos à instabilidade de um tipo ou outro. Por exemplo, na mudança de uma rota devido ao fato de que um dos enlaces está fora, várias atualizações podem ser necessárias antes que o sistema chegue a um novo conjunta de rotas coerentes. Em alguns algoritmos, esse comportamento é chamado ***Counting to Infinity***, normalmente requerendo aproximadamente 10 atualizações por roteador para que ocorra a convergência. A menos que sejam tomadas certas precauções, o resultado desse comportamento é uma alta quantidade de mensagens de atualização em um curto periodo de tempo. A maioria dessas mensagens de atualização são *broadcast* e isso pode saturar os processos de entradas de outros sistemas na rede. As redes que têm linhas de comunicação lentas podem ser inundadas por vários segundos.

Há várias medidas defensivas para esse problema. Uma delas é a construção de protocolos de roteamento que convergem rapidamente. Outra solução é introduzir um atraso no processo de atualização. Quando uma mudança de rota é realizada, ela é propagada para outros sistemas depois de um tempo de atraso calculado de tal forma que o tráfego de roteamento não inunde a rede .

Tabelas de roteamento erradas podem provocar um "loop" de roteamento (rota circular) para um dado destino. Um "loop" de roteamento pode consistir de dois roteadores (A e B) - o

roteador A tem como rota para o destino D o roteador B, e vice-versa - ou esse "loop" pode consistir de vários roteadores. Quando vários roteadores formam o "loop", cada um deles tem definido como rota para o destino D o próximo roteador no "loop". Assim um datagrama permanece no "loop" indefinidamente. Mas em redes TCP/IP, cada datagrama IP contém um contador que indica o seu tempo de vida na rede. Esse contador é o TTL. Um roteador decrementa o contador TTL quando ele processa o datagrama e o descarta quando o contador está em zero. O problema de rotas circulares geralmente acontece durante a recuperação de uma falha em um enlace, pois a maioria dos algoritmos de roteamento não garantem a entrega de mensagens que precisam passar perto do enlace com falhas[COM 91].

e) Colisões

A taxa de colisões é o principal parâmetro para avaliar o desempenho de uma rede Ethernet. Esse parâmetro está relacionado com o parâmetro utilização de uma rede, pois se a taxa de utilização sobrepasa 30 %, a taxa de colisões aumenta exponecialmente. Assim uma rede sobrecarregada apresentará uma taxa de utilização alta (acima de 35 ou 40 %) e consequentemente a taxa de colisões será maior que 2 % do seu tráfego. Uma rede com uma taxa de colisões menor que 2 %, apresenta um nível de desempenho aceitável. Se esse valor for superado, deve ser investigado onde está o problema antes que ele atinja 5 %, pois isso deve sempre ser evitado[NAS 94].

Pode-se concluir que para manter um nível de desempenho aceitável em uma rede, o nível de colisões deve ser cuidadosamente controlado. Até aqui foi apresentado que altas taxas de colisões podem surgir, seja por ter uma rede sobrecarregada (fora da sua capacidade) ou por algum problema físico como os mencionados em alguns dos item anteriores.

f) Tormenta de Pacotes *Broadcasts*

Uma tormenta de pacotes *broadcast* é um conjunto de pacotes enviados ao endereço *broadcast ethernet* em um volume suficientemente alto para causar problemas em muitos hosts na rede. Algumas vezes, esse problema acontece junto com um outro, conhecido como Avalanche de Pacotes, que ocorre quando um único pacote dispara uma avalanche de respostas.

Segundo [SPU 89], existem três principais causas que provocam uma tormenta *broadcast* e avalanche de pacotes: problemas nos protocolos, configuração e implementações com defeitos.

Problemas nos protocolos

O problemas de tormenta *broadcast* nos protocolos é devido principalmente a que muitas implementações de protocolos das camadas superiores da rede, dependem do uso de endereço *broadcast*, no lugar de *multicast*.

Um exemplo desse problema, pode ser visto no protocolo da *Silicon Graphics* que inclui em suas estações um jogo simulador de vôo, que faz uso do endereço *broadcast*. Esse jogo pode ser executado entre duas estações, mas devido ao uso de *broadcast*, todas as estações no segmento onde está sendo executado o programa, escutam cada movimento do jogo. Se for usado

um endereço IP *broadcast* incorreto, uma rápida seqüência de tormenta *broadcast* pode ser gerada.

No mecanismo ICMP para descobrir a máscara da rede, também pode surgir o problema de tormenta *broadcast*. Um *host* procurando pela máscara da rede pode enviar uma requisição ao endereço IP *broadcast*. A maioria dos *hosts* atuais estão programados para responder a essa requisição, produzindo uma rajada de respostas de máscaras da rede. Algumas vezes, as respostas a essas requisições de máscara são também enviadas ao endereço *broadcast*.

Má configuração

Em relação a má configuração, uma das principais razões de tormenta *broadcast*, segundo [SPU 89] é a de ter em uma rede TCP/IP estações UNIX com as versões BSD 4.2 e alguma outra versão como a BSD 4.3 ou superior. Isso devido a que na versão BSD 4.2 do UNIX, o endereço *broadcast* está definido com todos os bits que identificam os *hosts* em "zero". Nas versões BSD 4.3 e superiores, esse campo é definido com todos os bits em "um". Um exemplo desses endereços seria: 128.83.0.0 (endereço *broadcast* para a rede 128.83 com o UNIX BSD 4.2) e 128.83.255.255 (endereço *broadcast* para a rede 128.83 com o UNIX BSD 4.3).

O problema surge quando uma estação com a versão BSD 4.3, envia um pacote ao endereço *broadcast* 128.83.255.255. Isso, porque as estações com a versão BSD 4.2 não reconhecem o endereço como válido. Essas estações tentam resolver o endereço *broadcast* para elas desconhecido, enviando simultaneamente uma requisição ARP ao endereço *broadcast*. Essa requisição provoca uma avalanche de pacotes que produz uma tormenta *broadcast* de tamanho máxima.

Implementações com defeitos

Implementações de *software* com defeitos também podem gerar tormenta de pacotes *broadcast* e avalanche de pacotes. É possível surgirem *softwares* com defeitos, que usem *broadcast* indiscriminadamente.

5 Sistema OLHO VIVO

Usando os conceitos acima referidos foi projetado e implantado um sistema de apoio à Gerência Pró-ativa, agregado a um inteligente RMON.

A idéia do sistema é de que ele funcione como um vigilante da rede, observando certos indicadores de degradação de desempenho e buscando soluções para esses problemas. A solução dos indicadores foi feita a partir de um cuidadoso estudo dos problemas mais freqüentes nas redes. O sistema está dividido em módulos conforme a Figura 5.1.

5.1 Agente RMON

O agente RMON utilizado é conhecido com **btng**: *Beholder The Next Generation*. Esse agente é um *software* de domínio público que foi desenvolvido pelo grupo de pesquisa DNPAP (*Data Network Performance Analysis Project*) da universidade de Delft na Holanda[BEH 93]. O agente implementa a RMON MIB completa e alguns outros objetos que esse grupo de pesquisa adicionou.

A estrutura do *beholder* é de um conjunto de coletores, onde cada coletor vai ser responsável por um grupo de objetos RMON MIB. Por exemplo, o coletor *"stats"* coleta informações referentes ao grupo *statistics*. Neste trabalho, não foi necessário fazer uso de todos os coletores, usando somente os dos grupos *stats*, *history*, *host* e *matrix*. Isso porque esses grupos são os que fornecem informações relacionadas aos tipos de problemas identificados.

Antes da implementação do módulo inteligente, foi estudado o *beholder* (agente RMON) e com ele se fez uma monitoração por um tempo estendido, de um dos segmentos da rede do Instituto de Informática da UFRGS. Essa monitoração foi feita com o objetivo de criar uma *baseline*[2]. Isso serviu para identificar problemas naquele segmento, de tal forma que foi possível definir limites que formariam parte do conjunto de regras do modulo inteligente.

Da monitoração com o *beholder* se obteve uma *baseline* da operação da subrede 143.54.7.0 da rede do Instituto de Informática da UFRGS. Devido a que a informação obtida com essa monitoração é muito extensa, somente é apresentada aqui, a informação da monitoração de um dos dias mais significativos (ou seja aquele dia das duas semanas de monitoração no qual se identificaram os maiores valores dos contadores de erros e de utilização). Essa informação é apresentada nas Figuras da 5.2.a até a 5.2.f. Como pode-se observar nessas figuras, o nível de utilização (ver Fig. 5.2.f) e a taxa de colisões (ver Fig. 5.2.e) são bem baixos, o que indica que a rede está trabalhando dentro de parâmetros normais de operação, ainda sem comprometer o desempenho da rede.

---

[2] baseline é a função através da qual se pode conhecer a operação normal de uma rede. É o perfil de operação de uma rede.

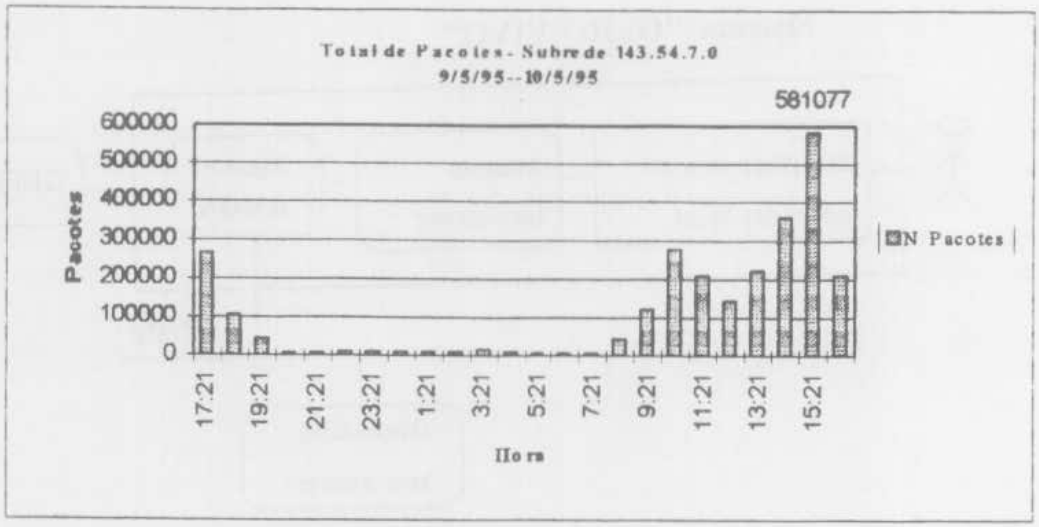

Figura 5.2a

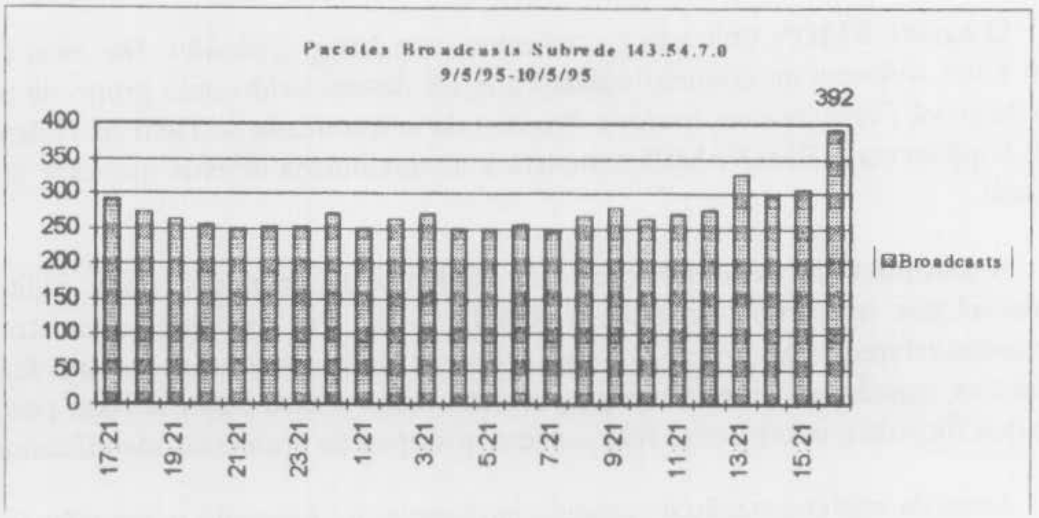

Figura 5.2b

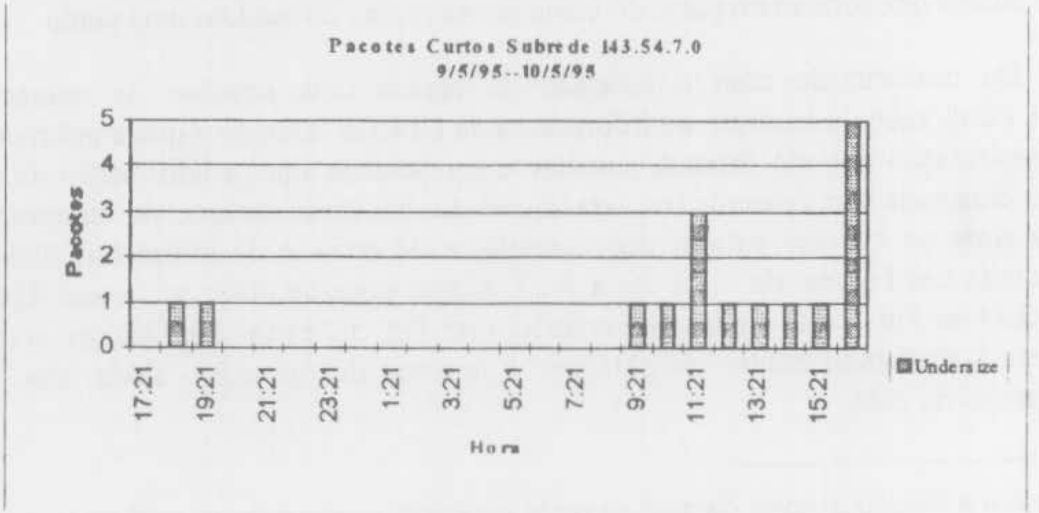

Figura 5.2c

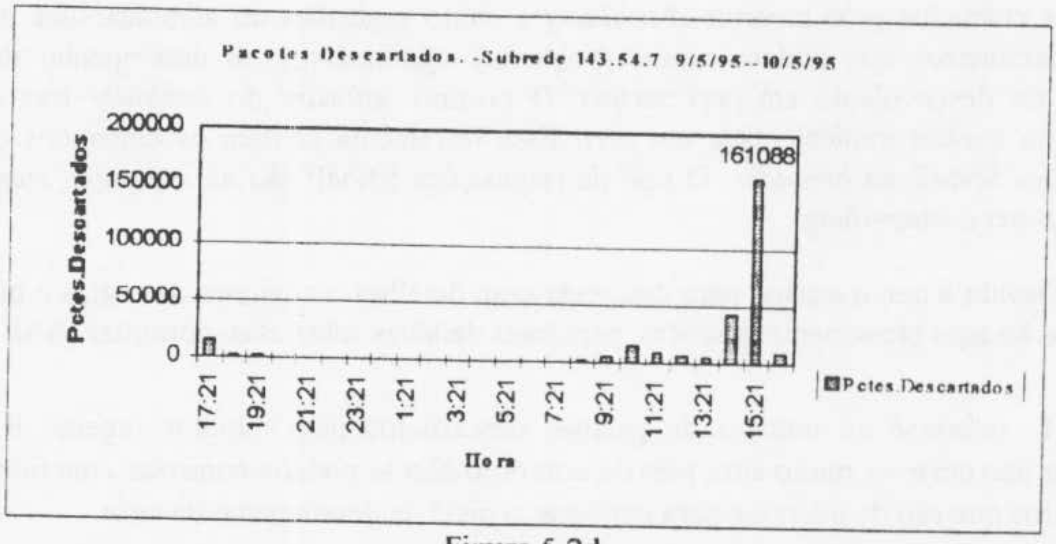

Figura 5.2d

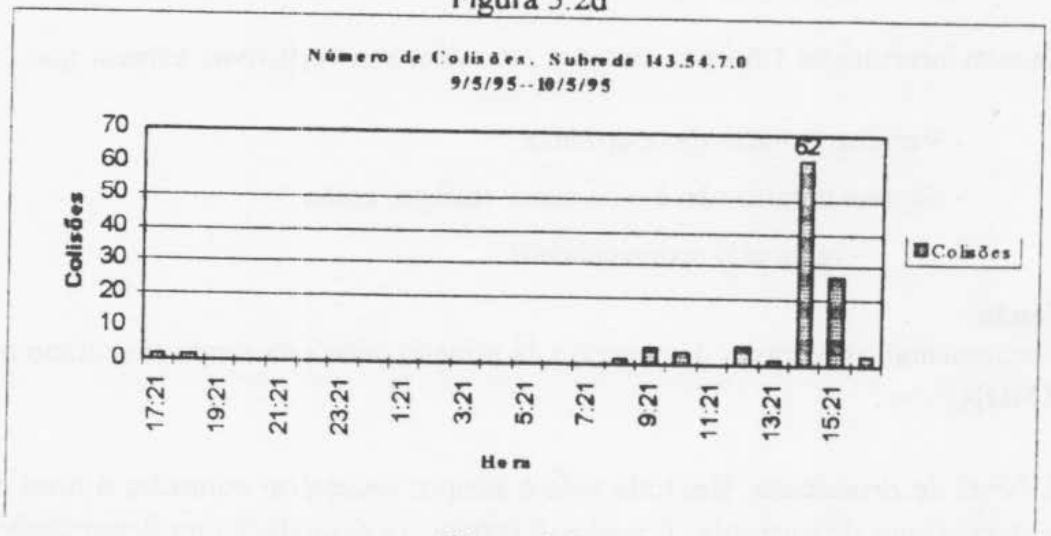

Figura 5.2e

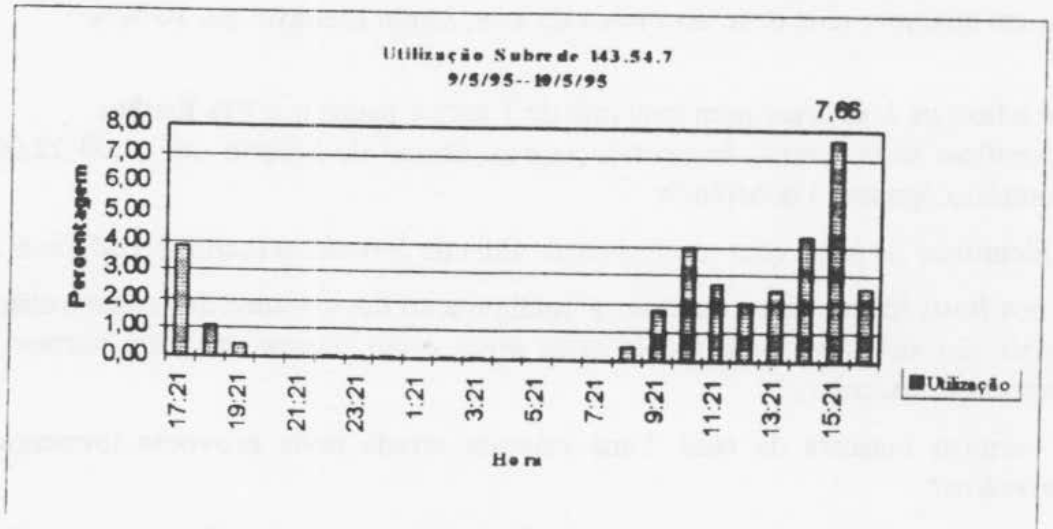

Figura 5.2f

5.2 Módulo Inteligente

O módulo inteligente é o principal componente do sistema Olho Vivo. Esse módulo analisa os dados coletados pelo monitor (*beholder*) e emite sugestões ao administrador da rede sobre certos parâmetros que podem causar problemas significativos no desempenho da rede. Esse módulo foi desenvolvido em *perl scripts*. O próprio *software* do *beholder* traz consigo uma ferramenta *tricklet* implementada em *perl*. Essa ferramenta contém os comandos que fazem as requisições SNMP ao *beholder*. O tipo de requisições SNMP são as *snmp-get*, *snmp-set*, *snmp-tbl*, *snmp-nxt* e *snmp-dump*.

Devido a que o espaço para descrever com detalhes o conjunto de regras é pequeno, essas regras serão aqui brevemente descritas, para mais detalhes sobre elas, consultar [SAE 96].

Regra 1: refere-se ao número de pacotes descartados pelo monitor (agente RMON). Esse contador não deve ser muito alto, pois do contrário não se poderia controlar com fidelidade outros parâmetros que são de interesse para conhecer o nível de desempenho da rede.

Se num intervalo de 1 hora o contador *etherHistoryDropEvents* é maior que 1 % **Então:**

- Verificar horário de ocorrência;
- Se esse horário não é o de maior tráfego, então:
  - ignorar o acontecimento;

Senão:
- incrementar os recursos de memória da estação onde está sendo executado o agente RMON.

Regra 2. Nível de *Broadcasts*. Em toda rede é sempre necessário controlar o nível de *broadcast*, pois o acontecimento de tormentas *broadcast*, provocará degradação no desempenho da rede. De uma ou outra maneira *broadcast* são necessários, mas em geral, segundo [NAS 94], o nivel de *broadcast* em qualquer rede deve ser menor de 8 %, sendo aceitável até 10 %%.

Se a taxa de *broadcast* num intervalo de 1 hora é maior que 8% **Então:**
- verificar se o horário da ocorrência está dentro do horário útil (7:00-22:00 horas), do contrário, ignorar a ocorrência;
- identificar os *hosts* com os niveis mais altos de *broadcast* (script broad_nivel.pl);
- nos *hosts* identificados, analisar a configuração do *software* de comunicação, para saber quais são as razões pelas quais esses *hosts* estão transmitindo um número tão alto de pacotes *broadcasts*;
- verificar máscara da rede. Uma máscara errada pode provocar tormenta de pacotes *broadcast*;
- verificar que na rede sendo monitorada não estejam estações com versões do UNIX incompativeis (BSD 4.2 junto com BSD 4.3 ou versões superiores).

Regra 3. Nível de *multicast*. Igual a regra 2, somente que isto só vai ser identificado em redes com equipamentos que suportam *multicast*.

Regra 4. Erros de alinhamento ou CRC (*Cyclic Redundancy Check*). Este tipo de erro normalmente ocorre quando há alguma placa controladora de rede, *transceiver* e/ou cabo coaxial com problemas. Segundo [NAS 94], uma percentagem de 2 % do tráfego total, deste tipo de erro é o suficientemente alto para afetar o desempenho da rede.

Se *etherHistoryCRCAlignErrors* é maior que 2 % **Então**:
- revisar o cabeamento do segmento da rede sendo monitorado;
- investigar os *hosts* com problemas. Isso deve ser feito manualmente com cada um dos *hosts* do segmento, pois infelizmente o agente não fornece informações desse tipo de erro por *host* individual. Em cada *host* o que deve ser verificado é o bom funcionamento das placas controladoras e dos *tranceivers*. Isso pode ser feito com *transceiver-tester*.

Regra 5, 6 e 7. Estas regras avaliam os contadores de pacotes com erros de comprimento, (pacotes curtos (menor que 64 bytes), pacotes longos (maior que 1518 bytes)), pacotes fragmentados e *jabbers*. Os tipos de erros que essas regras identificam, geralmente são causados por problemas no nível fisico da rede, por tal razão estas regras são semelhantes, tendo também como limiar 2 % do total do tráfego. Algumas das recomendações emitidas por estas regras são mencionadas a seguir:

- verificar o bom funcionamento de placas controladoras e *transceivers*;
- verificar a taxa de colisões (para a regra de pacotes curtos).

Erros de pacotes fragmentados:
- verificar configuração de roteadores;
- verificar o bom funcionamento da porta do repetidor ao qual o segmento sendo monitorado está conectado;
- verificar taxa de colisões.

Erros de pacotes *jabbering* , estes tipos de pacotes são errados por terem um comprimento maior que 1518 bytes e por também terem erros de *checksum*. Isso pode ser provocado por *transceivers* com defeitos ou por problemas no cabo:

- verificar o bom funcionamento do cabo coaxial;
- verificar o bom funcionamento dos *transceivers*.

Regra 8. Taxa de colisões.

Se a taxa de colisões é maior que 1 %, e se essa taxa for identificada repetidas vezes, principalmente no horário de maior tráfego da rede, **Então**:
- verificar o bom funcionamento do cabo coaxial, fazendo os testes de continuidade e condutividade com um multímetro.

- verificar taxa de utilização. A rede pode estar sobrecarregada produzindo uma alta taxa de colisões.

Regra 9. Taxa de utilização. Na maioria dos estudos feitos sobre o desempenho do protocolo CSMA/CD [BMSG 86], se diz que o desempenho da rede se mantém num nível de desempenho aceitável quanto a taxa de utilização não sobrepasa 40 %. Baseando-se nesses estudos, foi definida na regra 9 o valor de 40 % como o limite para indicar que o desempenho da rede ainda não está sendo degradado.

Se a taxa de utilização é maior que 40 %, **Então**:
- monitorar a rede com o objetivo de criar uma *baseline* da operação normal da rede, por um periodo mínimo de 5 dias, para identificar as mudanças;
- manter os seguintes parâmetros numa *baseline* com intervalos de tempo pequenos (de preferência entre 30 e 60 minutos):
  - número total de pacotes transmitidos;
  - erros detectados;
  - taxa de utilização dos *hosts* na rede. Esta é uma forma de identificar quais são os *hosts* que transmitem mais. Isso pode ser feito usando o *script* host_util.pl;
  - *hosts* que mais conversam entre si (matrix_talk.pl).

Com os parâmetros acima pode-se observar se houve alguma mudança na rede, por exemplo, o incremento significativo da taxa de utilização de um dia para outros.

Se depois de criada uma baseline da rede, constata-se que a rede está sobrecarregada **Então**:
- confinar o tráfego;
- se não for possível confinar o tráfego, é necessário pelo menos incrementar os recursos de memória e velocidade de processamento dos servidores de redes e arquivos.

5.3 Gerente SNM (Sun Net Management)

No sistema Olho Vivo (ver Fig.5.1), uma estação de gerenciamento SNM (*SunNetManagement*) é utilizada como gerente. Para integrar essa estação com o agente RMON, é necessário que o SNM acesse o *beholder*. Os arquivos do que identificam o *beholder* para SNM, vem junto com o software do beholder, a colocação adequada desses arquivos nos diretórios do SNM permite que o *beholder* seja consultado desde uma estação de gerenciamento SNM.

5.4 Interface com o operador local

A comunicação do sistema Olho Vivo com o operador ou administrador da rede (ver Figura 5.1) é através de e-mail. A primeira vez que uma das regras que o módulo inteligente implementa é verdadeira emite um mail: (a) descrevendo o problema identificado no intervalo de

tempo em que isso aconteceu e (b) é indicado o nome de um arquivo onde é mantido um histórico ou *baseline* da operação da rede. Isso pode ser posteriormente analisado pelo administrador, verificando se a rede está trabalhando dentro dos parâmetros de operação que garantem que o desempenho está num nível normal, satisfazendo dessa maneira os usuários finais.

5.5 Avaliação do Olho Vivo

Após implementado o módulo inteligente, este foi testado por duas semanas, com o objetivo de avaliar o conjunto de regras desse módulo. Esse teste foi feito em 4 subredes da UFRGS, e com isso pôde-se constatar que a rede está trabalhando dentro de parâmetros normais, não se identificando taxas de utilização maiores que 10 %, as maiores taxas de utilização identificadas oscilaram entre 3 e 7 %. Devido a que as taxas de utilização são baixas, as taxas de colisões não poderiam ser altas. Em nenhuma das subredes se identificou uma taxa de colisões mais que 0,5 %.

O único problema identificado foi o de pacotes descartados pelo agente RMON. Esse acontecimento se dá por falta de recursos de memória na estação onde está sendo executado o agente. De uma forma não muito significativa pôde-se perceber que as estações que tinham uma área de *swap* maior que a sua memória interna, apresentaram uma percentagem ligeiramente menor de pacotes descartados que aquelas estações onde essa área era igual a memória interna da estação. Assim sendo, pode-se dizer que na regra 1, seria interessante sugerir a verificação da área de *swap* naquela estações que estão rodando o agente RMON, isso de preferência deve ser feito se a estação mantém em execução muitos processos. Também no teste pôde-se constatar que devido a que alguns coletores (*matrix e hosts*, por exemplo) mantém tabelas de dados muito extensas a percentagem de pacotes descartados era incrementada. Por essa razão seria conveniente verificar se é necessário que tais coletores estejam ativos, senão for o caso, então eliminar esses coletores para diminuir a percentagem de pacotes descartados, e se essa percentagem ainda assim for alta, então o agente deve ser mudado para uma outra estação com maiores recursos de memória.

6 Conclusões

O objetivo principal deste trabalho foi desenvolver suporte para gerenciar uma rede de computadores de forma pró-ativa. Acredita-se que parte desse objetivo foi atingido. Porém, cabe mencionar uma das principais dificuldades encontradas neste trabalho. Essa dificuldade está relacionada a identificação de problemas que afetam o desempenho de uma rede. A melhor forma para obter uma descrição de problemas é através de consultas a especialistas da área (administradores e operadores de redes). Grande parte dos problemas neste trabalho descrito fora obtidos dessa maneira, porém seria interessante fazer uma pesquisa mais ampla, onde também possa ser incluída uma descrição dos problemas que os serviços pela rede oferecidos apresentam com o objetivo de desenvolver outras regras que venham à enriquecer o módulo inteligente do Olho Vivo.

7 Bibliografia

[BEH 93] BEHOLDER The Next Generation. Delft: University of Delft, Holanda. 1993. Software e documentação obtida por ftp Anônimos de "dnpap.ut.tudelft.nl" no diretório"./pub/dnpap/btng/

[BOS 88] BOSACK, L. HEDRICK, C. Bridges and Routers Observations, Comparisons and Choosing Problems in Large LANs. IEEE Network, v.2, n.1, p.49-56. January 1988.

[COM 91] COMER, Douglas E. Internetworking with TCP/IP vol I: Principles, Protocols, and Architecture. Segunda Edição, Prentice Hall 1991.

[HAY 85] HAYES-ROTH, Frederick. Rule Based Systems. Communications of the ACM. v.28, n.9, p. September 1985.

[ISO 92] ISO/IEC 8802-3. International Standard. ANSI/IEEE std. 802.3. Information Techology -Local and Metropolitan Area Networks-. Part 3: Carrier Sense Multiple Access Method and Phyxical Layer Specifications. Terceira Edição, 1992.

[JAN 93] JANDER, Mary. Proactive LAN Management: Tools that look for trouble to keep LANs out of danger. Data Communications, p.49-56. March 21, 1993.

[KLE 88] KLERER, S.Mark. The OSI Management Architecture: an Overview. IEEE Network, Vol2. No.2, March, 1988.

[LIE 88] LIEBOWITZ, J. Expert System Applications to telecommunications. Willey Interscience, 1988.

[NAS 94] NASSER, Dan. Network Optimization and TroubleShooting: Achieve Maximum Network Performance. NRP,1994.

[NEM 92] NEMZOW, Martin A.W. The Ethernet Management Guide: Keeping the Link. Segunda Edição, McGraw Hill, 1992.

[SAE 96] SÁENZ, Esmilda. Olho Vivo Sistema Inteligente para Gerência Pró-ativa Remota. Dissertação de Mestrado, CPGCC-UFRGS, Porto Alegre, 1996. (ainda não publicada).

[SPU 89] SPURGEON, Charles. Broadcasts Storm and Packets Avalanches on Campus Internets. University of Texas at Austin.(documento extraído da lista eletrônica BIG-LAN), July 1989.

[TAR 90] TAROUCO, Liane M.R. Inteligência Artificial Aplicada ao Gerenciamento de Redes de Computadores.São Paulo: USP-Escola Politécnica, 1990.

[WAL 91] WALDBUSSER.S. Remote Network Monitoring Management Information Base: RMON MIB. Request for Comment 1271 (documento obsoleto),1991.

[WAL 93] WALDBUSSER.S. Token Ring Extensions to the Remote Network Monitoring MIB. Request for Comment 1513,1993

[WAL 95] WALDBUSSER.S. Remote Network Monitoring Management Information Base: RMON MIB. Request for Comment 1757,1995

ANEXO A-1 Grupos da RMON MIB utilizados no Olho Vivo

Tabela 2.1 Grupo *Statistics*

etherStatsIndex	etherStatsCRCAlignErrors	etherStatsPkts65to127Octets
etherStatsDataSource	etherStatsUndersizePkts	etherStatsPkts128to255Octets
etherStatsDropEvents	etherStatsOversizePkts	etherStatsPkts256to511Octets
etherStatsOctets	etherStatsFragments	etherStatsPkts512to1023Octets
etherStatsPkts	etherStatsJabbers	etherStatsPkts1023to1518Octets
etherStatsBroadcastPks	etherStatsColisions	etherStatsOwner
etherStatsMulticastPkts	etherStatsPkts64Octets	etherStatsStatus

Tabela 2.2 Grupo *Matrix*

MatrixControlTable	**MatrixSDTable**	**MatrixDSTable**
matrixControlIndex	matrixSDSourceAddress	matrixDSSourceAddress
matrixControlDataSource	matrisSDDestAddress	matrisDSDestAddress
matrixControlTableSize	matrixSDIndex	matrixDSIndex
matrixControlLastDeleteTime	matrixSDPkts	matrixDSPkts
matrixControlOwner	matrixSDOctets	matrixDSOctets
matrixControlStatus	matrixSDErrors	matrixDSErrors

Tabela 2.3 Objetos do Grupo *History*

historyControlTable	**etherHistoryTable**
historyControlIndex	etherHistoryIndex
historyControlDataSource	etherHistorySampleIndex
historyControlBucketRequested	etherHistoryIntervalStart
historyControlBucketsGranted	etherHistoryDropEvents
historyControlInterval	etherHistoryOctets
historyControlOwner	etherHistoryPkts
historyControlStatus	etherHistoryBroadcastPkts
	etherHistoryMulticastPkts
	etherHistoryCRCAlignErrors
	etherHistoryUndersizePkts
	etherHistoryOversizePkts
	etherHistoryFragments
	etherHistoryJabbers
	etherHistoryCollisions
	etherHistoryUtilization

Tabela 2.4 hostControlTable

hostControlIndex
hostControlDataSource
hostControlTableSize
hostControlLastDeleteTime
hostControlOwner
hostControlStatus

Tabela 2.5 hostTable e hostTimeTable

HostTable	**HostTimeTable**
hostAddress	hostTimeAddress
hostCreationOrder	hsotTimeCreationOrder
hostIndex	hostTimeIndex
hostInPkts	hostTimeInPkts
hostOutPkts	hostTimeOutPkts
hostInOctets	hostTimeInOctets
hostOutOctets	hostTimeOutOctets
hostOutErrors	hostTimeOutErrors
hostOutBroadcastPkts	hostTimeOutBroadcastPkts
hostOutMulticastPkts	hostTimeOutMulticastPkts