Tratamento automatizado de alertas

Cristina Moreira Nunes *Liane M. R. Tarouco*

Pós-Graduação em Ciência da Computação
Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves 9500, Campus do Vale, Bloco IV
91501-970 Porto Alegre, RS
nunes@inf.ufrgs.br e liane@penta.ufrgs.br

Resumo

O trabalho descreve um sistema integrado de gerência de rede com um sistema de registro de problemas. Módulos especializados, orientados à análise de aspectos específicos do comportamento da rede efetuam uma análise das características e do status da mesma, filtrando eventos e tentando prover resposta automatizada e/ou recomendações sobre cursos de ação para as anormalidades percebidas.

Abstract

This work describes an integrated network management system coupled with a trouble ticketing system. Specialized modules, oriented to different network characteristics analyse its status, flitering events and providing automated response or/and advisoring on procedures to handle perceived abnormalities.

1 Introdução

As redes de computadores estão evoluindo rapidamente e cada vez mais sua utilidade tem-se tornado indispensável, dada a facilidade em transferir dados de um lugar para outro. Contudo, para seu perfeito funcionamento, é preciso haver um suporte à operação da rede e como esta é uma tarefa um tanto complexa foi especificado um Sistema de Trouble Ticket [MAD 94] e um Sistema de Alertas. O primeiro é utilizado para registrar e acompanhar problemas detectados na rede, enquanto o segundo visa permitir ao administrador da rede especificar condições de excessão sobre as quais deseja ser avisado ou para as quais um registro de problemas (Trouble Ticket) deva ser automaticamente criado. Ambos sistemas foram propostos para apoiar a organização e estruturação da gerência de rede da UFRGS.

No trabalho de [MAD 94] foi projetado e implantado um protótipo de sistema de trouble ticket orientado ao acesso cooperativo. O sistema foi designado CINEMA-Cooperative Integrated Network Management [MAD 93] e nele foram especificados módulos de manuseio de registros de problemas numa base de dados e uma plataforma básica para configurar a obtenção de informações sobre a rede (acesso a objetos gerenciados em agentes remotos).

Todavia, a integração entre estes dois módulos (Sistema de Trouble Ticket e de Alertas) é feita por um outro módulo, designado *Processador de Eventos*. Nele deverá ser armazenado o conhecimento necessário para avaliar as informações obtidas via sistema de alertas e discriminar os

eventos menores que não são dignos de registro, daqueles que precisam resultar na criação de um registro de eventos e precisam ser sinalizados, por outras maneiras, para o administrador da rede.

Este trabalho visa descrever um sistema especialista que atuará como um dos módulos do Sistema de Alertas, o módulo processador de eventos. Este módulo *Processador de Eventos* terá também integração com o Sistema de Trouble Ticket de modo a poder gerar registros automaticamente quando verficar a existência de condições de excessão previamente especificadas numa base de conhecimento a ele inerente.

O sistema resultante é passível de acesso a partir de vários pontos de gerência local da rede, funcionando como um elemento de integração entre os esforços dos administradores locais dos diferentes domínios de gerenciamento possivelmente existentes na rede, de forma a permitir a cooperação e integração entre as atividades de gerência dos administadores.

2 O Sistema de Alertas

O sistema proposto integra-se ao CINEMA e serve para que quando ocorra algum problema na rede da Universidade, o administrador da mesma seja avisado desse problema e realize a ação corretiva adequada. Sua principal característica é a busca de situações críticas com base nas informações que possui a sua disposição.

É muito importante para as pessoas encarregadas de administrar uma rede de computadores a utilização de ferramentas que monitoram a mesma. Essas ferramentas permitem avaliar o desempenho da rede de forma ágil, assim como tomar as atitudes cabíveis tão logo perspectivas de falhas surjam. Através das ferramentas de monitoração, o administrador não precisa se preocupar em coletar informações manualmente, onde é muito fácil de ocorrerem erros, além de levar um tempo consideravelmente grande. Dependendo do tipo de ferramenta utilizada, após a monitoração ser efetuada, o administrador é avisado dos problemas que estão ocorrendo na rede e então cabe a ele tentar resolvê-los. Essas ferramentas geralmente emitem relatórios sobre o estado atual da rede, fornecendo estatísticas proveitosas para o administrador.

Com base nessas características o sistema foi desenvolvido. A monitoração da rede é efetuada em objetos e máquinas definidas pelo administrador e após a coleta destas informações o sistema faz uma análise estatística, gerando eventos. Além disso, os eventos são verificados para se descobrir quais deles têm a necessária gravidade de se tornar um alerta. Quando alertas são gerados, o administrador fica sabendo que um problema ocorreu com a rede e que deve ser corrigido o mais rápido possível. Desta forma, o administrador pode ficar livre para realizar outras tarefas enquanto o sistema efetua as monitorações definidas por ele.

O sistema de Alertas visa permitir ao administrador especificar condições de excessão sobre as quais deseja ser avisado. Ele coleta e analisa estatísticamente as informações obtidas, auxilia na monitoração dos níveis de serviço e por conseqüência provê meios para a gerência de desempenho e de falhas. Gerência de desempenho é o processo de medir o desempenho de todos os elementos que compreendem uma rede de dados [LEI 93].

Através de monitorações contínuas é possível encontrar a utilização corrente dos enlaces e segmentos de rede, identificar áreas de possível congestionamento, isolar altas taxas de erros e examinar os padrões de tráfego da rede. Cada um desses valores pode auxiliar o gerente de rede a assegurar se a rede está ou não atingindo as expectativas de seus usuários.

As informações de desempenho devem ser analisadas sobre um determinado período de tempo. Esse período de tempo pode variar de poucos segundos para um mês ou mais.

O sistema coleta os dados em equipamentos definidos pelo usuário, filtra-os após uma análise estatística e gera eventos. Nesta análise é feita uma comparação do dado coletado com alguns parâmetros, denominados "limites", que são calculados de forma automática pelo sistema ou fornecidos estaticamente pelo usuário. Eventos de mesma natureza subsequentes não são gerados repetitivamente através do uso de um mecanismo de histerese tal como proposto em [WAL 91].

Os eventos gerados pelo sistema são analisados por um dos módulos do sistema, denominado processador de eventos. Usando uma base de regras é possível determinar quais eventos têm a necessária gravidade de se transformar em alertas e gerarem um registro de problemas.

Eventos são indicadores de situações anormais detectadas a partir da filtragem dos dados coletados junto aos equipamentos da rede. Enquanto alertas são indicadores de que estas situações anormais são críticas, e representam algum tipo de ameaça à manutenção do nível de qualidade dos serviços da rede.

A figura 1 mostra onde o processador de eventos se localiza dentro do Sistema de Alertas.

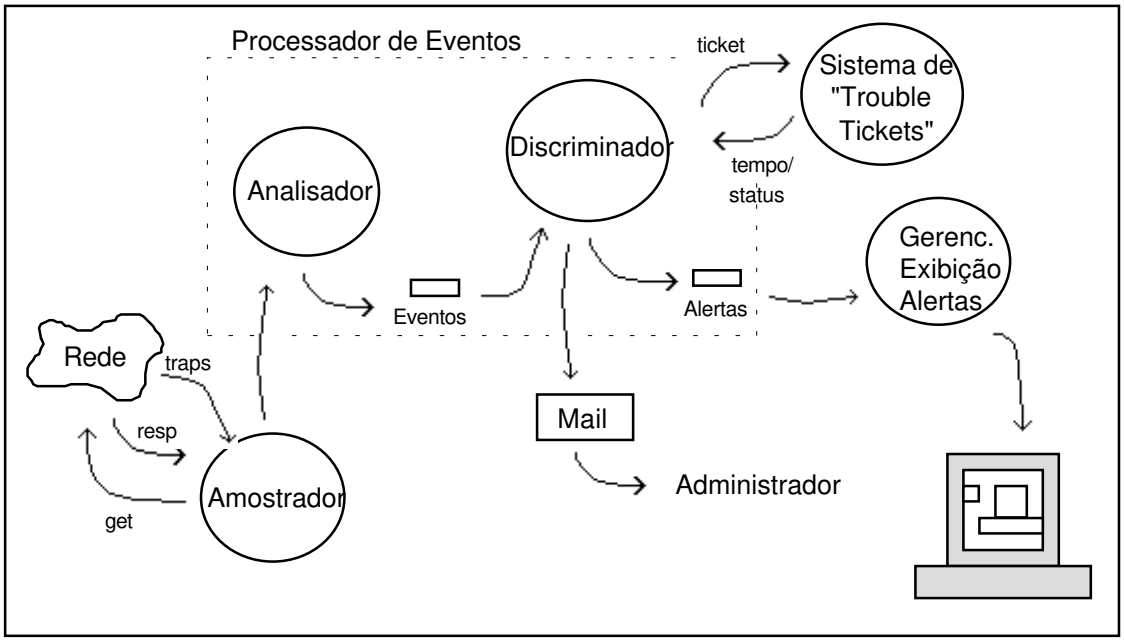

Figura 1

2.1 Processo de coleta de informações

A monitoração da rede se baseia em arquivo de configuração, onde devem são definidas as seguintes informações para cada instância de um objeto monitorado:

- Entidade Monitorada: equipamento a ser monitorado (roteador, ponte, estação de trabalho, etc).
- Objeto: nome completo do objeto a ser monitorado, incluindo a entrada que o objeto de interesse pertence (ex. interfaces.ifTable.ifEntry.ifInOctets.1).
- Expressão: O usuário pode fornecer expressões ao invés de objeto isolados tal como implementado em [KRA94].

• Janela de Amostragem: espaço de tempo utilizado para o cálculo automático dos limites (caso esta seja opção do usuário).
• Origem dos Limites: os limites podem ser calculados automaticamente pelo sistema ou fornecidos pelo usuário.
• Limites Superior e Inferior: valor dos limites informados pelo usuário (caso esta seja opção do usuário).
• Fatores Superior e Inferior: fatores utilizados no cálculo automático dos limites (caso esta seja opção do usuário).
• Tempo de Monitoração: tempo que o sistema ficará monitorando o objeto informado anteriormente.

Caso o administrador não saiba quais são os limites adequados, o sistema pode calculá-los dinamicamente, utilizando o mecanismo de Histerese, o qual será descrito mais adiante.

Além de objetos isolados, o usuário também pode fornecer expressões no arquivo de configuração. A partir de objetos da MIB II (Management Information Base) pode-se criar expressões com as quais se poderá analisar o desempenho da rede de uma forma bastante precisa, e com issso ajudar o administrador a tomar as medidas necessárias para melhorar o seu desempenho. Um exemplo desse tipo de expressão pode ser visto a seguir onde pode-se calcular a taxa de erros de datagramas IP, para tanto utiliza-se objetos da MIB II [MCC 91]:

$$taxa\ de\ erros\ ip = \frac{(ipInDiscards + ipInHdrErrors + ipInAddrErrors)}{ipIn\ Re ceives}$$

Cada expressão representa uma condição a ser inspecionada periodicamente para permitir detectar eventuais problemas na rede.

Após a leitura do arquivo de configuração, a monitoração começa a ser realizada. Cada valor amostrado é decrementado de seu antecessor, dando a certeza ao usuário de que valores obtidos, por serem diferenças, refletem a evolução do indicador desde o início da amostragem. Então esse valor é comparado com limites calculados dinamicamente pelo sistema. Ao efetuar-se a comparação e verificar-se que o valor monitorado está fora do intervalo definido como "normal" para aquela rede, um evento é gerado.

Através da análise dos eventos gerados podem ser ou não criados alertas ao administrador. Para cada alerta será criado um registro de problemas no Sistema de Trouble Ticket, bem como uma mensagem será enviada ao administrador por meio de um mail.

A análise dos eventos é realizada através de uma base de regras, e é ela que decide se um alerta deve ou não ser gerado. Assim, pode-se verificar o problema que realmente está ocorrendo, além de agregar uma recomendação.

2.2 O Mecanismo de Histerese

Este é um mecanismo proposto na RFC 1271 para o grupo de Alarmes da RMON MIB [WAL 91]. Seu funcionamento é definido da seguinte forma: dois limites são definidos para cada objeto monitorado, o limite de subida (ou superior) e o de descida (ou inferior). Para que limites de subida e descida sejam gerados é necessário aplicar certas condições. A figura 2 mostra o funcionamento desse mecanismo para um objeto qualquer que esteja sendo monitorado, este

objeto pode ser, por exemplo, o número de pacotes que estão entrando por uma determinada interface de algum roteador. A partir dos limites de subida e descida é possível criar eventos que avisem ao usuário que algo fora do normal está acontecendo. Os limites indicam o intervalo dentro do qual o valor de um objeto amostrado deve estar quando a rede está no seu estado "normal", ou seja, tudo está funcionando normalmente sem muito atraso e sem a rede estar muito lenta.

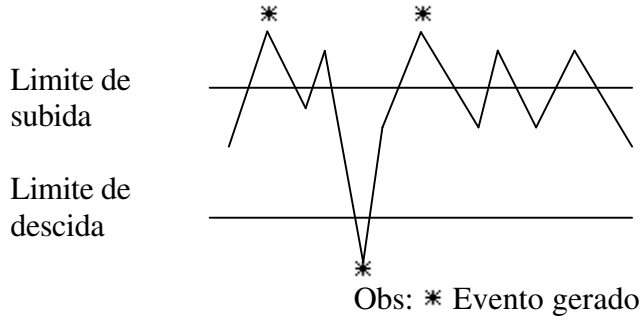

Figura 2 - O mecanismo de Histerese.

Segundo [MAD 94], um evento de subida (ou superior) é gerado se:

- o valor recebido é o primeiro desde o início da amostragem e o flag para ação inicial definido pelo usuário é EVENTO_DE_SUBIDA ou EVENTO_DE_SUBIDA_OU_DESCIDA;
- o valor recebido não é o primeiro a ser amostrado, e este valor é maior ou igual que o limite de subida, sendo que o último valor recebido até então era menor que o limite de subida e o último evento gerado foi um evento de descida.

De um modo complementar, um evento de descida (ou inferior) é gerado se:

- o valor recebido é o primeiro desde o início da amostragem e o flag para ação inicial definido pelo usuário é EVENTO_DE_DESCIDA ou EVENTO_DE_SUBIDA_OU_DESCIDA;
- o valor recebido não é o primeiro a ser amostrado, e este é menor ou igual que o limite de descida, sendo que o último valor recebido até então era maior que o limite de descida e o último evento gerado foi um evento de subida.

O sistema proposto também utiliza este mecanismo para calcular automaticamente o intervalo dentro do qual os valores amostrados devem encontrar-se. O sistema calcula esses limites a partir de uma base de dados que foi previamente criada para a rede, isto é, uma *baseline* (ver Anexo) da rede. Os valores típicos de uma série de objetos da MIB II se encontram nessa base de dados, então o sistema procura o objeto adequado e automaticamente calcula os limites para aquele objeto.

Caso o valor que acabou de ser amostrado encontrar-se fora desse intervalo, um evento indicando uma situação anormal é gerado, conforme algumas condições. Devido a estas condições, eventos idênticos não são gerados sequencialmente.

Conforme a figura 3, uma determinada instância de um objeto pode ultrapassar o mesmo limite várias vezes mas de forma esparsa no tempo sem que o limite oposto seja ultrapassado neste tempo. Levando isso em consideração, o sistema em questão faz algumas extensões ao mecanismo de histerese original. No mecanismo original, apenas um evento seria gerado quando essa situação ocorresse. Então para que mais de um evento possa ser gerado, o sistema considera o tempo decorrido desde o último evento gerado. Se o tempo decorrido for maior que o tempo de espera

definido pelo usuário, um evento idêntico pode ser gerado, caso as demais condições sejam satisfeitas.

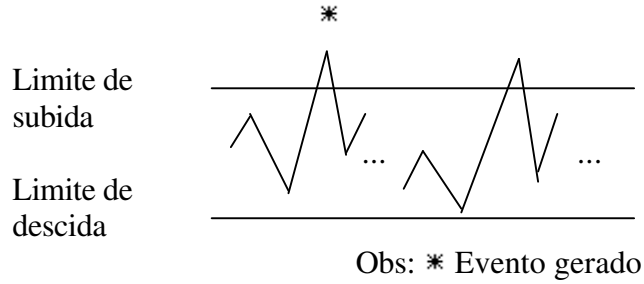

Figura 3 - Mecanismo de histerese original desconsidera tempo entre eventos idênticos.

Outra modificação com relação ao mecanismo original é a não implementação do *flag* de ação inicial. Neste caso, se o primeiro valor amostrado encontrar-se fora dos limites definidos pelo sistema, um flag de ação inicial constante e igual a EVENTO_DE_SUBIDA_OU_DESCIDA será gerado.

É importante também levar em consideração quando limites tendem ao infinito, sendo eles de subida ($+\infty$) ou de descida ($-\infty$), e ainda um igual ao outro. Em cada caso deve-se tomar bastante cuidado para que eventos sejam gerados no tempo correto e segundo as condições definidas inicialmente.

Ainda outra extensão desse mecanismo é o cálculo automático dos limites. As redes acadêmicas em geral não comportam-se da mesma forma durante todo o dia. Geralmente no início da manhã o número de pessoas que estão trabalhando em suas estações é bem menor que no final da manhã, bem como no início e final da tarde onde o tráfego também varia bastante.

Com base nisso o sistema é capaz de calcular os limites dinamicamente. O cálculo é feito de acordo com o tamanho da janela de amostragem definido pelo usuário no arquivo de configuração, que será explicado mais adiante. É muito importante compreender a real implicação de calcular os limites dinamicamente pelo sistema: os limites superior e inferior tornam-se flutuantes ao longo do tempo (principalmente para indicadores de tráfego e outros que sofrem variações no tempo). Além disso, também é importante notar que a distância entre os limites superior e inferior de um mesmo objeto monitorado sempre variam no tempo, exceto para objetos monitorados cujos valores para as instâncias são constantes ao longo do tempo.

Os limites de subida e descida são calculados através da média e desvio padrão dos valores que foram amostrados previamente e encontram-se em uma base de dados (a *baseline* da rede). Desta forma, as expressões utilizadas são as seguintes:

$$LS = \mu + (S \times \sigma)$$
$$LI = \mu + (I \times \sigma)$$

onde:

LS ⇨ limite superior (ou de subida);
LI ⇨ limite inferior (ou de descida);
μ ⇨ média aritmética para os valores amostrados contidos na janela de amostragem;
S ⇨ fator para cálculo do limite superior;
I ⇨ fator para cálculo do limite inferior;

σ ⇨ desvio padrão para valores amostrados contidos na janela de amostragem.

Os fatores de cálculo das expressões acima são providos de tal forma que transformam-se em pontos de customização para a administração da rede na obtenção dos limites. Estes fatores podem ser usados para alargar ou estreitar a faixa na qual os valores amostrados devem situar-se.

2.3 A Reinicialização de Contadores

A maioria dos objetos da MIB II são contadores. Esses objetos podem apenas aumentar seu valor, sendo portanto cumulativos. Mas existe um determinado momento em que eles são reinicializados, voltam para o zero e recomeçam a partir dali, processo conhecido como *wrap around*. Isto acontece quando o valor 2^{32} - 1 ou 4.294.967.295 é atingido conforme [MCC 90]. Um exemplo desse tipo de objeto é o número de octetos que saem de uma determinada interface (ifOutOctets). Este é apenas um motivo porque contadores são inicializados. Além disso, podem ocorrer interrupções durante a contagem de uma monitoração quando, por exemplo, um roteador é reinicializado. Neste caso também há uma reinicialização dos contadores, voltando todos a contagem a partir do valor zero. Ferramentas que se utilizam de monitorações constantes em objetos contadores devem se preocupar em ter alguma forma de determinar quando uma interrupção ocorreu.

Ainda outra situação onde pode ocorrer interrupção dos contadores é quando operações de escritas são efetuadas por diferentes gerentes em um mesmo contador não *read-only* [MAD 94]. Mas como é impossível determinar a ocorrência de operações de escrita em qualquer momento passado, esta situação não é considerada. Logo, é importante diferenciar quando ocorrem *resets* e quando ocorrem *wrap arounds*. Em ambos os casos, quando se estiver trabalhando com diferenciais de contadores em tempos diferentes (t_{i-1} e t_i) esses diferenciais não devem ser baseados em apenas uma subtração simples como $A(t_i) - A(t_{i-1})$, mas sim deve-se levar em consideração o valor obtido antes da reinicialização, com a seguinte expressão:

$$\Delta A_i = (A(Z) - A(t_{i-1})) + A(t_i)$$

onde:

$A(Z)$ ⇨ valor obtido imediatamente antes da reinicialização;
$A(t_{i-1})$ ⇨ valor obtido no instante da última amostragem realizada;
$A(t_i)$ ⇨ valor obtido no instante da amostragem sendo realizada (após a reinicialização).

A forma de detectar quando ocorreu um *reset* e um *wrap around* é bastante simples. Monitorando-se paralelamente o objeto SysUpTime, que indica a quanto tempo o agente está rodando, é possível determinar se uma máquina foi reinicializada ou não. Caso a instância desse objeto seja menor do que a última amostragem, então assume-se que houve a reinicialização do equipamento gerenciado, ou seja, ocorreu um *reset*. Caso contrário, assume-se que ocorreu um *wrap around* no contador.

Sabendo-se que ocorreu um *wrap around* e que isto ocorre quando um contador chega ao seu limite máximo 2^{32} - 1 (4.294.967.295) é possível utilizar-se da seguinte expressão para determinação dos diferenciais.

$$\Delta A_i = ((\text{VALOR MÁXIMO}) - A(t_{i-1})) + A(t_i)$$

Na verdade, é impossível determinar de forma determinística se mais de um *wrap around* ocorreu entre um instante e outro de monitoração se estes forem muito distantes em tempo. Para tanto, é muito importante dimensionar o mais adequadamente possível o intervalo de amostragem de forma que ocorra no máximo uma interrupção a cada instante de amostragem.

Da mesma forma é muito difícil determinar o instante exato antes de um *reset* ocorrer. Portanto, sempre que *resets* são detectados pelo sistema, o valor calculado como indicado acima é colocado em seu lugar.

Pelo que foi estudado e analisado, o intervalo mais adequado entre monitorações parece ser de uma em uma hora, sendo este o intervalo utilizado pelo sistema.

2.4 Janela de Amostragem

O cálculo das médias e desvios padrão descrito anteriormente deve ser feito de acordo com um tempo estipulado previamente no arquivo de configuração. É através do mecanismo de janela de amostragem que o espaço de tempo entre as monitorações é determinado.

Existem dois tipos de janelas de amostragem: as de tamanho cumulativo, incluindo todas amostras efetuadas desde o início da coleta de dados até o momento atual, bem como as de tamanho fixo. As janelas de tamanho fixo podem ainda ser estáticas, quando limites são calculados uma única vez e sem muita freqüência, ou deslizantes, quando limites são calculados cada vez que chegam novos dados à estação de gerência.

Os três tipos de janelas têm suas vantagens e desvantagens, sendo as janelas de amostragem deslizantes as utilizadas no sistema em questão. A principal vantagem desse tipo de janela é o fato dos parâmetros de filtragem sempre serem atuais e auto-adaptáveis a situações de mudança. Assim, o cálculo da média e desvio padrão sempre são recalculados a cada nova amostra recebida. Quando uma nova amostra é recebida e a janela se encontra cheia a amostra mais antiga é descartada e a nova é colocada em seu lugar.

3 Sistema Especialista

Um sistema especialista é um software de solução de problemas que incorpora conhecimento especializado em um domínio limitado para fazer um trabalho que geralmente é feito por uma pessoa muito bem treinada [CRO 88]. Um sistema especialista pode trabalhar com dados incompletos e inexatos, pode lidar com complexidade, pode fornecer explicações de suas conclusões e pode talvez aprender por experiência.

Nem todos os sistemas especialistas possuem todas essas características, mas pode-se dizer que um sistema especialista tem a capacidade, dado algum estado de um processo, de dizer o que fazer, baseado em um conhecimento que pode ser aplicado para aquela situação.

Em geral, sistemas especialistas aplicam técnicas de Inteligência Artificial, tendo como núcleo de tais sistemas uma base de conhecimento, que consiste numa coleção de fatos, definições e regras heirísticas, adquiridas diretamente do especialista humano [TAR 90].

O uso de heurísticas é uma característica chave de um sistema especialista, as quais servem para pesquisar as soluções de problemas. Essas heurísticas são basicamente "regras" que especialistas no problema utilizam para resolvê-los [PAS 86]. Além disso, a linguagem de especificação de um sistema especialista é tipicamente um conjunto de heurísticas na forma de regras IF-THEN. Essa linguagem é declarativa (isto é, ela especifica o que fazer, ao contrário das linguagens procedurais que especificam como fazer alguma coisa).

Em um sistema de produção, uma regra é, simplesmente, um par *condição-ação*; dada a existência da condição expressa, faça a ação. As regras são expressas da seguinte forma:

IF <*condição*> THEN <*ação*>

Sistemas especialistas são compostos de uma Base de Conhecimento, na forma de heurísticas ou um conjunto de regras, e utilizam um mecanismo que faz inferência através da aplicação de regras baseadas em conhecimento. É através de inferências, análises baseadas em fatos e premissas, que o sistema consegue tirar suas conclusões. Tendo um ambiente baseado em regras, a inferência determina quais regras são aplicáveis e quais destas regras deveriam ser usadas em uma determinda situação.

3.1 Gerenciamento de Redes utilizando Sistema Especialista

Um dos maiores avanços da área de gerenciamento de redes foi devido ao uso de sistemas especialistas. Por exemplo, na área de gerenciamento de falhas, sempre que um problema acontece, o sistema tenta tomar diversas decisões, analisando histórico de ocorrências do sistema, o que reduz em muito os trabalhos rotineiros executados por operadores de rede [BRI 93].

A utilização de sistemas especialistas no gerenciamento de uma rede pode ajudar em muito o administrador da mesma, dada a complexidade que é gerenciá-la. Através de sistemas especialistas é possível, por exemplo, prever a carga e performance de uma rede, com isso, problemas podem ser antecipados e ações corretivas podem ser tomadas antes do problema realmente acontecer.

O diagnóstico de um sistema especialista tenta inferir a causa de um problema através dos sintomas reconhecidos nos dados que foram coletados. Diagnósticos podem isolar falhas de uma rede e, uma vez uma falha tenha sido localizada, o usuário pode ser avisado para tomar a devida providência.

É importante que um sistema especialista armazene informações relativas a erros, falhas e outras condições problemáticas de uma rede. É também muito importante ter armazenado estas informações na forma de limiares, que quando ultrapassados, determinam uma sinalização ao operador ou o início de uma ação corretiva. Segundo [TAR 90], um determinado limiar pode ser aceitável numa situação de carga leve na rede, mas intolerável numa outra situação, de carga mais intensa, onde o número de retransmissões faria com que o tráfego total excedesse a capacidade do enlace, afetando seriamente o tempo de resposta. Portanto, deve-se dispor de estatísticas de erros em função do tráfego existente e não apenas de valores absolutos.

Além disso, cada nível de diagnóstico deve ter um roteiro próprio, tendo, assim, passos ou etapas a serem cumpridos em cada nível. Logo, pode-se ter a necessidade de emitir alertas se certos limiares forem atingidos sem que o problema seja solucionado. É importante para o gerenciamento de uma rede que o sistema consiga tomar ações corretivas e prover subsídios para apoiar a análise de tendências do comportamento dos componentes da rede.

Para se determinar tendências do comportamento de uma rede é preciso analisar os dados que foram coletados da mesma. Desta forma, pode-se evitar que problemas se tornem muito graves, antecipando medidas a serem tomadas. Assim, tem-se um sistema que "aprende" com o passar do tempo e com a ocorrência de eventos.

3.2 O Processador de Eventos

A medida que eventos são gerados pelo sistema, eles são passados para um módulo, o processador de eventos, onde é submetido a uma base de regras e então verificado se há necessidade de se gerar um alerta ou não. Estas regras são construídas com base na experiência prática dos administradores da rede.

Para cada evento analisado pode haver o incremento de contadores em função da chegada de eventos idênticos, a geração de alertas ou pode ocorrer o descarte daquele evento sendo analisado.

As regras do sistema são implementadas na forma de comandos *if...then...else*, e elas estão voltadas à gerência de desempenho e de falhas. Abaixo encontram-se algumas regras que foram implementadas para verificar o andamento da rede da UFRGS e foram analisadas no intervalo de uma hora:

• Verificação do estado de uma interface

É possível identificar quando uma interface está *down* através da observação de dois objetos da MIB II, ifOperStatus (estado operacional da interface) e ifAdminStatus (estado administrativo da interface). Para tanto a seguinte regra é utilizada:

IF (ifOperStatus = *down*)
THEN IF (ifAdminStatus = *up*)
THEN IF (após três tentativas em intervalos de cinco minutos a interface continuar *down*)
THEN (Manda um aviso ao administrador do sistema através de um mail dizendo que a interface de determinada máquina está *down* e a hora que isto foi observado.)

Quando uma interface está *down* pode-se dizer que tem-se uma situação bastante crítica e com isso diminuiu-se o tempo de observação daquela interface que encontra-se neste estado. Se nada de anormal é verificado com uma interface, o intervalo de monitoração continua sendo de uma hora. Uma interface pode estar *down* por motivos como: falta de portadora ou o "outro lado" não enviou um *keepalive* em dez segundos (isto é uma característica da CISCO onde ambos os lados enviam uma mensagem informando que estão vivos).

• Percentual de ocupação de uma interface

O percentual de ocupação de uma rede vai depender principalmente da capacidade efetiva de transmissão na rede, dependendo com isso da velocidade que uma determinada interface é capaz de transmitir. Para uma rede Ethernet constuma-se utilizar um percentual de até 30%, após este limite a rede pode tornar-se congestionada. Já no caso de se tratar de uma interface serial esse limite aumenta, passando para 60%. Este percentual é maior pois a velocidade de transmissão de uma interface serial é bem menor do que de uma Ethernet.

Quando percentuais acima desses limites forem verificados no sistema em questão, o administrador é avisado e caso o problema persista uma medida deverá ser tomada, tal como, confinamento do tráfego ou aumento de memória e velocidade de processamento nos servidores da rede.

Uma rede pode tornar-se sobrecarregada quando o número de usuários aumentar em determinado segmento de rede ou uma aplicação começar a ser muito utilizada.

• Percentual de erros de uma interface

Pacotes com erros recebidos ou enviados para uma interface podem ser devido a colisões, ou problemas em equipamentos como *tranceivers* ou *drivers* que controlam a placa de rede. Além disso, a linha pode estar com problemas ou então aquela interface pode estar enviando ou recebendo pacotes maiores do que ela é capaz de enviar ou receber. É possível verificar se isso está ocorrendo consultando a variavel ifMtu, a qual indica o tamanho do maior datagrama que pode ser enviado ou recebido pela interface [MCC 91]. Erros ocorrem mais frequentemente em linhas seriais, onde a maior causa deles é devido ao ruído.

A rede Ethernet, por ser mais confiável, deve possuir um percentual de no máximo 1% no intervalo de uma hora. O mesmo não ocorre com linha serial, onde a taxa de erros é mais elevada.

Levando isso em cosideração foi criada a seguinte regra:

IF (((interface = "Ethernet") AND (taxa de erros > 1% no intervalo de uma hora)) OR
(interface = "Serial") AND (taxa de erros > 7% no intervalo de uma hora)))
THEN envia aviso ao administrador recomendando-o verificar a taxa de colisões, pois elas podem estar gerando erros, bem como verificar a carga da rede. O percentual de erros pode aumentar a medida que o tráfego da rede aumenta.

A taxa de erros da interface serial foi definida levando-se em conta os valores obtidos da *baseline* da rede, sendo que esta taxa irá variar de segmento para segmento.

• Percentual de pacotes ICMP enviados ou recebidos por uma interface

Um número excessivo desse tipo de pacote pode degradar o desempenho de uma rede. Enquanto durante períodos de tráfego normal o poder de processamento consumido pode ser mínimo, em horas mais ocupadas, o envio de grandes números de pacotes ICMP podem requerer recursos suficientes para sensivelmente atrapalhar o desempenho da entidade, [LEI 93]. Para identificar o excesso desse tipo de pacote foi utilizada a seguinte regra:

IF ((taxa de pacotes ICMP > 2%) AND (após três tentativas em intervalos de cinco minutos a taxa continuar alta))
THEN (Avisar ao administrador que um número de pacotes ICMP está alto, gerando um registro de problemas)

Esta taxa pode aumentar quando uma máquina não consegue encontrar uma outra pois a interface que chega até ela está *down.*

É importante notar que nem sempre a recepção ou envio de pacotes ICMP podem significar que um problema de desempenho exista, mas possuir essas estatísticas pode ajudar a resolver um problema futuro.

• Verificação de reinicialização do roteador

Para gerência de falhas pode-se verificar se um roteador está sendo reinicializado muito freqüentemente. Isso pode estar ocorrendo devido a falta de energia elétrica, mas também pode ser devido a problemas internos ao roteador. A principal forma de descobrir isso é verificando o contador sysUpTime daquele roteador. Se for verificado que ele está sendo reinicializado em intervalos irregulares e freqüentes, deve-se verificar se outras máquinas, que estão próximas a ele, também foram reinicializadas. Se acontecer dessas máquinas não terem sido reinicializadas pode-

se dizer que o problema não é por falta de energia e sim por algum problema interno que está ocorrendo no roteador. Deve-se então avisar ao administrador da rede do problema encontrado além de criar um registro de problemas para ele.

Para tanto a seguinte regra é utilizada:

IF ((roteador_reinicializado) AND (não houve falta de energia))
THEN (Alerta o administrador que o roteador está sendo reinicializado por motivo interno e cria registro de problemas.)

Todas essas regras foram definidas levando-se em consideração os objetos da MIB II. Para tanto, foram selecionados objetos que são aplicados na gerência de desempenho e de falhas. Ainda é possível criar regras como por exemplo para taxa de pacotes *broadcast*, percentual de pacotes descartados, entre outras.

É muito importante saber identificar todas as possíveis causas pelas quais um problema pode ser gerado. O sistema deve fazer vários testes antes de determinar a causa de algum problema para então avisar o usuário. Assim, o administrador conseguirá corrigir o problema de uma maneira mais rápida e precisa, sem precisar ficar perdendo tempo testando vários elementos de rede para descobrir a causa do problema. Um exemplo disso pode ser a alta taxa de erros como mencionado anteriormente.

Uma variável estar fora dos limites pode acarretar em vários problemas e todos eles devem ser levados em consideração quando o usuário precisar ser avisado que algo de errado ocorreu. Só assim será possível corrigir o problema de modo que a rede a volte sua operação normal o mais rápido possível.

4 Aspectos da Implementação

O sistema de Alertas utiliza o protocolo SNMP (Simple Network Management Protocol) para consultar o valor dos objetos selecionados para as monitorações. O SNMP é um protocolo de "pedido/resposta", sendo os pedidos formados por operações, tais como *get*, *get-next* e *set*. Para sua utilização foi utilizado o pacote de software CMU da Carnegie Mellon University. Este é um pacote de domínio público que trabalha com a linguagem C e pode ser obtido por ftp anonymous.

Inicialmente o sistema desenvolvido possui uma base de dados com a *baseline* da rede para alguns objetos da MIB II. A tabela 1 apresenta a média de três objetos que fazem parte da *baseline* de um dos roteadores da UFRGS, o roteador “routcc.ufrgs.br”.

máquina	**hora**	**ifInUcastPkts**	**ifInOctets**	**ifOutErrors**
routcc.ufrgs.br	01-00	20671	669386668	0
routcc.ufrgs.br	02-01	16711	602228342	0
routcc.ufrgs.br	03-02	15076	1642590574	0
routcc.ufrgs.br	04-03	11451	331755135	0
routcc.ufrgs.br	05-04	12281	76863259	0
routcc.ufrgs.br	06-05	7805	1578565693	0
routcc.ufrgs.br	07-06	11718	332992787	0
routcc.ufrgs.br	08-07	20488	395949751	0
routcc.ufrgs.br	09-08	32862	927888820	0
routcc.ufrgs.br	10-09	67765	140418596	0

routcc.ufrgs.br	11-10	104142	651624254	0
routcc.ufrgs.br	12-11	101876	1486858883	0
routcc.ufrgs.br	13-12	79598	1077174639	0
routcc.ufrgs.br	14-13	98111	531428627	0
routcc.ufrgs.br	15-14	110952	961016833	0
routcc.ufrgs.br	16-15	134867	716550474	0
routcc.ufrgs.br	17-16	116927	276037381	0
routcc.ufrgs.br	18-17	80231	672475851	0
routcc.ufrgs.br	19-18	58956	660351667	0
routcc.ufrgs.br	20-19	37589	335617578	0
routcc.ufrgs.br	21-20	24142	664407946	0
routcc.ufrgs.br	22-21	23208	658692718	0
routcc.ufrgs.br	23-22	26082	69901629	0

Tabela 1 - Médias dos valores monitorados

Para tanto, foram feitas monitorações durante um mês, sendo o valor dos objetos consultados de uma em uma hora, através de operações de *get* do SNMP. Após cada dia de monitoração os valores foram decrementados uns dos outros para se obter o quanto realmente foi alterado de uma hora para outra e então esses valores decrementados foram armazenados em outra base de dados. Com isso, quando novos objetos devem ser monitorados, é essa base de dados que será utilizada para o cálculo das médias e desvios padrão.

Para formar a *baseline* foram selecionados objetos dos seguintes grupos: interfaces, ip, tcp, udp e icmp. Os objetos escolhidos para cada grupo foram aqueles relacionados a gerência de desempenho e de falhas, tal como, ifInUcastPkts, ifInOctets, ifOutOctets, ipInDiscards, entre outros. Tais monitorações foram efetuadas nos servidores principais da universidade, bem como em alguns roteadores.

Para monitorar o grupo de interfaces de cada máquina foi verificado, primeiramente, quantas interfaces cada máquina possuia e assim foi possível consultar todas as interfaces das máquinas em questão.

A medida que as monitorações são efetuadas, elas são armazenadas em um banco de dados. Para tanto, utilizou-se o sistema de banco de dados POSTGRES [STO 90], mantendo assim compatibilidade com o sistema de Trouble Tickets, que utiliza o mesmo sistema.

Depois da *baseline* da rede ter sido criada, utiliza-se o sistema para monitorar determinados objetos por um período de tempo e quando algo estiver fora do "normal", isto é, quando os dados consultados não estiverem de acordo com a *baseline*, um evento é criado podendo ser gerado um alerta ou não. Para tanto, deve-se preencher um arquivo de configuração, o qual está descrito na primeira seção. Caso o objeto que o usuário deseja monitorar não encontrar-se na *baseline* da rede, este pode ser monitorado durante um mês e então ser incluído na *baseline*. Além disso é possível substituir uma *baseline* que já está antiga, criada a alguns meses atrás, por outra mais atual.

Quando o sistema de Alertas consulta a base de regras e verifica que um alerta deve ser enviado ao usuário, ele também consulta a base de *trouble ticket* para verificar se o mesmo problema já aconteceu anteriormente e qual solução foi utilizada para corrigi-lo. Desta forma o sistema pode aprender maneiras diferentes para corrigir um mesmo problema.

A confiabilidade foi considerada pelo Sistema de Alertas. São utilizados arquivos de *log* para manter as informações sobre eventos e alertas que são gerados. Assim, evita-se de perder informações quando, por exemplo, ocorrer uma falta de energia elétrica. Sempre que o sistema

reinicia sua execução, ele procura por arquivos de *log* no disco. Caso estes arquivos existam, eventos serão os primeiros a serem processados e alertas serão indicados ao administrador da rede. Se o sistema terminar sua execução normalmente, arquivos de *log*, que foram criados anteriormente, serão removidos.

5 Conclusão

O sistema ora proposto visa oferecer as seguintes funcionalidades:

- Funcionar como uma espécie de planilha hospitalar ou memória/buffer para registro de ocorrências e da situação da rede a cada instante;
- Organização e atribuição de tarefas (quem esta fazendo o que, quem pode fazer o que);
- Indicação de pessoas responsáveis pelos diversos segmentos da rede e fornecer-lhes subsídios para que possam cooperar na solução dos problemas;
- Vigilância contínua sobre a rede, inspecionando parâmetros críticos e alertando as pessoas apropriadas quando necessário;
- Fornecer uma visão geral da situação a técnicos e usuários;
- Apoiar análise estatística e estudo de tendência da rede;
- Filtrar alertas para não sobrecarregar o operador da rede;
- Registrar o uso (duração e interrupções) das facilidades da rede para fins de contabilização;
- Prover uma imagem da rede para o Centro de Operações e Controle.

6 Bibliografia

[BRI 93] BRISA - Sociedade Brasileira para Interconexão de Sistemas Abertos. **Gerenciamento de Redes - Uma Abordagem de Sistemas Abertos**. São Paulo, Makron Books do Brasil Editora Ltda, 1993.

[CRO 88] CRONK, Robert N., CALLAHAN, P. H. e BERNSTEIN, L. Rule-Based Expert Systems for Network Management and Operations: An Introduction. **IEEE Network**. p. 7-21, Sep. 1988.

[KRA 94] KRAHE, Fernando. **Um Avaliador de Expressões para Gerenciamento de Desempenho**. Porto Alegre: CPGCC - UFRGS, 1994. Trabalho de Conclusão.

[LEI 93] LEINWAND, A. e FANG, K. **Network Management: a practical perspective**. EUA, Addison-Wesley, 1993.

[MCC 90] McCLOGHRIE, K.; ROSE, M. T. **Structure and Identification of Management Information for TCP/IP-based Internets**. DDN Network Information Center, SRI International. Request for Comments 1155, May, 1990. 91 p.

[MCC 91] McCLOGHRIE, K.; ROSE, M. T. **Management Information Base for Network Management of TCP/IP-based Internets: MIB-II**. DDN Network Information Center, SRI International. Request for Comments 1213, March, 1991. 70 p.

[MAD 94] MADRUGA, Ewerton L. **Ferramentas de Apoio à Gerência de Falhas e Desempenho em Contexto Distribuído**. Porto Alegre: CPGCC - UFRGS, 1994. Dissertação de Mestrado.

[MAD 93] MADRUGA, E. L e TAROUCO, L.M.R. Trouble Ticketing in a Cooperative Integrated Network Management Enviroment. In: IV IFIP/IEEE INTERNATIONAL WORKSHOP ON DISTRIBUTED SYSTEMS: OPERATIONS AND MANAGEMENT, Oct.5-6, 1993, Long Branch, NJ, EUA. **Case Studies**. Long Branch, NJ:[s.n], Oct, 1993, DSOM'93.

[PAS 86] PASQUALE, Joseph. **Knowledge-Based Distributed Systems Management**. University of California, Berkeley, 1986. (Relatório N° UCB/CSD 86/295).

[ROS 90] ROSE, Marshall. **The Simple Book: An Introduction on to Management of TCP/IP - based Internets**. Englewood Cliffs: Prentice-Hall, 1990.

[SNG 90] SNG, D.C.H. **Network Monitoring and Fault Detection on the University of Illinois at Urbana-Champaign Campus Computer Network**. Urbana-Champaign, IL, EUA: DCS/UIUC, 1990. (Relatório Técnico UIUCDCS-R-90-1595).

[STO 90] STONEBRAKER, M. **POSTGRES Reference Manual**. Berkeley, CA, EUA: University of California, 1990.

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

[WAL 91] WALDBUSSER, S. **Remote Network Monitoring Management Information Base**. Carnegie Mellon University. Request for Comments 1271, Nov. 1991. 81 p.