DESCENTRALIZAÇÃO DO MODELO SNMP ATRAVÉS DA UTILIZAÇÃO DE MULTI-GERENTES DISTRIBUÍDOS COM ÊNFASE A RELATÓRIO DE EVENTOS

Augusto José Venâncio Neto [1]
augusto@fesurv.br

Elizabeth Sueli Specialski[2]
beth@inf.ufsc.br

Eduardo Coelho Cerqueira[3]
dudu@fesurv.br

Hélio Corrêa Filho[4]
helio@fesurv.br

Maxwell Silva Carmo[5]
max@fesurv.br

[1,3,4,5]Fesurv – Fundação do Ensino Superior de Rio Verde
Departamento de Ciência da Computação
Setor Universitário - Cx. Postal 104
CEP 75901-970 Rio Verde - Goiás.

[2]UFSC-CTC-INE
CEP 88040-900 - Campus Universitário Cx.p. 476
Florianópolis S.C.

RESUMO

Este artigo tem como objetivo apresentar um modelo para descentralização da arquitetura de gerenciamento SNMP, utilizada em redes internet. Este processo se dá através do uso de multi-gerentes distintos, independentes e distribuídos em máquinas diferentes. Outro objetivo desta pesquisa é aplicar ao modelo, a ênfase à recepção de relatórios de eventos (*Traps/Informs*), e não à técnica de *polling* como é feito atualmente. A adoção destas técnicas permite aumentar a eficiência da gestão de redes IP, alcançando-se: diminuição de *overhead* nas estações de gerenciamento; realização simultânea de tarefas de gerenciamento em cada gerente; considerável diminuição do tempo para solucionar problemas; e aumentando do grau confiabilidade do ambiente. O estudo desta viabilidade foi realizado através da avaliação do uso de 6 protótipos desenvolvidos na linguagem Java. Destes, 5 são aplicações gerentes distintas, diferenciadas através das áreas funcionais de gerência definidas pela ISO (BRISA , 1993).

Palavras Chave: Gerência de redes de computadores, SNMP, Discriminador de repasse de eventos e Java.

ABSTRACT

This article shows a model in order to decentralize the SNMP management-architecture, used for IP-networks. This task is made trough the use of multi-managers different, independent geographically distributed machines. Another

objective of this research is apply to the model, the event report (Traps/Informs) emphasizing, different as made currently where the polling technique is frequently used. The Adoption of these technique enable the increase of network-management efficiency, getting: overhead decrease in management stations; simultaneous management tasks on each manager; decrease of time in order to solve problems; and increase of environment trust level. The study of this viability was made through the evaluation of 6 prototypes developed in java language, where 5 are different manager applications being differentiated through management functional areas, defined by ISO (BRISA , 1993).

1 INTRODUÇÃO

O contexto de gerenciamento de redes de computadores tem sido bastante explorado através de pesquisas tendo diversos modelos, arquiteturas e protocolos apresentados ao mercado. As adoções de novas tecnologias, aumento da quantidade de usuários e distribuição geográfica contribuem para o aumento da complexidade de uma rede, tornando imprescindível sua gestão.

Novas técnicas de Engenharia de Software, Inteligência Artificial, Redes Neurais, Raciocínio Baseado em Casos e Mineração de Dados, permitiram aplicar aos atuais NMS´s (Sistemas de Gerenciamento de Redes) alto grau de robustez e eficiência através da implementação de novas funcionalidades, como exemplo, tarefas automatizadas sem a necessidade da interação direta do administrador da rede. Entretanto, a execução de processos grandes e complexos contribui diretamente para o aumento de *overhead* da máquina hospedeira, necessitando a realização de um *upgrade* de processador e memória, a fim de evitar atrasos na solução de problemas identificados, podendo comprometer o funcionamento de todo o ambiente.

A solução aqui proposta é dividir o sistema de gerenciamento central em sub-sistemas, definir cada um através de domínios de gerenciamento e distribuí-los em máquinas diferentes. Com isso, além de conseguir um maior aproveitamento das máquinas e diminuição do *overhead*, torna-se possível que cada sub-sistema gerencie e solucione problemas da rede simultaneamente.

Uma outra técnica a ser incorporada ao modelo seria a ênfase à recepção de alarmes. Hoje em dia existe uma grande quantidade de alarmes proprietários capazes de fornecer informações muito mais precisas e abrangentes do que ficar monitorando o ambiente através de sucessivos e numerosos *pollings*, causando com isso maior tráfego de pacotes na rede e *overhead* na estação de gerenciamento, implicando na queda drástica do desempenho na tarefa de gerenciamento.

O problema em enfatizar o modelo SNMP à recepção de *Traps* está na ausência de uma ferramenta nativa para filtragem destes. Isto se dá através da associação entre regras de filtragem e conteúdo de cada *Trap*, para tomada de decisão. Esta decisão consiste na determinação do envio ou não e identificação de destinatários, no caso de envio. Com isso, evitar-se-ia o envio de *Traps* desnecessários a NMS´s de incompatíveis domínios de gerenciamento.

Reforçando a solução descrita acima, grandes empresas adicionaram em seus NMS´s módulos funcionais com características semelhantes. Tanto a *Sun* quanto a *IBM* oferecem poderosas ferramentas, o *SunNetManager* e o *Tivoli NetView* respectivamente (SUN, 1996). Contudo, ambos são executados diretamente na estação de gerenciamento não impedindo o recebimento de *Traps*, e consequentemente gargalos e *overhead* (IBM, 2000). Em contrapartida, um *applet* denominado *TrapConsole* pode estar em um equipamento diferente da estação de gerenciamento, impedindo com isso a chegada de determinados *Traps*, entretanto a ferramenta é incapaz de determinar, por si só, o(s) destinatário(s) necessitando da decisão do administrador (CSCARE , 1999).

O restante deste artigo está organizado da seguinte forma: o capítulo 2 mostra a arquitetura de gerenciamento SNMP, os tipos de *Traps* padronizados e tece conceitos de Discriminadores OSI; o capítulo 3 a solução proposta, sua implementação, considerações sobre seu desenvolvimento, sua arquitetura, funcionamento, os módulos componentes e detalha o processo de discriminação; o capítulo 4 descreve os testes, suas etapas e seus respectivos resultados para validação do protótipo; o capítulo 5 traz as conclusões, considerações finais e trabalhos futuros e o capítulo 6 apresenta a bibliografia utilizada.

2 SNMP E DISCRIMINADORES

No início de 1988, o IAO, órgão que regulamenta os padrões na Internet, adotou o SNMP como uma solução imediata, padronizando-o como protocolo para gerência de redes IP devido à sua simplicidade, e o CMOT como uma solução em longo prazo (SUN, 1996). Entretanto, logo se percebeu a completa inadequação do CMOT com a arquitetura Internet, dando todo o espaço para o SNMP.

2.1 A ARQUITETURA SNMP

A arquitetura SNMP é uma coleção de estações de gerenciamento, equipamentos que abrigam os módulos gerentes responsáveis pela monitoração e controle dos elementos gerenciáveis, e elementos de rede, equipamentos que abrigam os agentes responsáveis por coletar informações sobre as atividades relacionadas com a rede, armazenar estatísticas localmente e responder às solicitações do gerente (MAURO ; SCHMIDT , 2001). Seu modelo de gerenciamento está representado pela fig. 2.1.

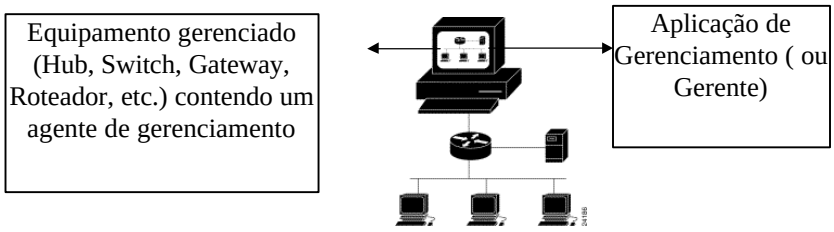


Figura 2.1 – Modelo de gerenciamento SNMP

A comunicação agente/gerente se dá através do SNMP de duas maneiras (BATES , 2002):

4. Polling: Solicitação ao agente para leitura ou escrito de variáveis de gerenciamento;
5. Relatório de Evento: envio de um alarme ao gerente.

Ambas são realizadas através de operações definidas pelo protocolo, denominadas PDU. Para as versões 1 e 2 do SNMP existem as seguintes (CASE , 1996): GetRequest, solicitação de leitura de uma variável de gerenciamento; GetNextRequest, leitura do valor da variável cujo OID é lexicograficamente o próximo daquele OID contido na PDU; SetRequest, alteração do conteúdo de uma variável; GetResponse, resposta a um Get, GetNext ou SetRequest; e Trap/inform, notificação de um evento fora do comum ocorrido na rede.

2.1.1 TIPOS DE TRAPS

Sete *Traps* foram padronizados (MAURO ; SCHMIDT , 2001):

- **coldStart (0)** – Reinicialização de um dispositivo com causa especificada;
- **warmStart (1)** – Reinicialização de um dispositivo sem causa especificada;

- **link-Down (2)** – Falha em um dos links de comunicação do dispositivo;
- **link-Up (3)** – Reativação do(s) link(s) de comunicação do dispositivo;
- **authenticationFailure (4)** – Detecção de uma mensagem SNMP com comunidade inválida;
- **egpNeighborLoss (5)** – Desativação do Link EGP entre *gateways*;
- **enterpriseSpecific (6)** – Evento proprietário do fabricante do dispositivo.

2.2 DISCRIMINADOR DE REPASSE DE EVENTOS

Trata-se de um elemento definido para ambientes OSI de gerenciamento, tendo como principais objetivos (CCITT , 1993):

4. Selecionar os relatórios de eve4ntos que devem ser enviados a um sistema de gerenciamento particular;
5. Determinar os destinatários para os quais os relatórios de eventos devem ser enviados;
6. Controlar (suspender e retomar) o repasse de relatórios de eventos;
7. Possibilitar que um sistema de gerenciamento externo modifique as condições de emissão de relatórios de eventos;
8. Designar endereços alternativos.

FIGURA 2.2 - Modelo do gerenciamento de relatórios de evento

3 SOLUÇÃO PROPOSTA

O modelo apresentado neste trabalho propõe o uso de mais de um gerente na rede e a necessidade de um mecanismo que permita determinar os destinatários para os quais um *Trap* particular deva ser encaminhado. Desta forma consegue-se distribuir o modelo SNMP e aumentar a ênfase à utilização da técnica de relatório de eventos.

3.1 MODELO GENÉRICO

O modelo proposto contempla a inserção de um discriminador no modelo SNMP. A principal dificuldade reside na inserção deste objeto como parte do código do agente. Devido a isso, optou-se pela inserção de um ou mais objetos discriminadores interceptando a comunicação gerente/agente. A vantagem é evitar qualquer modificação no código dos agentes e permitir a sua utilização em qualquer ambiente de gerenciamento SNMP, mesmo que já tenha sido instalado, com poucas configurações adicionais. É importante ressaltar que, quanto mais próximo fisicamente dos agentes um discriminador estiver, menor será o tráfego de informações desnecessárias na rede. A decisão da localização do discriminador é, portanto, fortemente dependente da configuração topológica da rede. A figura 3.1 ilustra o modelo genérico proposto.

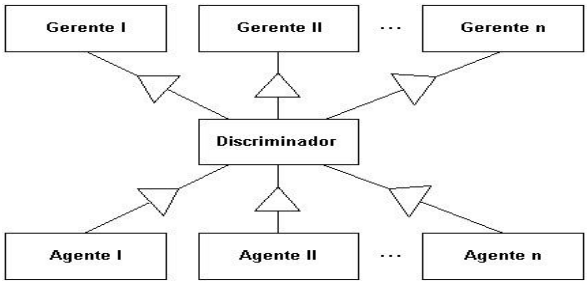


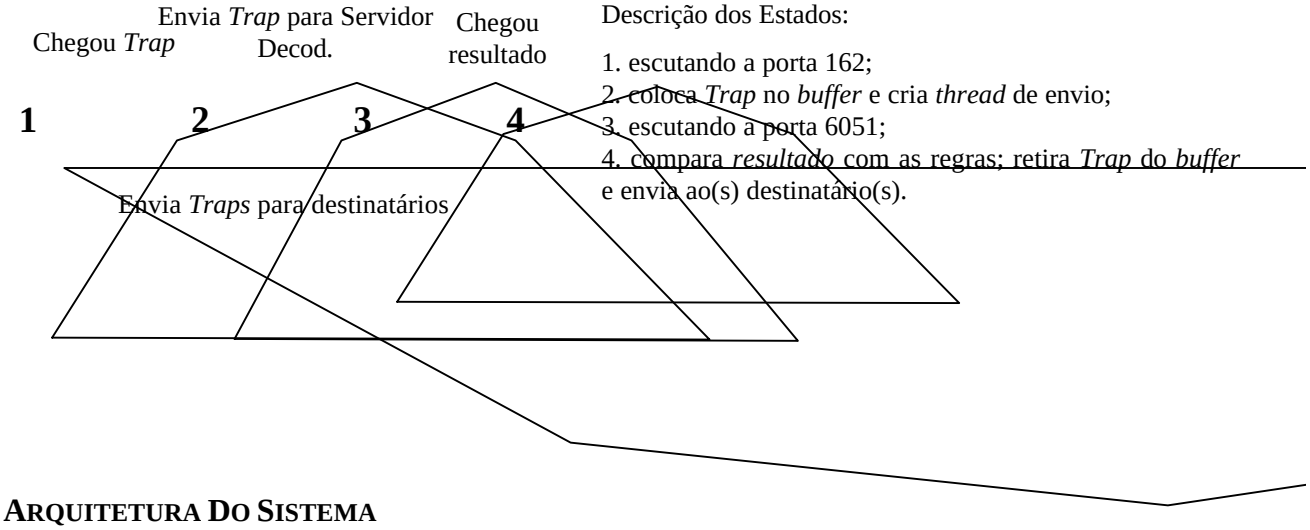
FIGURA 3.1 Modelo genérico de discriminação

Apesar de se propor um modelo distribuído, a adoção do discriminador mantém a visão de um modelo centralizado para os agentes e, a primeira questão sobre este modelo diz respeito ao desempenho do discriminador. Para resolver este problema, foi adicionado ao sistema um módulo que verifica a eficiência de discriminação obtida, em termos de *Traps* emitidos vs *Traps* processados. No caso de situações de carga excessiva (além da capacidade do discriminador), é recomendável a adição de outro discriminador e a divisão dos agentes em grupos. Esta prática, apesar de demandar um esforço extra de configuração dos agentes, garante o desempenho e a confiabilidade do sistema de discriminação.

O discriminador apresenta as seguintes características, funcionalidades e necessidades:

4. Um endereço IP para o qual os agentes vinculados deverão encaminhar os *Traps*;
5. Um critério de discriminação definido através de regras;

6. Transparência, o gerente recebe o *Trap* da mesma forma como receberia do agente, não implicando em nenhuma alteração no código da aplicação de gerenciamento receptora;
7. lexível, inserção de tantas regras quantas forem necessárias para discriminação;
8. Possibilidade de direcionamento do envio, ou não, de *Traps* para uma quantidade ilimitada de gerentes diretamente no discriminador, independente do agente.



3.2 ARQUITETURA DO SISTEMA

A complexidade de implementação dos códigos de estabelecimento/encerramento de comunicação, codificação/decodificação de mensagens e métodos de busca e tomada de decisão, implicaria em significativo grau de dificuldade da organização estrutural do código fonte do protótipo. Sendo assim, optou-se pela utilização de dois módulos independentes e distintos:

- Discriminador – Responsável pela recepção dos *Traps*, e aplicação das regras de discriminação com posterior determinação de destinatários.

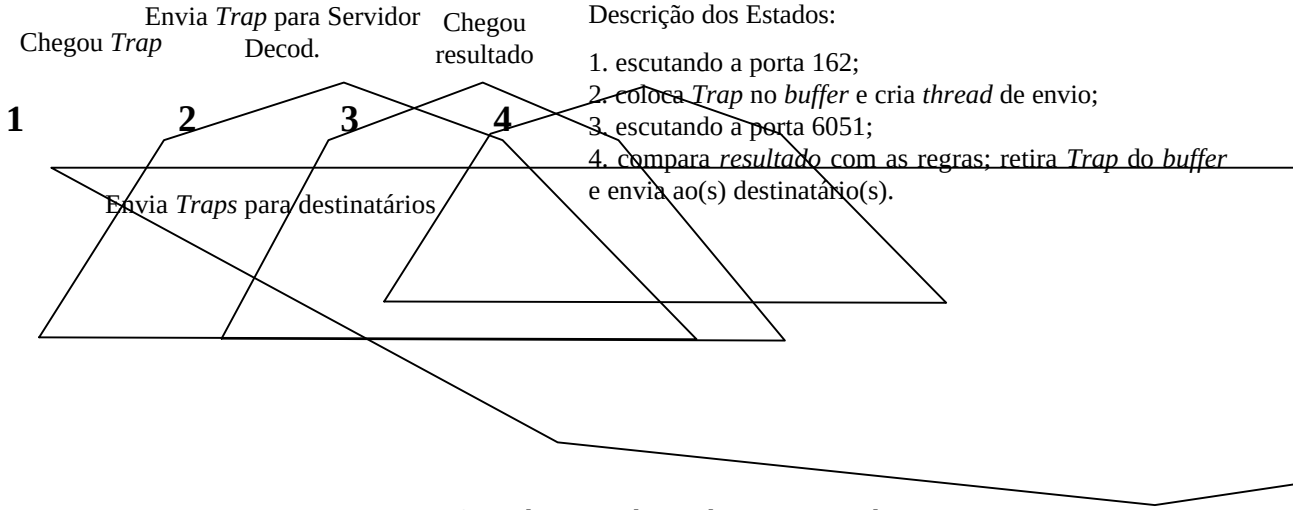


Figura 3.2 – diagrama de estados Discriminador

Ao ser iniciado, o Discriminador cria duas *threads*, uma para conexão com a porta 162 para chegada de *Traps*, e outra com a porta 6051 para chegada do objeto *resultado*. Na chegada de um *Trap*, primeiramente este é colocado em um *buffer* e depois é criada uma outra thread para enviar uma cópia deste *Trap* para o servidor decodificador sendo finalizada após o envio. Ao chegar o objeto *resultado*, uma *thread* é então criada para discriminação do *Trap*. Determinado o(s) destinatário(s), esta mesma *thread* envia o *Trap* para ele(s), finalizando-se ao final.

- Servidor Decodificador – Responsável por decodificar a mensagem SNMP devolvendo ao discriminador um pacote contendo o código genérico e específico do *Trap*.

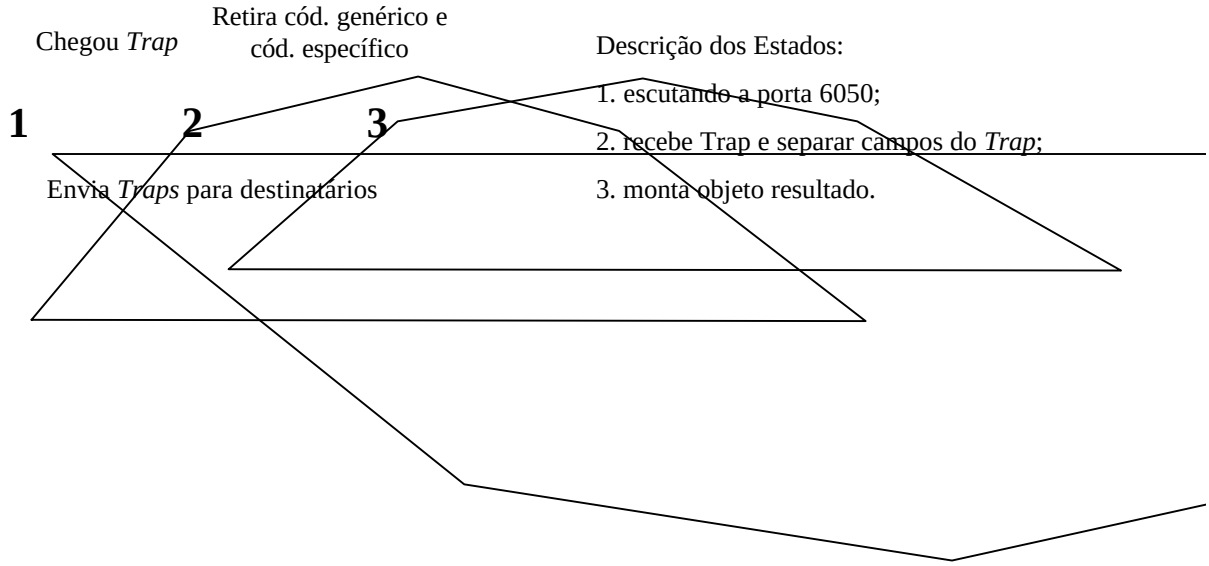


Figura 3.3 – Diagrama de estados Servidor Decodificador

Ao ser iniciado, é criada uma *thread* para conexão com a porta 6050 local para chegada do *Trap* enviado pelo discriminador. Ao Detectar a chegada, é criada uma outra *thread* para decodificar a mensagem SNMP, dividi-la em tokens, extrair o código genérico e código específico e montar um objeto denominado *resultado* o qual será enviado em seguida ao Discriminador, sendo finalizada em seguida. A *thread* com conexão à porta 6050 encerra-se apenas ao finalizar o módulo Servidor Decodificador, as demais são criadas ao detectar a chegada de novos *Traps* encerrando-se ao final de suas tarefas específicas.

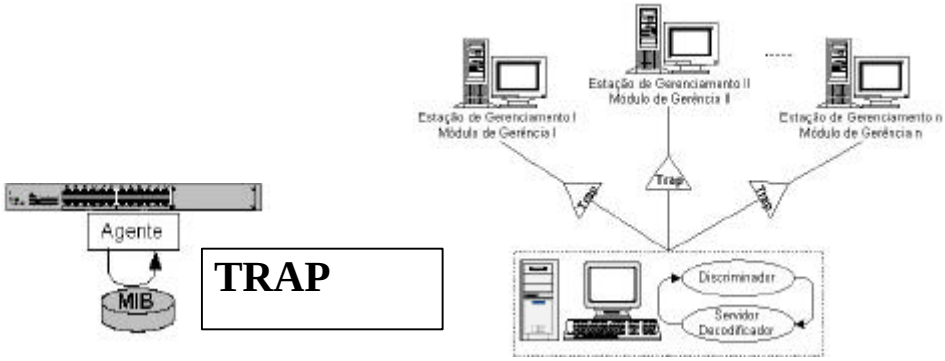


FIGURA 3.4: Arquitetura do Sistema

3.3 O PROCESSO DE DISCRIMINAÇÃO

A discriminação de um *Trap* se dá a partir da leitura do objeto *resultado*, o qual trata-se de uma *string* com apenas dois valores numéricos, código genérico e específico que identificam o conteúdo do *Trap*. Lido os valores, dá-se então início ao processo de busca da regra que definirá o destino da mensagem. Este processo é feito através de códigos de condição onde, ao atingir uma situação verdadeira, uma ação será disparada pelo Discriminador, envio ou não ao(s) gerente(s).

Código Genérico	;	**Código Específico**

FIGURA 3.5: formato do objeto resultado

Exclusivamente para esta pesquisa, a tabela a seguir mostra as definições para formulação das regras de discriminação.

Trap	Resultado	Área Funcional	Ação/Enviar para:
ColdStart	0/0	Configuração	Descartar
WarmStart	1/0	Configuração	Descartar
LinkDown	2/0	Desempenho/Falhas	Ger.Desempenho/Ger. Falhas
LinkUp	3/0	Desempenho/Falhas	Ger.Desempenho/Ger. Falhas
AuthenticaTion-Failure	4/0	Segurança	Ger. Segurança
EgpNeighBorLoss	5/0	Falhas/Desempenho	Ger.Desempenho/Ger. Falhas
EnterpriseSpecific	6/0		
Syslog	6/1	Segurança	Ger. Segurança

Tabela 3.1 Relação *Traps*/Destinatários

4 TESTES E VALIDAÇÃO DO PROTÓTIPO

Para validação do protótipo, os testes foram realizados em duas etapas, sendo que cada etapa possui duas fases de testes em situações distintas. As etapas estão detalhadas na tabela 4.1.

<table>
<tr><th colspan="2">ETAPA I</th></tr>
<tr><td colspan="2">Local: LRCSD: Laboratório de Redes de Computadores e Sistemas Distribuídos – FESURV
Dispositivo: Roteador Cisco 7500 acesso dial up
Equipamento: Intel Pentium 233 MMX, 64 Mb de RAM, WNT 4.0 Server.</td></tr>
<tr><th colspan="2">ETAPA II</th></tr>
<tr><td colspan="2">Local: Perdigão – Gerência de Informática GIN
Dispositivo: 81 roteadores Cisco 2500,1 Cisco 7500
swtches catalyst 1900, 1 catalyst 5000
o: Estação Sun, processador Risc, 128 MB de RAM, Solaris 7.</td></tr>
<tr><th>FASE I</th><th>FASE II</th></tr>
<tr><td>Gerentes, Discriminador e Serv. Decod. No mesmo micro.</td><td>Gerentes em micros diferentes; discriminador e Serv. Deocd. no mesmo micro.</td></tr>
</table>

Tabela 4.1 – Etapas de testes e suas fases

A eficácia do funcionamento do protótipo será alcançada caso a fórmula abaixo seja verdadeira.

$$\sum A = \sum B = \sum C$$

A = Traps Enviados pelo(s) Equipamento(s)
B = Traps Recebidos pelo(s) Discriminador(s)
C = Traps Discriminados

Para determinação da quantidade de *Traps* originados/recebidos, no início dos testes primeiramente é zerada a variável da MIB do equipamento gerenciado a qual informa o total de *Traps* enviados por este, depois são iniciados os testes. Ao final da fase de testes, são coletados os valores totais para os *Traps* enviados pelo equipamento gerenciado e os recebidos pelo discriminador. Estes valores são confrontados e depois emitidos os resultados. Para os testes de eficiência do protótipo em alta carga, utilizou-se o *SNMP Simulator*.

4.1 ETAPA I

Com duração de aproximadamente 3 meses, os horários de testes foram variados não seguindo um padrão devido a constante utilização dos equipamentos por estudantes vinculados ao

laboratório. Totalizando os dias e horários utilizados para testes possibilitou-se calcular as médias de utilização mostradas na tabela 4.2.

	FASE I	**FASE II**
Média Tempo decorrido	12	12
Média *Traps* recebidos	300	300
Média *Traps* Discriminador	300	300
Porcentagem de Acerto	100	100
Qtd. máxima de *Traps* processados simultaneamente	32	32

Tabela 4.2 – Resultados Apurados na Etapa I

4.2 ETAPA II

Nesta etapa houve a possibilidade de apuração exata do tempo decorrido de testes. Os testes tiveram início às 7hs da manhã, antes dos funcionários chegarem, e finalizaram às 19 hs, quando todos os funcionários já haviam saído.

	FASE I	**FASE II**
Média Tempo decorrido	12	12
Média *Traps* recebidos	1024	983
Média *Traps* Discriminador	1024	983
Porcentagem de Acerto	100	100
Qtd. máxima de *Traps* processados simultaneamente	38	38

Tabela 4.3 – Resultados Apurados na Etapa II

5 CONCLUSÕES

A comparação do desempenho entre o ambiente SNMP centralizado e distribuído pode ser realizada através da confrontação de algumas variáveis coletadas tanto no gerente quanto no discriminador: Instante de chegada do *Trap*; Instante de ação disparada; e Latência. Constatou-se que a utilização de gerentes distribuídos é mais eficiente.

A latência do gerente significa a quantidade de tempo que o gerente leva desde a chegada do *Trap* até sua tomada de decisão. A latência do Discriminador diz respeito ao tempo total de realização do ciclo de tarefas no protótipo, desde a chegada do *Trap* até sua discriminação.

Variáveis Coletadas	Qtd. *Traps* enviados simultanemante	Centralizado	Distribuídos
Latência Gerente	20	45 ms	7 ms
Latência Discriminador	20	9 ms	17 ms

Tabela 5.1 – Comparação entre as médias de latências

5.1 CONSIDERAÇÕES FINAIS

O modelo SNMP distribuído possibilita que várias ações sejam disparadas ao mesmo tempo por diferentes gerentes em diferentes equipamentos gerenciados, dessa forma, vários problemas podem ser solucionados simultaneamente. Já em ambientes centralizados, a concorrência da grande quantidade de processos em uma mesma máquina prejudica o desempenho do sistema aumentando muito o tempo de resolução de problemas e latência.

Enfatizar a recepção de alarmes é uma técnica bastante eficaz para solução de problemas isolados, contribuindo também para diminuição do tráfego de pacotes na rede e diminuição do *overhead* de processamento nas estações de gerenciamento. Hoje em dia existem centenas de tipos de *Traps* proprietários bastante eficazes, capazes inclusive de relatar informações referentes tanto a hardware como software, possibilitando um gerenciamento completo do equipamento.

Nenhuma incompatibilidade foi identificada nos ambientes e situações de testes. Conseguiram-se excelentes resultados nas discriminações e encaminhamentos dos *Traps* chegando a uma taxa de 100% de acerto.

Para melhoria do protótipo as seguintes soluções são propostas:

- Atribuição de prioridades aos *Traps* (baixa, média, alta, por exemplo). Alguns relatórios de eventos, com maior nível de importância, seriam encaminhados de forma prioritária em relação aqueles com níveis inferiores, com isso, problemas mais críticos seriam tratados mais rapidamente;
- Implementação de um esquema de *Log/Accounting* a fim de totalizar a quantidade de *Traps* recebidos/encaminhados, gerando estatísticas diárias e emissão de relatórios;
- Implementação de interfaces gráficas, possibilitando a configuração do sistema, inclusão de novas regras de discriminação, geração de relatórios, visualização de *logs*, etc;

- Utilização de técnicas de Inteligência Artificial para desenvolvimento de uma base de conhecimento e um motor de inferência para uma busca eficaz, dando a possibilidade de um grande número de regras, não acarretando em perda de desempenho;
- Substituição da técnica de comunicação via socket para RMI, a fim de obter um meio mais confiável e com desempenho superior.

6 BIBLIOGRAFIA

BRISA. **Gerenciamento de Redes – Uma abordagem de Sistemas Abertos**, São Paulo: MAKRON Books do Brasil Editora Ltda., 1993.

CASE,j.; FEDOR, M.; SCHOFFSTALL, M.; DAVIN, J**.; Simple Network Management Protocol**, RFC 1157, maio de 1990.

CASE,j.; MCCLOGHRIE, K.; ROSE, M.; WALDBUSSER, **S. Protocol Operations for Version 2 of the Simple Network Management Protocol** (SNMPv2), RFC 1905, janeiro de 1996.

CISCO URL http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/fun_r/frprt3/frd3001.htm, agosto de 2000.

CISCO. URL: http://www.cisco.com/univercd/cc/td/doc/product/lan/cat1700/c1700/c17inbnd.htm, setembro de 2000.

HARNEDY, S. **Total SNMP: Exploring the Simple Network Management Protocol**. Second Edition. New Jersey: Prentice Hall PRT, 1997.

CCITT; Event Report Management Function, X0734, 1993.

SUN Sun Microsystems, Inc. Solstice Administration Guide: Site/SunNet/Domain Manager. 1996

STALLINGS, W. SNMP, **SNMPv2 and RMON: Pratical Network Management**. Second Edition. United States of America: Addison Wesley, Inc., 1996.

IBM. URL: http://www.tivoli.com, dezembro de 2000.

CS Software. URL: http://www.cscare.com/TrapConsole/, setembro de 1999

MAURO, R. DOUGLAS; SCHMIDT, KEVIN J. **SNMP Essencial**. . Editora Campus. Rio de Janeiro, 2001.

BATES, REGIS J. **Network Management SNMP**. Digital. Janeiro 2002.