DISCRIMINADOR DE REPASSE DE EVENTOS EM AMBIENTES SNMP

V. Augusto, S. Elizabeth
UFSC – Universidade Federal de Santa Catarina, Campus Universitário Trindade
Departamento de Informática e Estatística, Centro Tecnológico, 88040-900
Florianópolis, SC

RESUMO

A utilização de mais de uma estação de gerenciamento, diferenciadas de acordo com as cinco áreas funcionais de gerência definidas pela ISO (falhas, configuração, contabilização, desempenho e segurança), destaca-se como uma ótima solução para gerência de redes complexas de telecomunicações (OSI). Neste contexto, este artigo pesquisa a possibilidade do uso desta técnica de gerenciamento em ambientes SNMP, atualmente inviabilizada devido a ausência de uma ferramenta nativa que possibilite filtrar relatórios de eventos para determinação de destinatários. Um protótipo foi desenvolvido para exercer este papel de filtro de relatórios de eventos (*Traps/Informs*), de acordo com as especificações constantes na norma X-734 [6] para objetos discriminadores de repasse de eventos em ambientes de gerencia OSI.

1. INTRODUÇÃO

A adoção de novas tecnologias, o aumento da quantidade de usuários e sua distribuição geográfica, implicam no aumento da complexidade de uma rede de computadores, tornando necessário seu gerenciamento a fim de garantir a todos usuários sua disponibilidade com níveis aceitáveis de desempenho e segurança [1].

Em redes de telecomunicações é comum a utilização de mais de uma estação de gerenciamento, onde cada uma abriga um módulo de software gerente específico de uma área funcional de gerência, sendo incumbido de receber e tratar apenas relatórios de eventos de acordo com sua área de gerenciamento [1]. Os relatórios de eventos e seus destinatários são definidos em políticas de gerenciamento, sendo estas realizadas de acordo com as especificações dos relatórios de eventos, das áreas funcionais e das necessidades do ambiente. Como exemplo, uma aplicação de gerência poderia apenas receber relatórios referentes à falhas, ou então ser definido como destinatário também de relatórios referentes à desempenho ou segurança, dependendo do ambiente. Isto se torna possível devido a existência de uma ferramenta denominada Objeto Discriminador de Repasse de Eventos, o qual tem como objetivo filtrar os relatórios de eventos, descartando ou determinando destinatário(s) [6].

A inexistência de uma ferramenta nativa que assuma esta funcionalidade no ambiente SNMP motiva este trabalho. As pesquisas foram realizadas considerando a versão 1 do protocolo SNMP, devido a sua larga utilização.

O restante deste artigo está organizado da seguinte forma: o capítulo 2 descreve a arquitetura de gerenciamento SNMP e os tipos de *Traps* padronizados; o capítulo 3 apresenta uma justificativa para a implementação desta pesquisa, detalhando o problema e também algumas iniciativas existentes no contexto deste trabalho; o capítulo 4 mostra o sistema implementado, tecendo considerações sobre seu desenvolvimento, sua arquitetura, funcionamento, os módulos componentes e detalhando o processo de discriminação; o capítulo 5 descreve os testes, suas etapas e seus respectivos resultados para validação do protótipo; o capítulo 6 traz as conclusões, considerações finais e trabalhos futuros.

2. O PROTOCOLO SNMP

Para entendimento do contexto desta pesquisa, é necessário conhecer o ambiente a ser utilizado. Este capítulo é dedicado a conceituação do protocolo SNMP, sua arquitetura e a PDU *Trap*, sendo esta última o vetor principal utilizado nesta pesquisa.

2.1 A ARQUITETURA SNMP

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 a longo prazo [8].

A documentação do SNMP, mais especificamente o RFC 1157, diz que “a arquitetura de gerenciamento SNMP é uma coleção de estações de gerenciamento e elementos de rede”, onde estações de gerenciamento são equipamentos que abrigam os módulos gerentes, o qual executam aplicações de gerenciamento e são responsáveis pela monitoração e controle dos elementos gerenciáveis, e os elementos de redes são equipamentos que abrigam os agentes, destinados às tarefas de coletar informações sobre as atividades relacionadas com a rede, armazenar estatísticas localmente e responder às solicitações do gerente [2]. Seu modelo de gerenciamento está representado pela fig. 2.1

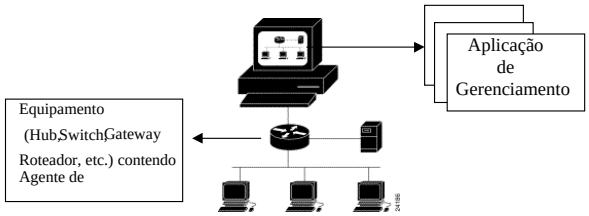


Figura 2.1 – Modelo de gerenciamento SNMP

A comunicação entre os agentes e o(s) gerente(s) é realizada através do SNMP, existindo duas maneiras de transmissão de informações [8]:

- polling: Por iniciativa do gerente, trata-se da solicitação de uma ação a ser realizada pelo agente, retornando uma resposta com o resultado.
  Ex. leitura ou escrita de variáveis de gerenciamento.

- Relatório de Evento: Por iniciativa do agente, consiste em relatar, ao gerente, a ocorrência de um determinado evento no ambiente. No SNMP v1 é realizado através de *Traps*, e no SNMP v2, além dos *Traps*, existem também os *Informs*.

2.2 TIPOS DE TRAPS

Sete *Traps* foram padronizados [2]:

- **coldStart (0)** – Significa que um dispositivo gerenciável reinicializou-se devido à alterações na configuração do agente ou a implementação da entidade de protocolo;
- **warmStart (1)** – Significa que um dispositivo gerenciável reinicializou-se, mas não especificamente por mudanças da configuração do agente ou da implementação da entidade de protocolo;
- **link-Down (2)** – Significa que um dispositivo gerenciável identifica uma falha em um dos links de comunicação contidos na configuração do agente;
- **link-Up (3)** – Significa que um dispositivo gerenciável reconhece que um dos links de comunicação, contidos na configuração do agente, está ativo novamente;
- **authenticationFailure (4)** – Detecção de uma mensagem SNMP com comunidade inválida;
- **egpNeighborLoss (5)** - Significa a não comunicação entre *gateways* através do protocolo EGP devido à queda do link de comunicação de um deles, ou de ambos.
- **enterpriseSpecific (6)** – Significa que o dispositivo gerenciável reconhece algum evento proprietário do fabricante. O campo *specific-trap* identifica o tipo do Trap proprietário ocorrido. Cada fabricante implementa seus próprios Traps em seus equipamento de acordo com suas necessidades específicas. Um exemplo seria o Trap do tipo *SysLog* (1), implementado pela Cisco, relatando o acesso de um equipamento ao seu sistema.

3. JUSTIFICATIVA

Atualmente um agente SNMP é incapaz de determinar, por si mesmo, destinatários específicos para cada tipo de Trap, devido à ausência de um filtro nativo. Um protótipo desenvolvido em java implementa algumas funcionalidades utilizadas por objetos discriminadores de repasse de eventos em ambientes OSI. Este protótipo tem a finalidade de descartar ou determinar destinatários para os Traps através de regras formuladas em políticas de gerenciamento.

3.1 TRABALHOS CORRELATOS

Atualmente existem algumas aplicações que implementam as funcionalidades dos discriminadores OSI. Tanto a *Sun* quanto a *IBM* oferecem uma poderosa ferramenta, o *SunNetManager* com o *Trap Deamon* e o *Tivoli NetView* respectivamente [9]. Contudo, ambos são executados diretamente na estação de gerenciamento não impedindo o recebimento de *Traps*, implicando no consumo largura de banda e *overhead* de processamento para descartar os *Traps* desnecessários aquela estação de gerenciamento específica [7]. Já o *applet TrapConsole* [10] 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) destinatários(s) já que, ao chegar um *Trap*, a aplicação fica aguardando a decisão do administrador [10].

O protótipo implementado nesta pesquisa diferencia-se das aplicações supracitadas por ficar em uma máquina distinta das estações de gerenciamento, sendo capaz de descartar *Traps* indesejados ou determinar os destinatários mais adequados para um determinado tipo de *Trap*. Com isso, assegura-se que apenas determinados *Traps* serão enviados a estações pré-definidas, implicando em uma melhor utilização da largura de banda e diminuição de *overhead* nas estações, evitando que estas recebam *Traps* pertencentes a outros gerentes descartando-os em seguida.

4. O SISTEMA IMPLEMENTADO

Para o estudo da viabilidade de utilização de discriminadores de repasse de eventos em ambientes SNMP com mais de uma estação de gerenciamento, foi necessário construir um protótipo que implemente as funcionalidades definidas na norma x-734 [6].

Nesta fase do projeto, foi implementado um esquema de determinação de destinatários através de regras definidas exclusivamente para esta pesquisa. A definição das regras foi feita através da comparação das características da área funcional com a informação trazida pelo *Trap*, ou seja, de acordo com o conteúdo do *Trap* identifica-se em qual(s) área(s) de gerência ele estaria enquadrado, enviando-o em seguida para o gerente determinado.

4.1 CONSIDERAÇÕES SOBRE O DESENVOLVIMENTO

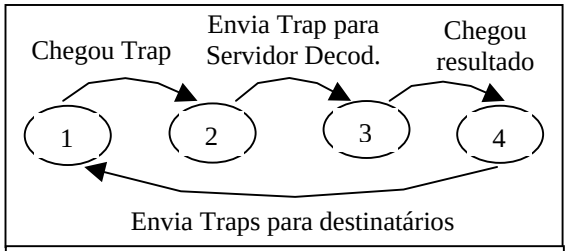
O protótipo foi desenvolvido utilizando a linguagem Java com as seguintes justificativas:

- Devido à independência de plataforma, pode ser utilizado em diversos ambientes, desde o Unix ao Windows, requerendo apenas um Web Browser ou a máquina virtual Java;
- A existência de diversas Api´s abertas SNMP, facilitaram bastante o desenvolvimento deste protótipo;
- A interface amigável e grande facilidade de implementação.

4.2 ARQUITETURA DO SISTEMA

O sistema possui dois módulos de software: o Discriminador e o Servidor Decodificador, cujos diagramas de estado são apresentados, respectivamente nas figuras 4.1 e 4.2:

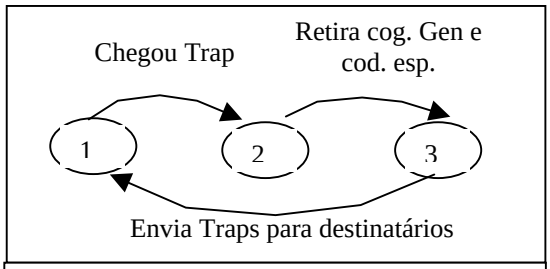
- Discriminador – Responsável pela recepção dos Traps, e aplicação das regras de discriminação com posterior determinação de destinatários.



estados:
1. escutando a porta 162
2. coloca Trap no buffer e cria thread de envio
3. escutando a porta 6051
4. compara resultado com as regras, retira Trap do buffer e envia ao(s) destinatário(s).

Figura 4.1 – diagrama de estados Discriminador

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



estados:
1. escutando a porta 6050
2. recebe Trap e separar campos do *Trap*.
3. monta objeto resultado.

Figura 4.2 – Diagrama de estados Servidor Decodificador

4.3 FUNCIONAMENTO DO PROTÓTIPO

A seguir estão descritas, de maneira detalhada, o funcionamento lógico dos módulos componentes do protótipo.

4.3.1 MÓDULO DISCRIMINADOR

Ao ser executado, o Discriminador cria duas *threads*, uma que estabelece conexão com a porta 162 local, aguardando a chegada de *Traps* dos agentes, e outra com a porta 6051, aguardando a chegada do objeto *resultado*, ambas através de *socket*. Ao detectar a chegada de um *Trap*, primeiramente este é colocado temporariamente em um *buffer* e em seguida é criado uma outra *thread* com a função de enviar uma cópia deste *Trap* para o servidor decodificador através de um *socket* para a porta 6050 deste último, onde após o envio, a *thread* é finalizada. Ao chegar o objeto *resultado* na porta 6051, uma *thread* é então criada para determinação do(s) destinatário(s) do *Trap* de acordo com regras de discriminação. Determinado o(s) destinatário(s), esta mesma *thread* envia o *Trap* para ele(s) e finalizando-se ao final. As *threads* com conexão na porta 162 e 6051 são apenas finalizadas ao encerrar o Discriminador, as demais são criadas ao detectarem a chegada do *Trap* e do objeto *resultado*, encerrando-se ao final de suas tarefas específicas.

4.3.2 MÓDULO SERVIDOR DECODIFICADOR

Ao ser iniciado, o Servidor Decodificador cria uma *thread* a qual estabelece uma conexão com a porta 6050 local aguardando a chegada de um *Trap*. Detectada a chegada deste, é então criada uma outra *thread* que tem o objetivo de decodificar a mensagem SNMP, dividi-la em pedaços, extrair o código genérico e código específico e montar um objeto denominado *resultado* o qual será enviado em seguida ao Discriminador através de um socket para a porta 6051 do Discriminador. Depois a *thread* é finalizada. 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.

4.3.3 DISCRIMINAÇÃO

A discriminação de um Trap é realizada através da implementação de regras de discriminação, que trabalham associando os valores dos códigos genérico e específico do Trap com as regras definidas pelos administradores do ambiente.

Como exemplo podemos destacar:
se o código genérico = 1 (*Link-UP*) e código específico = 0 (porque não se trata de um Trap do tipo *Enterprise Specific*) então enviar Trap para gerente de Falhas e Desempenho.
Exclusivamente para esta pesquisa, a tabela a seguir mostra as definições para formulação das regras de discriminação.

Tabela 4.1 Relação *Traps*/Destinatários

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

5. 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 5.1

Tabela 5.1 – Etapas de testes e suas fases

ETAPA I	
Local: LISA: Laboratório de Tratamento de Incerteza e Sistemas Adaptativos - UFSC Dispositivo: Roteador Cisco 7500 acesso dial up Equipamento: Intel Pentium 233 MMX, 64 Mb de RAM, WNT 4.0 Server.	
ETAPA II	
Local: Casan – Gerência de Informática GIN Dispositivo: 81 roteadores Cisco 2500,1 Cisco 7500 21 swtches catalyst 1900, 1 catalyst 5000 Equipamento: Estação Sun, processador Risc, 128 Mb de RAM, Solaris 7.	
FASE I	**FASE II**
Gerentes, Discriminador e Serv. Decod. no mesmo micro.	Gerentes em micros diferentes; discriminador e Serv. Decod. no mesmo micro.

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

A determinação da quantidade de *Traps* originados/ recebidos é feita da seguinte forma:

- Antes de iniciar os testes, primeiramente é zerada a variável da MIB do equipamento gerenciado a qual informa o total de Traps enviados por este;
- Após este procedimento, iniciam-se os testes com os valores nulos (*Traps* enviados/recebidos);
- Ao final da fase de testes constata-se o valor total dos Traps enviados pelo equipamento gerenciado em seu IOS e os Traps recebidos pelo discriminador através de um contador implementado neste;
- Confrontam-se os valores para emissão dos resultado da fase de testes.

5.1 ETAPA I

Esta etapa teve duração de aproximadamente 3 meses, onde os horários de testes foram variados não seguindo um padrão devido a constante utilização dos equipamentos por estudantes vinculados ao laboratório. Sendo assim, totalizando os dias e horários utilizados para testes possibilitou-se calcular as médias de utilização mostradas na tabela 5.2.

Tabela 5.2 – Resultados Apurados na Etapa I

	FASE I	**FASE II**
Média Tempo decorrido	12	12
Média Traps recebidos	300	300
Média Traps Discriminador	300	300
Porcentagem de Acerto	100	100

5.2 ETAPA II

Nesta etapa de testes houve a possibilidade de apuração exata do tempo decorrido de testes, 12 horas em 2 dias. Os testes iniciaram às 7 horas da manhã, antes dos funcionários chegarem, e finalizaram às 19 horas, quando todos os funcionários já haviam saído. Os resultados obtidos estão demonstrados na tabela 5.3.

Tabela 5.3 – Resultados Apurados na Etapa II

	FASE I	**FASE II**
Média Tempo decorrido	12	12
Média Traps recebidos	1024	983
Média Traps Discriminador	1024	983
Porcentagem de Acerto	100	100

6. CONCLUSÕES

As fases de testes possibilitaram avaliar e validar o protótipo, constatando-se sua completa aplicabilidade e utilidade para ambientes SNMP.

Nem todos os tipos de *Traps* foram originados pelos equipamentos gerenciados devido a não constatação de situações que exigissem o envio destes. Os *Traps* genéricos detectados foram: *coldStart*, *warmStart*, *linkUp*, *linkDown*, *authenticationFailure* e *enterpriseSpecific*. Dos *Traps* específicos foram detectados apenas: *sysLog* e *ISDN*.

A não detecção de todos os tipos de *Traps* genéricos não implica em limitação de discriminação dos *Traps* recebidos já que foram implementadas regras de discriminação para todos os tipos de *Traps* genéricos. Devido a grande quantidade de *Traps* específicos desconhecidos este protótipo limitou-se apenas aqueles conhecidos pelo autor, entretanto o protótipo está aberto para a inclusão de tantas regras quantas forem necessárias, aumentando a gama de *Traps* específicos suportados.

Como se trata de um protótipo, não foi levado em consideração o desempenho na busca das regras para determinação de destinatários implicando em uma limitação considerável devido ao método sequencial de busca.

6.1. CONSIDERAÇÕES FINAIS

Este protótipo demonstrou ser uma ferramenta bastante útil no suporte à administração de redes IP complexas, com mais de uma estação de gerenciamento e diversos dispositivos de conectividade. A filtragem de Traps e o encaminhamento à estações de gerenciamento específicas é uma maneira 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.

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.

6.2 TRABALHOS FUTUROS

No decorrer das fases desta pesquisa, foi possível identificar algumas implementações que implicariam em melhorias ao protótipo.

- 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.

8. BIBLIOGRAFIA

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

[2] Case,j.; Fedor, M.; Schoffstall, M.; Davin, J.; Simple Network Management Protocol, RFC 1157, maio de 1990.

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

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

[5] Harnedy, S. Total SNMP: Exploring the Simple Network Management Protocol. Second Edition. New Jersey: Prentice Hall PRT, 1997.

[6] CCITT; Event Report Management Function, X0734, 1993.

[7] Sun Microsystems, Inc. Solstice Administration Guide: Site/SunNet/Domain Manager. 1996

[8] Stallings, W. SNMP, SNMPv2 and RMON: Pratical Network Management. Second Edition. United States of America: Addison Wesley, Inc., 1996.

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

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