MONITORAMENTO DE TAXA DE ERROS EM TRANSMISSÃO DE REDES USANDO SOFTWARE LIVRE

Marcos Valnes Abadi[1], Carlos Henrique da Costa Cano[1],Rubem Dutra Ribeiro Fagundes[2]

[1]Instituto Porto Alegre da Igreja Metodista
Centro Universitario Metodista
Joaquim Pedro Salgado, 80 CEP: 90420-060 Porto Alegre - RS - Brasil[1]

Pontifícia Universidade Católica do Rio Grande Do Sul - PUCRS
Avenida Ipiranga, 6681 - Predio 30
Bloco B - Sala 101 CEP: 90619-900 Porto Alegre - RS - Brasil[2]

{marcos.abadi , carlos.cano}@metodistadosul.edu.br, rubemdrf@pucrs.br

Abstract. *The main objective on this work is to provide a better way to understand and monitor CRC error in current networks. Our study implemmented a open source plug-in for Nagios, leveraging its capabilities and providing a better way for the open source community monitor and gather information about CRC error on its networks.*

Resumo. *O objetivo deste trabalho é o de apresentar uma forma melhor de monitoramento de erros de dentro das redes de comunicação. Neste estudo foi implementado um módulo adicional ao Nagios que permite o controle e monitoramento de erros de CRC em redes e equipamentos de redes, aumentando assim consideravelmente as capacidades do Nagios e provendo uma melhoria para a comunidade de Software Livre.*

1. Introdução

O monitoramento de ambientes heterogêneos não é uma tarefa muito fácil [BRISA(1993)]. Sabe-se também que existe uma grande quantidade de softwares de monitoramento de dispositivos de rede, onde o foco está mais voltado em saber se o dispositivo está operacional, qual a taxa de transmissão no momento, o quanto já foi transferido, e muitos outros, no entanto muitos deles não se preocupam em disponibilizar recursos para checar com que qualidade os dados passam por uma determinada interface, ou seja, se há ou não erros na transmissão e quais suas proporções.

A detecção de erros ocorre através do uso de CRC ou Cyclic Redundancy Check, processo este que faz um cálculo polinomial[IEEE(2005)], gerando um resultado. Caso o resultado da transmissão seja diferente do mesmo processo na recepção isto sinaliza que um erro de transmissão ou recepção ocorreu[TANENBAUM(1994)]. Quando ocorre este fato os contadores de "erros de CRC" são incrementados.

Em muitos equipamentos só é possível ver com detalhes suas taxas de erros acessando suas interfaces de gerenciamento. Mas isso se torna inviável quando se tem uma enorme quantidade de equipamentos.

Neste estudo será utilizada a API do protocolo amplamente utilizado para o gerenciamento de redes chamado SNMPv2 *(Simple Network Management Protocol) [Miller(1999)],* este protocolo trabalha em nível da camada de aplicação do modelo OSI e encontra-se disponível em diversos dispositivos que formam uma rede [KORZENIOWSKI(1994)].

A inciativa é a criação de um plugin simples, onde informaremos apenas três parâmetros: Nome do host, comunidade snmp e número de identificação da interface. O plugin será integrado a ferramenta de gerenciamento NAGIOS[NAGIOS()]. Uma prévia análise foi feita na API do NAGIOS para que os mesmos trabalhem de forma correta e possam nos fornecer informações as quais serão usadas para identificar os problemas e solucioná-los da melhor forma possível.

Objetivando desta forma a criação de um plug-in adicional para a ferramenta de monitoria NAGIOS, contribuindo para uma melhoria dentro da comunidade de Software Livre.

2. Metodologia

Para obter os dados necessários para as análises foram usados alguns objetos do grupo de gerência de performance dentro da hierarquia SNMP como: ifInerrors, ifInUcastPkts, ifInNUcastPkts, MIB-II [PRAS(1997)] segundo a RFC2863[IETF(a)].

Abaixo segue uma tabela com uma breve descrição de cada um dos objetos utilizados:

Tabela 1. Objetos SNMP utilizados

Objeto	Descrição
IfInErrors	Número de pacotes recebidos que continham erros (Impedimento para repassa-los para protocolos das camadas superiores).
IfInUcastPkts	Número de pacotes unicast repassados para protocolos de camadas superiores.
IfInNUcastPkts	Número de pacotes não-unicast (broadcast ou multicast) repassados para protocolos de camadas superiores.

A estrutura de objetos SNMP segue uma ordem hierárquica, indo dos objetos mais genéricos até os mais específicos conforme a figura 1.

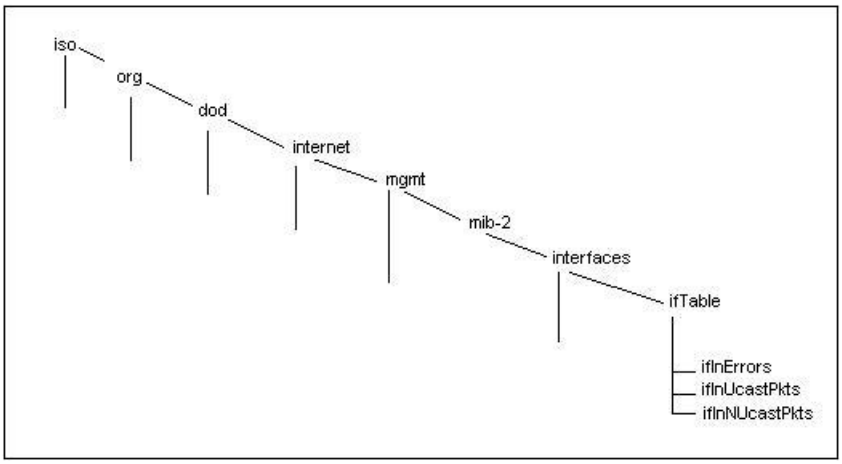

Figura 1. Ramificação dos objetos MIB utilizados

2.1. Cálculo da perda de pacotes

A fórmula usada para calcular a taxa de erro e usualmente expressa em percentagem:

$$Erros = \frac{IfInErrors * 100}{(ifInUcastPkts + ifInNUcastPkts)} \quad (1)$$

Observe que os erros de saída não são considerados como taxa de erro nas fórmulas de precisão. Pois teoricamente um equipamento operando em condicoes normais nunca poderia colocar pacotes com erro na rede, e as taxas de erro da interface de saída nunca iriam aumentar. Sendo assim o tráfego de entrada e os erros são as únicas medidas contabilizadas para os erros de interface.

Esta fórmula reflete o erro e a precisão em termos da norma MIB II interface [IETF(b)] para contadores genéricos. O resultado é expressado em termos de uma percentagem que compara erros ao total de pacotes enviados e recebidos. O cálculo de uma transmissão eficiente está relacionada com o número de pacotes inválidos.

2.2. Infra-Estrutura de rede

A infra-estrutura de rede que foi utilizada para certificar o funcionamento do plugin está descrita abaixo onde efetuamos vários testes em diferentes arquiteturas de equipamentos:

- Servidores com Sistema operacional Linux.
- Servidores com Sistema operacional Windows Server.
- D-Link Access Point.
- Rádio Alvarion - BreezeACCESS VL.
- Switch 3Com 5500.
- Switch D-Link DES-3052 Fast Ethernet.
- Switch Dell PowerConnect 3448.

Disponibilizamos uma máquina com sistema operacional Linux para a instalação da ferramenta de monitoramento também conhecida como de gerência de rede, denominada NAGIOS de código livre o que possibilita a modificação do código fonte.

3. Implementação

Para melhor identificação e integração com ferramentas de Software Livre foi desenhado e implementando um plug-in para o NAGIOS, plug-in este desenvolvido em C e em conformidade com as APIs do Nagios.

Na criação deste plug-in foram necessários alguns requisitos conforme o guia de desenvolvimento do Nagios. Também na criação foi seguido o modelo GNU, onde qualquer sistema operacional que suporte GNU possa compilar o plug-in.

Como compilador da linguagem C foi utilizado o compilador GCC 4.1.2 o qual já veio padrão na distribuição. Seguindo o modelo de desenvolvimento para o Nagios o plug-in criado tinha como resultado somente uma linha de texto com as informações de eficiência das portas monitoradas conforme a tabela 2.

Tabela 2. Códigos de Retorno do Plug-In

Valor	Estado	Descrição
0	OK	O plug-in verifica que o estado está normal
1	Warning	O plug-in verifica o serviço, mas o estado está em alerta
2	Critical	O estado do serviço é crítico
3	Unknow	O plug-in não conseguiu verificar o serviço

Colocando-se o plug-in ativo com gatilho de 20 erros de CRC por hora, foi possível identificar principalmente dois cenários de falhas de comunicação e/ou CRC.

3.1. Primeiro Cenário: Configuração das Portas

As redes Ethernet operavam em modo half-duplex[TANENBAUM(1994)], onde era possível realizar uma transmissão por vez, tanto para envio quanto para recebimento de dados, mas nunca simultaneamente. Nestas redes um protocolo é muito utilizado para detecção de colisões chamado de CSMA/CD (Carrier Sense Multiple Access with Collision Detection). Tendo em vista esse conceito foi feita a seguinte análise:

Por padrão de fábrica todos os switches vem com o modo de auto negociação ativados nas suas respectivas portas. Isso pode ser um problema a ser analisado se detectado uma grande quantidade de pacotes com erros em uma determinada porta. Ao trabalhar-se com o modo de auto negociação ativo corre-se o risco do mesmo não funcionar adequadamente com o dispositivo da ponta, ou seja, quando uma porta em modo half-duplex tem-se o recurso do controle de colisões do protocolo CSMA/CD, já se ocasionalmente em ambas as portas estiverem funcionando em modos diferentes por algum problema como por exemplo: uma em modo half-duplex e outra em full-duplex , onde neste último modo não temos o recurso de detecção de colisões podemos ter uma potencial perda de pacotes.

Ao ser analisado que uma determinada porta em um switch apresentam esses erros, é identificado que a mesma está operacional e em modo de auto negociação, então neste caso pode-se sugerir algumas mudanças como:

Verificar se a Auto-negociação está ativada nas portas, caso sim, altere para modo full-duplex e desabilite o controle do fluxo de pacotes.

É preciso lembrar que não basta ter comunicação ponto-a-ponto para que o modo full-duplex seja habilitado. Ambos os lados devem ser configurados para esse modo de operação, seja através de um procedimento manual ou através de auto negociação. Não se pode misturar os modos half-duplex e full-duplex, um em cada lado do enlace. Isso resultaria em erros de vários tipos, incluindo mais colisões, colisões tardias e erros de CRC.

Para usuários de sistemas operacionais Linux é indicado o uso das seguintes ferramentas para verificação das configurações das interfaces:

- mii-tool (Permite visualizar, manipular o estado de uma interface independente da media).
- ethtool (Permite visualizar ou realizar mudanças nas configurações de placas Ethernet) .

3.2. Segundo Cenário: Problemas Físicos

No segundo caso de análise é de suma importância a verificação de cabos, conectores, patch cords, que interligam os equipamentos de rede, ter a certeza de que a certificação do cabeamento foi realizada adequadamente. Verificar se os cabos de lógica não estão próximos aos fios de alta tensão o que também pode causar problemas nas transmissões de dados.

4. Resultados obtidos

A partir deste estudo foram feitos testes em campo e verificou-se que:

- Portas que apresentavam sinais elevados de erro de CRC estavam indevidamente configuradas
- Portas que estavam sofrendo alguma forma de interferência também apresentavam altos índices de erros de CRC

Neste artigo apresentou-se uma proposta da melhoria da qualidade do gerenciamento de dispositivos de rede através do uso de ferramentas livres, onde foi abordada a criação de um plug-in para identificação de erros de CRC nas interfaces de dispositivos de rede. De um modo centralizado foram obtidos os seguintes ganhos:

- Centralização das informações de erros em interfaces em uma única ferramenta.
- Informação do percentual de erros.
- Alertas quando o percentual está muito alto.
- Guiar na identificação e solução de possíveis problemas.
- Monitoramento da qualidade de Link de missão crítica.
- Garantir a informação de um percentual da eficiência do Link.

Através dos alertas foi possível detectar falhas transientes que antes eram ignoradas, pois sumiam em poucos minutos. Também é importante salientar que uma maior consciência da rede foi percebida conforme os erros alertados pelos sistemas foram sendo solucionados, gerando uma melhoria significativa de desempenho.

5. Conclusão

Como conclusão deste artigo aprendemos a usar os valores disponibilizados pelos objetos da MIB, para calcular com precisão a taxa de erros em qualquer interface de rede que suporte o padrão especificado pela [IETF(a)]. O plugin em conjunto com a ferramenta de monitoramento de redes nos possibilita identificar problemas e chegar mais facilmente ao ponto de falha com as análises sugeridas na metodologia.

Ainda mais importante salientar é que o uso de software livre nos possibilitou modificar diretamente o software escolhido NAGIOS e também estudar outros códigos e processos para usarmos em nossa implementação, algo que seria impossível no mundo de software proprietário e fechado.

Preocupar-se com a qualidade da comunicação entre dispositivos de rede é garantir que os mesmos estejam trabalhando adequadamente, ou seja, a retransmissão dos pacotes usa a banda que poderia ser usada para enviar novos dados.

Pretende-se futuramente aperfeiçoar os trabalhos em cima do plugin adicionando novos recursos e disponibilizando-os sob General Public License versão 2, para a comunidade open-source. Gerando assim um retorno para a comunidade de Software Livre, para que esta mesma utilize e até melhore o sistema apresentado.

Referências

S. B. p. I. d. S. A. BRISA, *GERENCIAMENTO DE REDES: Uma abordagem de sistemas abertos*, MAKRON, Ed. MAKRON Books, 1993.

IEEE, "Ieee std 802.3-2005," *IEEE Standards*, 2005.

IETF, "The interfaces group mib using smiv2," no. RFC2233.

——, "The interfaces group mib," *IETF*, no. RFC2863.

P. KORZENIOWSKI, "Monitorando sua rede," *BYTE*, pp. 100–102, 1994.

M. A. Miller, *Managing internetworks with SNMP*, M. Books, Ed., 1999.

NAGIOS. Nagios. [Online]. Available: http://www.nagios.org

H. H. E. PRAS, A. HAZENWINKEL, "Management of the world-wide web," *Anais 15ª SBRC*, pp. 340–345, 1997.

A. TANENBAUM, *Redes de Computadores*. Editora Campus, 1994.