Gerenciamento de Roteamento BGP em Pontos de Troca de Tráfego

Ana Benso da Silva[1], Andrey Vedana Andreoli[1], Fábio Falci Rodrigues[1], Leandro Márcio Bertholdo[2], Liane Tarouco[2]

[1]Faculdade de Informática – Universidade Católica do Rio Grande do Sul (PUCRS)
Avenida Ipiranga, 6681 – 91.501-970 – Porto Alegre – RS – Brazil

[2]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{benso,ji202379}@inf.pucrs.br,{andrey,berthold,liane}@penta.ufrgs.br

Abstract. *This work describes a system for management of protocol BGP (Border Gateway Protocol), applied to interconnecting of large networks and backbones and NAPs (Network Access Points). Thus being this work uses the architecture of NAPs to analyze the BGP traffic identifying the requirements of management of this protocol and elaborating a management proposal. The results of this work are an specification and implementation of an experimental MIB to BGP in addition to MIB BGP and implementation of SNMP agents for executing the necessary management tasks.*

Resumo. *Este trabalho apresenta uma proposta de gerenciamento do protocolo de roteamento BGP (*Border Gateway Protocol*), aplicado na interconexão de grandes* backbones *e PTTs (Pontos de Troca de Tráfego). Assim sendo, essa proposta utiliza a arquitetura de um PTT para analisar o tráfego BGP, identificar os requisitos de gerenciamento do protocolo e elaborar um conjunto de informações úteis ao seu gerenciamento. Como resultados do trabalho obteve-se a especificação de uma MIB experimental para o BGP, baseada em MIBs já existentes e a implementação de agentes SNMP para execução das tarefas de gerenciamento necessárias.*

1. Introdução

Os elementos de interconexão de redes são essenciais para o funcionamento da Internet, em especial os roteadores. Eles estão ligados entre si de forma que uma mensagem possa chegar ao seu destino rapidamente e com eficácia. Logo, o processo de configuração do roteamento deve ser robusto e eficaz. Para ter controle sobre toda a rede, é necessário o uso efetivo do gerenciamento. Com a complexidade e a probabilidade de falhas crescendo cada vez mais, a atividade de gerência torna-se essencial para o funcionamento da rede.

Assim sendo, o objetivo principal deste trabalho foi desenvolver um sistema de gerenciamento do protocolo de roteamento BGP, para utilização em locais de redes de computadores de domínios administrativos diferentes. Neste contexto, o sistema tem o visa identificar possíveis falhas e situações anômalas ou suspeitas a respeito das operações do protocolo BGP, mantendo a equipe de suporte informada sobre esses

eventos. O trabalho envolveu o estudo de arquiteturas e protocolos de gerenciamento de redes de computadores, do protocolo de roteamento BGP e de arquiteturas de PTTs (Pontos de Troca de Tráfego), que representam umas das formas de interconexão das redes de diferentes domínios administrativos.

2. O protocolo BGP

O BGP é um protocolo de roteamento dinâmico utilizado para comunicação entre sistemas autônomos (ASs). Baseados nestas informações, os sistemas autônomos conseguem trocar informações e determinar o melhor caminho para as redes que formam a Internet. Tal papel é muito importante sabendo que a todo o momento essas redes podem sofrer alterações, terem perda de conectividade, receberem anúncios inválidos, possuirem alterações em suas políticas de roteamento [SAM00], entre outros. Mesmo assim o protocolo deve adaptar-se rapidamente para manter seus anúncios de forma eficiente. Para se ter uma idéia, a tabela de roteamento BGP completa da Internet no início do ano de 2002 possuía aproximadamente 107.000 rotas de acordo com [CID02]. Já em Janeiro de 2003 é de 139.000 rotas [CAI03][CID03].

De acordo com [RFC1771] a negociação de uma sessão BGP passa por diversos estados até o momento que é propriamente estabelecida e é iniciada a troca de anúncios de prefixos de cada vizinho BGP.

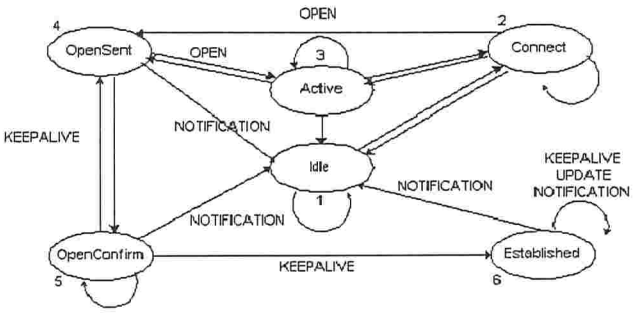

Figura 1 - Máquina de estados do protocolo BGP

A partir da máquina de estados apresentada anteriormente, é possível saber qual o *status* de uma sessão BGP entre dois roteadores, podendo também iniciar uma investigação sobre qual problema pode estar ocorrendo em alguma sessão. O objetivo esperado é que todas as sessões BGP de um roteador mantenham-se no estado *ESTABLISHED*, visto que somente neste estado ocorre a troca de anúncios com o roteador vizinho. Para a administração do PTT esse é o único mecanismo confiável para avaliar a alcançabilidade de um de seus participantes. Essa é a única forma de gerenciar problemas como "NAPs sobre redes Metropolitanas" citados em [KRA01]

Outros problemas como o "Backbone fragmentado" e o acompanhamento das políticas de roteamento aplicadas pelos participantes do PTT, também citado em [KRA01] devem ser monitorados observando-se as variações nos anúncios enviados por cada um dos participantes, tendo-se consciência que muitas das políticas aplicadas relacionam diretamente o volume de tráfego trocado com seus anúncios realizados.

A fim de acompanhar o crescimento da Internet e para atender às novas tecnologias como *Multicast* e o surgimento do Ipv6, foram propostas também

funcionalidades adicionais ao protocolo BGP (*Multiprotocol BGP* [RFC2283]. O sistema proposto nesse artigo também prevê a sua utilização.

3. A Organização da Internet em PTTs

Desde os primórdios da Internet, quando apenas alguns centros de pesquisa dos EUA foram interconectados, com o intuito de compartilhar informações acadêmicas, não se imaginou que a grande rede mundial de computadores, a Internet, chegasse ao ponto que se encontra hoje. O crescimento apresentado assemelha-se a uma curva exponencial. Com esse crescimento tão rápido, a Internet foi deixando de ser uma única rede homogênea para se tornar uma grande rede heterogênea, formada por diversos *backbones* pertencentes a diferentes empresas. Com o objetivo de garantir a ligação regional entre todos essas redes, independente de qual *backbone* estivessem conectados, foi necessário criar pontos de interconexão entre esses *backbones* para não isolá-los dos demais.

Fisicamente um PTT [REC00] é formado por um conjunto de roteadores localizados em um único ponto neutro, formando uma rede local de alta velocidade, geralmente com tecnologias de nível 2 como *Ethernet* ou *ATM* [COM98]. Cada roteador representa um sistema autônomo que deseja trocar tráfego com pelo menos um dos outros participantes. Para a conexão entre todos os roteadores, existe um comutador ou *switch* com alto poder de processamento.

As ferramentas utilizadas em PTTs são basicamente *softwares* que implementam o protocolo BGP e tem por objetivo manter o funcionamento e divulgação de rotas de cada participante. Para tanto são utilizados softwares como o Zebra [ZEB02] e/ou Gated [GAT02] no papel de *route-servers*.

Hoje existem quatro pontos de troca de tráfego conhecidos no Brasil, o PTT-ANSP [ANS03], o Optix [OPT03], o RSiX [RSI03] e o Fix. [FIX03] Outras iniciativas de criação de pontos de troca de tráfego estão em andamento, como o Prix.

4. O sistema de Gerenciamento do protocolo BGP em PPTs

Neste trabalho é apresentado um sistema de gerenciamento SNMP para monitoração do tráfego BGP em PTTs. O objetivo deste trabalho é fornecer informações relevantes para o gerenciamento das operações de roteamento entre sistemas autônomos, através da monitoração, coleta e processamento de informações a respeito do protocolo BGP e características da rede.

No gerenciamento de um PTT, as informações que as MIBs existentes fornecem são importantes [RFC1773] [RFC1657], mas além de encontrarem-se em entidades de diferentes sistemas autônomos, não são suficientemente específicas para a efetiva administração do PTT. Assim sendo este trabalho propõe a implementação de um processo de gerenciamento que facilite a monitoração do roteamento BGP em PTTs. Para alcançar esse objetivo, o processo proposto procura:

- Coletar informações relevantes dessas MIBs em diferentes entidades do PTT.
- Fornecer uma nova MIB contendo apenas as informações específicas e importantes ao gerenciamento do BGP no contexto de PTTs, além de outros objetos coletados diretamente junto do *Route Server*

- Fornecer um sistema de *traps* que observa e reporta comportamentos anormais no dia-a-dia de um PTT.

5. A Arquitetura do Sistema

A arquitetura desenvolvida neste trabalho é apresentada na Figura 2. Nesta arquitetura são representadas todas as entidades das quais obtém-se informações para o gerenciamento do sistema e as demais entidades existentes.

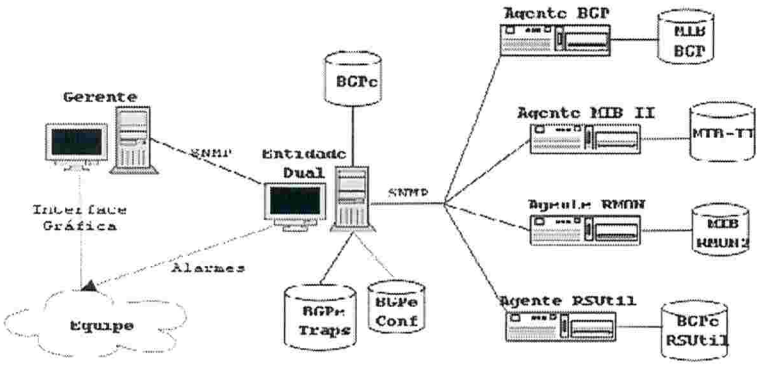

Figura 2 - Arquitetura do sistema proposto

Como pode-se verificar na Figura 2, a entidade Dual é quem fornece as MIBs resultantes do sistema, buscando para isso as informações de 4 agentes localizados em diferentes componentes no PTT via protocolo SNMP. Os resultados obtidos são fornecidos via SNMP para uma estação de Gerência, chamada de entidade Gerente. Através da entidade Gerente a equipe do suporte ao PTT poderá efetivamente fazer uso das informações fornecidas pelo sistema. A função de cada uma das entidades é descrita a seguir:

- Entidade Gerente: Esta entidade é quem dispara solicitações e obtém as informações do sistema. O gerente faz a interface entre a equipe do PTT e o sistema implementado, utilizando o conjunto de informações de diversos sistemas A localização desta entidade geralmente é uma estação de trabalho que tem como função principal o próprio sistema de gerência da rede. O gerente é implementado utilizando recursos/ferramentas já existentes.
- Entidade Dual: Esta é a entidade principal do sistema, pois faz a união das informações que se encontram dispersas nos diversos agentes existentes. Essa entidade implementa três MIBs (BGPe, BGPe Conf e BGPe Traps) propostas pelo trabalho, e faz acesso a diversas MIBs, entre elas a MIB RSUtil. A localização desta entidade pode ser em uma estação ou computador localizados em qualquer rede, desde que tenha permissões para acessar os agentes. Também poderia ser definido como local de utilização deste sistema o próprio gerente, ou ainda, em caso extremo, em algum dos *Route Servers* com alguns ajustes necessários. No protótipo esta entidade está presente em uma máquina exclusiva.
- Agentes: Estas entidades possuem a capacidade de coletar os dados necessários para o sistema. Cada um dos agentes representa um grupo de informações necessárias para a entidade dual. As MIBs BGP, RMON e MIB II já se encontram implementados nos próprios equipamentos utilizados em um PTT. Já o agente BGPe RSUtil foi totalmente implementado. O agente BGP (MIB BGP) é

localizado nos *Route Servers*. Este módulo é baseado na integração entre o *software* responsável pelo roteamento BGP e o *software* NET-SNMP que implementa um servidor SNMP. A MIB BGP utilizada neste agente já é definida e suportada por estas ferramentas. O agente BGPe RSUtil também está presente nos *Route Servers*, sendo implementado para fornecer os *logs* provenientes do *software* de roteamento BGP e algumas informações do próprio sistema operacional dos *Route Servers*. Já os agentes RMON e MIB II estão presentes no próprio *Switch* do PTT, onde todos os participantes estão conectados.

No conjunto destas entidades, um dos benefícios é a unificação de configuração e obtenção de informações para o gerenciamento através da entidade Dual. A visão a partir do gerente e da equipe de suporte é simplificada aos objetos presentes nas MIBs BGPe, BGPeTraps e BgpeConf.

O restante dos agentes pelo qual as informações são obtidas são consultados pela entidade Dual que além de coletar, processa e fornece apenas as informações definidas nas MIBs. Uma das facilidades desta solução é a configuração através da MIB BGPeConf, que pode ser configurada via SNMP a partir de qualquer outro *host* da Internet. Como desvantagem pode-se citar a entidade Dual como ponto único de falha do sistema de gerência, já que se esta falhar, todo o sistema de gerência estará comprometido.

5. A proposta de MIBs BGP complementares

Para a implementação do sistema proposto, foi necessária a criação de um agente que implementa a MIB BGP e a MIB-II. O agente que foi escolhido, e que disponibilizará essas MIBs é o da UCD-SNMP [UCD01] (nas versões mais novas chama-se NET-SNMP). Através dessa integração é possível acessar dados da tabela de roteamento e das sessões estabelecidas entre os *peers*.

5.1 MIBs utilizadas

As principais MIBs já existentes onde essas informações são extraídas são a MIB II, MIB BGP e MIB RMON. Tais MIBs são implementadas na maior parte de equipamentos de rede utilizados atualmente. Parte das informações importantes ao gerenciamento do protocolo BGP já existem, mas para atender as necessidades específicas de um PTT as informações são abrangentes demais e não chegam ao nível de mostrar apenas as informações efetivamente úteis, discriminando-as.

Sendo assim, algumas informações são coletadas das MIBs já existentes e apenas organizadas de acordo com as necessidades do PTT, enquanto outros dados sofrem alguma espécie de análise ou contabilização para depois serem apresentados na forma de uma nova MIB. As MIBs que são utilizadas como base de informações ao sistema são apresentadas a seguir.

5.1.1 A MIB II

Sua principal utilização é na coleta de informações sobre o tráfego de entrada e saída de cada participante do PTT, alterações nessa informação podem ser provocadas por problemas em um dos participantes do PTT, conforme relatados em [KRA01]. Entre os

subgrupos existentes, é utilizado o grupo Interfaces que possui os objetos *IfInOctets* e *IfOutOctets* que retornam tais informações de cada interface.

5.1.2 RMON II

Como já foi visto, o tráfego de entrada e saída de cada participante é importante, da mesma forma é importante saber para onde flui o tráfego entre os participantes. Mediante tal necessidade será usado o grupo *AlMatrix* da MIB RMON para mostrar o tráfego entre pares de participantes. Este grupo possui três tabelas, uma de controle e duas de dados. Essas tabelas de dados são usadas para armazenar informações sobre o tráfego de um *host* origem para um número de *hosts* destinos. A outra tabela contém as mesmas informações, mas é indexada pelo host destino. Com essa informação, é possível montar gráficos mostrando o tráfego entre cada um dos participantes do PTT. Diferentes VLANs são utilizadas para separar diferentes tipos de tráfego no PTT, tais como IPv4, IPv6 e Multicast.

5.1.3 A MIB BGP

Nesta MIB diversos objetos são utilizados. As informações constantes neste agente são utilizadas para verificar todas as informações do protocolo BGP pertinentes a cada participante do PTT. Alguns destes dados são utilizados no processamento dos dados. Exemplos que podem ser dados de objetos utilizados seriam: *bgpPeerState*; *bgpPeerRemoteAS*; *bgpPeerInUpdates*; *bgpPeerOutUpdates*; *bgpPeerLastError*; *bgp4PathAttrPeer*; *bgp4PathAttrIpAddrPrefixLen*; *bgp4PathAttrIpAddrPrefix*; *bgp4PathAttrOrigin*; *bgp4PathAttrASPathSegment*; *bgp4PathAttrNextHop*.

Um exemplo para ilustrar essa MIB é apresentado pela Figura 3, que ilustra um roteador (10.0.0.2) com sessões BGP com três outros sistemas autônomos.

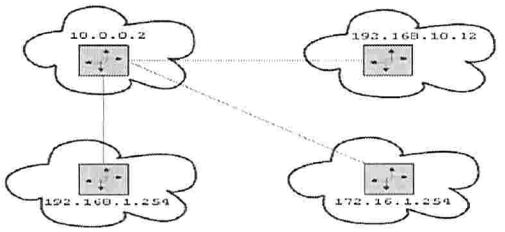

Figura 3- Exemplo de utilização da MIB BGP

A partir deste roteador, é verificado através do objeto *bgpPeerState* o *status* dessas conexões BGP.

```
[root@gerencia]# snmpwalk 10.0.0.2 community .1.3.6.1.2.1.15.3.1.2
15.3.1.2.192.168.1.254 = 6
15.3.1.2.172.16.1.254 = 6
15.3.1.2.192.168.10.12 = 6
```

Figura 4 - Saída do *snmpwalk*

Como se pode verificar na figura 4, o *status* das sessões BGP com os quatro *peers* é de *Established*, definido pelo código 6, de acordo com a seguinte codificação: (1) *Idle*, (2) *Connect*, (3) *Active*, (4) *OpenSent*, (5) *OpenConfirm* e (6) *Established*. Nesse estado a sessão BGP está ativa e ocorre de fato a troca de anúncios, e por conseqüência, a troca de tráfego.

5.2 MIBs propostas

Como resultado do processamento do sistema são disponibilizadas quatro MIBs: BGPe, MIB BGPe Conf, MIB BGPe Traps, MIB RSUtil. Todas elas estão localizadas no grupo *experimental*, que é reservada para MIBs em teste.

5.2.1 A MIB BGPe

Essa MIB fornece informações sobre o estado global do PTT, bem como o estado de cada participante. As informações fornecidas por essa MIB são o resultado do processamento dos dados obtidos através dos agentes. O resultado dessas informações pode ser analisado pelo gerente na forma mais adequada que ele desejar. A MIB BGPe, tem os seguintes objetos, com mostra a Figura 5.

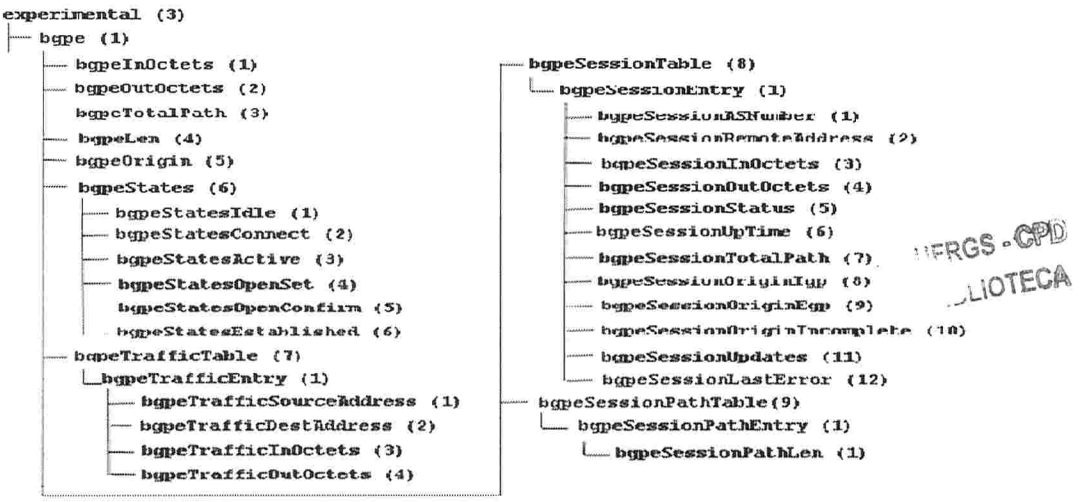

Figura 5 - Especificação da MIB BGPe

Em resumo, a MIB BGPe pode ser dividida em duas partes principais. A primeira delas possui informações globais ao PTT, ou seja, informações que refletem a situação e comportamento de todos os participantes agrupados, sem distinção. Já a segunda parte agrupa informações particulares para cada participante, sendo útil na análise particular de informações de determinado participante.

5.2.2 A MIB BGPeTrap

Este módulo é responsável pelo controle de comportamento anormais no funcionamento do protocolo BGP e dos *Route Servers*. A comunicação é realizada mediante o envio de *traps* ao gerente e/ou envio de e-mail. Neste caso, a equipe do PTT receberá o aviso do evento tão logo ele ocorra, sem atrasos e com a agilidade necessária. Além disso, também é possível registrar em arquivo de LOG todos os eventos reportados por esse módulo para análise posterior e programar o envio de um relatório periódico dos dados mais significativos. A definição de quais eventos serão reportados é feita na MIB de configuração BGPe Conf, que será apresentada adiante.

É apresentado a seguir a definição em formato ASN.1 da MIB BgpeTrap.

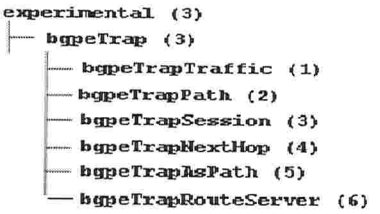

Figura 6 - Especificação da MIB BGPeTrap

```
bgpeTrap    OBJECT IDENTIFIER ::= { experimental 4 }
bgpeTrapTraffic NOTIFICATION-TYPE
 OBJECTS {         bgpeConfSessionBandwithMax    bgpeConfSessionBandwithMin
                   bgpeInOctets                  bgpeOutOctets        }
 STATUS  current
 DESCRIPTION "Gera o trap quando o bgpeInOctets ou bgpOutOctets sair do limite imposto
or bgpeConfSessionBandwithMax e bgpeConfSessionBandwithMin"
 ::= { bgpeTrap 1 }
bgpeTrapPath NOTIFICATION-TYPE
 OBJECTS {         bgpeTotalPath       bgpeConfMaxPath       bgpeConfMinPath      }
 STATUS  current
 DESCRIPTION "Gera o trap quando bgpTotalPath sair do limite imposto por bpeConfMaxPath
e bgpeConfMinPath"
 ::= { bgpeTrap 2 }
bgpeTrapSession NOTIFICATION-TYPE
 OBJECTS {         bgpeSessionStatus    }
 STATUS  current
 DESCRIPTION "Quando o link estiver down, conseguido através do bgpeSessionStatus"
 ::= { bgpeTrap 3 }
bgpeTrapNextHop NOTIFICATION-TYPE
 OBJECTS {  bgpeConfASDeny }
 STATUS  current
 DESCRIPTION " Quando alguém tiver como nexthop algum de bgpeConfASDeny "
 ::= { bgpeTrap 4 }
bgpeTrapAsPath NOTIFICATION-TYPE
 OBJECTS { bgpeConfASDeny       }
 STATUS  current
 DESCRIPTION " Quando alguém tiver como path algum de bgpeConfASDeny "
 ::= { bgpeTrap 5 }
bgpeTrapRouteServer NOTIFICATION-TYPE
 OBJECTS {        bgpeConfRouteServerCPUMax      bgpeConfRouteServerMemoryMax
                  bgpeConfRouteServerProcess     bgpeLogRouteServerCPUMax
                  bgpeLogRouteServerMemoryMax    bgpeLogRouteServerProcess     }
 STATUS  current
 DESCRIPTION " Quando alguém tiver como path algum de bgpeConfASDeny "
 ::= { bgpeTrap 56}
```

Figura 7 – Especificação em formato ASN.1 da MIB BGPeTrap.

5.2.3 A MIB BGPeConf

Esta MIB é responsável pela determinação dos dados de cada participante e do PTT de forma global. Através desses dados são repassados ao sistema as informações necessárias para coleta e processamento dos dados a serem disponibilizados pela BGPe. Também serão utilizadas essas informações para a determinação dos parâmetros que definem comportamentos anormais ou aceitáveis, permitindo que a BGPe Trap gere *traps* ou alarmes. A especificação desta MIB é exibida na figura 8. Para configurar tais informações pode ser utilizado um MIB *Browser*, utilizando para isso uma *community* com permissão de escrita e leitura.

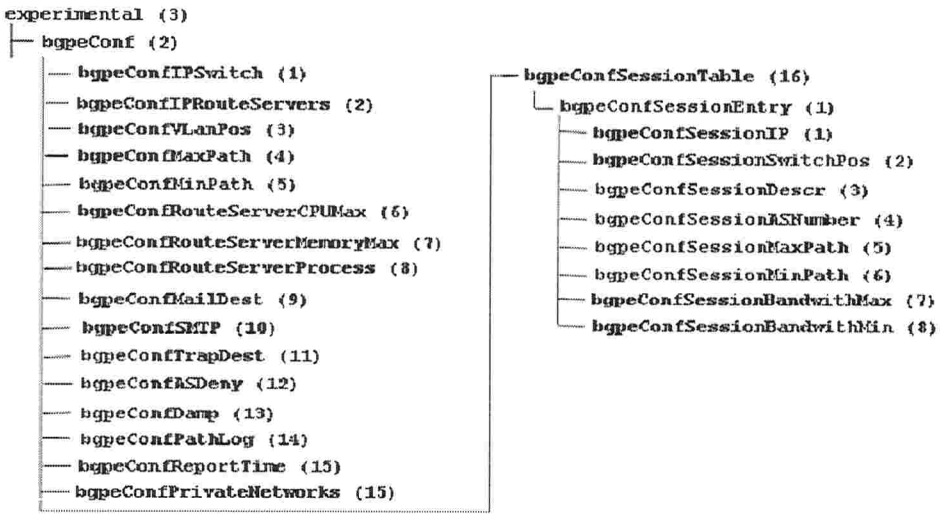

Figura 8 - Especificação da MIB BGPeConf

Os objetos contidos nessa MIB são:

BgpeConfIPSwitch: Determina o endereço IP do *Switch* do PTT.

BgpeConfIPRouteServers: Determina o endereço IP dos *Routes Server*.

BgpeConfVlanPos: Determina a referência da VLAN no *Switch* que engloba todas as portas dos participantes do PTT.

BgpeConfMaxPath: Determina o número máximo de prefixos aceitável no PTT.

BgpeConfMinPath: Determina o número mínimo aceitável de prefixos do PTT.

BgpeConfRouteServerCPUMax: Utilização de CPU máxima esperada dos *RSDs*.

BgpeConfRouteServerMemoryMax: Utilização de memória máxima esperada dos *RSDs*.

BgpeConfRouteServerProcessMax Número máximo de processos esperados dos *RSDs*.

BgpeConfMailDest: Determina o mail de destino da equipe de suporte ao PTT.

BgpeConfSMTP: Determina o IP do servidor SMTP para postagem dos alarmes.

BgpeConfTrapDest: Determina o IP da estação de gerência que recebe as *TRAPs*.

BgpeConfASDeny: Determina o número dos ASs que não deverão constar em nenhum campo AS_PATH de anúncios provenientes de participantes do PTT.

BgpeConfDamp: Ativa o envio de alarmes quando alguma rota entrar em *dampening*.

BgpeConfPathLog: Determina a localização do arquivo geral de *log* do sistema no *host* que a ferramenta estiver sendo executada.

BgpeConfReportTime: Determina o intervalo de tempo que deve ser enviado um relatório sobre as atividades mais importantes do PTT.

BgpeConfPrivateNetworks: Determina as redes privadas que não devem estar incluídas nos anúncios dos participantes dos PTTs.

BgpeConfSessionTable: Tabela que possui dados importantes de cada participante, que são utilizados para o acesso a informações e envio de possíveis alarmes. Para cada um

dos participantes, serão preenchidos os objetos *SessionIP*, *SwitchPos*, *Descr*, *AsNumber*, *MaxPath*, *MinPath*, *BandwithMax* e *BandwithMin*, que significam respectivamente: IP do roteador utilizado na sessão BGP, posição no *Switch*, descrição para identificar o participante, número do AS, número máximo e mínimo de anúncios esperados e banda do circuito até o PTT.

5.2.4 A MIB RSUtil

Essa MIB é totalmente implementada baseada em dados extraídos dos *Route Servers*. Os *Route Servers* tem o poder de controlar diversas informações, como por exemplo, o número máximo e mínimo de anúncios de determinado participante, entradas/saídas de anúncios de rotas que podem resultar em *dampening*, aspectos de segurança e estabilidade dos próprios *Route Servers*, entre outros. Esses dados são explorados e reportados por essa MIB. A Figura 9 apresenta os objetos da MIB em formato ASN.1.

```
bgpeRSUtil DEFINITIONS ::= BEGIN
IMPORTS;
bgpeRSUtil::= { experimental 3  }
bgpeRSUtilDampennedNetworks OBJECT-TYPE
 SYNTAX DisplayString
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Representa as redes que estão incluídas em dampenning "
 ::= { bgpeRSUtil 1 }
bgpeRSUtilMaxAnnounc OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION "Registro do AS cujo número de anúncios máximos foi excedido."
 ::= { bgpeRSUtil 2 }
bgpeRSUtilRouteServerCPUUser OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Registra informações da utilização da CPU do sistema operacional
dos Route Servers pelo usuário "
 ::= { bgpeRSUtil 3 }
bgpeRSUtilRouteServerCPUSystemOBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Registra informações da utilização da CPU do sistema operacional
dos Route Servers  pelo sistema"
 ::= { bgpeRSUtil 4 }
bgpeRSUtilRouteServerCPUIdle OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Registra informações do utilização da CPU do sistema operacional
dos Route Servers  em Idle"
 ::= { bgpeRSUtil 5 }
bgpeRSUtilRouteServerMemoryUsed OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Registra informações do uso de memoria do sistema operacional dos
RSDs"
::= { bgpeRSUtil 6 }
bgpeRSUtilRouteServerMemoryFree OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
```

```
 STATUS mandatory
 DESCRIPTION " Registra informações do uso de memoria do sistema operacional dos
RSDs"
 ::= { bgpeRSUtil 7 }
bgpeRSUtilRouteServerProcess OBJECT-TYPE
 SYNTAX INTEGER
 ACCESS read
 STATUS mandatory
 DESCRIPTION " Registra informações do número de processos do sistema operacional
dos Route Servers"
 ::= { bgpeRSUtil 8 }
bgpeRSUtilRouteServerLogGeneral OBJECT-TYPE
 SYNTAX DisplayString
 ACCESS read
 STATUS mandatory
 DESCRIPTION "Trechos de LOGs relevante registrado pelos RSDs"
 ::= { bgpeRSUtil 9 }
END
```

Figura 9 – Especificação em formato ASN.1 da MIB BGPeRSUtil.

6. Implementação do sistema

A modelagem das entidades SNMP baseou-se em [STA99], que aborda detalhadamente este tipo de entidade. A implementação foi feita usando a linguagem de programação Java, utilizando a versão 1.4 do *Java Development Kit*.

Para a implementação do sistema foram criadas duas entidades, a citar: entidade Dual e Agente. Como a diferença de operações entre elas é minima [DAN98] [ROS90], implementou-se um modelo orientado a objetos onde a diferença é de apenas algumas classes, sendo que a estrutura básica das entidades pode ser considerada a mesma. As principais classes que formam as entidades Dual e Agente, são citadas a seguir:

- *Classe Pendente*: Esta classe, apesar de não estar prevista na modelagem presente em [STA 99], foi necessária pela razão que a solicitação de algumas variáveis exige um processamento especial, ou seja, fazer o envio de outras requisições para então processar tais valores e formar a resposta.
- *Classe CommandGenerator*: Esta classe é responsável em iniciar as mensagens de *Get*, *GetNext*, *GetBulk* e *Set*. Ela também vai processar as respostas que vierem dessas requisições. O seu uso mais evidente é no caso de uma aplicação Gerente, pois acessa diretamente o *CommandGenerator* para mandar requisições.
- *Classe CommandResponder*: A classe CommandResponder recebe as solicitações SNMP da classe Dispatcher e baseada na versão da Pdu recebida, repassa para a classe V1Methods se for SNMPv1, classe V2Methods se for SNMPv2 ou a classe V3Methods caso a versão seja SNMPv3. Essas classes recebem os dados da Pdu e executam as operações necessárias, de acordo com a versão especificada, retornando os dados solicitados. Atualmente o sistema encontra-se preparado para aceitar requisições versão 1 e 2, sendo necessários alguns ajustes para a versão 3.

6. Validando o Sistema

Tendo o intuito de buscar uma forma de simulação de um PTT para testes, foi montada uma estrutura com 7 computadores com sistema operacional FreeBSD, rodando o *software* Zebra, afim de simular um PTT baseado em acordos multilaterais.

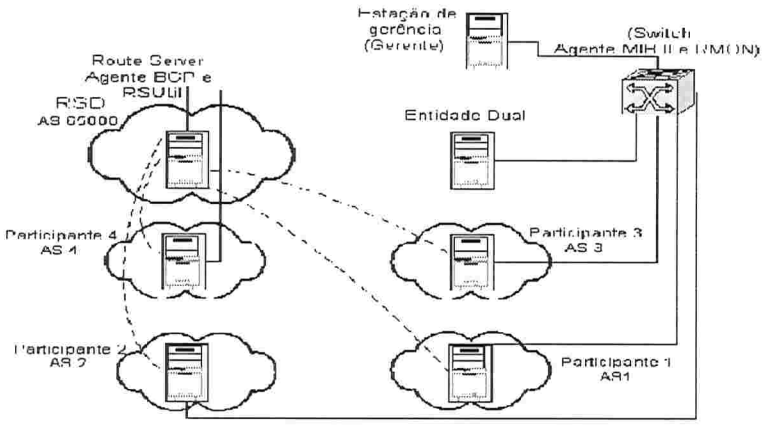

Figura 10 - Estrutura para simulação de um PTT

Além do software Zebra, citado como uma das ferramentas utilizadas em PTTs, os seguintes *softwares* foram instalados para possibilitar a implementação do protótipo como um PTT:

- *JDK*: O *Java Development Kit* compilador e interpretador para a linguagem Java. Utilizado para rodar os componentes implementados no trabalho.
- *NET-SNMP*: Conhecido também como UCD-SNMP, é uma ferramenta que implementa o daemon SNMPD, utilizado para o acesso e obtenção de informações em determinado *host* via SNMP. Também implementa um cliente para consultas SNMP em outros *hosts*. Possui também outras ferramentas, como um receptor de *traps*.
- *RRDTool* e *NRG*: Sucessor do popular MRTG, é utilizado para representar graficamente informações obtidas via SNMP. É um facilitador para visualizar a ocorrência de eventos anormais.
- *JOpenEyes*: Implementado em Java com recursos gráficos, este programa tem funcionalidades semelhantes a sistemas de gerência comerciais como *Openview*, *Netview*, entre outros. Utilizado principalmente para o recebimento de *traps*.

6.1 Simulação de problemas comuns em PTTs.

Alguns dos problemas mais comuns em PTTs foram reproduzidos afim de validar o sistema no protótipo apresentado, como mostrado a seguir:6.1.1 Queda da sessão BGP de algum participante

A queda da sessão BGP de algum participante faz com que não sejam mais feitos anúncios das redes do participante e por conseqüência, não troque mais tráfego no PTT. Em ocasiões normais, a verificação do *status* dos participantes seria feita através da console de um *Route Server* ou pela monitoração de objetos da MIB BGP. Ambos os métodos não são eficientes pois apenas sinalizam o evento, necessitando que a equipe de suporte analise continuamente tais valores para diagnosticar problemas

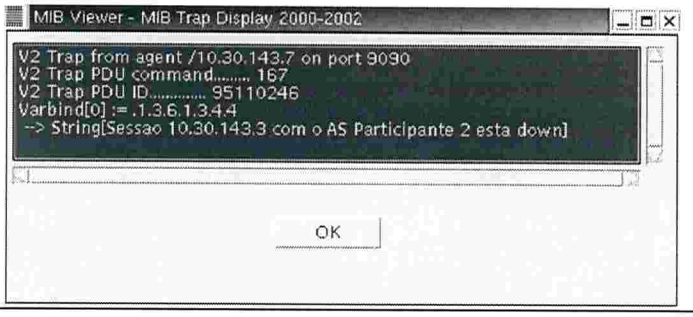

Figura 11 - Trap comunicando problemas com participante

6.1.2 Quantificação da profundidade de anúncios

Mesmo sabendo do número de anúncios de determinado participante, não se sabe a quantidade de anúncios de acordo com sua profundidade, ou seja, se os anúncios são mais ou menos específicos. Em geral, são atribuídos a sistemas autônomos blocos CIDR de tamanho 20 (ou seja, /20 em notação CIDR), mas é muito comum que estes blocos sejam anunciados em sub-blocos menores. Esse detalhe pode parecer simples, mas facilita o troubleshooting da rede, já que muitos participantes e, em alguns casos, o próprio PTT limite a profundidade.

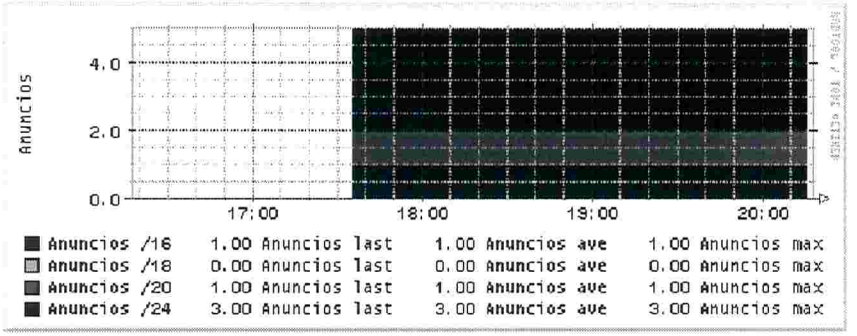

Figura 12 - Gráfico do número de anúncios classificados por profundidade.

6.1.3 Número de anúncios fora do intervalo esperado

Este problema pode ser muito comum em PTTs e deve ser rapidamente detectado para verificar o que está provocando tal ação. Em geral, quando um participante entra em um PTT, a administração limita o número de blocos que poderão ser anunciados pelo participante através de filtros nos *Route Servers*, essa estratégia evita que anúncios indevidos sejam repassado aos demais participantes. Mesmo com esses filtros, o problema acaba sendo apenas remediado, visto que os anúncios em excesso continuarão a ocorrer, sendo descartados pelo *Route Server*, muitas vezes erroneamente.

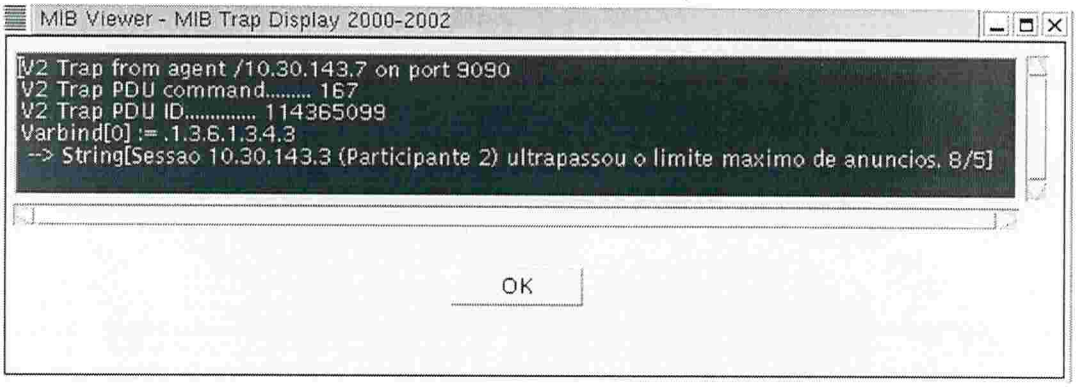

Figura 13 - Trap que comunica o número total de anúncios excedido.

6.1.4 Número de anúncios e histórico de anúncios de participantes

O número blocos anunciado por cada participante é um dado importante para analisar o comportamento de um AS em um PTT. Através da MIB BGP não é possível obter o total de blocos anunciados por determinado AS, sendo possível obter, no máximo, o conjunto total de anúncios no PTT por todos os participantes. A única forma de obtenção dessas informações seria através da console do software Zebra, que através do comando "show ip bgp summary" fornece informações, entre estas, o total de blocos anunciados. Mesmo com isso, não é possível manter nenhum histórico sobre essa informação de cada participante. Através da MIB implementada isso se torna possível.

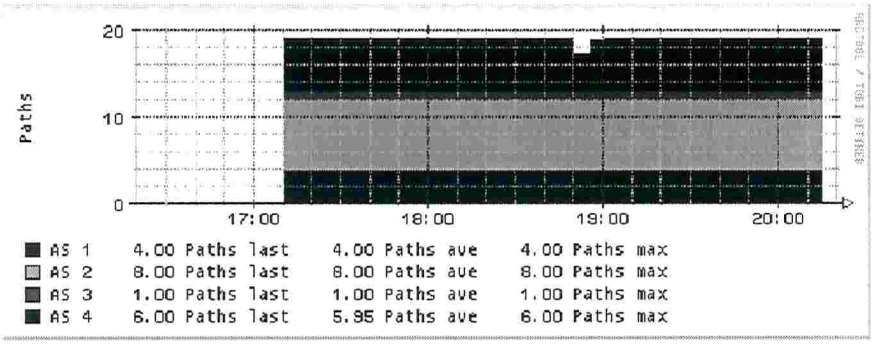

Figura 14 - Gráfico que mostra o total de anúncios de cada participante.

7. Conclusões

Os objetivos propostos neste trabalho foram alcançados e resultaram nas seguintes contribuições:

- Implementação de um agente SNMP versão 2, preparado para ser estendido para a versão 3;
- Implementação de um gerente SNMP versão 2, também preparado para ser estendido para a versão 3;
- Implementação de um MIB compiler, que pode agregar novas MIBs em sua base de dados;
- Definição de informações importantes ao gerenciamento local de PTTs.
- Especificação de MIBs experimentais que podem ser utilizadas em outros PTTs para o gerenciamento do protocolo BGP;

Como um dos resultados alcançados, e provando a utilização do sistema e sua significância sobre o tema abordado. Um dos PTTs existentes no país, o RSiX, está avaliando sua implantação como forma de gerência específica para essa variedade de problemas.

Adicionalmente, como o trabalho ainda é tido como um protótipo, alguns trabalhos futuros ainda precisam ser realizados, entre eles, pode-se citar:

- Testes de performance sobre a capacidade de processamento que o sistema proposto atinge.
- Utilização do sistema de forma didática para auxiliar os alunos da disciplina de redes sobre os tópicos abordados pelo trabalho, a citar: roteamento BGP, gerência através de SNMP e Pontos de troca de tráfego.
- Integração total com o software NET-SNMP, de forma a permitir a distribuição da MIB junto com um software padrão.
- Disponibilização dos fontes do sistema a comunidade acadêmica para dar continuidade ao trabalho.
- Adaptação e aplicação do sistema em PTTs nacionais, como é o caso do RSIX.

Referências

[ANS03]Rede ANSP - An Academic Network at São Paulo – on line – 2003 – http://www.ansp.br

[CAI03] *The Caida Website* – on line – 2003 – http://www.caida.org

[CID02] Today CIDR Report – on line – 2002 – http://www.employees.org/~tbates/cidr-report.html

[CID03] CIDR Report – on line – 2003 - http://www.cidr-report.org/as1221/index.html

[COM98] DOUGLAS E. COMER. Interligação em rede com TCI/IP. Rio de Janeiro : Campus, 1998. Volume I e II.

[DAN98] DANIELE, M., WIJNEN, B., FRANCISCO, D. Agent Extensibility Protocolo (Agent X). RFC 2257. IETF, Janeiro, 1998.

[FIX03] Ponto de Troca de Tráfego de Brasília – FIX – on line - 2003 http://www.rnp.br/noticias/2002/not-021119b.html

[GAT02] *Software* GATED – on line – 2002 - http://www.nexthop.com/products/gated.shtml

[GTE03] Grupo de Trabalho em Engenharia de Redes – GTER – on line - http://www.gt-er.cg.org.br/

[KRA01] Segurança em Pontos de Troca de Tráfego – 14ª Reunião do Grupo de Trabalho em Engenharia de Redes – GTER 14 – 2001 – on line - ftp://ftp.registro.br/pub/gter/gter14/aspectos_sec_ptt_bertholdo.ppt e http://www.rsix.tche.br/artigos/artigo-ptt.pdf

[OPT03] Optiglobe – OPTix-LA – on line – 2003 – http://www.optiglobe.com

[PRI03] Ponto de Troca de Tráfego do POP-PR/RNP – PRIX – on line – 2003 - http://prix.pop-pr.rnp.br/

[REC00] Comitê Gestor Internet/BR – Grupo de Trabalho em Engenharia de Redes - Operação e Administração de PTTs no Brasil – on line – 2000 - http://www.cg.org.br/grupo/operacao_ptt_v1.1.htm

[RFC1657] Definitions of Managements Objects for version of the Border Gateway Protocol using SMIv2 – on line – 1994 - http://www.ietf.org/rfc/rfc1657.txt

[RFC1771] A Border Gateway Protocol 4 (BGP-4) – on line – 1995 - http://www.ietf.org/rfc/rfc1771.txt

[RFC1773] Experience with the BGP-4 protocol - on line – 1995 - http://www.faqs.org/rfcs/rfc1773.html

[RFC2283] Multiprotocol Extensions for BGP-4 – on line – 1998 - http://www.ietf.org/rfc/rfc2283.txt

[ROS90] ROSE, Marshall, McCLOGHRIE, Keith. Structure and Identification of Management of TCP/IP based internets. Request for Comments 1155, DDN Network Information Center, SRI International, Maio, 1990.

[RSI03] RSIX - Ponto de Troca de Tráfego Internet – on line - 2003 – http://www.rsix.tche.br

[STA99] STALLINGS, Willian. SNMP, SNMPv2, SNMPv3 and RMON 1 and 2. Addison-Wesley, 3ª Edition, 1999.

[SAM00] SAM HALABI, DANNY MCPERSON. Internet Routing Architectures, Second Edition. Indianapolis – USA : Cisco Press, 2000

[UCD01]*Software* UCD-SNMP – on line – 2002 - http://www.net-snmp.org

[ZEB02]*Software* Zebra – on line – 2002 - http://www.zebra.org