**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**Ricardo Gazola**

# FAILOVER DE LINKS COM ROTEADOR MIKROTIK

**FLORIANÓPOLIS**
**2013**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**Ricardo Gazola**

# FAILOVER DE LINKS COM ROTEADOR MIKROTIK

Trabalho de Conclusão do Curso de Graduação em Sistemas de Informação, do Centro de Ciências Tecnológicas da Universidade Federal de Santa Catarina, requisito parcial à obtenção do título de Bacharel em Sistemas da Informação. Orientação de: Prof. Dr. Carlos Becker Westphall.

**FLORIANÓPOLIS**
**2013**

**Acadêmico: Ricardo Gazola**

**Título: Failover de Links com roteador Mikrotik**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação, do Centro Tecnológico da Universidade Federal de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação, aprovado com nota ____.

Florianópolis, _____ de _____ de 2013 .

_____

Dr. Carlos Becker Westphall, UFSC

Professor Orientador

_____

Dra. Carla Merkle Westphall, UFSC

Membro da Banca Examinadora

_____

Dr. João Bosco M. Sobral, UFSC

Membro da Banca Examinadora

# RESUMO

Com o crescimento de serviços baseados na web, a necessidade de manter estes serviços em funcionamento vinte e quatro horas por dia tornou-se primordial, assim como a complexidade para tal. O alto custo e a complexidade fazem com que diversas empresas pequem na implantação destas tecnologias de infraestrutura de redes e deixem este quesito em segundo plano. Este trabalho de conclusão de curso irá apresentar uma técnica de implantação de redundância de links chamada failover de links, que mostrará através da utilização de scripts em roteadores da Mikrotik, de baixo custo e alta flexibilidade, uma forma de manter o ambiente redundante e com baixa indisponibilidade.



**Palavras-chaves**: Failover; Links; Redes; Roteador.

# ABSTRACT

With the growth of web-based services, the need to keep these services up and running 24 hours a day has become paramount, as well as the complexity for such. The high cost and complexity make several firms sin deployment of these technologies in network infrastructure and leave this question in the background. This work of course completion will present a technique for deploying redundant links called failover link, which will show through the use of script in the Mikrotik router, low cost and high flexibility, a way to keep the environment redundant and low unavailability.



**Keywords**: Failover; Link; Network; Router.

# LISTA DE FIGURAS

## SUMÁRIO

# 1 INTRODUÇÃO

A relação entre o sucesso de uma aplicação e o fracasso da mesma dentro de uma organização está fortemente relacionada com a disponibilidade do ambiente em que se localizam. Com o advento da informação e a rápida expansão que proporciona, as aplicações dependem de uma infraestrutura de redes confiável e que atinja o máximo possível de disponibilidade. O uso de múltiplos links de acesso à internet está cada vez mais comum e mais necessário, porém com o grande número de acessos que acarreta, traz consigo uma complexidade elevada, por isso o uso de ferramentas e técnicas de redundância e automatização do ambiente de redes torna-se uma das peças chaves do sucesso ou fracasso da aplicação e indiretamente da organização.

A escolha das tecnologias utilizadas para tais procedimentos muitas vezes são complexas e exigem muito conhecimento da estrutura em que será implantada, a grande maioria dos equipamentos ou conjuntos de equipamentos que realizam estas técnicas possuem um custo elevado pelo grau de complexidade que possuem, em contrapartida, são mais completas ou possuem um suporte maior que tecnologias gratuitas ou de baixo custo.

Este trabalho foi implantado utilizando tecnologia paga sem grande custo aquisitivo, porém fornece grande flexibilidade nas operações de roteamento de redes, disponibilidade e segurança, que será usada para criar um ambiente de links de acesso à internet redundante, chamado de failover de links, o que proporcionará um ambiente com maior disponibilidade e menor tempo de resposta a um evento de queda.

## 1.1 MOTIVAÇÕES E JUSTIFICATIVA

O ambiente corporativo exige flexibilidade e ao mesmo tempo segurança, assim como manter a disponibilidade de todos os serviços de forma ininterrupta. Para isso a infraestrutura de redes tem que ser bem planejada para atender a tudo o que é exigido dela. Deve ser estruturado para atender a milhares de acessos simultâneos, deve ser capaz de suportar publicações com restrições de destino e fonte, ser possível controlar tráfego

e banda, assim como limitar recursos e priorizar serviços.

Um grande problema quando a estrutura tem diversos links de comunicação, cada qual com serviços específicos, este é o cenário em que o trabalho foi aplicado, resolver um problema de contingencia e disponibilidade dos diferentes links de comunicação e prover o mínimo possível de indisponibilidade.

## 1.2 OBJETIVOS

Os objetivos foram estabelecidos de acordo com a prioridade da empresa, a principal prioridade é a troca de forma automatizada dos links quando houver quedas, o que antes era considerado um ponto de falha na empresa. Os objetivos específicos, não menos importantes que o principal, são relacionados aos serviços que a empresa fornece, como publicações, serviço de VPN, e a segurança, que será necessária, empregar em cada um dos roteadores, para garantir o controle de entrada.

### 1.2.1 GERAL

Após análise da atual estrutura de redes e dos serviços da empresa, será verificado o processo de alteração de link, quando da ocorrência de quedas nos links de comunicação que são muito lentas, além de alterações nos servidores da empresa, serão necessárias alterações em aplicações e nos roteadores, o que totalizará um tempo de indisponibilidade nos serviços muito alto, prejudicando indiretamente o suporte aos clientes, pois muito dos acessos a clientes são feitos através de VPNs que dependem de uma conexão com a internet e algumas vezes de um link especifico.

O objetivo principal deste trabalho é a automatização do processo de mudança de links quando há quedas de comunicação, visando à diminuição do tempo de indisponibilidade dos serviços da empresa.

Para automatizar o processo será usado à linguagem de script própria dos roteadores, que fornece todos os meios necessários para cumprir a tarefa, sem que seja necessária a compra de ferramentas extras para a automatização deste processo.

Para cumprir o objetivo será realizado um profundo estudo da

tecnologia que será utilizada, pelo fato dos roteadores já estarem sendo utilizados na empresa. O estudo focará em como realizar está tarefa da melhor forma possível, usando os recursos disponíveis, sem a necessidade da compra de outros equipamentos ou da troca dos mesmos.

## 1.2.2 ESPECÍFICOS

Partindo do sucesso do objetivo principal, que é a automatização do processo de mudança de link, houve a necessidade de outros objetivos mais específicos que serão descritos adiante, os quais são tão importantes quanto o objetivo principal. Sem estes objetivos, a seguir, a continuação do trabalho se tornaria praticamente impossível, pois não basta apenas trocar os links e manter a conectividade, mas também manter todos os serviços em funcionamento, para isso os objetivos a seguir servem para manter os serviços da empresa em funcionamento, assim como a segurança em toda a rede.

**a) Disponibilidade das publicações:**

Além da automatização do processo de troca de links, era necessário que esta troca não fosse demorada, para isso seria preciso repensar a forma como algumas aplicações eram acessadas externamente, para que quando o script de automatização de troca fosse invocado, não dependesse de qualquer outra troca manual, tanto em servidores como em aplicações.

Para cumprir este objetivo será necessário que as aplicações não dependem de um link de comunicação especifico, para isso será adicionado um IP de rede privada nos servidores e aplicações que necessitam ser acessadas externamente. Todos os roteadores também possuem um IP da mesma rede privada, possibilitando assim a comunicação entre estes serviços, e a troca dos links sem prejudicar o acesso às aplicações e publicações.

**b) Segurança de acesso:**

Para o objetivo anterior ser totalmente atingido será necessário garantir os acessos às aplicações e as publicações através de mecanismos de firewall, como NAT, source NAT, destination NAT e regras de aceitação, assim como controle de restrições.

Para manter a confiabilidade dos serviços será necessária a configuração dos firewalls em todos os roteadores envolvidos no projeto. Para tal procedimento será utilizado à interface gráfica dos roteadores, ou seja, não se utilizará scripts para o processo de segurança.

**c) Reestabelecimento de conexões VPN:**

Um dos pontos importantes para que a solução seja completa, é o restabelecimento das conexões VPN, para isso é necessário manter os serviços de VPN funcionando nos roteadores a cada troca de link. Para manter as conexões VPN estabelecidas é preciso também reestabelecer os serviços de NAT de cada uma das VPNs conectadas ao roteador.

## 1.3 ORGANIZAÇÕES DO TRABALHO

A organização do trabalho está dividida em capítulos, cada capítulo tratará de um determinado assunto, sendo o primeiro um capitulo introdutório ao trabalho.

O capítulo dois trata dos assuntos referentes à fundamentação teórica, a qual esclarece assuntos e tecnologias que são necessárias conhecer e entender para que seja possível alcançar o objetivo principal e os específicos.

O capitulo três trata da principal parte do trabalho, que é a proposta em si, explicando como o trabalho foi realizado e quais as metodologias usadas, assim como os processos e tecnologias. Também mostra os resultados obtidos e analises da proposta.

O capitulo quatro trata das conclusões do trabalho, demonstra de forma resumida os passos seguidos para chegar aos resultados, algumas dificuldades encontradas e complementações que não puderam ser implementadas neste momento.

Os demais capítulos não são textuais.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentada toda a fundamentação teórica necessária para o entendimento do trabalho, priorizando os aspectos essenciais empregados na implementação da solução aqui estudada, e tendo como referencia as funcionalidades do roteador Mikrotik, que foi a ferramenta principal usada em todo o projeto. Não será abrangido de forma avançada, apenas explicada as principais fundamentações que foram necessárias estudar para manter os serviços da empresa disponíveis durante e após a implementação do trabalho.

Os assuntos aqui abordados serão sobre a ferramenta utilizada, sistema operacional da Mikrotik nos roteadores, serviços de VPN, roteamento estático e dinâmico, apesar do trabalho utilizar apenas roteamento estático. Também serão comentados brevemente sobre roteamento dinâmico, aspectos importantes de redes como, gateway, IP, NAT e DNS, aspectos de segurança empregados no trabalho como firewall, regras e cadeias, input, output e forward.

## 2.1 MIKROTIK

*Mikrotik* é um sistema operacional para servidores e roteadores *“routerboard”,* que são equipamentos oferecidos já com o sistema operacional instalado. O *Mikrotik* pode ser instalado em um desktop, por exemplo, ou em uma máquina virtual, que foi aplicado neste trabalho pelo fato de trazer uma alta flexibilidade e redundância do sistema como um todo.

O *Mikrotik* é um sistema completo, com todas as ferramentas necessárias para roteamento dinâmico ou estático, para criação de VPNs de qualquer tipo, possui *firewall* completo com todas as funcionalidades necessárias para marcação de pacotes, bloqueios e controle de acesso.

A principal característica destes roteadores, além da sua ampla flexibilidade e uso, é a possibilidade de criação de scripts utilizando a própria linguagem do sistema operacional, o sistema vem com uma console a qual é possível executar inúmeros comandos, dentre eles a formação de *arrays*,

*strings*, variáveis globais e locais, laços como *while* e *for*, IF e *else*, entre outras semelhantes à linguagem de *scripting* como php ou *VBscript*.

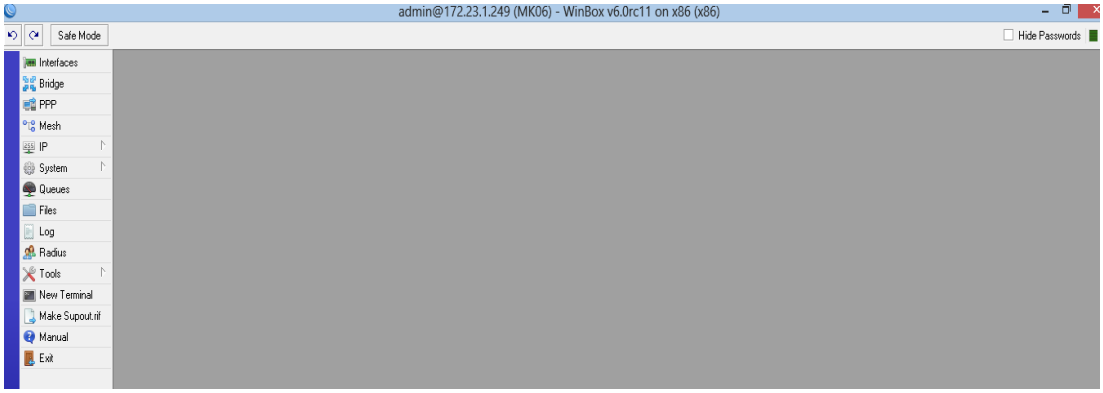


*Figura 1 – Roteador Mikrotik*

## 2.2 ROTEAMENTO

O roteamento é a principal forma utilizada na Internet para a entrega de pacotes de dados entre hosts (equipamentos de rede de uma forma geral, incluindo computadores, roteadores etc.). É a tarefa de escolher o melhor e mais rápido caminho para entrega de uma determinada informação, para esta tarefa é necessário que o roteador conheça diversos caminhos, endereços que, senão para entrega direta, venham a levar a outro roteador que possivelmente saberá a quem entregar.


*Figura 2 – Roteamento*

Se desejarmos enviar uma mensagem de qualquer terminal (Figura 2) para a internet em um determinado endereço, o caminho que a mensagem fará, com certeza deve passar pelo roteador, e este roteador deve conhecer o caminho, ou conhecer outro roteador que leve a tal endereço, disto se conclui, de uma maneira geral, que roteadores devem ter um mapa, ou uma tabela onde possam armazenar estes endereços de saída, está tabela pode ser dinâmica (automática) ou estática (manual).

Roteador, basicamente, é um computador, dedicado ou não, ligado a uma ou mais redes fisicamente ou virtualmente. O trabalho do roteador é verificar para qual interface de rede deverá ser enviada a informação, bem como unir redes com diferentes protocolos. Ao receber um datagrama (interno ou externo), o roteador retira de dentro do *frame* (encapsulado pelo endereço físico da placa de rede – *Mac Address*) e verifica o endereço de destino (contido no datagrama). Caso não seja o seu próprio endereço ou de uma das redes internas pela qual seja responsável, o roteador descarta o datagrama. Entretanto, caso contrário, se pertencer ao mesmo, o roteador verifica para qual interface de rede deve encaminhar o datagrama.

## 2.2.1 ROTEAMENTO ESTÁTICO

O roteamento estático normalmente é configurado manualmente quando uma tabela de roteamento estático é construída manualmente pelo administrador do sistema, uma rede com um número limitado de roteadores para outras redes poderem ser configuradas com roteamento estático, e pode ou não ser divulgada para outros dispositivos de roteamento na rede.

Tabelas estáticas não se ajustam automaticamente a alterações na rede, portanto devem ser utilizadas somente onde as rotas não sofrerem alterações. Algumas vantagens do roteamento estático são a segurança obtida pela não divulgação de rotas que devem permanecer escondidas e a redução do *overhead* introduzido pela troca de mensagens de roteamento na rede.

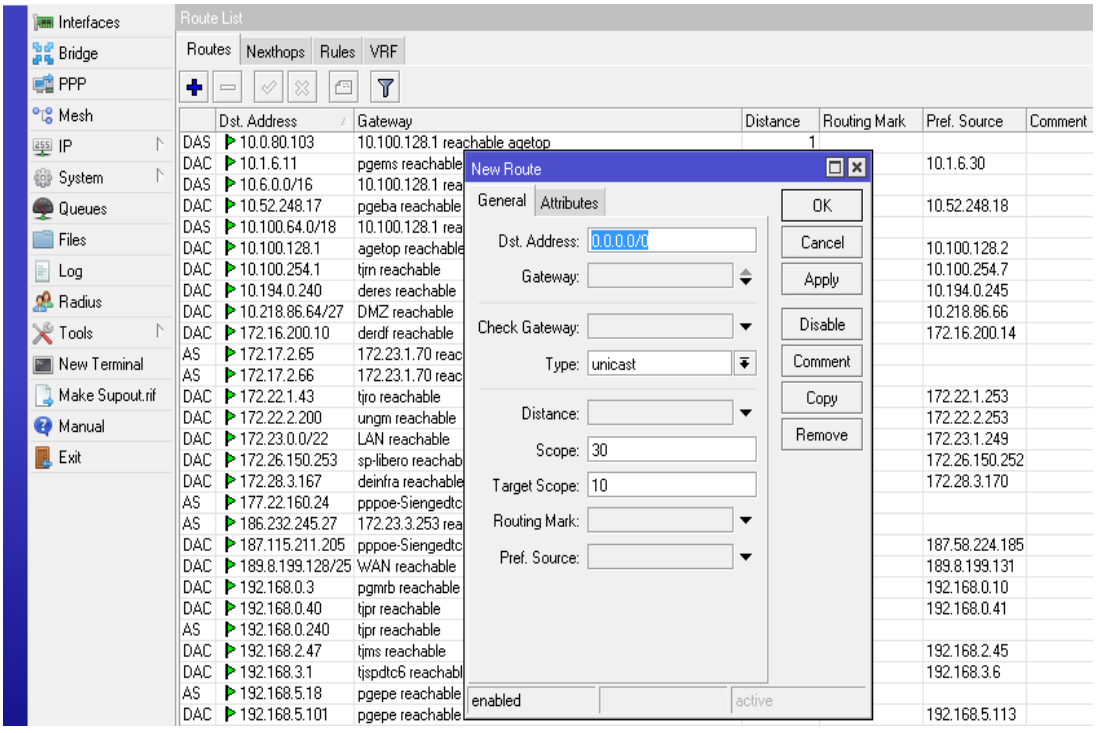


*Figura 3 – Roteamento estático Mikrotik*

## 2.2.2 ROTEAMENTO DINÂMICO

São redes com mais de uma rota possível para o mesmo ponto, devem utilizar roteamento dinâmico. Uma tabela de roteamento dinâmico é

construída a partir de informações trocadas entre protocolos de roteamento.

Os protocolos são desenvolvidos para distribuir informações que ajustam rotas dinamicamente para refletir alterações nas condições da rede. Protocolos de roteamento podem resolver situações complexas de roteamento mais rápida e eficientemente que o administrador do sistema, eles são desenvolvidos para trocar para uma rota alternativa quando a rota primária se torna inoperável e para decidir qual é a rota preferida para um destino, em redes onde existem várias alternativas de rotas para um destino onde devem ser utilizados.

## 2.3 FIREWALL

*Firewall* é um dispositivo ou uma ferramenta, dependendo do contexto, de uma rede de computadores que tem por objetivo aplicar uma política de segurança a um determinado ponto de controle da rede. Sua função consiste em regular o tráfego de dados entre redes distintas e impedir a transmissão e/ou recepção de acessos nocivos ou não autorizados de uma rede para outra. Esse conceito inclui os equipamentos de filtros de pacotes e de *Proxy* de aplicações, comumente associados a redes TCP/IP.

Existe na forma de *software* e *hardware*, ou na combinação de ambos, chamado de *appliance*. No *Mikrotik* temos tudo isso de forma personalizada, sendo possível desenvolver políticas baseadas em uma rede de arquitetura TCP/IP, com os benefícios de um sistema operacional para roteador.

Mecanismos de *firewall* são muito usados na prática para aumentar à segurança de redes ligadas a internet, uma espécie de barreira de proteção, segundo Luiz Fernando Gomes Soares, Guido Lemos e Sérgio Colcher, a segurança normalmente é inversamente proporcional à complexidade, assim proteger computadores e dispositivos em uma rede com diferentes sistemas operacionais, de pequeno e grande porte, geralmente é uma tarefa complicada, por isso fica muito mais fácil proteger a rede isolando estes dispositivos e computadores de uso geral do acesso ao mundo externo, utilizando, para tal, barreiras de proteção.

O principio desta simplicidade tem como consequência a seguinte consideração: para diminuir os riscos, a configuração dos *firewalls* deve ser minimizada, excluindo tudo o que não seja estritamente necessário, este foi

o fundamento utilizado nestes roteadores.

Um *firewall* é definido de acordo com Cheswick (1994), como uma coleção de componentes, colocada entre duas redes que coletivamente possui as seguintes propriedades:

**a)** Todo o tráfego de dentro para fora da rede, e vice-versa, passa pelo *firewall*;

**b)** Só o tráfego autorizado pela política de segurança pode atravessar o *firewall*;

**c)** O firewall deve ser a prova de violações.

Um *firewall* pode ser visto como um monitor de referências para uma rede, sendo seu objetivo garantir a integridade dos recursos ligados a ela. A centralização demanda uma administração mais cuidadosa, por parte dos administradores do sistema, das máquinas que implementam o *firewall*.

Enquanto as máquinas de uso geral são configuradas para aperfeiçoar o desempenho e a facilidade de utilização. No *firewall* tudo isso passa para o segundo plano, cedendo lugar ao seu objetivo principal no sistema: a segurança.

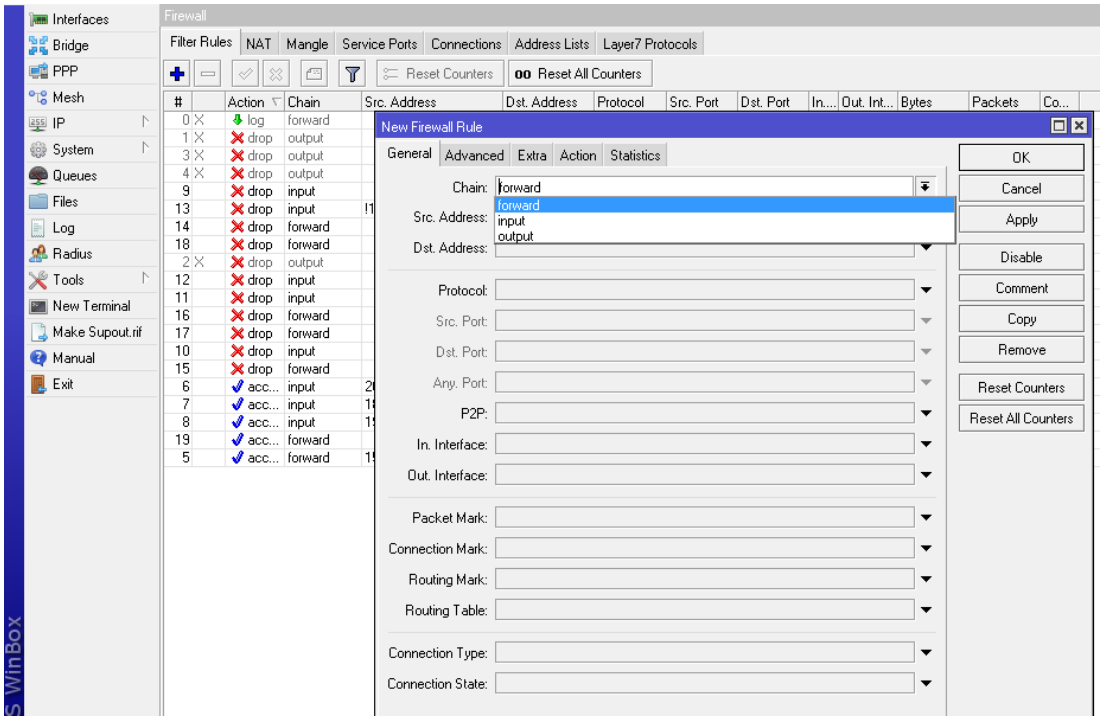


*Figura 4 – Firewall no Mikrotik*

### 2.3.1 REGRAS, CADEIAS E AÇÕES

O *firewall* é fundamentado por meio de regras, uma regra é uma expressão lógica que diz ao roteador o que fazer com um determinado tipo particular de pacote, o *firewall* do sistema operacional *Mikrotik* é semelhante ao *firewall* aplicado em sistemas operacionais *Linux*, chamado *iptables*.

As regras são organizadas em cadeias e existem as seguintes cadeias pré-definidas:

- **Forward:** Responsável pelo tráfego que passa pelo roteador
- **Input**: Responsável pelo tráfego que vai para o roteador
- **Output:** Responsável pelo tráfego que sai do roteador

## 2.4 NAT

O NAT é um mecanismo que visa economizar endereços IP públicos e simplificar as tarefas de gerenciamento do endereçamento IP. Quando um pacote é roteado através de um dispositivo de rede, geralmente um *firewall* ou um roteador de borda, o endereço IP interno (privado) é traduzido para um endereço IP externo (público). Isso permite que o pacote seja transportado por redes públicas como a Internet. Em seguida, o endereço IP externo de resposta é retraduzido para o endereço IP interno que originou o pacote, para ser entregue dentro da rede interna.

Os endereços IPs internos e externos citados acima são definidos por algumas nomenclaturas, aqui se utilizará a mesma nomenclatura usada pela Cisco:

a) Endereço local interno: Endereço IP atribuído a um *host* da rede interna. Definido pelo administrador da rede local e provavelmente um

dos endereços privados especificados na RFC 1918;

b) Endereço global interno: Endereço IP público atribuído pelo provedor de serviço. Este endereço pode representar um ou mais endereços IP locais internos para o mundo exterior, endereço local externo: Endereço IP de um *host* externo, tal como é conhecido pelos hosts da rede interna;

c) Endereço global externo: Endereço IP atribuído a um host da rede externa. O proprietário do *host* atribui esse endereço.

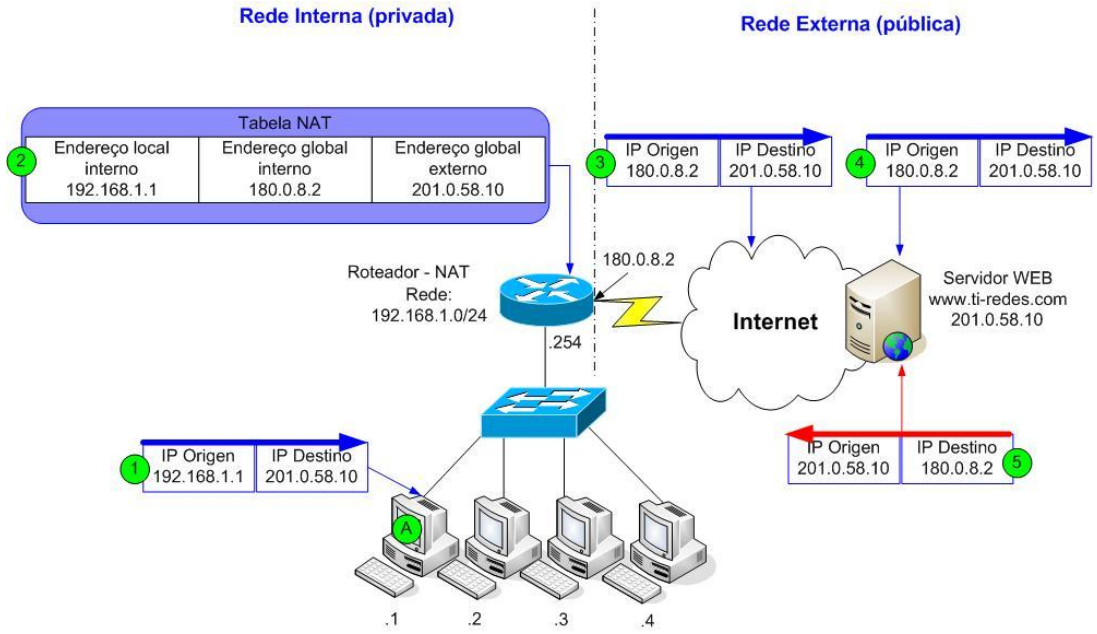


*Figura 5 – Exemplo de NAT*

Analisando o fluxo de dados da figura 5, pode-se perceber que o pacote no momento "1" sai da estação de trabalho “A” com o endereço local interno “192.168.1.1” e em seguida no momento "2" é traduzido para o endereço IP global interno “180.0.8.2”, isto permite que o pacote que saiu da estação de trabalho “A” seja roteado através de uma rede pública chegando ao seu destino, o servidor WEB com o endereço IP global externo “201.0.58.10”. Quando o pacote retorna do servidor WEB, o mesmo terá um endereço IP de destino “180.0.8.2”, que será retraduzido pelo roteador para a rede interna com o endereço IP local interno “192.168.1.1”. O NAT permitiu que o pacote que foi originado na rede interna retornasse corretamente, pois

o endereço de origem “180.0.8.2” atribuído pelo processo de NAT no momento "3" é conhecido na rede pública, possibilitando o seu retorno.

## 2.4.1 SOURCE NAT

É um tipo de NAT, também conhecido como NAT de origem, usado pelo roteador para reescrever um IP de origem e/ou a porta por um outro IP de destino. Também chamado de *Srcnat*, processa o tráfego mandado a partir do roteador através do roteador, depois que ele sai de output e *forward*. Exemplo utilizado em um roteador *Mikrotik* para uma conexão com cliente que é utilizada em nossa rede para suporte há um cliente:

chain=srcnat action=src-nat to-addresses=10.218.86.66
dst-address=10.116.1.0/24 out-interface=DMZ

Neste exemplo todo o tráfego da rede 10.116.1.0/24 sairá da rede local com o IP 10.218.86.66, o qual é o endereço que o cliente libera no roteador para acesso a rede do cliente. Neste caso não está sendo considerada a tradução deste IP para o IP público de acesso externo, considere o exemplo como a tradução de um roteador para outro dentro da mesma rede.

Outra opção do source Nat é “mascarar” uma determinada rede, assim qualquer pacote que sair desta rede sairá com um determinado IP, como no exemplo a baixo:

chain=srcnat action=masquerade out-interface=VPNcliente1

Neste exemplo já existe uma conexão VPN de um cliente configurada no roteador, VPN com o nome VPNcliente o qual é uma interface. No exemplo, todo o tráfego direcionado para esta interface sairá com o mesmo IP, que foi recebido ao estabelecer a VPN.

## 2.4.2 DESTINATION NAT

É um tipo de NAT, também conhecido como NAT de destino, é usado quando o roteador reescreve o endereço e/ou a porta de destino. Também chamado de *Dstnat*, processa o tráfego mandado para o roteador através do roteador, antes que ele seja dividido em input ou *forward*.

Exemplo utilizado na rede estudada:

chain=dstnat action=dst-nat to-addresses=192.168.23.70 to-ports=21 protocol=tcp in-interface=pppoe-Ft dst-port=21

Neste exemplo todo o tráfego que for utilizado na porta 21 será direcionado para o IP 192.168.23.70, empresa, a qual precisou redirecionar para um determinado link quando da necessidade de uso do FTP.

## 2.5 VPN

A ideia de utilizar uma rede pública como a Internet em vez de linhas privativas para implementar redes corporativas é denominada de Virtual Private Network (VPN) ou Rede Privada Virtual.

Virtual Private Network, as VPNs são túneis de criptografia entre pontos autorizados, criados através da Internet ou outras redes públicas e/ou privadas para transferência de informações, de modo seguro, entre redes corporativas ou usuários remotos.

A segurança é a primeira e mais importante função da VPN. Uma vez que dados privados serão transmitidos pela Internet, que é um meio de transmissão inseguro, eles devem ser protegidos de forma a não permitir que sejam modificados ou interceptados.

Outro serviço oferecido pelas VPNs é a conexão entre corporações (*Extranets*) através da Internet, além de possibilitar conexões *dial-up* criptografadas que podem ser muito úteis para usuários móveis ou remotos, bem como filiais distantes de uma empresa.

Uma das grandes vantagens decorrentes do uso das VPNs é a

redução de custos com comunicações corporativas, pois elimina a necessidade de links dedicados de longa distância que podem ser substituídos pela Internet. As LANs podem, através de links dedicados ou discados, conectar-se a algum provedor de acesso local e interligar-se a outras LANs, possibilitando o fluxo de dados através da Internet. Esta solução pode ser bastante interessante sob o ponto de vista econômico, sobretudo nos casos em que enlaces internacionais ou nacionais de longa distância estão envolvidos. Outro fator que simplifica a operacionalização da WAN é que a conexão LAN-Internet-LAN fica parcialmente a cargo dos provedores de acesso.

As redes virtuais privadas baseiam-se na tecnologia de tunelamento cuja existência é anterior às VPNs. Ele pode ser definido como processo de encapsular um protocolo dentro de outro. O uso do tunelamento nas VPNs incorpora um novo componente a esta técnica: antes de encapsular o pacote que será transportado, este é criptografado de forma a ficar ilegível caso seja interceptado durante o seu transporte. O pacote criptografado e encapsulado viaja através da Internet até alcançar seu destino onde é desencapsulado e decriptografado, retornando ao seu formato original. Uma característica importante é que pacotes de um determinado protocolo podem ser encapsulados em pacotes de protocolos diferentes. Por exemplo, pacotes de protocolo IPX podem ser encapsulados e transportados dentro de pacotes TCP/IP.

O protocolo de tunelamento encapsula o pacote com um cabeçalho adicional que contém informações de roteamento que permitem a travessia dos pacotes ao longo da rede intermediária. Os pacotes encapsulados são roteados entre as extremidades do túnel na rede intermediária. Túnel é a denominação do caminho lógico percorrido pelo pacote ao longo da rede intermediária. Após alcançar o seu destino na rede intermediária, o pacote é desencapsulado e encaminhado ao seu destino final. A rede intermediária por, onde o pacote trafegará, pode ser qualquer rede pública ou privada.

As VPNs podem se constituir numa alternativa segura para transmissão de dados através de redes públicas ou privadas, uma vez que já oferecem recursos de autenticação e criptografia com níveis variados de segurança, possibilitando eliminar os links dedicados de longa distância, de alto custo, na conexão de WANs.

Entretanto, em aplicações onde o tempo de transmissão é crítico, o uso de VPNs através de redes externas ainda deve ser analisado com muito cuidado, pois podem ocorrer problemas de desempenho e atrasos na transmissão sobre os quais a organização não tem nenhum tipo de gerência ou controle, comprometendo a qualidade desejada nos serviços corporativos.

## 2.6 GATEWAY

*Gateways* são componentes indispensáveis para alcançar as comunicações entre terminais ligados a redes heterogêneas que usam protocolos diferentes. São equipamentos que podem ser um computador com duas (ou mais) placas de rede, ou um dispositivo dedicado, cujo objetivo é permitir a comunicação entre duas redes com arquiteturas diferentes, como também compartilhar uma conexão com a Internet entre várias estações. Esse equipamento permite traduzir os endereços e os formatos de mensagens presentes em redes diferentes.

Um *gateway* de rede pode ser completamente implementado em *software*, totalmente em *hardware*, ou como uma combinação de ambos. Atua em todas as camadas do modelo OSI e está associado a roteadores, *switches*, *firewalls* e servidores *proxy*. Um roteador usa cabeçalhos e tabelas de encaminhamento para resolver o destino onde dados ou pacotes devem ser enviados e fornece caminho por meio do qual as informações podem ser enviadas dentro e fora do *gateway*.

Os *gateways* são usualmente classificados em dois tipos: *gateways* conversores de meio (*media-conversion gateway*) e *gateways* tradutores de protocolos (*protocol-translation gateway*).

Os *gateways* conversores de meio são os mais simples. Bastante utilizados em inter-redes que oferecem o serviço de datagrama, suas funções resumem-se em receber um pacote do nível inferior, tratar o cabeçalho inter-redes do pacote, descobrindo o roteamento necessário, construir novo pacote com novo cabeçalho inter-redes, se necessário, e enviar esse novo pacote ao próximo destino, segundo o protocolo da rede local em que este se encontra, este *gateway* também pode ser chamado de

roteador.

Os *gateways* tradutores de protocolos são mais utilizados em inter-redes que utilizam circuitos virtuais passo a passo. Eles atuam traduzindo mensagem de uma rede, em mensagens da outra rede, com a mesma semântica de protocolo. Nem todos os protocolos podem ser mapeados entre si, e o subconjunto formado pela intersecção dos serviços comuns é o serviço que deverá ser oferecido como base para a interligação. As dificuldades na tradução dos protocolos tornam bastante complexas e de difícil realização os *gateways* tradutores de protocolos, o que pode aumentar em muito o custo da interligação. Esse tipo de *gateway* pode atuar em qualquer nível acima do enlace, de acordo com o modelo OSI.

Quando os *gateways* interligam duas redes cuja administração pertence a duas organizações diferentes, possivelmente em países diferentes, a operação do *gateway* pode causar sérios problemas. Como a estrutura de ligação em cada uma das redes é completamente independente, para facilitar a implementação e a operação, é comum separar essas entidades também fisicamente, a cada uma dessas interfaces denominamos *half-gateway.*

Cada uma das metades lembra a estrutura requerida por uma estação qualquer da inter-rede. As metades se comunicam através de um sistema de comunicação mais adequado a velocidade de comunicação e a distancia entre as redes. Nesse caso cabe a cada *half-gateway* a realização do protocolo de comunicação entre eles. Ao dividirmos o *gateway* estamos tornando seu projeto mais simples e estruturado, além de contarmos com maior flexibilidade quanto a distancia física das redes.

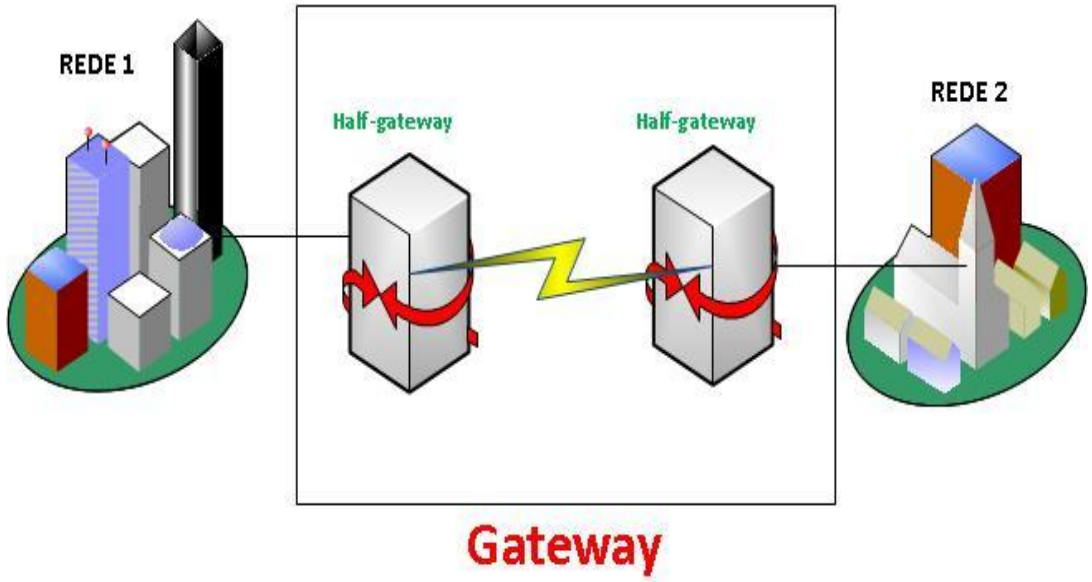


*Figura 6 – Half-gateway*

## 2.7 DNS – DOMAIN NAME SYSTEM

O DNS é um esquema de gerenciamento de nomes, hierárquico e distribuído. O DNS define a sintaxe dos nomes usados na internet, regras para delegação de autoridade na definição de nomes. Um banco de dados distribuído que associa nomes a atributos (entre eles o endereço IP) e um algoritmo distribuído para mapear nomes em endereços.

Foi criado para auxiliar os seres humanos a armazenar os nomes de domínios relacionados com seus respectivos endereços IP, criado pelo NIC (Network Information Center), que cotinha uma base de dados contendo todos os nomes de domínio da internet, assim ao buscarmos um determinado site não digitamos seu IP e sim seu nome de rede. Inicialmente o "*resolvedor"* de nomes será consultado, este buscará em sua base de dados o IP associado e reencaminhará a mensagem ao computador solicitante, que por sua vez, o armazenará no seu *cache* – armazenamento local.

Desta forma surgiu o DNS, cujas principais características, de acordo com Marques (2000) são:

a) Associar a cada IP um nome de domínio;
b) Banco de dados distribuídos: contendo, basicamente, informações a respeito de estações que pertencem a sua rede, bem como o

endereço de outros DNS;

c) Hierarquia de nomes de profundidade arbitraria de acordo com a Figura 7:

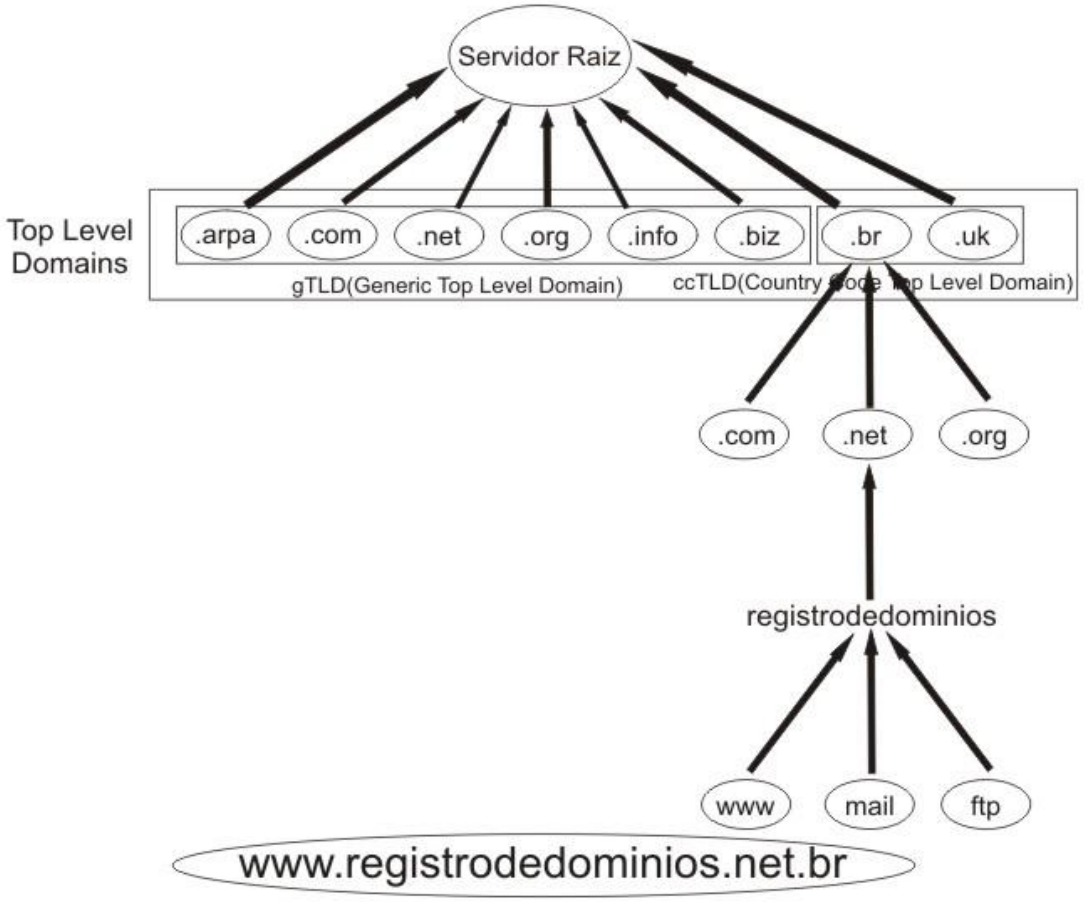


*Figura 7 – Estrutura de um DNS*

d) A distribuição da informação é controlada pela própria base de dados;

e) O serviço de DNS é baseado em UDP – *User Datagram Procotol*, e eventualmente em TCP;

f) A estrutura de DNS de acordo com a Figura 7 é dividida em níveis.

Não é necessário existir apenas um servidor de nomes BR, por exemplo, normalmente existem no mínimo dois, é importante salientar que todos devem ter em sua base de dados o endereço dos demais servidores, formando assim uma hierarquia de buscas.

A finalidade principal e primordial do DNS para grandes redes é a

tradução de nomes, o que torna fácil a compreensão por parte dos seres humanos, senão teríamos que usar números, endereços IP para acessarmos um determinado site por exemplo.

Um conjunto de servidores de nomes mantém o banco de dados com os nomes e endereços das máquinas conectadas na internet. Na realidade este é apenas um tipo de informação armazenada no DNS. É usado um conjunto de servidores interconectados, ao invés de um único servidor centralizado. Existem atualmente tantas instituições conectadas na internet que seria impraticável exigir que elas notificassem uma autoridade central toda vez que uma máquina fosse instalada ou trocasse de lugar. Assim, a autoridade para atribuição de nomes é delegada a instituições individuais.

## 2.8 INTERNET PROTOCOL

Protocolo de internet, responsável pelo endereçamento e roteamento da mensagem e quando necessário, pela fragmentação em datagramas. A fragmentação de uma mensagem é a responsabilidade das camadas de transporte ou TCP e de internet ou camada de IP.

Segundo a RFC 791, o *Internet Protocol* foi projetado para permitir a interconexão de redes de computadores que utilizam a tecnologia de comutação de pacotes, ou seja, qualquer rede, seja ela de pequeno ou grande porte, somente interna ou pública, todas elas são baseadas em IP.

O ambiente inter-rede consiste em *hosts* (qualquer dispositivo ou computador ligado à rede) conectados a redes que por sua vez são interligadas através de *gateways.*

O protocolo IP é um protocolo sem conexões, sua função é transferir blocos de dados datagramas da origem para o destino, onde a origem e o destino são hosts identificados por endereços IP. O protocolo IP também fornece o serviço de fragmentação e remontagem de datagramas longos, quando necessário, para que eles possam ser transmitidos através de redes onde o tamanho máximo permitido para os pacotes é pequeno

O serviço oferecido pelo IP é sem conexão. Portanto, cada datagrama IP é tratado como uma unidade independente que não possui nenhuma relação com qualquer outro datagrama. A comunicação é não confiável, não sendo usados reconhecimentos fim a fim ou entre nós intermediários. Nenhum mecanismo de controle de erros nos dados transmitidos é utilizado,

exceto um *checksum* do cabeçalho que garante que as informações nele contidas, que são usadas pelos *gateways* pra encaminhar os datagramas, estão corretas. Nenhum mecanismo de controle de fluxo é empregado.

Algumas das principais características desse protocolo são:

a) Serviço de datagrama não confiável;
b) Endereçamento hierárquico;
c) Facilidade de fragmentação e remontagem de pacotes;
d) Identificação da importância do datagrama e do nível de confiabilidade exigido;
e) Identificação da urgência de entrega e da ocorrência futura ou não de pacotes na mesma direção (pré-alocação, controle de congestionamento);
f) Campo especial indicando qual o protocolo de transporte a ser utilizado no nível superior;
g) Roteamento adaptativo distribuído nos *gateways*;
h) Descarte e controle de tempo de vida dos pacotes inter-redes no *gateway*.

# 3 PROPOSTA

A partir de uma análise realizada na estrutura da rede, foi observado que vários problemas relacionados com a conectividade estavam impactando nos serviços da empresa. A partir destas constatações, passou-se a analisar a estrutura da rede e seus componentes, em principal os roteadores, os quais são a base para toda a rede de comunicação.

Antes da realização deste trabalho todo o processo de troca de links e manutenção das publicações e serviços da empresa eram feitos de forma manual, descritos abaixo:

a) Troca manual do *gateway* default do roteador MK254 da Figura 8, dependendo de qual dos links estivesse com queda;
b) Troca do *gateway* default do roteador MK09 (Figura 8) caso o link que estivesse com queda era o link das publicações (como site da empresa, serviço de e-mail ou outros sistemas externos dependentes

deste link), Link Y de acordo com a Figura 8;

c) Caso a queda fosse Link Y, era necessário acessar os servidores que possuíam hospedados as publicações e alterar determinados parâmetros para que fosse necessário reestabelecer os serviços;
d) Necessário acessar o servidor de DNS da empresa e trocar o nome das publicações e dos serviços de VPN para outro link.

Estes processos manuais citados acima foram os principais pontos críticos que fizeram com que a realização desse trabalho fosse de extrema importância para situação atual da rede da empresa, além destes processos, eram realizados outros que exigiam muitas vezes a presença de duas pessoas, o que totalizava uma demora de minutos para realização de toda a troca de link e restabelecimento dos principais serviços, outros serviços secundários que não eram priorizados nos momentos de queda ficavam prejudicados, pois não era viável nem possível trocar toda a estrutura, o que prejudicava também o suporte aos clientes da empresa.

Este processo, todo realizado de forma manual, elevava as chances de erros. Para isso foi proposto à criação de uma forma automatizada para fazer tal processo, com o uso de scripts da linguagem dos sistemas operacionais dos roteadores *Mikrotik*, utilizando e mantendo as mesmas estruturas, sem que fossem necessárias mudanças na rede, nem a compra de novos equipamentos, objetivando poupar recursos e explorar as ferramentas dos roteadores já empregados na empresa, tirando o máximo proveito da tecnologia sem desperdício de recursos.

A proposta foca na resolução de um problema recorrente na empresa: a demora na troca dos links de comunicação quando há quedas. Está demora causa alta indisponibilidade nos serviços da empresa, como suporte a clientes, aplicações externas e publicações que dependem da disponibilidade da internet. Nas próximas sessões será explicada de forma detalhada cada parte deste processo de desenvolvimento dos scripts, assim como a explicação da lógica e dos principais recursos utilizados neles.

## 3.1 ESTRUTURAS DA REDE

A estrutura da rede é baseada em quatro roteadores de acordo com a

Figura 8, sendo três deles alimentados com scripts de failover para este trabalho. Cada um dos três roteadores possui um link de comunicação diferente, com o objetivo de criar alta disponibilidade e separar os serviços da empresa da seguinte forma:

a) MK254, *gateway* da rede local, toda rede local da empresa passa por primeiro neste roteador, este roteador por ser apenas local e não ter link com internet não possui script de *failover*;
b) MK05, roteador que possui o link principal para navegação na internet, e possui um serviço de VPN, para que os usuários da empresa possam trabalhar remotamente;
c) MK06, roteador que possui a grande maioria das VPNs de acesso aos clientes, usados para prestar suporte aos clientes;
d) MK09, roteador que possui todas as publicações externas da empresa, como site, e-mail, e alguns sistemas que precisam ser acessados publicamente, este roteador é o principal roteador da empresa por tratar da grande maioria dos acessos externos há empresa.

Os roteadores são virtualizados, ou seja, são estruturas virtuais hospedadas em plataformas de alta disponibilidade, utilizando tecnologias de alto desempenho, fornecendo assim ampla flexibilidade no manuseio, maior segurança e obviamente maior disponibilidade.

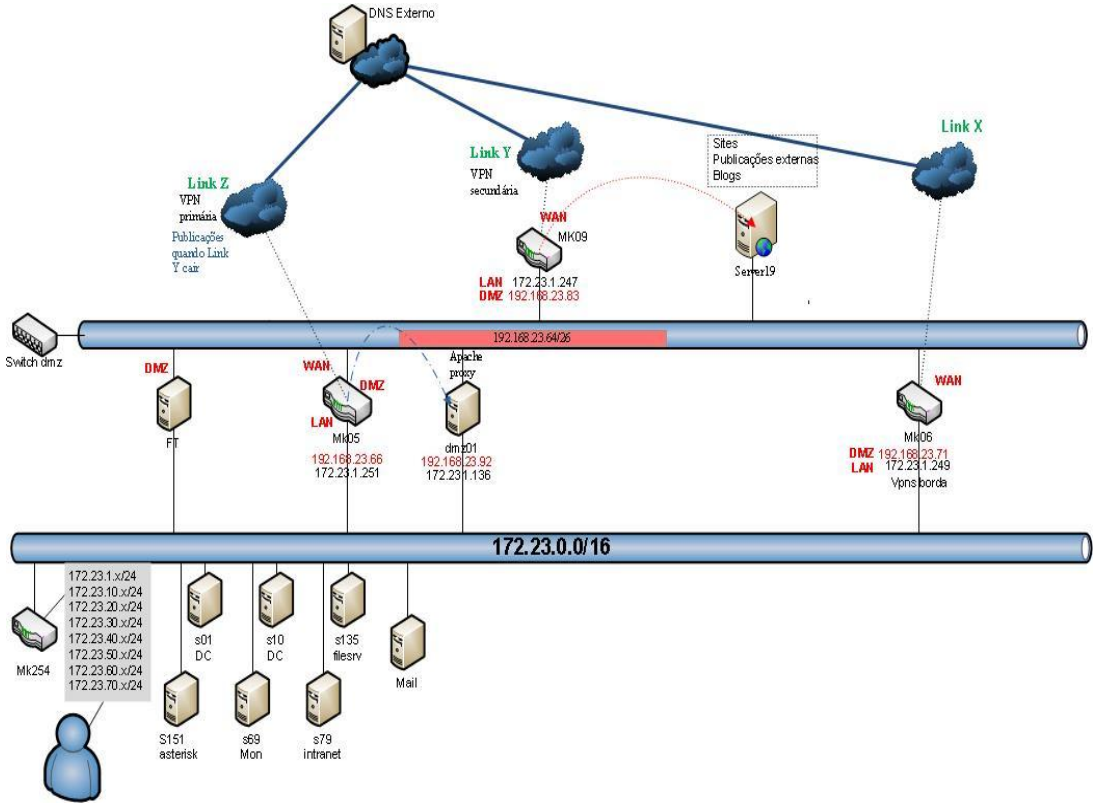


*Figura 8 – Mapa da rede*

## 3.2 OBJETIVOS DOS SCRIPTS

A escolha na utilização dos scripts se deu por causa da maior possibilidade de realização das tarefas, em especial as tarefas relacionadas aos testes de conectividade, estes que precisavam ter uma flexibilidade maior, pelo fato de ser necessário testar as diversas formas de monitoração, alterar parâmetros de TTL, *delay* e *ping*, parâmetros estes que o roteador não fornecia na forma como precisava.

Com o uso de scripts, é possível criar um leque de possibilidades maior, pelo fato da linguagem usada nos scripts ser a própria linguagem do sistema operacional do roteador. A estrutura dos scripts não é trivial, por se tratar de uma linguagem própria, foi necessário um estudo de toda sua estrutura, apesar da semelhança com sistemas operacionais *Unix*, a forma de utiliza-los difere em vários pontos. A seguir será mostrado as principais estruturas usadas na implementação:

**a) Instruções condicionais:**

```
do..while: do { <comandos> } while=( <condições> ); :while ( <condições> )
do={ <comandos> };
for :for <var> from=<int> to=<int> step=<int> do={ <comandos> }
foreach :foreach <var> in=<array> do={ <comandos> };
```

**b) Declaração condicional:**

```
if    :if(<condição>) do={<comando>} else={<comando>} <expressão>
```

O exemplo abaixo demonstra estas duas principais estruturas utilizadas no trabalho, a parte do script mostrado a seguir tem objetivo de tratar a adição de uma rota. Quando uma rota é adicionada é necessário que a sua máscara de rede seja adicionada junto, esta parte do script tem como função principal verificar se há “/” caso tenha, desconsiderar, caso não tenha, seta, a rota com “/32”, que significa ser apenas um IP não um range de IPs:

```
:foreach b in=$rotas do={
        :if ([find $b "/"]>=0) do={nothing} else={:set b "$b/32"};
                :if ([/ip route find dst-address="$b" !routing-mark]="") do={
                                        /ip route add comment="$a add by
LinkFailover script" disabled=no distance=1 dst-address="$b"
gateway="$gwpingok" scope=30 target-scope=10;
        :put "rota adicionada=$b descricao=$a"
                                    } else={
                                        /ip route set [find dst-
address="$b" !routing-mark] gateway=$gwpingok;
                                    }
                                }
```

Todo o trabalho se baseia na lógica utilizada nos scripts, todo o funcionamento depende dos seguintes componentes:

a) Um servidor externo confiável para monitoramento, chamado de DNS externo de acordo com a Figura 6;
b) Fatores internos que são os roteadores e toda a rede local da empresa;
c) Um fator externo chamado de servidor alvo de acordo com a Figura 9, que serve como alvo de monitoramento no processo de teste de conectividade.

A principal funcionalidade dos scripts é a mudança de gateway quanto há quedas, ou seja, é o objetivo principal do trabalho. Cada roteador tem um gateway principal, representado nos scripts com o seguinte parâmetro:

```
:if ($a="defaultgw") do={
        :set gw1 189.8.199.254; :set rotas {0.0.0.0/0;};
};
```

Este parâmetro simplesmente seta o *gateway* default do roteador para a saída do seu link principal, mostrado na Figura 9 abaixo, na forma gráfica do roteador.

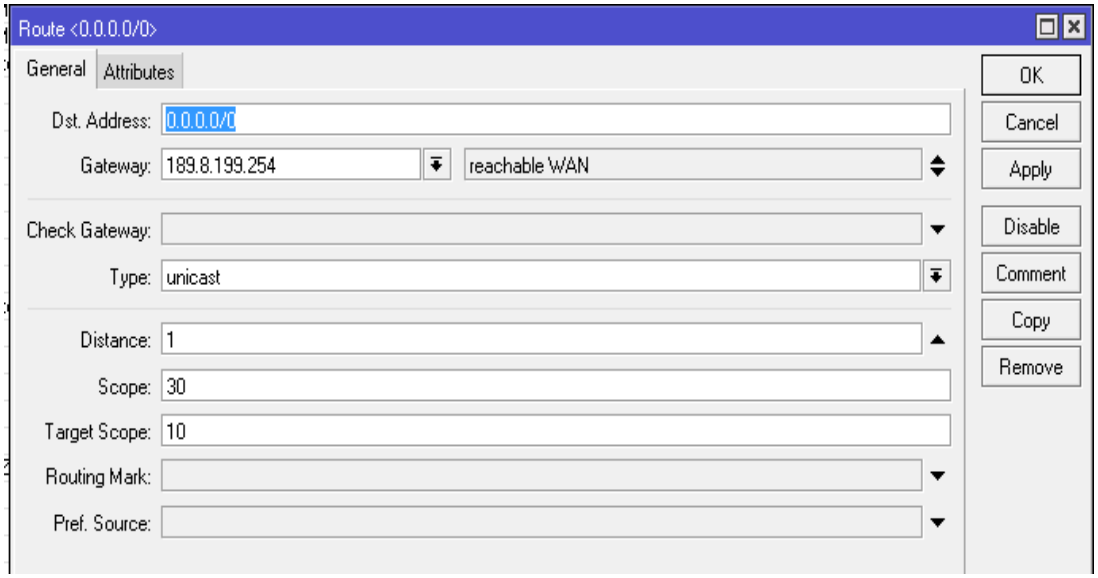


*Figura 9 – Gateway default*

Como mostrado na Figura 9, à rota default do roteador direciona todo o tráfego deste roteador para um determinado link de internet com uma banda contratada, e um conjunto de links secundário com os outros *gateways* (roteadores) para serem usados quando houver quedas, de acordo com o trecho de código abaixo, a qual apenas está definindo a ordem de troca de links, caso o link principal caia, importante salientar que cada roteador considera o seu link como sendo o principal e restante como secundários:

```
# Todos os gateways da rede definindo ordem de preferência
:local gws
{192.168.23.66;192.168.23.83;189.8.199.254;192.168.23.65;192.168.23.67;1
92.168.23.72;};
```

Note que repito o *gateway* default na terceira vez, isto é feito propositalmente para verificar se a conectividade com o link principal deste roteador já está normalizado, caso os dois primeiros *gateways* falhem, as quais são as rotas para os outros dois roteadores preferenciais, os demais *gateways* são usados apenas em caso extremo de queda simultânea dos três links principais.

Todos os roteadores possuem rotas entre eles semelhante a uma sub-rede local, cada roteador tem um link de conexão diferente provendo um serviço diferente ou como redundância de outro, com diferentes operadoras, ou seja, cada um dos roteadores possui uma operadora com contrato diferente um do outro, total de três roteadores cada um com um script com funcionamento semelhante, possuindo a mesma lógica de verificação.

O funcionamento do script tem como ferramenta central a verificação da conectividade, a qual é o principal componente no processo de automatização, toda a funcionalidade e lógica está baseada no constate monitoramento das conexões em cada roteador, a sua explicação será apresentada na sessão a seguir.

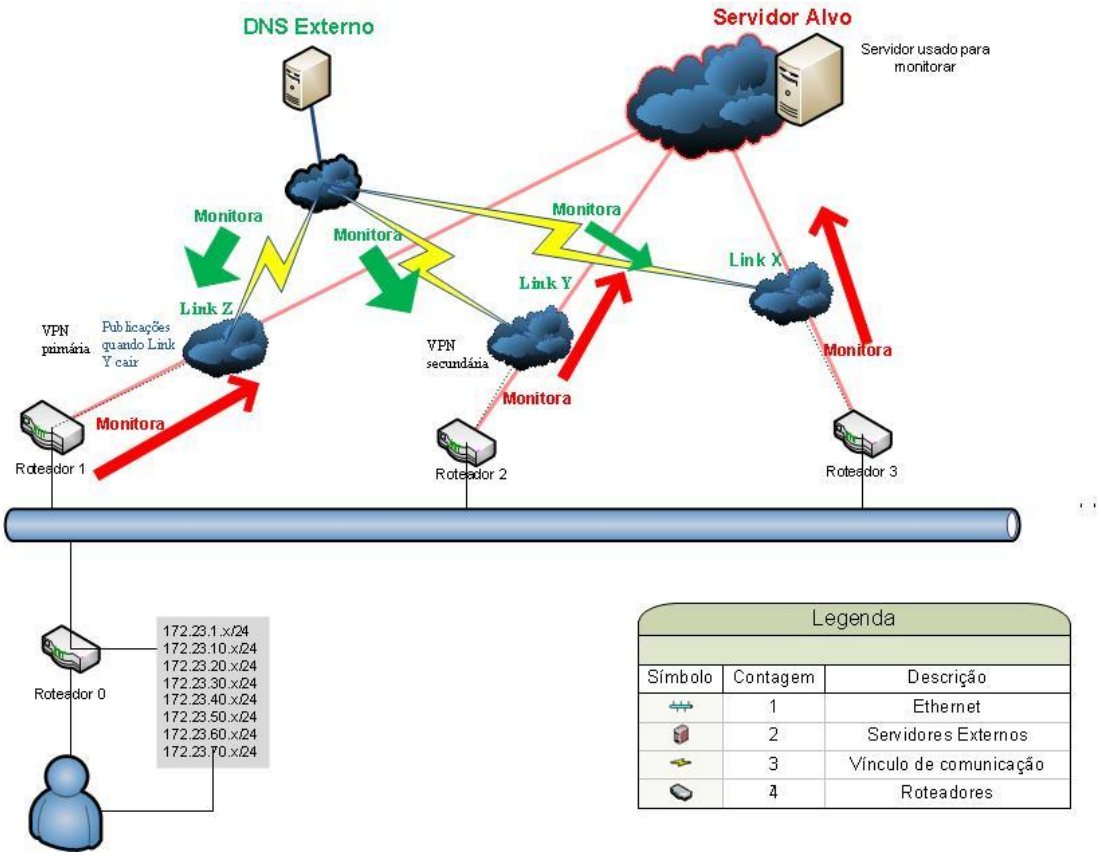


*Figura 10 – Mapa do script*

## 3.2.1 VERIFICAÇÃO DE CONECTIVIDADE

Se tivesse que escolher uma palavra que descrevesse o objetivo dos scripts seria monitoramento, toda a estrutura dos scripts foi pensada objetivando a melhor forma de monitorar os links de cada roteador, por este motivo esta parte do trabalho foi a que mais demandou tempo e testes. Todos os testes precisavam ser feitos constantemente e alguns testes duravam dias, só assim poderia ter a certeza, ou quase certeza de que os parâmetros utilizados eram os mais corretos possíveis, causando o mínimo de falsos positivos.

Para que os scripts pudessem ser executados de forma confiável, era necessário que eles se baseassem em um servidor externo na internet (servidor alvo, Figura 10) que estivesse sempre disponível. Para isso foi necessário um período de pesquisas e testes para encontrar qual o servidor era ideal ou o mais confiável possível.

Esta parte do processo empregado no script, não possui total eficácia, pelo seguinte motivo, não é possível garantir que o servidor alvo não fique

indisponível, pois ele depende de outros fatores que ficam de fora do escopo do projeto, para tentar garantir o máximo de eficiência possível é necessário constante monitoramento. E qualquer eventual queda neste servidor utilizado implicará em um falso positivo nos roteadores, o que acarretará na ativação do script e troca do link.

Em um primeiro momento foi utilizado alguns servidores próprios da empresa como servidor alvo, hospedados fora da rede. Mas com o passar dos testes notou-se que não eram os ideais, por sofrerem constantes quedas e serem servidores utilizados para manutenção. Passou-se então a pesquisar alguns servidores mais conhecidos, como servidores de operadoras de telecomunicações, serviços de DNS como a que a Google fornece. Foi aplicado diversos testes de monitoramento e passou-se a utilizar com mais frequência o DNS 8.8.8.8 e 8.8.4.4 fornecido pela *Google*, pois ficou 100% do tempo online. Outros servidores que foram utilizados são os servidores das próprias operadoras de telecomunicações, os quais também se mostraram confiáveis.

Após determinar qual servidor alvo utilizar, iniciou-se o planejamento e realização dos testes com parâmetros de monitoramento. Para monitoramentos iniciais utilizou-se apenas o utilitário ping de acordo com exemplo abaixo:

```
:if ([/ping 8.8.8.8 src-address=172.23.1.249 count=3]=0) do={
/tool e-mail send to=monitor@empresa subject="Queda de Link";
} else {
:log info “Link UP";
}
```

No exemplo acima a verificação é feita de forma básica, é realizado um ping para o IP da *Google*, através do IP do roteador, *“src-address”*, ou seja, endereço fonte é o endereço do roteador, e o servidor alvo é o endereço da *Google*, assim se este ping retornasse 0, que é o equivalente a não receber resposta alguma, um e-mail era enviado informando queda de link. Este procedimento básico possibilitou diversos testes iniciais. Assim

percebeu-se que apenas este parâmetro, *"count=3"* não era necessário, três pings são insuficientes para identificar uma queda, muitas vezes o link pode estar apenas oscilando, ou com intermitência, nestes casos não é necessário trocar toda a estrutura.

O script utiliza o protocolo de redes ICMP com o utilitário ping em um servidor externo confiável (ou o mais confiável possível), assim a cada intervalo de tempo o script pinga através de cada um dos links de conexão para monitorar, ou seja, o script define como rota default do roteador o gateway de um dos links e defini o servidor externo alvo que irá monitorar:

```
…
:if ([/ip route find dst-address="$ipmon/32"]="") do={/ip route add
comment="LinkFailover $a monitored IP" disabled=no distance=1 dst-
address="$ipmon/32" gateway=$gw1 scope=30 target-scope=10;}
…
```

No trecho de script acima, por exemplo, para todos os roteadores, ele procura pela rota do servidor alvo na tabela de roteamento do roteador, caso não encontre, ele adiciona a rota, o IP do servidor alvo está salvo na variável *"ipmon", que* altera o *gateway* default para o *gateway* principal que está na variável *"gw1".* Os outros parâmetros são apenas para a alteração ser aceita pelo roteador, a rota configurada ficará como na figura 11 abaixo.

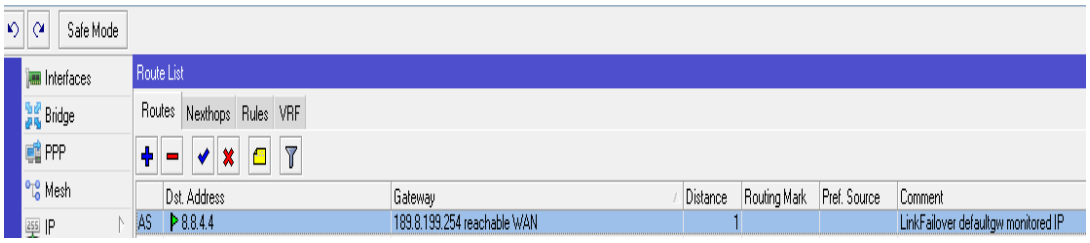


*Figura 11 – Rota default para monitoramento do servidor alvo*

Os parâmetros citados anteriormente definem a estrutura para que seja possível testar a conectividade. Para testar efetivamente não será utilizado somente o ping, como no começo do estudo, se utilizará o ping e

suas variações, como delay, e TTL, para determinar se o link está ativo ou se está com problemas, este mesmo processo é utilizado em todos os roteadores para cada um dos links de comunicação, o teste de conectividade respeita uma ordem de prioridade, que é definida em cada script e varia de roteador para roteador, pois cada roteador possui um link de comunicação diferente, cada um destes links de comunicação é definido como *gateway* default para as verificações de conexão.

Os parâmetros de monitoramento são configurados em variáveis locais no começo de cada script, para que possam ser alterados mais facilmente caso apresente muitos falsos positivos, a seguir pode-se ver o trecho do script que realiza os testes de conectividade:

```
:if ($2gatewaysonly) do={:set gwslist [(:put "$gw1", "$gw2")]} else={:set
gwslist [(:put "$gw1", "$gw2","$gws")];};
        :set continue true; :set counter 0; :set gwpingok "";
:while ($continue) do={
                :if ([:len $gwslist]>$counter) do={
                  :set gwcandidato [:tostr [:pick $gwslist $counter]];
                :put "----- Testando $a gateway[$counter] IP na linha
abaixo"; :put [:pick $gwslist $counter];
                        :if ([/interface find name="$gwcandidato"]!="" and
[/interface find name="$gwcandidato" disabled=no]="") do={
                        :put "Gateway é uma interface  e esta desabilitada ou
não existe";
                                :set counter ($counter+1);
                        } else={
#A lista de gws contem gw1 e gw2 nas 2 primeiras posições porem os outros
gateways podem conter gw1 e gw2 repetidos deve-se pular pois já foram
testados
```

```
                    :while (($counter>1 and "$gwcandidato"=$gw1) or
($counter>1 and "$gwcandidato"=$gw2) or ("$gwcandidato"="")) do={
                    :put "Gateway duplicado ou branco[$counter]"; :put
"$gwcandidato"; :set counter ($counter+1);
                        };
#Altera rota ipmon via gateway da lista array posicao counter para tentar
ping e ver se responde
                /ip route set [find dst-address="$ipmon/32"]
gateway="$gwcandidato";                    :local statusrota [/ip route find
dst-address="$ipmon/32"];
                :if ([/ip route get $statusrota active]=true) do={
                :if ([:pick $a 0 4]="ClienteVPN1") do={:set pingcount 10;}; :set
pingcountOK 0;
                :set pingcountOK [/ping $ipmon count=$pingcount
interval=1]; :if ($pingcountOK>=1) do={:nothing} else={delay $sleep; :set
pingcountOK [/ping $ipmon count=$pingcount interval=1]}
                :if ( $pingcountOK >= 1 ) do={
                :set gwpingok "$gwcandidato";
                :set continue false;
#Remove $a do array $alloff se $a estiver listado dentro dele
                :if ([:len $alloff]>0) do={
                    :local alloffaux {};
                    :foreach e in=$alloff do={
                        :if ($e!=$a) do={set alloffaux ($alloffaux,$e)};};
                  :if ([:len $alloff] != [:len $alloffaux]) do={
                      :put "$mk LinkFailover [$a] NORMALIZADO via
```

```
gateway [$gwpingok] antes todos gateways estavam indisponíveis";
/tool e-mail send to=$mailto subject="$mk LinkFailover $inicio $a
NORMALIZADO Via $gwpingok antes TODOS GWs estavam indisponíveis"
body="Vazio";
                                                }
                                                :set alloff $alloffaux;
                                        }
                                }
                        } else={
                                :put "Gateway unreachable"
                        }
                        :set counter ($counter+1);
#ClienteVPN2 nao aceita ping dentro da vpn, então assume-se o gw default
dele
                        :if ($gwpingok="") do={:if ("$gwcandidato"="
ClienteVPN2") do={:set gwpingok ClienteVPN2; :set continue false; :put
"Gateway [ClienteVPN2] foi assumido";};}
                                }
                } else={:set continue false;};
        }
```

Como observado acima, este procedimento além de testar a conectividade dos links de comunicação, testa a conectividade dos clientes VPN que estão configurados no roteador, a lógica para as verificações das VPNs é a mesma. Como cada VPN estabelecida cria uma interface no roteador, mostrado na Figura 12, basta utilizar como *gateway* de teste um IP utilizado em algum dos acessos dentro da VPN, como a empresa possui servidores hospedados nestes clientes basta monitorar um destes servidores

através da VPN, ou seja, através da interface desta VPN, que funciona como um servidor alvo do cliente VPN.

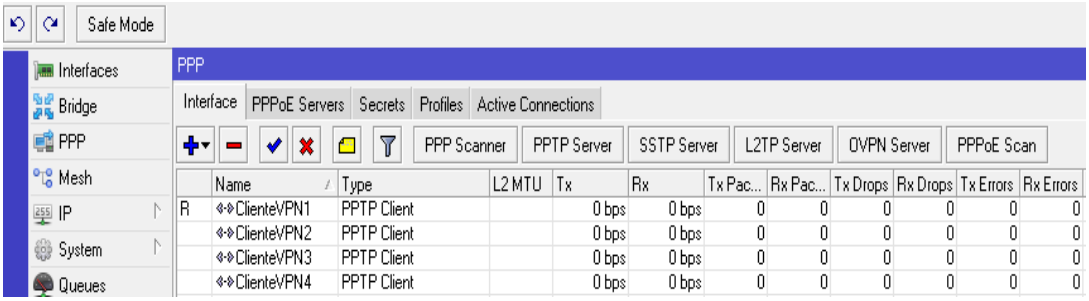


*Figura 12 – Clientes VPN configurados no roteador*

A Figura 12 é uma ilustração de como que um cliente VPN fica estabelecido nos roteadores da *Mikrotik*, o restante do comportamento das VPNs é semelhante a um roteamento normal, como um roteamento default usado aqui para monitorar o link principal. Da mesma maneira que é possível monitorar o link principal, é também possível monitorar as VPNs, está flexibilidade auxilia no processo de monitoramento, pois é usada a mesma lógica em todo o processo, facilitando a perfeição dos testes.

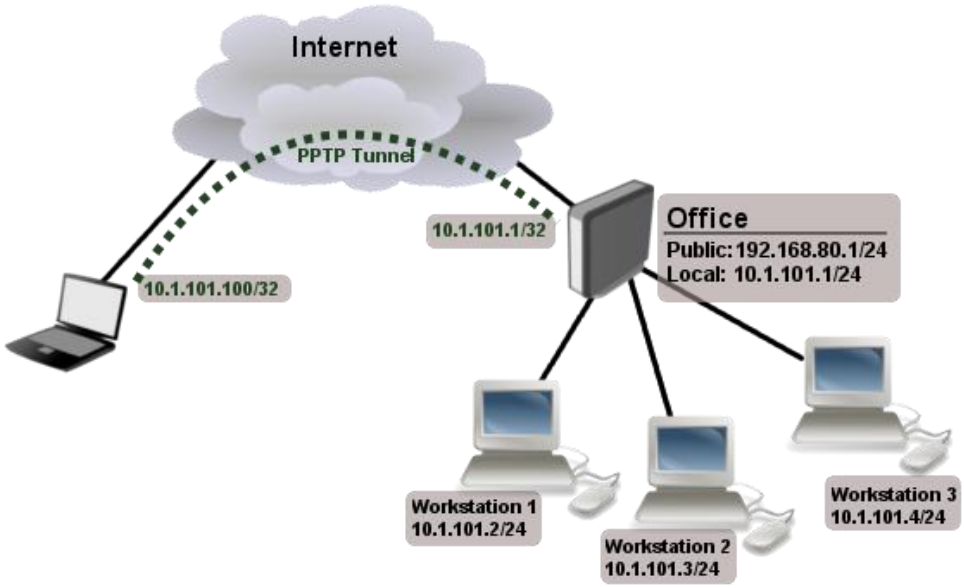


*Figura 13 – VPN pptp cliente*

Cada alteração realizada de troca de link, queda, ou quando o link volta para seu estado padrão, é informado via e-mail para conhecimento de toda a equipe e para fins de comparação com outros monitoramentos, como o script de DNS, por exemplo, que tem um processo de monitoramento semelhante e também notifica via e-mail quando realizada trocas, este script de DNS é um processo adicional ao trabalho e será explicada detalhadamente mais adiante.

## 3.3 ESQUEMATIZAÇÕES DOS TESTES

Para que pudesse testar em um ambiente de homologação, sem prejudicar os serviços da empresa, foi necessária à criação de roteadores de testes para simulação dos scripts, simulando o ambiente da empresa.

Para este processo de testes foi utilizado à tecnologia de virtualização, que fornece a flexibilidade necessária para criar e destruir facilmente um ambiente, no caso, o ambiente de rede semelhante ao real.

No ambiente de testes foram utilizados os mesmos links conectados ao ambiente real, para que pudessem ser testadas de forma efetiva as quedas dos links. Os roteadores utilizados foram todos virtualizados em apenas um servidor físico, poupando assim a aquisição de outros equipamentos. Os roteadores virtuais foram ligados no mesmo barramento dos roteadores reais, para que o caminho percorrido por eles pudesse ser o mesmo dos reais, confirmando assim a efetividade dos testes. Toda a simplicidade da estrutura forneceu toda esta flexibilidade, conforme ilustrado na Figura 14.

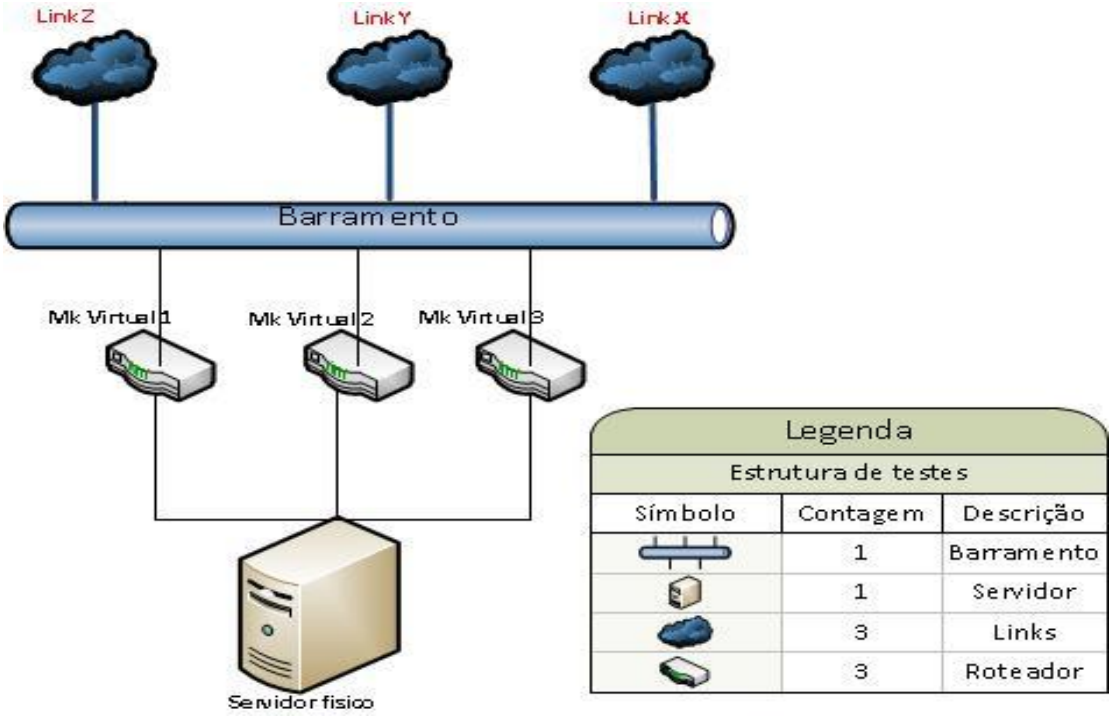


*Figura 14 – Estrutura de testes*

Com a flexibilidade que os roteadores *Mikrotik* fornecem, foi possível simular todo o ambiente de forma virtualizada, utilizando os mesmo links de comunicação, tornando assim os testes efetivos e reais, o que proporcionou mais eficácia na aplicação posterior dos scripts nos roteadores de produção.

O período de testes foi intercalado com a aplicação nos roteadores em produção, e não foi criado nenhum cronograma de testes, apenas foi testado o que era necessário de acordo com o andamento do desenvolvimento e das dúvidas que surgiam. Os testes ocorreram da seguinte maneira:

a) Depois de implantado o script em um roteador teste, foi simulado a queda, bloqueando o endereço de monitoramento no *firewall* (Dst. *Address* na Figura 15), assim o script percebia a queda e trocava os serviços, para que fosse possível verificar os parâmetros, esta mesma regra pode ser alterada para qualquer destino, facilitando os trabalhos;

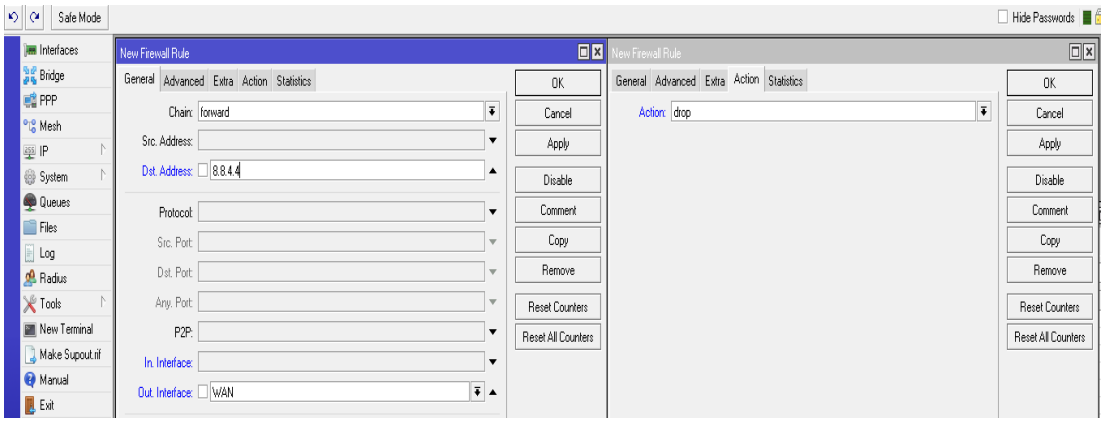


*Figura 15 – Regra de bloqueio*

b) Os serviços testados inicialmente eram apenas a internet, foram trocados apenas o *gateway* default do roteador testado. Foi decidido inicialmente testar apenas os links de acesso a internet, pois o restante dos serviços, como as publicações, eram necessárias outras alterações na estrutura da empresa, o que poderia causar indisponibilidade de alguns serviços em produção;

c) Após os testes iniciais com a troca apenas da internet, passou-se a considerar os serviços hospedados no roteador, o primeiro serviço que foi testado foi o restabelecimento de VPNs, tal teste fez com que fosse percebido que era necessário reiniciar todos os serviços relacionados à VPN quando houvesse troca de links, este impacto, apesar de não ter resolução, impacta nos acessos aos clientes, pois não é possível manter a conexão estabelecida durante a troca, porém o restabelecimento da VPN é rápido e compensa a falha;

d) Os outros serviços como alterações de rotas e publicações também foram aplicados em roteadores de testes, assim como o processo adicional de DNS foi aplicado antes em um servidor de homologação com uma cópia real das bases de dados do DNS.

Cada fase destes testes demorava no mínimo um dia, para que se pudesse ter certeza que quando se aplicasse no ambiente real o número de falhas fosse o mais baixo possível, a demora nos testes também era proposital, pois se estava testando a internet e a internet não pode ser considerada estável, por tal motivo alguns dos testes foram deixamos por semanas até que se tivesse total segurança dos parâmetros, principalmente

ao que se refere ao servidor alvo.

Os três roteadores de testes eram com o mesmo sistema operacional Mikrotik empregado nos rotadores reais, foi utilizado um servidor Linux com sistema operacional CentOS para o serviço de DNS, o qual testava o script de DNS com a copia das bases DNS, outras aplicações que eram necessários testes, foi utilizado um IP extra para as publicações externas, assim não era necessário alterá-las na produção.

## 3.4 PROCESSOS ADICIONAIS

Para que o trabalho ficasse completo e para que pudesse cumprir de forma satisfatória os seus objetivos, foram necessários alguns processos adicionais à solução inicial, estes processos adicionais foram também implementados em um ambiente já existente, sem a necessidade de aquisição extra, todo o processo adicional aqui referido foi criado em um servidor hospedado fora da empresa, o qual é um servidor que hospeda o domínio da empresa, ou seja, possui toda a estrutura de DNS da empresa.

Como processo adicional, foi criado script de DNS (vide anexo A) em um servidor hospedado fora da rede da empresa, com o objetivo principal de monitorar os links de comunicação da empresa no sentido internet-empresa, e quando ocorrer à queda de algum dos links possa trocar a chamada de DNS das publicações, de forma automatizada, para que não dependêssemos mais de apenas um IP amarrado a um link e sim uma lista de IPs e consequentemente uma lista de links.

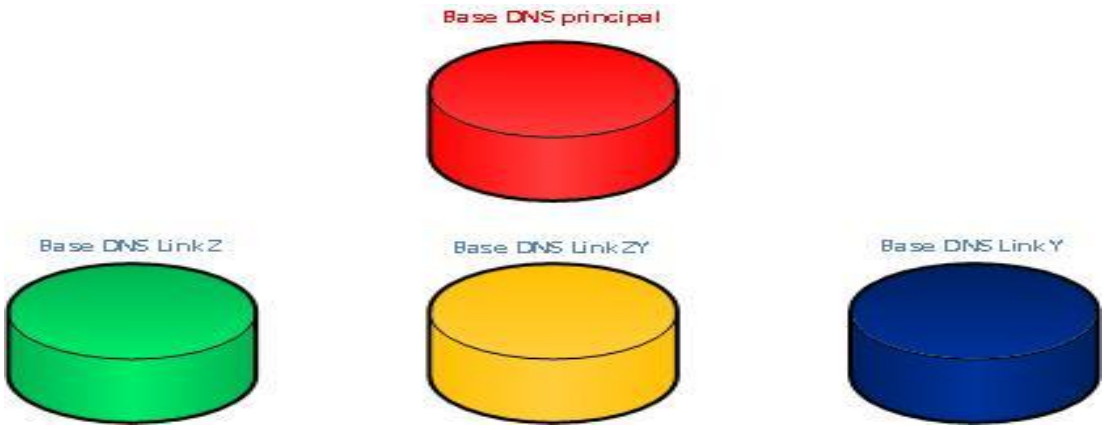


*Figura 16 – Bases DNS para o script*

O script possui um objetivo claro e simples, monitorar a cada determinado intervalo de tempo os links de comunicação da empresa (este intervalo de tempo é relativo, pode ser trocado quando forem detectados falsos positivos ou se acharmos que o tempo de monitoração estiver errado), para que quando haja alguma queda não apenas os serviços internos da empresa continuem com acesso, como também as publicações externas e demais serviços que dependam de um link de comunicação especifico, ou seja, que possuem um endereço externo fixo.

```
pinga() {
        pingcount=3; sleeptime=120; waitsegs=10;
        if $simula; then
                echo Pingando Link1 e Link2 [Aguarda 2x 10segundos pelo
ping e mais 1.10mins se detectar erro - se ambos links down leva 2.3mins
para executar]
        fi
#usando ping para testar
        ip=$l1ip; ping -c $pingcount -w $waitsegs $ip >/dev/null 2>&1; if [ $?
== 0 ]; then status=up; else sleep $sleeptime; ping -c $pingcount -w
$waitsegs $ip >/dev/null 2>&1 && status=up || status=down; fi; l1=$status
#usando nc para testar
        ip=$l1ip; /usr/bin/nc -w 45 -z $ip 8291 >/dev/null 2>&1; if [ $? == 0 ];
then status=up; else sleep $sleeptime; /usr/bin/nc -w 45 -z $ip
8291 >/dev/null 2>&1 && status=up || status=down; fi; l1=$status;
#compara ping com nc
        ip=$l1ip; ping -c $pingcount $ip >/dev/null 2>&1; if [ $? == 0 ]; then
status=up; else sleep $sleeptime; ping -c $pingcount $ip >/dev/null 2>&1 &&
status=up || status=down; fi; l1aux=$status;
```

```
#usando ping para testar
        ip=$l2ip; ping -c $pingcount -w $waitsegs $ip >/dev/null 2>&1; if [ $?
== 0 ]; then status=up; else sleep $sleeptime; ping -c $pingcount -w
$waitsegs $ip >/dev/null 2>&1 && status=up || status=down; fi; l2=$status;
#usando nc para testar
        ip=$l2ip; /usr/bin/nc -w 45 -z $ip 443 >/dev/null 2>&1; if [ $? == 0 ];
then status=up; else sleep $sleeptime; /usr/bin/nc -w 45 -z $ip 443 >/dev/null
2>&1 && status=up || status=down; fi; l2=$status;
#compara ping com nc
        ip=$l2ip; ping -c $pingcount $ip >/dev/null 2>&1; if [ $? == 0 ]; then
status=up; else sleep $sleeptime; ping -c $pingcount $ip >/dev/null 2>&1 &&
status=up || status=down; fi; l2aux=$status;
```

O trecho de script acima faz toda a verificação de conectividade, diferente dos scripts nos roteadores, este script testa, além do ping, um serviço em si, usando *telnet* (comando *nc* em Linux) em um serviço na rede da empresa, como por exemplo, o serviço de VPN na porta 1723, ou uma porta aberta no próprio roteador exclusivamente para este teste, assim o script pode comparar os dois testes, ping e telnet, e ter a certeza de que o link realmente está com queda.

Após verificar a conectividade e detectar a queda, as bases são migradas para outro link assumindo assim outro endereço de IP externo e mantendo o mesmo nome, mantendo o funcionamento das publicações e alta disponibilidade. Esta migração é simplesmente a cópia de uma base já pronta, com os apontamentos de nomes para um determinado link, esta copia é transferida para substituir a base atual e assim restabelecer os serviços externos, conforme a Figura 16, o servidor DNS possui quatro bases, uma base principal com a estrutura padrão, uma base com a copia da base principal, mas com os IPs de um link, no caso, do link Z, outra base

também com a cópia da base principal, mas com os IPs de outro link, neste caso, do link Y, e outra base que é apenas uma cópia da principal, esta terceira cópia é para fins de criação de DNS, para facilitar apenas a manutenção.

```
# Função usada para alterar a base DNS
changeto() {
        echo LastActiveLink=$1 > $LastActiveLinkFile
        msgvar=msg$1
        if [ $1 != L1L2DOWN ]; then
        cd $wkdir/$1
        for x in `ls db.*`; do cp -f -p $x $wkdir; done;
        /etc/init.d/named restart
# Este comando gera novo serial para o DNS
                echo `date +"%d/%m/%Y %T"` Alterado para [$1] - `echo
${!msgvar}` - `$SOASerialChange` >> $log
                subject="QUEDA DE LINK: Dns master externo LinkATIVO [$1]
[`date +"%d/%m/%Y %T"`]"
        else
                echo `date +"%d/%m/%Y %T"` Alterado para [$1] - `echo
${!msgvar}` >> $log
                subject="QUEDA DE LINK: Dns master externo - AMBOS os
LINKs DOWN [$1] [`date +"%d/%m/%Y %T"`]"
        fi
          /usr/bin/tail -v -n 20 $log | /bin/mail -s "$subject" $mailto
}
```

Como pode ser notado, foram tratados apenas dois links da empresa

neste processo, são os dois principais links para publicações externas e com os serviços de VPN da empresa, serviço este usado pelos colaboradores da empresa, não é para suporte aos clientes como o mostrado no script dos roteadores internos na rede local, mas poderiam ser acrescidos outros links também, basta seguir a mesma lógica, o funcionamento seria muito semelhante.

Como processo auxiliar e muito importante, o script de DNS, além de monitorar a conectividade dos links da empresa para troca de DNS, monitora a conectividade com a finalidade de garantir menos falsos positivos no monitorando interno da empresa, este monitoramento interno feito como um dos processos dos scripts utiliza um servidor externo para que possa ser verificada a queda de um link, e este servidor externo precisa ser confiável, o que não podemos garantir com 100% de certeza, por isso o script de DNS externo ajuda neste processo, assim podemos comparar as quedas identificadas pelos scripts internos com as quedas identificadas pelo script externo, porém quando o script interno detecta uma queda e altera o link, o script externo acompanha este processo para que os serviços se mantenham sempre constantes.

Outro processo adicional usado, implementado no mesmo servidor do DNS, é um script básico que monitora se o script principal está rodando assim como todos os processos dependentes dele, e caso não esteja, o script habilita, vide Anexo B.

## 3.5 DETALHAMENTO E LÓGICA DOS SCRIPTS

Neste capítulo será detalhada as principais partes dos scripts que foram necessários para a solução automatizada de troca de links, implementados nos roteadores da empresa, cada script tem como função principal monitorar seu link principal, cada roteador tem uma particularidade diferente, como publicações, navegação na internet, VPNs, entre outros, algumas destas particularidades são tratadas nos scripts, outras são tratadas no script de DNS como processo adicional.

A verificação da conectividade é a mesma em todos eles, assim como

as variáveis utilizadas neles, como quantidade de pings, tempo de espera de cada um dos pings, gateway um e gateway 2 utilizado apenas para os roteadores que possuem VPNs com redundância , ou seja, que podem estar estabelecidas em dois locais diferentes, uma lista de gateways usado para seguir uma ordem de testes e prioridades, variais para configurar as rotas, variável para configurar o IP do servidor alvo, entre outras variáveis que são utilizadas pra controles internos do script.

```
:local inicio [/system clock get time];
:local pingcount 5; :local sleep 10; :local pingcountOK ""; :local gw1 ""; :local
gw2 ""; :local rotas ""; :local ipmon ""; :local gwpingok ""; :local gwslist
""; :local continue "";
:local counter 255; :local troca ""; :local vpnreset ""; :local 2gatewaysonly
""; :local gwcandidato "";
:global alloff; :local alertaalloff sim; :local RotaGwsEncontrados false; :local
contagws 0;
```

Com pequenas variações, as principais características diferenciais dos scripts em cada roteador serão mostradas abaixo.

**a) Script roteador 1 link X: Anexo C**

Este script é considerado o principal script da rede local por tratar, além de um link de comunicação, tratar também do acesso e o estabelecimento da maioria das conexões VPNs com os clientes, diretamente é o que mais causa impacto nos serviços prestados pela empresa.

A principal diferença em relação aos outros scripts é que este trata as conexões VPN que estão criadas neste roteador, estas conexões são necessárias para prestar suporte aos clientes da empresa, por este motivo é considerado o mais impactante. Primeiramente todo script tem uma lista de gateways definidos por ordem de prioridade.

```
:local gws
{192.168.23.66;172.23.1.247;189.8.199.254;192.168.23.65;192.168.23.67;192.168.23.72;};
```

A pós definir a ordem de prioridade foi necessário configurar os parâmetros de quais VPNs foram testados.

```
...
:foreach a in=defaultgw,ClienteVPN2,ClienteVPN4,webex,sienge-dtc,ClienteVPN1,rotaestatica-linkz,rotaestatica-ebt,rotaestatica-linky
do={
...
```

Para cada um dos itens acima, configurados após o sinal de igual, será realizado o processo de verificação de conectividade e failover, tanto para o link de conexão com a internet quanto para os serviços de VPN.

```
:if ($a="ClienteVPN2") do={:set 2gatewaysonly true; :set ipmon 10.1.24.5; :set gw2 ClienteVPN2; :set rotas {10.1.0.0/16;10.2.0.0/16;192.168.10.0/24;}}
        :if ($a="ClienteVPN3") do={:set 2gatewaysonly true; :set gw2 172.23.1.52; :set ipmon 10.46.60.2; :set rotas {10.46.252.0/24;10.47.60.0/24;10.47.70.0/24;10.47.100.0/24;10.47.252.0/24;}}
        :if ($a="ClienteVPN4") do={:set 2gatewaysonly true; :set ipmon 192.168.0.28; :set gw2 ClienteVPN4; :set rotas {192.168.0.0/16;}}
        :if ($a="ClienteVPN1") do={:set 2gatewaysonly true; :set ipmon 10.33.192.130; :set gw1 172.23.3.253; :set gw2
```

```
"ClienteVPN1dtc1,ClienteVPN1dtc2,ClienteVPN1dtc3,ClienteVPN1dtc
4,ClienteVPN1dtc5,ClienteVPN1dtc6"; :set rotas
{10.33.0.0/16;10.34.0.0/16;10.96.0.0/16;10.97.0.0/16;10.98.0.0/16;10.
200.0.0/16;}}
        :if ($a="webex") do={:set gw1 172.23.3.253; :set gw2 pppoe-
Webex; :set rotas
{173.243.0.0/20;114.29.192.0/19;64.68.96.0/19;66.114.160.0/20;66.16
3.32.0/20;209.197.192.0/19;208.8.81.0/24;210.4.192.0/20;62.109.192.
0/18;173.243.0.0/20;};}
        :if ($a="sienge-dtc") do={:set gw1 172.23.3.253; :set gw2
pppoe-Siengedtc; :set rotas {50.22.1.226/32…
```

Como mencionado anteriormente, há alguns casos em que uma VPN pode ter redundância, nestes casos a opção *"2gatewaysonly"* é configurada com o parâmetro "*true"*, ou seja, em outro local há outra VPN que pode ser usada, então quando houver quedas neste cliente a opção dois será outro local, que está indicado com a variável gw2, nos outros casos onde está configurado gw1, e gw2, é apenas para dar prioridade a um roteador, por isso estão configurados um *gateway* principal como gw1 e um *gateway* secundário com gw2.

Há outros parâmetros que foram configurados, como diferença em relação aos outros, que também foram adicionados por causa dos serviços de VPN.

```
#Reseta variáveis comuns
        :set gw1 172.23.3.253; :set gw2 192.168.23.66; :set rotas
""; :set ipmon 8.8.4.4; :set troca false; :set vpnreset ""; :set
2gatewaysonly false;
        :if ($a="defaultgw") do={
```

```
                :set gw1 189.8.199.254; :set rotas
{0.0.0.0/0;187.50.39.75/32;187.50.39.76/32;};
                :set vpnreset
{"ClienteVPN5";"ClienteVPN6";"ClienteVPN7";"ClienteVPN8";"Cliente
VPN9";"ClienteVPN10";"ClienteVPN11";"ClienteVPN12";"ClienteVPN2
";"ClienteVPN4";"ClienteVPN13";"ClienteVPN14";"ClienteVPN15"};
        };
#Redirecionar pelo link Ebt.
        :if ($a="rotaestatica-ebt") do={
                :set gw1 172.23.3.253; :set gw2 192.168.23.66; :set
rotas
{200.19.194.34/32;201.49.164.48/32;200.101.66.117/32;201.90.224.1
12/28;};
                :set vpnreset
{"ClienteVPN16";"ClienteVPN17";"ClienteVPN18";"ClienteVPN19";};
        };
#Redirecionar pelo link z.
        :if ($a="rotaestatica-linkz") do={
                :set gw1 192.168.23.66; :set gw2 192.168.23.83; :set
rotas {177.43.91.132/32;200.142.86.221/32;};
                :set vpnreset
{"ClienteVPN20";"ClienteVPN1dtc1";"ClienteVPN1dtc2";"ClienteVPN1
dtc3";"ClienteVPN1dtc4";"ClienteVPN1dtc5";"ClienteVPN1dtc6";};
        };
#Redirecionar pelo link y.
        :if ($a="rotaestatica-linky") do={
```

```
                :set gw1 192.168.23.83; :set gw2 192.168.23.66; :set
rotas {1.1.1.1/32;};
```

Nesta parte são criados parâmetros para facilitar a troca de *gateway* de alguns clientes VPN, alguns destes precisam estar configurados em um determinado link, regras impostas pelo cliente, por este motivo apenas foi necessário estas alterações.

**b) Script roteador 2 link Y: Anexo D**

Roteador com o link das principais publicações da empresa, como serviços de e-mail, sites, aplicações com acesso externo e serviço de VPN secundário, para que os colaboradores da empresa possam acessar a rede remotamente. Este roteador também é considerado muito importante, pois trata de acessos externos, o que acarreta na indisponibilidade de serviços importantes que clientes precisam acessar via site por exemplo.

Não possui nenhuma característica diferencial em relação aos outros, apenas a ordem de preferência dos *gateways*.

```
:local gws
{192.168.23.71;192.168.23.65;192.168.23.67;192.168.23.72;};
```

Um fator importante é que cada um dos roteadores possui um servidor alvo diferente do outro, para ter maior confiança e porque notasse algumas falhas quando foi feito o monitorando no mesmo servidor alvo.

```
:foreach a in=defaultgw do={
#Reseta vars comuns
        :set gw1 10.111.22.1; :set gw2 192.168.23.66; :set rotas ""; :set
ipmon 69.43.160.154; :set troca false; :set vpnreset ""; :set
2gatewaysonly false;
        :if ($a="defaultgw") do={:set rotas {0.0.0.0/0;};};
```

**c) Script roteador 3 link Z: Anexo E**

Roteador com as publicações de redundância quando da queda do link principal, link Y, e com a saída de acesso principal a internet da empresa e serviço de VPN primário.

```
:local gws
{192.168.23.71;192.168.23.65;192.168.23.67;192.168.23.72;
172.23.3.253;};
```

Decidiu-se que este link seria o link de redundância para todos os serviços externos que estão por padrão direcionado pelo link Y, como configuramos o script DNS para realizar o failover de links dos dois principais links, internamente fizemos o mesmo, não quer dizer que não possa ser adicionado outros links, mas a principio apenas dois links fazem todo o processo de publicações externas.

```
:foreach a in=defaultgw do={
#Reseta vars comuns
        :set gw1 PPPoE; :set gw2 192.168.23.83; :set rotas ""; :set
ipmon 216.146.39.70; :set troca false; :set vpnreset ""; :set
2gatewaysonly false;
```

Todos os links servem como redundância para acesso a internet da empresa, quando mais de um link estiver com problemas, pois cada um dos roteadores tem acesso direto à internet através de um link contratado.

Se forem descartados todos os serviços, publicações e particularidades de cada um dos roteadores, se poderia simplificar todo o script em apenas algumas linhas, monitorando o link e alterando para outros de forma muito mais clara, e utilizando apenas um roteador, o *gateway* default da rede local, o qual atualmente não possui link direto para internet,

porém como toda estrutura é, de cera forma, complexa, isto não seria possível sem alterar a estrutura da rede, o que iria contra uma das restrições impostas para realização do trabalho.

Todo o restante dos scripts é igual, as únicas diferenças foram informadas nesta sessão, e as outras partes dos scripts são apenas para tratamento de alguns parâmetros, como o abaixo que trata da fila de *gateways* testados.

```
#A lista de gws contem gw1 e gw2 nas 2 primeiras posições porem os
demais gateways podem conter gw1 e gw2 repetidos deve-se pular pois já
foram tentados
                                :while (($counter>1 and "$gwcandidato"=$gw1) or
($counter>1 and "$gwcandidato"=$gw2) or ("$gwcandidato"="")) do={
                                        :put "Gateway duplicado ou
branco[$counter]"; :put "$gwcandidato"; :set counter ($counter+1);
                                };
```

Outro fator de tratamento importante que foi utilizado no script é sobre o escalonamento dos scripts. Antes de começar cada script é executado um comando que desativa a execução automática do script, para que o script possa demorar mais tempo, e só após o término das verificações é que o script volta à forma automática.

```
#Desabilitar schedule
/system scheduler set [find name=LinkFailover] disabled=yes


#Habilitar a schedule
/system scheduler set [find name=LinkFailover] disabled=no
```

## 3.6 RESULTADOS E ANÁLISES

Desde o principio do trabalho, o resultado principal esperado era a automatização do processo de troca de links, usando os roteadores *Mikrotik* da empresa, para tal resultado foi necessário uma série de estudos e procedimentos, em conjunto com alguns processos adicionais externos aos roteadores para que a solução chegasse mais perto possível do sucesso.

A Diminuição da indisponibilidade dos links de dados, é a consequência do objetivo principal alcançado, assim como o sucesso na automatização do processo de troca de links, mas não são os principais resultados, decorrente do sucesso deste processo, outros resultados foram importantes, a seguir será citado cada um deles, assim como uma análise dos principais fatores encontrados durante a implementação do trabalho.

Outro resultado importante deste trabalho foi o aumento da eficiência no suporte aos clientes, com o processo de automatização foi possível criar redundância de acesso aos clientes e filiais, criando, por exemplo, sempre duas formas de se conectar a um cliente, o que antes dependia sempre de um link especifico, agora não tem mais está dependência, antes era preciso trocar o acesso manualmente, gerando problemas no suporte, hoje se conseguiu fazer esta troca junto com a solução de automatização de troca de links, utilizando os mesmos scripts para tal processo.

Um fator negativo no processo é a dificuldade de manter a estrutura caso ela cresça de forma desorganizada, está solução é especifica para resolver estes problemas na empresa, é um processo custoso manter toda a estrutura caso seja adicionado, por exemplo, mais três ou mais links, está adição faz com que seja necessário implementar scripts nestes novos links, apesar dos scripts serem padrões, eles dependem de manutenção constante e monitoramentos diários, cada nova alteração é necessário adicionar ou revisar os scripts, estas adições de links além destes fatores que citei referente ao controle, criaria mais pontos de falhas, ou seja, a cada nova adição de links cria-se um ponto de falha também pelo custo adicional de manutenção e complexidade.

Outro fator que ajudou a manter um monitoramento confiável é solicitar relatórios as operadoras de links, afim de comparação com as quedas, para determinar se houve falso positivo na troca de links, as quais

dependem de um servidor externo sempre disponível e este fato é um ponto em que não é possível gerenciar de forma segura, pois o servidor externo é de terceiros e não se possui o domínio dele.

Um problema que foi encontrado foi no reestabelecimento das conexões VPN quando há trocas de links, toda vez que a troca de links é acionada é necessário reiniciar o serviço de VPN, acarretando um ponto de falha no processo, como a VPN está estabelecida através de um determinado gateway, quando houver uma troca de links o gateway será alterado forçando assim o reestabelecimento das VPNs, por não se possível manter o serviço de NAT necessário para os clientes VPN, força o reinicio dos serviços de NAT também, causando perda parcial da conexão, tal perda foi considerada insignificante pelo fato do ganho ser muito maior e ser possível reestabelecer de forma rápida a conexão.

Outro fator importante no processo foi o baixo custo da solução, a qual praticamente não teve custo, além das horas gastas no estudo e na implementação. O baixo custo se deve ao fato de poder usar estes roteadores de forma virtualizada pagando apenas a licença do sistema operacional, esta forma de uso não foi por acaso, foi escolhida para usar a tecnologia de virtualização por proporcionar uma flexibilidade maior, assim como maior segurança e facilidade nos backups e na fase de testes, por isso nas fases de testes foi possível montar um ambiente praticamente idêntico ao real, o que proporcionou maior confiança em todo o processo de criação dos scripts.

A utilização de um servidor externo para auxiliar no processo de monitoramento e manutenção das publicações, foi de grande ajuda, pois sem ele não poderia garantir que todas as publicações externas fossem mantidas, quando trocasse o link localmente nos roteadores. A utilização só foi possível, pois já possuía servidores externos que eram usados prara hospedar alguns serviços da empresa, um deles foi o servidor de DNS da empresa, que hospedou os scripts de DNS mencionados neste trabalho, caso não houvesse um serviço de DNS externo, este tipo de alternativa não seria possível, pois continuaria sem poder trocar as bases DNS, que são essenciais para o funcionamento dos serviços externos.

# 4 CONCLUSÃO

O objetivo principal desde o inicio do trabalho, era à busca de uma solução para um grande problema que a empresa possuía que era a troca manual de links quando houvesse problemas de conectividade. Para alcançar este objetivo, seguindo as restrições impostas pela empresa, e utilizando as tecnologias já empregadas na estrutura de redes, sem que fosse necessária a compra de equipamentos, foi realizado um estudo avançado em roteadores Mikrotik, os quais são os utilizados na empresa, para descobrir uma forma automatizada de realizar o failover de links de internet e manter a estrutura redundante.

O Começo do estudo foi dificultado pela falta de materiais completos e científicos que abrangessem tal problema, com a falta de orientação técnica no processo, resolveu-se criar uma forma própria para resolver o problema, que foi a utilização de scripts na linguagem própria do roteador, para isso foi estudado toda a sintaxe da linguagem a fim de entender sua estrutura para que se pudesse criar de forma segura e com sucesso.

O desenvolvimento dos scripts proporcionou um aprendizado de forma aprofundada dos fundamentos de rede, assim como na busca das melhores soluções, citando como um ponto importante à busca da solução para o problema da monitoração dos links, ou seja, qual o servidor externo usar, quanto tempo deixar monitorando para ter certeza que não era um falso positivo ou ter a certeza de que o problema não era o próprio servidor monitorado, mesmo que este processo de monitoramento possa ser um ponto de falha, ainda assim é o processo mais importante, por isso foi a parte da implementação que gerou o maior número de testes e horas gastas.

Outro ponto importante da solução aqui proposta, foi a utilização de scripts auxiliares no servidor de DNS, hospedado em um *data center* fora do país, o qual proporcionou uma segurança maior na verificação da conectividade, assim podia ter maior certeza de que os scripts na empresa estavam alterando o link no momento certo, evitando na maioria das vezes falsos positivos.

Como processo complementar e importante, o script de DNS ajudou a manter disponíveis as publicações externas da empresa, as quais não precisavam mais ser fixas a um IP externo e sim a uma lista de IPs, cada um

de um link diferente.

Como os recursos eram limitados e os materiais técnicos específicos para o assunto eram escassos, foi priorizado a solução do problema principal, e tentado automatizar da melhor maneira possível, deixando para trabalhos futuros outras complementações que eram também muito importantes, como a priorização do tráfego e a distribuição de banda de forma igualitária, o que tornaria o tempo de implementação muito maior do que o exigido para este trabalho.

## 4.1 PRINCIPAIS CONTRIBUIÇÕES

Devido à particularidade do trabalho, que foi desenvolvido para tratar de um problema especifico, poderia ter poucas contribuições, mas o trabalho pode contribuir tanto para pesquisas relacionadas ao assunto, quanto para implementações semelhantes em roteadores de qualquer porte, é claro, respeitando as funcionalidades de cada um.

Em se tratando dos fundamentos teóricos que podem ser aprendidos com este trabalho, posso citar os fundamentos de redes necessários para implantar alta disponibilidade de links, o conhecimento em roteamento e conhecimentos em segurança de firewall, para proteger toda a rede. O tema de alta disponibilidade e garantia de estruturas de rede está cada vez mais relevante nas empresas, ninguém quer ter seu sistema, ou seu site indisponível, ou se eventualmente isto acontecer querem que o seu restabelecimento seja rápido. Este trabalho contribui com uma lógica simples que pode ser implementada em inúmeros cenários.

A simples ideia de monitorar um serviço externo através da rede local, passando pelo principal link de comunicação, já pode ser de grande contribuição para quem deseja começar a estruturar algum sistema de alta disponibilidade, o fato de manter um constante monitoramento é outro ponto importante, muita empresa esquece-se de manter algum sistema de monitoramento confiável, ou acreditam que o monitoramento por si só não ajuda. Porém as medidas que podem ser tomadas pelo constante monitoramento é que são os principais pontos, monitorar e tomar uma decisão quando algo estiver errado, este é o principal ponto de partida para

a construção de um sistema eficaz de alta disponibilidade.

Outra contribuição importante deste trabalho, é que não se deve pensar somente no controle da rede local, controlando a rede local, é claro, teremos inúmeros benefícios, porém não devemos crer em cem por cento neste controle, pois estamos analisando de dentro para fora, analisando apenas as circunstâncias internas que sempre, ou quase sempre, são diferentes das externas, as circunstâncias externas também precisam ser analisadas pois elas impactam nos serviços também, por isso neste trabalho utilizamos um serviço externo para auxiliar em todo o processo, tornando-se fundamental para os objetivos estabelecidos no começo do trabalho.

Um controle externo visando à rede da empresa e seus serviços podem ser considerados como a segunda contribuição mais importante deste trabalho, um sistema de DNS tem um poder muito grande perante qualquer estrutura de redes, é um serviço fundamental para qualquer serviço público na internet e sem ele é praticamente inviável manter alta disponibilidade de serviços na internet.

## 4.2 TRABALHOS FUTUROS

Com aplicação deste trabalho foi possível observar que há várias complementações possíveis, por se tratar de uma aplicação especifica para solução de um problema, utilizando uma ferramenta ampla e completa de um sistema operacional para roteadores, é possível complementar a solução de diversas formas, dentre elas duas em particular que pretende-se implementar futuramente, que é a priorização de tráfego e distribuição de banda de forma igualitária entre os utilizadores de uma rede.

A priorização do tráfego funcionaria com o objetivo de estabelecer determinados serviços que são essenciais para a empresa ou para determinadas aplicações, e utilizando qualidade de serviço (QOS) priorizar a banda de saída para estes serviços, um bom exemplo seria a priorização do tráfego para telefonia VOIP a qual não pode ter interferência para que funcione de forma correta.

A distribuição de banda de forma igualitária funcionaria para manter o uso correto da internet dentro de uma organização, distribuindo a banda de

forma igual para cada dispositivo ou usuário, tornando assim a uma rede mais estável e de certo modo seguro, cada usuário poderia ter por exemplo 1MB para download e 512KB para upload, deixando a rede democrática e sem abusos, o que sem está priorização penas uma pessoa ou dispositivo poderia consumir toda a internet, gerando assim conflitos e reclamações de lentidão.

# 5 REFERÊNCIAS BIBLIOGRÁFICAS


MARQUES, Wilson Soler. TCP-IP Projetando Redes. Rio de Janeiro: Brasport, 2000.

CHESSWICK, W.R. e Bellovin, S.M. Firewalls and internet security: repelling the wily hacker. Addison-Wesley Publishing Company. 1994.

BARION, Rogério. Mikrotik RouterOS: guia prático. Porto Alegre: Sulina, 2011.

MIKROTIK. Mikrotik RouterOS. Disponível em: <http://wiki.Mikrotik.com/wiki/Manual:Scripting>. Acesso em: 20 dez. 2012.

MIKROTIK. Mikrotik RouterOS. Disponível em: <http://forum.Mikrotik.com/viewtopic.php?f=9&t=38683 >. Acesso em: 20 dez. 2012.

MIKROTIK. Mikrotik RouterOS. Disponível em: <http://www.Mikrotik.com>. Acesso em: 20 dez. 2012.

TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 2. ed. São Paulo, SP:Pearson Education do Brasil, 2005. 695p.

C.SIEWERT, Vanderson. Firewall suas características e vulnerabilidades. Disponível em: <http://artigocientifico.tebas.kinghost.net/uploads/artc_1202930083_51.pdf>. Acesso em: 15 mar. 2013.

MOURA, Alex Soares de. Roteamento: O que é Importante Saber. Disponível em: <http://www.rnp.br/newsgen/9705/n1-1.html>. Acesso

em: 23 mar. 2013.

CHIN, Liou Kuo. Rede Privada Virtual - VPN. Disponível em: <http://www.rnp.br/newsgen/9811/vpn.html>. Acesso em: 23 mar. 2013.

SALVO, Rodrigo. NAT - Introdução. Disponível em: <http://www.ti-redes.com/roteamento/nat/introducao/>. Acesso em: 14 jun. 2013.

DOMINIOS, Registro de. Servidor de Nome de Domínio - DNS. Disponível em: <http://www.registrodedominios.net.br/dominios/servidor-de-nome-de-dominio-dns.html>. Acesso em: 02 nov. 2013.

MARTINEZ, Marina. Gateway. Disponível em: <http://www.infoescola.com/redes-de-computadores/gateway/>. Acesso em: 02 nov. 2013.

P. Mockapetris, "Domain Names - Implementation and Specification", RFC 883, USC/Information Sciences Institute, November 1983.

Crocker, D., "Standard for the Format of ARPA Internet Text Messages," RFC 822, Department of Electrical Engineering, University of Delaware, August 1982.

Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification", RFC 792, USC/Information Sciences Institute, September 1981.

Postel, J. (ed.), "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, USC/Information Sciences Institute, September 1981.

Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

P. Mockapetris, "Domain System Changes and Observations", RFC-973, USC/Information Sciences Institute, January 1986.

# 6 GLOSSÁRIO

ICMP - Internet Control Message Protocol;

NC – netcat, comando em Linux para verificar se uma porta está aberta;

Ping – Utilitário que usa o protocolo ICMP para testar a conectividade entre equipamentos;

Delay – Retardo de sinais;

TTL – Time to Live, número de saltos entre máquinas que os pacotes podem demorar numa rede de computadores antes de serem descartados;

FTP – File transfer protocol, protocolo para transferência de arquivos;

TCP – Transmission control protocol;

UDP - User Datagram Protocol;

RFC – Request For Comments são um conjunto de documentos de referência;

# ANEXO

Anexo A – Script DNS

```
---------- Começo do script ----------------------------------------------------------------
---
#!/bin/bash
wkdir=/var/named/chroot/var/named
mailto=monitor@empresa.com.br
runinterval=10
log=/scripts/dnsfailover.log
SOASerialChange=/scripts/dnsSOASerialChange.sh
msgL1L2='NORMALIZADO LinkY=UP e LinkZ=UP Inbound ALL via LinkY e
VPN[acr] via LinkZ'
msgL2='LinkY=DOWN LinkZ=UP Inbound ALL via LinkZ'
msgL1='LinkZ=DOWN LinkY=UP Inbound ALL via LinkY inclusive VPN[acr]
via LinkY'
msgL1L2DOWN='TODOS LINKS DOWN LinkY=DOWN e LinkZ=DOWN'
LastActiveLinkFile=/scripts/dnsfailover_LastActiveLink.sh
SimulaLastActiveLinkFile=/scripts/dnsfailover_SimulaLastActiveLink.sh

#--- functions
# Função usada para alterar a base DNS
changeto() {
        echo LastActiveLink=$1 > $LastActiveLinkFile
        msgvar=msg$1
        if [ $1 != L1L2DOWN ]; then
        cd $wkdir/$1
        for x in `ls db.*`; do cp -f -p $x $wkdir; done;
        /etc/init.d/named restart
                # Este comando gera novo serial para o DNS
                echo `date +"%d/%m/%Y %T"` Alterado para [$1] - `echo
${!msgvar}` - `$SOASerialChange` >> $log
                subject="QUEDA DE LINK: Dns master externo LinkATIVO [$1]
[`date +"%d/%m/%Y %T"`]"
        else
                echo `date +"%d/%m/%Y %T"` Alterado para [$1] - `echo
${!msgvar}` >> $log
                subject="QUEDA DE LINK: Dns master externo - AMBOS os
LINKs DOWN [$1] [`date +"%d/%m/%Y %T"`]"
        fi
          /usr/bin/tail -v -n 20 $log | /bin/mail -s "$subject" $mailto
}

# Função usada para simulação
simula_notifyonly() {
        echo LastActiveLink=$1 > $SimulaLastActiveLinkFile
        echo SIMULACAO QUEDA DE LINK: Alterado para [$1]
        msgvar=msg$1
        echo `date +"%d/%m/%Y %T"` SIMULACAO alterado para [$1] -
`echo ${!msgvar}` >> $log
```

```
    /bin/mail -s "SIMULACAO QUEDA DE LINK: Dns master externo
LinkATIVO [$1] MailOnlyTo=[$mailtosimula] [`date +"%d/%m/%Y %T"`]"
$mailtosimula < $log
}

#Funcção de verificação de conectivadade
pinga() {
        pingcount=3; sleeptime=120; waitsegs=10;
        if $simula; then
                echo Pingando Link1 e Link2 [Aguarda 2x 10segundos pelo
ping e mais 1.10mins se detectar erro - se ambos links down leva 2.3mins
para executar]
        fi
#usando ping para testar
        ip=$l1ip; ping -c $pingcount -w $waitsegs $ip >/dev/null 2>&1; if [ $?
== 0 ]; then status=up; else sleep $sleeptime; ping -c $pingcount -w
$waitsegs $ip >/dev/null 2>&1 && status=up || status=down; fi; l1=$status
#usando nc para testar
        ip=$l1ip; /usr/bin/nc -w 45 -z $ip 8291 >/dev/null 2>&1; if [ $? == 0 ];
then status=up; else sleep $sleeptime; /usr/bin/nc -w 45 -z $ip
8291 >/dev/null 2>&1 && status=up || status=down; fi; l1=$status;
#compara ping com nc
        ip=$l1ip; ping -c $pingcount $ip >/dev/null 2>&1; if [ $? == 0 ]; then
status=up; else sleep $sleeptime; ping -c $pingcount $ip >/dev/null 2>&1 &&
status=up || status=down; fi; l1aux=$status;

#usando ping para testar
        ip=$l2ip; ping -c $pingcount -w $waitsegs $ip >/dev/null 2>&1; if [ $?
== 0 ]; then status=up; else sleep $sleeptime; ping -c $pingcount -w
$waitsegs $ip >/dev/null 2>&1 && status=up || status=down; fi; l2=$status;
#usando nc para testar
        ip=$l2ip; /usr/bin/nc -w 45 -z $ip 443 >/dev/null 2>&1; if [ $? == 0 ];
then status=up; else sleep $sleeptime; /usr/bin/nc -w 45 -z $ip 443 >/dev/null
2>&1 && status=up || status=down; fi; l2=$status;
#compara ping com nc
        ip=$l2ip; ping -c $pingcount $ip >/dev/null 2>&1; if [ $? == 0 ]; then
status=up; else sleep $sleeptime; ping -c $pingcount $ip >/dev/null 2>&1 &&
status=up || status=down; fi; l2aux=$status;

        if $simula; then echo Resultado NC Link1=[$l1] Link2=[$l2]; fi
        resultado=0
#$usel1l2 indica se tem apontamentos DNS nos dois links ex: www e acr, ou
nao existe o padrao L1L2
        if [ $usel1l2 == true ]; then
                if [ $l1 == up -a $l2 == up ]; then resultado=L1L2; fi
        fi
        if [ $resultado != L1L2 ]; then
                if [ $l1 == down -a $l2 == down ]; then
#$l1 e l2 down, vai chamar a funcao changeto mas dentro dela nao altera o
dns neste caso
                        resultado=L1L2DOWN
                elif [ $l1 == down ]; then
```

```
#$l1 (Link y) esta em down, logo deve assumir L2 (LinkY=down e LinkZ=UP)
                        resultado=L2;
                else
#$l2(link z) esta em down, logo deve assumir L1 (LLivre=Up e LinkZ=Down)
                        resultado=L1;
                fi
        fi

#LastActiveLinkFile.sh seta variavel  LastActiveLink em L1 ou L2 ou L1L2 ou
L1L2DOWN
        if $simula; then
                LastActiveLink=`head -1 $SimulaLastActiveLinkFile | cut -d"=" -
f2`;
                echo Resultado detectado [$resultado] que sera comparado
com LastActiveLink [$LastActiveLink];
        else
                LastActiveLink=`head -1 $LastActiveLinkFile | cut -d"=" -f2`;
        fi

        if [ $resultado != $LastActiveLink ]; then
                if $simula; then
                        simula_notifyonly $resultado;
                else
                        changeto $resultado;
                fi
        elif $simula; then
                echo Nada foi alterado em relacao ao ultimo estado;
        fi
}

# --- main
if [ ! -f $LastActiveLinkFile ]; then echo LastActiveLink=L1L2DOWN >
$LastActiveLinkFile; chmod a+x $LastActiveLinkFile; fi
if [ ! -f $SimulaLastActiveLinkFile ]; then echo LastActiveLink=L1L2DOWN >
$SimulaLastActiveLinkFile; chmod a+x $SimulaLastActiveLinkFile; fi
if [ ! -f $log ]; then touch $log; fi

#Parametro passado via prompt
if [ "$1" == "simula" ]; then
        simula=true;
else
        simula=false;
fi

if $simula; then
        echo ----------------------------------------------------------------------------------------
----------
        echo Script executando em modo simulacao;
        echo Matando o processo dsnfailover.sh que roda em WHILE
INFINITO;
        for x in `ps -eo pid,cmd | fgrep "dnsfailover.sh" | egrep -iv "simula|grep
|vi |vim " | cut -d "/" -f1`; do
```

```
                if [ -n $x ]; then echo Process killed [`ps -fp $x | fgrep "$x"`]; kill -
9 $x; fi
        done
        echo ----------------------------------------------------------------------------
----------
fi

while [ 1 ]
do
#pinga (1=IpMonitorarLink1, 2=NomeDnsACheckar, 3=dirseLink1OK,
4=ipseLink1OK, 5=dirseLink1Out, 6=IpseLink1Out, 7=IpLink2)
#Se LinkY LinkOUT assume LinkZ e LinkY reassume se LinkOk. Compara
pelo IN A www.
        l1ip=187.49.235.161; l1dir=L1; l2ip=186.215.116.36; l2dir=L2;
usel1l2=true;
        pinga;

        if $simula; then
                echo ----------------------------------------------------------------------
------------------
                echo [ /scripts/dnsfailover.sh \& ]   *Volta o script para WHILE
INFINITO ou a crontab fara isso a cada 30 mins
                echo ----------------------------------------------------------------------
------------------
                exit 1;
        else
                /bin/sleep $runinterval;
        fi
done
---------- Fim do script ------------------------------------------------------------------
-
```

Anexo B – Script monitora DNS

```
---------- Começo do script ---------------------------------------------------------------
---
#/bin/bash
mailto=@empresa.com.br
log=/scripts/dnsfailover_checkisrunning.log
processo=dnsfailover.sh
# --- main
touch $log
if [ `ps -ef | grep -v grep | grep "$processo" | grep -v "vi" | wc | cut -c7` -ne "1" ]
then
        echo "----- `date +"%d/%m/%Y %T"`" >> $log;
        echo "Esse script é executado quando o processo dnsfailover.sh não
é detectado"
        echo "Comando executado: ps -ef | grep -v grep | grep $processo -
saida e num de processos seguem" >> $log;
        ps -ef | grep -v grep | grep "$processo" | grep -v "vi" >> $log;
        ps -ef | grep -v grep | grep "$processo" | grep -v "vi" | wc | cut -c7 >>
$log;
```

```
        /bin/mail -s "$processo nao estava rodando reiniciei em
dns.empresa.com.br" $mailto < $log;
        killall $processo > /dev/null 2>&1
        /scripts/$processo &
fi
---------- Fim do script ----------------------------------------------------------------
-
```

Anexo C – Script Roteador 1 link X

```
---------- Começo do script ----------------------------------------------------------
---
#/system script run LinkFailOver
:local inicio [/system clock get time];
:local pingcount 5; :local sleep 10; :local pingcountOK ""; :local gw1 ""; :local
gw2 ""; :local rotas ""; :local ipmon ""; :local gwpingok ""; :local gwslist
""; :local continue "";
:local counter 255; :local troca ""; :local vpnreset ""; :local 2gatewaysonly
""; :local gwcandidato "";
:global alloff; :local alertaalloff sim; :local RotaGwsEncontrados false; :local
contagws 0;

#------------------ INICIO       Alterar neste bloco
:local mk Mk1.249; :local mailto monitor01@empresa.com.br;
#Informar todos os gateways da rede definindo ordem de preferencia
:local gws
{192.168.23.66;172.23.1.247;189.8.199.254;192.168.23.65;192.168.23.67;19
2.168.23.72;};

:foreach a in=defaultgw,ClienteVPN2,ClienteVPN4,webex,sienge-
dtc,ClienteVPN1,rotaestatica-linkz,rotaestatica-ebt,rotaestatica-linky do={

#Reseta variáveis comuns
        :set gw1 172.23.3.253; :set gw2 192.168.23.66; :set rotas ""; :set
ipmon 8.8.4.4; :set troca false; :set vpnreset ""; :set 2gatewaysonly false;

        :if ($a="defaultgw") do={
                :set gw1 189.8.199.254; :set rotas
{0.0.0.0/0;187.50.39.75/32;187.50.39.76/32;};
                :set vpnreset
{"ClienteVPN5";"ClienteVPN6";"ClienteVPN7";"ClienteVPN8";"ClienteVPN9";"
ClienteVPN10";"ClienteVPN11";"ClienteVPN12";"ClienteVPN2";"ClienteVPN4
";"ClienteVPN13";"ClienteVPN14";"ClienteVPN15"};
        };

#Redirecionar pelo link Ebt.
        :if ($a="rotaestatica-ebt") do={
                :set gw1 172.23.3.253; :set gw2 192.168.23.66; :set rotas
{200.19.194.34/32;201.49.164.48/32;200.101.66.117/32;201.90.224.112/28;};
                :set vpnreset
{"ClienteVPN16";"ClienteVPN17";"ClienteVPN18";"ClienteVPN19";};
        };
```

```
#Redirecionar pelo link z.
        :if ($a="rotaestatica-linkz") do={
                :set gw1 192.168.23.66; :set gw2 192.168.23.83; :set rotas
{177.43.91.132/32;200.142.86.221/32;};
                :set vpnreset
{"ClienteVPN20";"ClienteVPN1dtc1";"ClienteVPN1dtc2";"ClienteVPN1dtc3";"
ClienteVPN1dtc4";"ClienteVPN1dtc5";"ClienteVPN1dtc6";};
        };

#Redirecionar pelo link y.
        :if ($a="rotaestatica-linky") do={
                :set gw1 192.168.23.83; :set gw2 192.168.23.66; :set rotas
{1.1.1.1/32;};
        };
        :if ($a="ClienteVPN2") do={:set 2gatewaysonly true; :set ipmon
10.1.24.5; :set gw2 ClienteVPN2; :set rotas
{10.1.0.0/16;10.2.0.0/16;192.168.10.0/24;}}
        :if ($a="ClienteVPN3") do={:set 2gatewaysonly true; :set gw2
172.23.1.52; :set ipmon 10.46.60.2; :set rotas
{10.46.252.0/24;10.47.60.0/24;10.47.70.0/24;10.47.100.0/24;10.47.252.0/24;
}}
        :if ($a="ClienteVPN4") do={:set 2gatewaysonly true; :set ipmon
192.168.0.28; :set gw2 ClienteVPN4; :set rotas {192.168.0.0/16;}}
        :if ($a="ClienteVPN1") do={:set 2gatewaysonly true; :set ipmon
10.33.192.130; :set gw1 172.23.3.253; :set gw2
"ClienteVPN1dtc1,ClienteVPN1dtc2,ClienteVPN1dtc3,ClienteVPN1dtc4,Clien
teVPN1dtc5,ClienteVPN1dtc6"; :set rotas
{10.33.0.0/16;10.34.0.0/16;10.96.0.0/16;10.97.0.0/16;10.98.0.0/16;10.200.0.0
/16;}}
        :if ($a="webex") do={:set gw1 172.23.3.253; :set gw2 pppoe-
Webex; :set rotas
{173.243.0.0/20;114.29.192.0/19;64.68.96.0/19;66.114.160.0/20;66.163.32.0/
20;209.197.192.0/19;208.8.81.0/24;210.4.192.0/20;62.109.192.0/18;173.243.
0.0/20;};}
        :if ($a="sienge-dtc") do={:set gw1 172.23.3.253; :set gw2 pppoe-
Siengedtc; :set rotas
{50.22.1.226/32;50.22.1.227/32;50.22.1.230/32;50.22.1.232/32;108.168.192.
72/32;108.168.192.73/32;50.97.128.12/32;50.97.128.3/32;74.52.48.146/32;7
4.54.79.243/32;74.55.82.250/32;74.55.83.66/32;74.55.114.154/32;75.125.10
7.170/32;75.125.175.138/32;108.168.192.67/32;108.168.220.66/32;173.193.
140.66/32;173.193.148.98/32;173.193.148.99/32;174.37.189.3/32;174.37.18
9.4/32;174.120.140.2/32;174.122.168.186/32;174.123.177.154/32;174.132.2
25.162/32;174.133.28.130/32;174.133.45.226/32;174.133.71.82/32;207.218.
202.66/32;108.168.220.68/32;50.22.1.299/32;50.22.1.236/32;50.22.4.202/32;
50.97.128.6/32;50.97.128.7/32;108.168.192.68/32;108.168.192.69/32;108.16
8.192.70/32;208.43.227.2/32;50.97.128.4/32;50.22.1.229/32;50.22.1.236/32;
50.22.4.202/32;50.97.128.5/32;108.168.220.69/32;50.22.1.229/32;108.168.1
92.66/32;108.168.162.71/32;108.168.220.70/32;108.168.220.71/32;50.22.1.2
33/32;50.22.1.237/32;50.97.128.11/32;};};
#------------------ FIM          Alterar neste bloco

#Desabilita a schedule, assim o script pode levar mais de 1 minuto para
```

```
executar,caso necessário
/system scheduler set [find name=LinkFailover] disabled=yes

#Cria rota para IP de monitoramento(servidor alvo) se nao existir precisa desta entrada para a lógica do script
	:if ([/ip route find dst-address="$ipmon/32"]="") do={/ip route add comment="LinkFailover $a monitored IP" disabled=no distance=1 dst-address="$ipmon/32" gateway=$gw1 scope=30 target-scope=10;}
	:if ($2gatewaysonly) do={:set gwslist [(:put "$gw1", "$gw2")]} else={:set gwslist [(:put "$gw1", "$gw2","$gws")];};
	:set continue true; :set counter 0; :set gwpingok "";
	:while ($continue) do={
		:if ([:len $gwslist]>$counter) do={
			:set gwcandidato [:tostr [:pick $gwslist $counter]];
			:put "----- Testando $a gateway[$counter] IP na linha abaixo"; :put [:pick $gwslist $counter];
			:if ([/interface find name="$gwcandidato"]!="" and [/interface find name="$gwcandidato" disabled=no]="") do={
				:put "Gateway eh uma interface  e esta desabilitada ou nao existe";
				:set counter ($counter+1);
			} else={
#A lista de gws contem gw1 e gw2 nas 2 primeiras posicoes porem os demais gateways podem conter gw1 e gw2 repetidos deve-se pular pois ja foram tentados
				:while (($counter>1 and "$gwcandidato"=$gw1) or ($counter>1 and "$gwcandidato"=$gw2) or ("$gwcandidato"="")) do={
					:put "Gateway duplicado ou branco[$counter]"; :put "$gwcandidato"; :set counter ($counter+1);
				};
#Altera rota ipmon via gateway da lista posicao counter, para testar a conectividade
				/ip route set [find dst-address="$ipmon/32"] gateway="$gwcandidato";
				:local statusrota [/ip route find dst-address="$ipmon/32"];
				:if ([/ip route get $statusrota active]=true) do={
#Cliente diferenciado, utilizamos outros paramentros de ping, diferentes do padrão, por isso deve ser alterado aqui
					:if ([:pick $a 0 4]="ClienteVPN1") do={:set pingcount 10;}; :set pingcountOK 0;
					:set pingcountOK [/ping $ipmon count=$pingcount interval=1]; :if ($pingcountOK>=1) do={:nothing} else={delay $sleep; :set pingcountOK [/ping $ipmon count=$pingcount interval=1]}
					:if ( $pingcountOK >= 1 ) do={
						:set gwpingok "$gwcandidato";
						:set continue false;
#Remove $a do array $alloff se $a estiver listado dentro dele
						:if ([:len $alloff]>0) do={
							:local alloffaux {};
							:foreach e in=$alloff do={:if
```

```
($e!=$a) do={set alloffaux ($alloffaux,$e)};};
                                                :if ([:len $alloff] != [:len
$alloffaux]) do={
                                                        :put "$mk LinkFailover
[$a] NORMALIZADO via gateway [$gwpingok] antes todos gateways
estavam indisponiveis";
                                                        /tool e-mail send
to=$mailto subject="$mk LinkFailover $inicio $a NORMALIZADO Via
$gwpingok antes TODOS GWs estavam indisponiveis" body="Vazio";
                                                }
                                                :set alloff $alloffaux;
                                        }
                                }
                        } else={
                                :put "Gateway unreachable"
                        }
                        :set counter ($counter+1);
#ClienteVPN2 nao aceita ping dentro da vpn, se gwlist chegou em
ClienteVPN2 sabe-se que gw1=1.100 nao funcionou assim assume-se a vpn
ClienteVPN2 e sai do laço, nao tem outra opcao de gw
                        :if ($gwpingok="") do={:if
("$gwcandidato"="ClienteVPN2") do={:set gwpingok ClienteVPN2; :set
continue false; :put "Gateway [ClienteVPN2] foi assumido";};}
                }
        } else={:set continue false;};
    }
    :if ($gwpingok!="") do={
#Se a rota1 do array rotas ja estiver pelo gwpingok nao precisa alterar nada
        :set RotaGwsEncontrados false;
        :if ([/ip route find dst-address=[:pick $rotas 0] and !routing-
mark]="") do={:nothing} else={
                :if ([find $gwpingok ","]>=0) do={
                        :local gwpingokAUX [:toarray $gwpingok];
                        :local gwsgravados [:toarray [ip route get [ip route
find where dst-address=[:pick $rotas 0] !routing-mark] gateway]];
                        :if ([:len $gwsgravados]>1) do={
                                :local numofgws [:len $gwsgravados]; :set
contagws 0;
                                :foreach f in=$gwpingokAUX do={:foreach g
in=$gwsgravados do={:if ($f=$g) do={:set contagws ($contagws+1);};};};
                                :if ($numofgws=$contagws) do={:set
RotaGwsEncontrados true;};
                        }
                } else={
#gwpingok é um unico gateway e nao uma lista de gateways
                        :if ([/ip route find dst-address=[:pick $rotas
0] !routing-mark gateway="$gwpingok"]="") do={:nothing} else={:set
RotaGwsEncontrados true};
                }
        }
        :if ( $RotaGwsEncontrados=false ) do={
                :if ([:len $rotas]>0) do={
```

```
                                :foreach b in=$rotas do={
#Se rota nao tem / colocar /32 no final
                                :if ([find $b "/"]>=0) do={nothing} else={:set
b "$b/32"};
                                :if ([/ip route find dst-address="$b" !routing-
mark]="") do={
                                        /ip route add comment="$a add by
LinkFailover script" disabled=no distance=1 dst-address="$b"
gateway="$gwpingok" scope=30 target-scope=10;
                                        :put "rota adicionada=$b
descricao=$a"
                                } else={
                                        /ip route set [find dst-
address="$b" !routing-mark] gateway=$gwpingok;
                                }
                        }
                        :set troca true;
                        :log info "$mk LinkFailover $a setado via
$gwpingok";
                        /tool e-mail send to=$mailto subject="$mk
LinkFailover $inicio $a setado via $gwpingok" body="Vazio";
#Reseta vpns interfaces e nats
                        :if ($vpnreset != "") do={
#Todas ifaces vpn  :foreach c in=[/interface find where !disabled &&
type!="ether"]
                                :foreach c in=$vpnreset do={
                                        :if ([/interface find name="$c"
disabled=no]="") do={
                                                :put "Vpn NAO resetada: [$c]
nao existe ou esta desabilitada";
                                        } else={
                                                :local x [/interface get $c
name];
                                                :if ([/ip address print count-
only where interface=$x]>=2) do={
                                                        :put "$x estava com 2
ou mais Ips";
                                                        /ip address remove [/ip
address find where interface=$x];
                                                };
                                                /interface disable $c;
                                                /interface enable $c; :put "[$c]
VPN resetada";
                                                :if ([/ip firewall nat find out-
interface="$c" disabled=no]!="") do={
                                                        /ip firewall nat disable
[/ip firewall nat find out-interface="$c"];
                                                        /ip firewall nat enable
[/ip firewall nat find out-interface="$c"];
                                                        :put "[$c] NAT regra
out-interface=[$c] resetada";
                                                }
```

```
                                        }
                                    }
                                }
                            }
                    } else={:put "$gwpingok ja estava setado nas rotas de $a nao
foi necessario alterar"};
            } else={
                    :set alertaalloff sim;
#Se o link $a esta no array é porque na ultima execucao do script ja detectou
AllOFF para o link$a
                    :if ([:len $alloff]>0) do={:foreach d in=$alloff do={:if ($d=$a)
do={:set alertaalloff nao;};};};
                    :if ($alertaalloff = "sim") do={
                            :put "$mk LinkFailover $a GRAVE todos os gateways
INDISPONIVEIS";
                            /tool e-mail send to=$mailto subject="$mk LinkFailover
$inicio [$a] GRAVE todos os gateways INDISPONIVEIS" body="Vazio";
                            :set alloff ($alloff,$a);
                    } else={
                            :put "CONTINUA - $mk LinkFailover [$a] GRAVE todos
os gateways INDISPONIVEIS";
                    }
            };
            :put "$a trocaGw=$troca gwnew=$gwpingok ipmon=$ipmon   Rotas
afetadas - Ordem de GWs - Vpns para resetar nas 2 linhas seguintes";
            :put "$rotas";
            :put $gwslist;
            :put "$vpnreset";
}
:put "Tempo de execucao"
:put $inicio;
:put [/system clock get time];
# Volta a schedule. Vai executar 1 minuto a frente
/system scheduler set [find name=LinkFailover] disabled=no
---------- Fim do script ----------------------------------------------------------------
-
```

Anexo D – Script roteador 2 link Y

```
---------- Começo do script ----------------------------------------------------------
---
#/system script run LinkFailOver
:local inicio [/system clock get time];
:local pingcount 5; :local pingcountOK ""; :local gw1 ""; :local gw2 ""; :local
rotas ""; :local ipmon ""; :local gwpingok ""; :local gwslist ""; :local continue
""; :local counter 255; :local troca ""; :local vpnreset ""; local 2gatewaysonly "";
:global alloff; :local alertaalloff sim;

#------------------ INICIO       Alterar neste bloco
:local mk Mk1.247; :local mailto monitor01@empresa.com.br;
#       Informar todos os gateways da rede definindo ordem de preferencia
:local gws {192.168.23.71;192.168.23.65;192.168.23.67;192.168.23.72;};
```

```
#:foreach a in=teste do={
:foreach a in=defaultgw do={
#Reseta vars comuns
	:set gw1 10.111.22.1; :set gw2 192.168.23.66; :set rotas ""; :set ipmon
69.43.160.154; :set troca false; :set vpnreset ""; :set 2gatewaysonly false;
	:if ($a="defaultgw") do={:set rotas {0.0.0.0/0;};};
	:if ($a="teste") do={:set ipmon 69.163.197.192; :set rotas
{200.162.176.4/32;}; :set vpnreset {"ugm1";};}
#------------------- FIM		Alterar neste bloco

#Desabilita a schedule, assim o script pode levar mais de 1 minuto para
executar,caso necessário
/system scheduler set [find name=LinkFailover] disabled=yes

#Cria rota para IP de monitoramento(servidor alvo) se nao existir precisa
desta entrada para a lógica do script
	:if ([/ip route find dst-address="$ipmon/32"]="") do={/ip route add
comment="LinkFailover $a monitored IP" disabled=no distance=1 dst-
address="$ipmon/32" gateway=$gw1 scope=30 target-scope=10;}
	:if ($2gatewaysonly) do={:set gwslist [(:put "$gw1", "$gw2")]} else={:set
gwslist [(:put "$gw1", "$gw2","$gws")];};
	:set continue true; :set counter 0; :set gwpingok "";
	:while ($continue) do={
		:if ([:len $gwslist]>$counter) do={
			:set gwcandidato [:tostr [:pick $gwslist $counter]];
			:put "----- Testando $a gateway[$counter] IP na linha
abaixo"; :put [:pick $gwslist $counter];
			:if ([/interface find name="$gwcandidato"]!="" and
[/interface find name="$gwcandidato" disabled=no]="") do={
				:put "Gateway eh uma interface  e esta
desabilitada ou nao existe";
				:set counter ($counter+1);
			} else={
#A lista de gws contem gw1 e gw2 nas 2 primeiras posicoes porem os
demais gateways podem conter gw1 e gw2 repetidos deve-se pular pois ja
foram tentados
				:while (($counter>1 and "$gwcandidato"=$gw1) or
($counter>1 and "$gwcandidato"=$gw2) or ("$gwcandidato"="")) do={
					:put "Gateway duplicado ou
branco[$counter]"; :put "$gwcandidato"; :set counter ($counter+1);
				};
#Altera rota ipmon via gateway da lista posicao counter, para testar a
conectividade
				/ip route set [find dst-address="$ipmon/32"]
gateway="$gwcandidato";
				:local statusrota [/ip route find dst-
address="$ipmon/32"];
				:if ([/ip route get $statusrota active]=true) do={
#Cliente diferenciado, utilizamos outros paramentros de ping, diferentes do
padrão, por isso deve ser alterado aqui
					:if ([:pick $a 0 4]="ClienteVPN1") do={:set
pingcount 10;}; :set pingcountOK 0;
```

```
                                        :set pingcountOK [/ping $ipmon
count=$pingcount interval=1]; :if ($pingcountOK>=1) do={:nothing}
else={delay $sleep; :set pingcountOK [/ping $ipmon count=$pingcount
interval=1]}
                                        :if ( $pingcountOK >= 1 ) do={
                                                :set gwpingok "$gwcandidato";
                                                :set continue false;
#Remove $a do array $alloff se $a estiver listado dentro dele
                                                :if ([:len $alloff]>0) do={
                                                        :local alloffaux {};
                                                        :foreach e in=$alloff do={:if
($e!=$a) do={set alloffaux ($alloffaux,$e)};};
                                                        :if ([:len $alloff] != [:len
$alloffaux]) do={
                                                                :put "$mk LinkFailover
[$a] NORMALIZADO via gateway [$gwpingok] antes todos gateways
estavam indisponiveis";
                                                                /tool e-mail send
to=$mailto subject="$mk LinkFailover $inicio $a NORMALIZADO Via
$gwpingok antes TODOS GWs estavam indisponiveis" body="Vazio";
                                                        }
                                                        :set alloff $alloffaux;
                                                }
                                        }
                                } else={
                                        :put "Gateway unreachable"
                                }
                                :set counter ($counter+1);
#ClienteVPN2 nao aceita ping dentro da vpn, se gwlist chegou em
ClienteVPN2 sabe-se que gw1=1.100 nao funcionou assim assume-se a vpn
ClienteVPN2 e sai do laço, nao tem outra opcao de gw
                                :if ($gwpingok="") do={:if
("$gwcandidato"="ClienteVPN2") do={:set gwpingok ClienteVPN2; :set
continue false; :put "Gateway [ClienteVPN2] foi assumido";};}
                        }
                } else={:set continue false;};
        }
        :if ($gwpingok!="") do={
#Se a rota1 do array rotas ja estiver pelo gwpingok nao precisa alterar nada
                :set RotaGwsEncontrados false;
                :if ([/ip route find dst-address=[:pick $rotas 0] and !routing-
mark]="") do={:nothing} else={
                        :if ([find $gwpingok ","]>=0) do={
                                :local gwpingokAUX [:toarray $gwpingok];
                                :local gwsgravados [:toarray [ip route get [ip route
find where dst-address=[:pick $rotas 0] !routing-mark] gateway]];
                                :if ([:len $gwsgravados]>1) do={
                                        :local numofgws [:len $gwsgravados]; :set
contagws 0;
                                        :foreach f in=$gwpingokAUX do={:foreach g
in=$gwsgravados do={:if ($f=$g) do={:set contagws ($contagws+1);};};};
                                        :if ($numofgws=$contagws) do={:set
```

```
RotaGwsEncontrados true;};
                                        }
                                } else={
#gwpingok é um unico gateway e nao uma lista de gateways
                                        :if ([/ip route find dst-address=[:pick $rotas
0] !routing-mark gateway="$gwpingok"]="") do={:nothing} else={:set
RotaGwsEncontrados true};
                                }
                        }
                        :if ( $RotaGwsEncontrados=false ) do={
                                :if ([:len $rotas]>0) do={
                                        :foreach b in=$rotas do={
#Se rota nao tem / colocar /32 no final
                                                :if ([find $b "/"]>=0) do={nothing} else={:set
b "$b/32"};
                                                :if ([/ip route find dst-address="$b" !routing-
mark]="") do={
                                                        /ip route add comment="$a add by
LinkFailover script" disabled=no distance=1 dst-address="$b"
gateway="$gwpingok" scope=30 target-scope=10;
                                                        :put "rota adicionada=$b
descricao=$a"
                                                } else={
                                                        /ip route set [find dst-
address="$b" !routing-mark] gateway=$gwpingok;
                                                }
                                        }
                                        :set troca true;
                                        :log info "$mk LinkFailover $a setado via
$gwpingok";
                                        /tool e-mail send to=$mailto subject="$mk
LinkFailover $inicio $a setado via $gwpingok" body="Vazio";
#Reseta vpns interfaces e nats
                                        :if ($vpnreset != "") do={
#Todas ifaces vpn  :foreach c in=[/interface find where !disabled &&
type!="ether"]
                                                :foreach c in=$vpnreset do={
                                                        :if ([/interface find name="$c"
disabled=no]="") do={
                                                                :put "Vpn NAO resetada: [$c]
nao existe ou esta desabilitada";
                                                        } else={
                                                                :local x [/interface get $c
name];
                                                                :if ([/ip address print count-
only where interface=$x]>=2) do={
                                                                        :put "$x estava com 2
ou mais Ips";
                                                                        /ip address remove [/ip
address find where interface=$x];
                                                                };
                                                                /interface disable $c;
```

```
                                                        /interface enable $c; :put "[$c]
VPN resetada";
                                                        :if ([/ip firewall nat find out-
interface="$c" disabled=no]!="") do={
                                                                /ip firewall nat disable
[/ip firewall nat find out-interface="$c"];
                                                                /ip firewall nat enable
[/ip firewall nat find out-interface="$c"];
                                                                :put "[$c] NAT regra
out-interface=[$c] resetada";
                                                        }
                                                }
                                        }
                                }
                        }
                } else={:put "$gwpingok ja estava setado nas rotas de $a nao
foi necessario alterar"};
        } else={
                :set alertaalloff sim;
#Se o link $a esta no array é porque na ultima execucao do script ja detectou
AllOFF para o link$a
                :if ([:len $alloff]>0) do={:foreach d in=$alloff do={:if ($d=$a)
do={:set alertaalloff nao;};};};
                :if ($alertaalloff = "sim") do={
                        :put "$mk LinkFailover $a GRAVE todos os gateways
INDISPONIVEIS";
                        /tool e-mail send to=$mailto subject="$mk LinkFailover
$inicio [$a] GRAVE todos os gateways INDISPONIVEIS" body="Vazio";
                        :set alloff ($alloff,$a);
                } else={
                        :put "CONTINUA - $mk LinkFailover [$a] GRAVE todos
os gateways INDISPONIVEIS";
                }
        };
        :put "$a trocaGw=$troca gwnew=$gwpingok ipmon=$ipmon   Rotas
afetadas - Ordem de GWs - Vpns para resetar nas 2 linhas seguintes";
        :put "$rotas";
        :put $gwslist;
        :put "$vpnreset";
}
:put "Tempo de execucao"
:put $inicio;
:put [/system clock get time];
# Volta a schedule. Vai executar 1 minuto a frente
/system scheduler set [find name=LinkFailover] disabled=no
---------- Fim do script ----------------------------------------------------------------
-
```

Anexo E – Script roteador 3 link Z

```
---------- Começo do script ----------------------------------------------------------
---
```

```
#/system script run LinkFailOver
:local inicio [/system clock get time];
:local pingcount 5; :local sleep 10; :local pingcountOK ""; :local gw1 ""; :local
gw2 ""; :local rotas ""; :local ipmon ""; :local gwpingok ""; :local gwslist
""; :local continue "";
:local counter 255; :local troca ""; :local vpnreset ""; :local 2gatewaysonly
""; :local gwcandidato "";
:global alloff; :local alertaalloff sim; :local RotaGwsEncontrados false; :local
contagws 0;

#------------------ INICIO       Alterar neste bloco
:local mk Mk1.251; :local mailto monitor01@empresa.com.br;
#Informar todos os gateways da rede definindo ordem de preferencia
:local gws {192.168.23.71;192.168.23.65;192.168.23.67;192.168.23.72;
172.23.3.253;};
#:foreach a in=teste do={
:foreach a in=defaultgw do={
#Reseta vars comuns
        :set gw1 PPPoE; :set gw2 192.168.23.83; :set rotas ""; :set ipmon
216.146.39.70; :set troca false; :set vpnreset ""; :set 2gatewaysonly false;
        :if ($a="defaultgw") do={:set gw1 PPPoE; :set gw2
"192.168.23.83,192.168.23.71,172.23.3.253"; :set rotas {0.0.0.0/0;};};
#------------------- FIM          Alterar neste bloco

#Desabilita a schedule, assim o script pode levar mais de 1 minuto para
executar,caso necessário
/system scheduler set [find name=LinkFailover] disabled=yes

#Cria rota para IP de monitoramento(servidor alvo) se nao existir precisa
desta entrada para a lógica do script
        :if ([/ip route find dst-address="$ipmon/32"]="") do={/ip route add
comment="LinkFailover $a monitored IP" disabled=no distance=1 dst-
address="$ipmon/32" gateway=$gw1 scope=30 target-scope=10;}
        :if ($2gatewaysonly) do={:set gwslist [(:put "$gw1", "$gw2")]} else={:set
gwslist [(:put "$gw1", "$gw2","$gws")];};
        :set continue true; :set counter 0; :set gwpingok "";
        :while ($continue) do={
                :if ([:len $gwslist]>$counter) do={
                        :set gwcandidato [:tostr [:pick $gwslist $counter]];
                        :put "----- Testando $a gateway[$counter] IP na linha
abaixo"; :put [:pick $gwslist $counter];
                        :if ([/interface find name="$gwcandidato"]!="" and
[/interface find name="$gwcandidato" disabled=no]="") do={
                                :put "Gateway eh uma interface  e esta
desabilitada ou nao existe";
                                :set counter ($counter+1);
                        } else={
#A lista de gws contem gw1 e gw2 nas 2 primeiras posicoes porem os
demais gateways podem conter gw1 e gw2 repetidos deve-se pular pois ja
foram tentados
                                :while (($counter>1 and "$gwcandidato"=$gw1) or
($counter>1 and "$gwcandidato"=$gw2) or ("$gwcandidato"="")) do={
```

```
                                        :put "Gateway duplicado ou
branco[$counter]"; :put "$gwcandidato"; :set counter ($counter+1);
                                    };
#Altera rota ipmon via gateway da lista posicao counter, para testar a
conectividade
                                    /ip route set [find dst-address="$ipmon/32"]
gateway="$gwcandidato";
                                    :local statusrota [/ip route find dst-
address="$ipmon/32"];
                                    :if ([/ip route get $statusrota active]=true) do={
#Cliente diferenciado, utilizamos outros paramentros de ping, diferentes do
padrão, por isso deve ser alterado aqui
                                        :if ([:pick $a 0 4]="ClienteVPN1") do={:set
pingcount 10;}; :set pingcountOK 0;
                                        :set pingcountOK [/ping $ipmon
count=$pingcount interval=1]; :if ($pingcountOK>=1) do={:nothing}
else={delay $sleep; :set pingcountOK [/ping $ipmon count=$pingcount
interval=1]}
                                        :if ( $pingcountOK >= 1 ) do={
                                            :set gwpingok "$gwcandidato";
                                            :set continue false;
#Remove $a do array $alloff se $a estiver listado dentro dele
                                            :if ([:len $alloff]>0) do={
                                                :local alloffaux {};
                                                :foreach e in=$alloff do={:if
($e!=$a) do={set alloffaux ($alloffaux,$e)};};
                                                :if ([:len $alloff] != [:len
$alloffaux]) do={
                                                    :put "$mk LinkFailover
[$a] NORMALIZADO via gateway [$gwpingok] antes todos gateways
estavam indisponiveis";
                                                    /tool e-mail send
to=$mailto subject="$mk LinkFailover $inicio $a NORMALIZADO Via
$gwpingok antes TODOS GWs estavam indisponiveis" body="Vazio";
                                                }
                                                :set alloff $alloffaux;
                                            }
                                        }
                                    } else={
                                        :put "Gateway unreachable"
                                    }
                                    :set counter ($counter+1);
#ClienteVPN2 nao aceita ping dentro da vpn, se gwlist chegou em
ClienteVPN2 sabe-se que gw1=1.100 nao funcionou assim assume-se a vpn
ClienteVPN2 e sai do laço, nao tem outra opcao de gw
                                    :if ($gwpingok="") do={:if
("$gwcandidato"="ClienteVPN2") do={:set gwpingok ClienteVPN2; :set
continue false; :put "Gateway [ClienteVPN2] foi assumido";};}
                                }
                            } else={:set continue false;};
                        }
                        :if ($gwpingok!="") do={
```

```
#Se a rota1 do array rotas ja estiver pelo gwpingok nao precisa alterar nada
                :set RotaGwsEncontrados false;
                :if ([/ip route find dst-address=[:pick $rotas 0] and !routing-
mark]="") do={:nothing} else={
                        :if ([find $gwpingok ","]>=0) do={
                                :local gwpingokAUX [:toarray $gwpingok];
                                :local gwsgravados [:toarray [ip route get [ip route
find where dst-address=[:pick $rotas 0] !routing-mark] gateway]];
                                :if ([:len $gwsgravados]>1) do={
                                        :local numofgws [:len $gwsgravados]; :set
contagws 0;
                                        :foreach f in=$gwpingokAUX do={:foreach g
in=$gwsgravados do={:if ($f=$g) do={:set contagws ($contagws+1);};};};
                                        :if ($numofgws=$contagws) do={:set
RotaGwsEncontrados true;};
                                }
                        } else={
#gwpingok é um unico gateway e nao uma lista de gateways
                                :if ([/ip route find dst-address=[:pick $rotas
0] !routing-mark gateway="$gwpingok"]="") do={:nothing} else={:set
RotaGwsEncontrados true};
                        }
                }
                :if ( $RotaGwsEncontrados=false ) do={
                        :if ([:len $rotas]>0) do={
                                :foreach b in=$rotas do={
#Se rota nao tem / colocar /32 no final
                                        :if ([find $b "/"]>=0) do={nothing} else={:set
b "$b/32"};
                                        :if ([/ip route find dst-address="$b" !routing-
mark]="") do={
                                                /ip route add comment="$a add by
LinkFailover script" disabled=no distance=1 dst-address="$b"
gateway="$gwpingok" scope=30 target-scope=10;
                                                :put "rota adicionada=$b
descricao=$a"
                                        } else={
                                                /ip route set [find dst-
address="$b" !routing-mark] gateway=$gwpingok;
                                        }
                                }
                                :set troca true;
                                :log info "$mk LinkFailover $a setado via
$gwpingok";
                                /tool e-mail send to=$mailto subject="$mk
LinkFailover $inicio $a setado via $gwpingok" body="Vazio";
#Reseta vpns interfaces e nats
                                :if ($vpnreset != "") do={
#Todas ifaces vpn  :foreach c in=[/interface find where !disabled &&
type!="ether"]
                                        :foreach c in=$vpnreset do={
                                                :if ([/interface find name="$c"
```

```
disabled=no]="") do={
                                        :put "Vpn NAO resetada: [$c]
nao existe ou esta desabilitada";
                                } else={
                                        :local x [/interface get $c
name];
                                        :if ([/ip address print count-
only where interface=$x]>=2) do={
                                                :put "$x estava com 2
ou mais Ips";
                                                /ip address remove [/ip
address find where interface=$x];
                                        };
                                        /interface disable $c;
                                        /interface enable $c; :put "[$c]
VPN resetada";
                                        :if ([/ip firewall nat find out-
interface="$c" disabled=no]!="") do={
                                                /ip firewall nat disable
[/ip firewall nat find out-interface="$c"];
                                                /ip firewall nat enable
[/ip firewall nat find out-interface="$c"];
                                                :put "[$c] NAT regra
out-interface=[$c] resetada";
                                        }
                                }
                        }
                }
        } else={:put "$gwpingok ja estava setado nas rotas de $a nao
foi necessario alterar"};
    } else={
        :set alertaalloff sim;
#Se o link $a esta no array é porque na ultima execucao do script ja detectou
AllOFF para o link$a
        :if ([:len $alloff]>0) do={:foreach d in=$alloff do={:if ($d=$a)
do={:set alertaalloff nao;};};};
        :if ($alertaalloff = "sim") do={
            :put "$mk LinkFailover $a GRAVE todos os gateways
INDISPONIVEIS";
            /tool e-mail send to=$mailto subject="$mk LinkFailover
$inicio [$a] GRAVE todos os gateways INDISPONIVEIS" body="Vazio";
            :set alloff ($alloff,$a);
        } else={
            :put "CONTINUA - $mk LinkFailover [$a] GRAVE todos
os gateways INDISPONIVEIS";
        }
    };
    :put "$a trocaGw=$troca gwnew=$gwpingok ipmon=$ipmon   Rotas
afetadas - Ordem de GWs - Vpns para resetar nas 2 linhas seguintes";
    :put "$rotas";
    :put $gwslist;
```

```
        :put "$vpnreset";
}
:put "Tempo de execucao"
:put $inicio;
:put [/system clock get time];
# Volta a schedule. Vai executar 1 minuto a frente
/system scheduler set [find name=LinkFailover] disabled=no
---------- Fim do script ----------------------------------------------------------------------
```

Anexo F – Artigo

# Failover de Links com roteador Mikrotik

**Ricardo Gazola**

Instituto de Informática e estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88040-900 – Florianópolis – SC – Brasil

gazolacobain@gmail.com

***Resumo.*** *Failover de links é um processo de restabelecimento de um serviço quando há falhas, neste artigo apresentarei uma forma de automatização do processo de troca de links, através de scripts, utilizando roteadores Mikrotik. Este processo foi baseado em decorrentes problemas que afetavam a estrutura de redes da empresa, tal processo de restabelecimento de serviços da empresa era feito de forma manual, após a implementação deste trabalho, todo o processo de restabelecimento de links foi automatizado.*

***Abstract.*** *Failover links is a process of restoring service when there are failures, in this article I will present a way to automate the link exchange process through scripts using Mikrotik routers. This process was based on arising problems affecting the structure of enterprise networks, this process of restoring the company's services was done manually, after the implementation of this work, the whole process of re-establishment of links was automated.*

## Introdução

A complexidade das redes de computadores e a grande demanda por serviços ininterruptos faz com que as empresas pensem cada vez mais em como manter toda a estrutura funcionando o maior tempo possível. Tecnologias de alta disponibilidade geralmente são caras, demandam altos investimentos e requerem um grande esforço e conhecimento. Observando a estrutura da empresa percebi alguns problemas referentes à queda nos links de comunicação, estas quedas impactam em diversos pontos: suporte a clientes, impacto nos serviços realizados pelos próprios colaboradores, acesso a diversos sistemas, os quais ficavam indisponíveis por um grande período tempo após as quedas nos links, etc.

Após a análise da estrutura e das tecnologias empregadas na empresa iniciei um processo de automatização do processo de troca de links e restabelecimento dos serviços da empresa, utilizando como ferramenta os roteadores e a própria linguagem de script do sistema Mikrotik. Para esta tarefa, algumas restrições foram impostas: utilizar as ferramentas já empregadas na empresa, não alterar a estrutura de redes e a não aquisição de novos equipamentos.

O sistema Mikrotik fornece grande flexibilidade nas operações de roteamento de redes, disponibilidade e segurança. Esta flexibilidade aliada com as ferramentas providas pelo roteador será usada para criar um ambiente de links de acesso à internet redundante chamado de failover de links, o que proporcionará um ambiente com maior disponibilidade e menor tempo de resposta a um evento de queda.

## Roteadores Mikrotik

O ambiente corporativo exige flexibilidade e ao mesmo tempo segurança. O fator flexibilidade e a forma como o roteador trabalha e se comporta foram pontos importantes em tomar a decisão de usar o sistema operacional Mikrotik, que também é nome da empresa que fornece o produto, para a implementação deste trabalho.

Mikrotik é um sistema operacional para servidores e roteadores "routerboard", que são equipamentos oferecidos já com o sistema operacional instalado. O Mikrotik pode ser instalado em um desktop, por exemplo, ou em uma máquina virtual, que foi o aplicado neste trabalho pelo fato de trazer uma alta flexibilidade e redundância do sistema como um todo. O Mikrotik é um sistema completo, com todas as ferramentas necessárias para roteamento dinâmico ou estático, para criação de VPNs de qualquer tipo, possui firewall completo com todas as funcionalidades necessárias para marcação de pacotes, bloqueios e controle de acesso.

A principal característica destes roteadores, além da sua ampla flexibilidade e uso, é a possibilidade de criação de scripts utilizando a própria linguagem do sistema operacional. O sistema vem com uma console na qual é possível executar inúmeros comandos, todos estes comandos possuem uma estrutura própria da linguagem do sistema que precisaram ser estudados para este trabalho, apesar de várias das estruturas serem parecidas com linguagem de scripting bash, por exemplo.

## Estrutura da rede

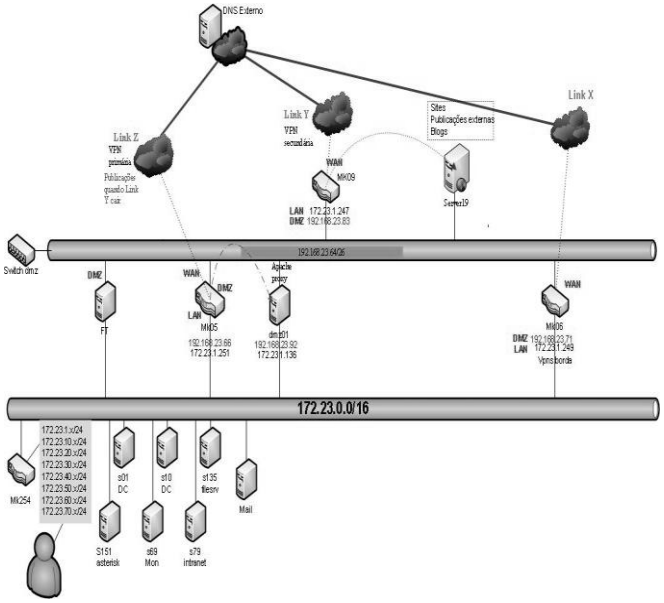


**Figura 1. Mapa da rede**

A estrutura da rede é baseada em quatro roteadores de acordo com o mostrado na Figura 1, sendo três deles alimentados com scripts de failover para este trabalho. Cada um dos três roteadores possui um link de comunicação diferente, com o objetivo de criar alta disponibilidade e separar os serviços da empresa da seguinte forma:

MK254, gateway da rede local, toda rede local da empresa passa primeiro por este roteador, o qual por ser apenas local e não ter link com internet não possui script de failover;

MK05, roteador que possui o link principal para navegação na internet, e possui um serviço de VPN, para que os usuários da empresa possam trabalhar remotamente;

MK06, roteador que possui a grande maioria das VPNs de acesso aos clientes, usados para prestar suporte aos clientes;

MK09, roteador que possui todas as publicações externas da empresa, como site, e-mail, e alguns sistemas que precisam ser acessados publicamente, este roteador é o principal roteador da empresa por tratar da grande maioria dos acessos externos há

empresa.

## Objetivos dos scripts

A escolha na utilização dos scripts se deu por causa da maior possibilidade de realização das tarefas, em especial as tarefas relacionadas aos testes de conectividade, estes que precisavam ter uma flexibilidade maior, pelo fato de ser necessário testar as diversas formas de monitoramento, alterar parâmetros de TTL, delay e ping, parâmetros estes que o roteador não fornecia na forma como precisava.

Com o uso de scripts, é possível criar um leque de possibilidades maior, pelo fato da linguagem usada nos scripts ser a própria linguagem do sistema operacional do roteador. A estrutura dos scripts não é trivial, por se tratar de uma linguagem própria foi necessário o estudo de toda sua estrutura, apesar da semelhança com sistemas operacionais Unix, a forma de utilizá-los difere em vários pontos.

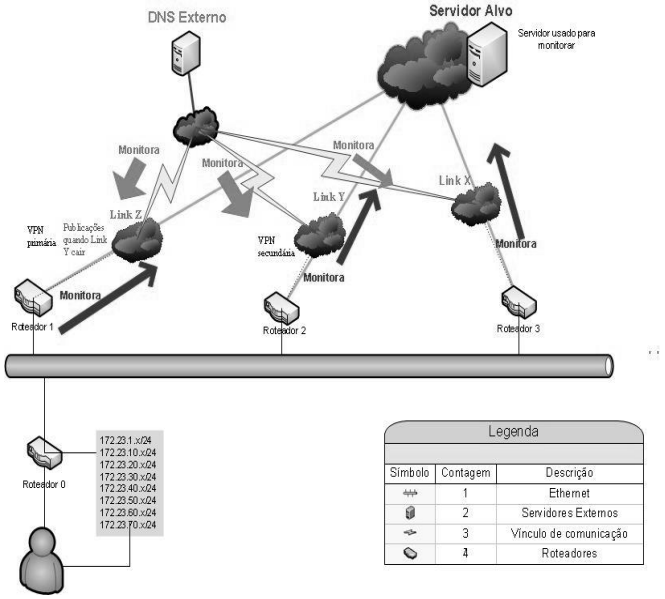


**Figura 2. Mapa dos scripts**

A principal tarefa do script é a verificação da conectividade, toda a estrutura dos scripts foi pensada objetivando a melhor forma de monitorar os links de cada roteador. Para que os scripts pudessem ser executados de forma confiável, era necessário que eles se baseassem em um servidor externo na internet (servidor alvo Figura 2) que estivesse sempre disponível. Esta parte do processo empregado no script não possui total eficácia, pois não é possível garantir que o servidor alvo não fique indisponível, ele depende de outros fatores que ficam de fora do escopo do projeto, para tentar garantir o máximo de eficiência possível é necessário constante monitoramento, qualquer eventual queda neste servidor utilizado implicará em um falso positivo nos roteadores, o que acarretará na ativação do script e troca do link.

Após determinar qual servidor alvo utilizar, iniciamos o planejamento e realização dos testes com parâmetros de monitoramento. Para monitoramentos iniciais utilizei apenas o utilitário ping de acordo com exemplo abaixo:

```
:if ([/ping 8.8.8.8 src-address=172.23.1.249 count=3]=0) do={
/tool e-mail send to=monitor@empresa subject="Queda de Link";
} else {
:log info “Link UP";
}
```

No exemplo acima a verificação é feita de forma básica: é realizado um ping para o

IP da Google, através do IP do roteador, “src-address”, ou seja, endereço fonte é o endereço do roteador, e o servidor alvo é o endereço da Google, assim se este ping retornasse 0, que é o equivalente a não receber resposta alguma, um e-mail era enviado informando queda de link, este procedimento básico possibilitou diversos testes iniciais, assim percebemos que apenas este parâmetro, “count=3” não era necessário, três pings são insuficientes para identificar uma queda, muitas vezes o link pode estar apenas oscilando ou com intermitência, nestes casos não é necessário trocar toda a estrutura.
O script utiliza o protocolo de redes ICMP com o utilitário ping em um servidor externo confiável (ou o mais confiável possível), assim, a cada intervalo de tempo o script pinga através de cada um dos links de conexão para monitorar, ou seja, o script define como rota default do roteador o gateway de um dos links e define o servidor externo alvo que irá monitorar.

Como processo adicional, criei um script de DNS em um servidor hospedado externamente à rede da empresa, com o objetivo de monitorar os links de comunicação da empresa no sentido internet-empresa, e quando ocorrer a queda de algum dos links se possa trocar a chamada de DNS das publicações para que não dependêssemos mais de apenas um IP amarrado a um link e sim uma lista de IPs, e consequentemente uma lista de links.
O script possui um objetivo claro e simples, monitorar a cada determinado intervalo de tempo os links de comunicação da empresa (este intervalo de tempo é relativo, pois pode ser trocado quando forem detectados falsos positivos ou se acharmos que o tempo de monitoramento estiver errado), para que quando haja alguma queda não apenas os serviços internos da empresa continuem com acesso, mas como também as publicações externas e demais serviços que dependam de um link de comunicação especifico, ou seja, que possuem um endereço externo fixo.

## Conclusão

O objetivo principal do trabalho era à busca de uma solução para automatizar a troca de links quando houvesse problemas de conectividade. Para alcançar este objetivo, seguindo as restrições impostas pela empresa e utilizando as tecnologias já empregadas na estrutura de redes, foi realizado um estudo avançado em roteadores Mikrotik, para descobrir uma forma automatizada de realizar o failover de links de internet e manter a estrutura redundante.

Como os recursos eram limitados e os materiais técnicos específicos eram escassos para o assunto, a solução foi utilizar scripts para automatizar da melhor maneira possível. Como o trabalho focou em automatizar o processo de troca de links, deixei para trabalhos futuros outras complementações que eram também muito importantes, como a priorização do tráfego e a distribuição de banda de forma igualitária para usuários da rede.

Ao fim do trabalho foi possível perceber uma grande melhora nas atividades de suporte, tanto para equipe interna, quanto para suporte aos clientes, assim como um aumento da disponibilidade dos serviços oferecidos pela empresa. Outro ponto importante do trabalho foi o aprendizado que proporcionou, sendo possível aplicar alguns dos principais fundamentos de rede e a utilização de diversos protocolos, como ICMP, DNS, IP entre outros.

## Referencias

BARION, Rogério. Mikrotik RouterOS: guia prático. Porto Alegre: Sulina, (2011).
MIKROTIK. Mikrotik RouterOS. Disponível em:
<http://wiki.Mikrotik.com/wiki/Manual:Scripting>. Acesso em: 29 nov. 2013.
MIKROTIK. Mikrotik RouterOS. Disponível em: <http://www.Mikrotik.com>.

Acesso
em: 29 nov. 2013.