

UNIVERSIDADE LUTERANA DO BRASIL

ULBRA – *CAMPUS* GUAÍBA

CURSO DE SISTEMAS DE INFORMAÇÃO



**OTIMIZAÇÃO WAN – ACELERADORES:
TRABALHO DE CONCLUSÃO DE CURSO I**

ANDERSON SILVA PETERSEN

André Peres
Orientador

Guaíba, dezembro de 2009.

DADOS DE IDENTIFICAÇÃO

Acadêmico: Anderson Silva Petersen

E-mail: aspetersen@gmail.com

Professor Orientador: André Peres

E-mail: peres@guaiba.ulbra.tche.br

Título do Projeto: Otimização WAN - Aceleradores

Período de realização: Agosto de 2009 à Dezembro de 2009

SUMÁRIO

1	DEFINIÇÃO DO TEMA	5
1.1	Tema.....	5
1.2	Delimitação do Tema	5
2	PROBLEMA DE PESQUISA	5
3	HIPÓTESES DE SOLUÇÃO	6
4	OBJETIVOS	6
5	JUSTIFICATIVA	6
6	FUNDAMENTAÇÃO TEÓRICA	7
6.1	TCP	7
6.1.1	<i>Características técnicas</i>	8
6.1.2	<i>Funcionamento</i>	9
6.2	UDP	10
6.2.1	<i>Cabeçalho UDP</i>	11
6.3	IP	11
6.4	WAN	12
6.4.1	<i>Protocolos WAN</i>	13
6.4.2	<i>Segurança em WAN</i>	15
6.4.3	<i>Otimização de WAN</i>	23
6.5	ACELERADORES WAN	30
6.5.1	<i>WAAS</i>	30
6.5.2	<i>Vtun</i>	34
7	METODOLOGIA	36
7.1	PLANEJANDO A IMPLEMENTAÇÃO DA SOLUÇÃO VTUN	36
7.1.1	<i>Pré-requisitos</i>	36
7.1.2	<i>Instalando o Vtun</i>	37
7.1.3	<i>Configurando o Vtun</i>	38
7.1.4	<i>Ambiente a ser implementado</i>	41
7.2	PLANEJANDO A IMPLEMENTAÇÃO DA SOLUÇÃO WAAS	41
7.2.1	<i>Requisitos necessários</i>	43
7.2.2	<i>Ambiente a ser implementado</i>	43
7.2.3	<i>Monitorando e Configurando o WAAS</i>	44

7.3 TESTES A SEREM REALIZADOS.....	52
8 RESULTADOS	53
9 TRABALHO DE CONCLUSÃO DE CURSO - II.....	53
10 REFERÊNCIAS	53

1 DEFINIÇÃO DO TEMA

1.1 Tema

O trabalho está sendo desenvolvido na área de redes, visando mais especificamente à questão de otimização WAN, trazendo o conceito de aceleração na comunicação entre duas redes geograficamente distribuídas. No início serão abordados conceitos básicos de rede, segurança, alguns breves comentários sobre as ferramentas já utilizadas para o melhor aproveitamento dos recursos de rede e o estudo de duas tecnologias diferentes que possuem a mesma finalidade.

1.2 Delimitação do Tema

Este trabalho visa conhecer e descrever duas tecnologias diferentes de aceleração WAN. Uma das tecnologias será a WAAS (*Wide Area Application Services*) criada pela Cisco Systems e a outra será uma solução Linux chamada Vtun, utilizada para estabelecer comunicação entre duas redes geograficamente distribuídas.

2 PROBLEMA DE PESQUISA

A baixa performance dos meios de comunicação, aliada ao alto custo das mesmas, dificultam a implementação de novos sistemas nas grandes empresas que atuam em localidades diferentes. Com *links* mais rápidos é possível o desenvolvimento de novas ferramentas, a centralização de mecanismos de backup, servidores e profissionais de tecnologia da informação. Tudo isso obtendo um melhor desempenho, favorecendo principalmente os usuários finais.

Uma forma de tornar mais rápida essa comunicação é utilizando soluções capazes de acelerar a transmissão de dados entre as redes. O conceito de acelerador WAN pode prover esta maior velocidade na comunicação, sendo esta solução, comercializada por várias empresas de tecnologia e tendo disponíveis também algumas soluções gratuitas.

A maior dificuldade encontrada no desenvolvimento deste trabalho é a escassez de documentações sobre a tecnologia, bem como a dificuldade de encontrar casos que nos permitam definir qual a melhor solução a ser implantada.

3 HIPÓTESES DE SOLUÇÃO

Após a pesquisa inicial sobre os problemas ocasionados por *links* de baixa qualidade e a possibilidade de implementar aceleradores de comunicação entre redes distantes geograficamente, serão estudadas duas alternativas diferentes que podem prover esta solução:

- Solução paga (requer investimento financeiro): será abordado o conceito desenvolvido pela Cisco Systems denominada WAAS (*Wide Area Application Services*);

- Solução gratuita: será utilizada a ferramenta Vtun, sendo executada sobre uma distribuição Linux.

4 OBJETIVOS

Criar uma documentação detalhada sobre aceleradores WAN, capaz de auxiliar um profissional da área de TI a escolher uma das alternativas, bem como instruí-lo a implementar qualquer uma das duas tecnologias descritas neste trabalho.

5 JUSTIFICATIVA

O mercado de desenvolvimento de softwares avança a cada dia que passa, desenvolvendo sistemas cada vez mais complexos e mais exigentes, tudo para atender aos requisitos expostos pelos usuários. Para os profissionais de infraestrutura não é diferente, pois os usuários necessitam ter cada vez mais performance, tanto das suas estações de trabalho como na transmissão de arquivos pela rede. Quando se trata de uma empresa de grande porte, que possui filiais espalhadas por regiões geográficas distantes, é necessário estabelecer uma comunicação entre estas filiais, para que os dados possam ser compartilhados, tornando-os comuns e acessíveis a qualquer funcionário, independente de sua região. Cabe a estes profissionais de infraestrutura buscar alternativas para suportar

a execução destes novos sistemas, bem como interligar todos os escritórios remotos.

Existe uma diversidade muito grande de tecnologias, porém todas envolvem um alto valor de implementação e mensalidades custosas. Quanto maior a banda contratada com a operadora de telecomunicação, maior o valor da mensalidade. Cria-se então mais um desafio para os profissionais de TI: estabelecer uma conexão segura entre filiais geograficamente distantes, proporcionando uma comunicação rápida e sem custos elevados.

6 FUNDAMENTAÇÃO TEÓRICA

6.1 TCP

O TCP (*Transmission Control Protocol*) é um dos protocolos sob os quais assenta o núcleo da Internet. O TCP possui controle de erros, verificando se os dados são enviados de forma correta, na sequência apropriada e sem erros. Este protocolo é do nível da camada de transporte do Modelo OSI. Podemos citar como principais portas as seguintes:

- HTTP (*HyperText Transfer Protocol Secure*): é o principal protocolo da Internet, usado para acesso às páginas *web*. Por padrão utiliza a porta 80.
- HTTPS: permite transmitir dados de forma segura, encriptados usando o SSL. Ele é usado por Bancos (Financeiras) e todo tipo de *site* de comércio eletrônico ou que armazene informações confidenciais. Utiliza a porta 443.
- FTP (*File Transfer Protocol*): é um dos protocolos de transferência de arquivos mais antigos e ainda assim um dos mais usados. O ponto fraco do FTP é a questão da segurança: todas as informações, incluindo as senhas trafegam em texto puro e podem ser capturadas por qualquer um que tenha acesso à transmissão. Utiliza normalmente a porta 21.
- SSH (*Secure Shell*): é a chave mestra da administração remota em servidores Linux. Inicialmente o SSH permitia executar apenas comandos de texto remotamente; depois passou a permitir executar também aplicativos gráficos e, em seguida, ganhou também um módulo para transferência de arquivos, o SFTP. A vantagem do SSH sobre o Telnet e o FTP é que tudo é feito através

de um canal encriptado, com uma excelente segurança. O SSH pode ser usado também para encapsular outros protocolos, criando um túnel seguro para a passagem dos dados. Criando túneis, é possível acessar servidores de FTP, *proxy*, e-mail, *rsync*, etc. de forma segura.

6.1.1 Características técnicas

Podemos citar como principais características do protocolo TCP:

- Orientado à conexão – primeiramente a aplicação envia um pedido de conexão para o destino e usa a "conexão" para transferir dados;
- Ponto a ponto - uma conexão TCP é estabelecida entre dois pontos;
- Confiabilidade - o TCP usa várias técnicas para proporcionar uma de dados confiável. O TCP permite recuperar pacotes perdidos, eliminar pacotes duplicados, recuperar dados corrompidos, e estabelecer novamente a ligação em caso de problemas no sistema e na rede.
- *Full duplex* - transferência simultânea em ambas as direções (cliente-servidor).
- *Handshake* – Utilizado para estabelecer e finalizar uma conexão a três e quatro tempos, o que permite a autenticação e encerramento de uma sessão completa. O TCP garante que todos os pacotes foram recebidos com sucesso.
- Entrega ordenada – o TCP garante a entrega ordenada dos pacotes através de números de sequência, mesmo que os pacotes fiquem desordenados ao longo do trajeto, pelo protocolo de encaminhamento.
- Controle de fluxo - O TCP usa o campo janela (ou *window*) para fazer o controle do fluxo. O receptor, à medida que recebe os dados, envia mensagens ACK confirmando o recebimento de um segmento. Estas mensagens podem especificar o tamanho máximo do *buffer* no campo (janela) do segmento TCP, determinando a quantidade máxima de *bytes* que o receptor aceitará. O transmissor pode transmitir segmentos com um número de *bytes* que deverá estar confinado ao tamanho da janela permitido.

6.1.2 Funcionamento

Conforme Tanenbaum (2003), o protocolo TCP possui três fases durante uma conexão: estabelecimento da ligação, transferência e término de ligação.

- Estabelecimento da ligação:** Em uma ligação TCP existe o 'cliente' e o 'servidor'. O cliente inicia a ligação enviando um pacote com a *flag* SYN ativa e espera que o servidor aceite a ligação enviando um pacote SYN+ACK. Se esse pacote não for recebido no tempo esperado ocorre um *timeout* (tempo esgotado) e o cliente reenvia o SYN. O estabelecimento da ligação é concluído por parte do cliente quando o mesmo responde com um pacote ACK. Durante estas trocas, são trocados números de sequência iniciais entre os interlocutores que irão servir para identificar os dados ao longo do fluxo, bem como servir de contador de *bytes* transmitidos durante a fase de transferência de dados. No final desta fase, o servidor registra o cliente como uma ligação estabelecida numa tabela própria que contém um limite de conexões, o *backlog*. Se o *backlog* fica completamente preenchido, o servidor rejeita as ligações, ignorando todos os pacotes SYN's subsequentes. O exemplo prático de como funciona o estabelecimento desta conexão pode ser visto na Figura 1 que segue logo abaixo.

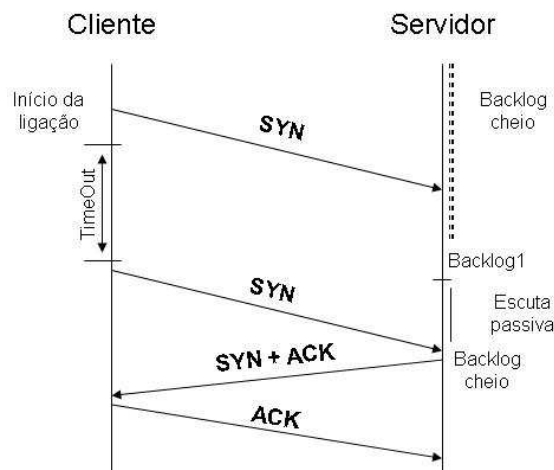


Figura 1 – Exemplo de conexão TCP

- Transferência de dados (sessão):** O TCP possui mecanismos que garantem a transmissão dos dados. Conforme o cabeçalho TCP, existe um par de números, denominados como *número de sequência* e *número de*

confirmação. O emissor determina o seu próprio número de sequência e o receptor confirma o segmento usando como número ACK o número de sequência do emissor. O receptor confirma os segmentos indicando que recebeu um determinado número de *bytes* contíguos. Uma das melhorias introduzidas no TCP foi a possibilidade do receptor confirmar blocos fora da ordem esperada. Esta característica é chamada SACK, ou *selective ACK*.

- **Adequação de parâmetros:** O cabeçalho TCP possui um parâmetro que permite indicar o espaço livre atual do receptor. Assim, o emissor fica sabendo que só poderá ter em trânsito aquela quantidade de informação até esperar o ACK de um dos pacotes.
- **Término da ligação:** o encerramento da conexão possui 4 fases, em que cada extremo se responsabiliza pelo encerramento da sua ligação. Quando um deles pretende finalizar a sessão, envia um pacote com a *flag* FIN ativa, ao qual deverá receber uma resposta ACK. O outro extremo irá proceder da mesma forma. Pode ocorrer, no entanto, que um dos lados não encerre a sessão. Este evento chama-se conexão *semiaberta*. O lado que não encerrou a sessão poderá continuar a enviar informação pela conexão, mas o outro lado não. Um exemplo de como isso ocorre pode ser visto na figura 2.

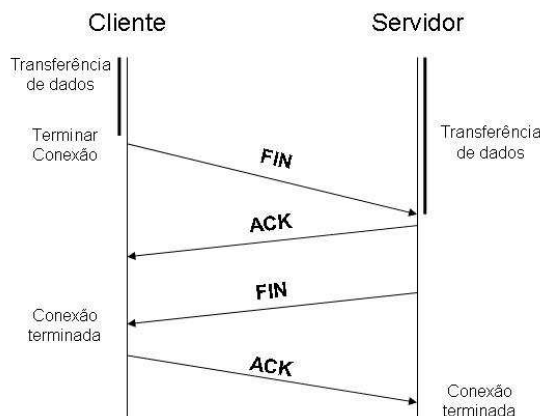


Figura 2 - Término de Conexão

6.2 UDP

O *User Datagram Protocol* (UDP) pertence à camada de transporte. Ele permite que a aplicação escreva um datagrama encapsulado em pacotes IPv4 e IPv6 para que seja enviado ao destino. O protocolo UDP torna-se não confiável por

não existir garantia de chegada do pacote ao destino. *Timeouts*, retransmissões, *acknowledgments*, controle de fluxo, etc., podem ser aplicados para tornar o UDP mais confiável.

O UDP, ao contrário do TCP, não utiliza serviço de conexão, pois não mantém um relacionamento longo entre o cliente e o servidor. O UDP trabalha criando *socket* que permite enviar um datagrama para um servidor e imediatamente enviar outro datagrama com o mesmo *socket* para outro servidor. Da mesma forma, um servidor poderia ler datagramas vindos de clientes diferentes, usando o mesmo *socket*.

A grande diferença entre o UDP e o TCP é o fato do TCP possuir mecanismos para controle de erros, ao contrário do UDP (utilizado geralmente em sistemas de áudio e vídeo conferência). Podemos citar como exemplo de portas UDP: DNS (*Domain Name Server*) e NTP (*Network Time Protocol*).

6.2.1 Cabeçalho UDP

O cabeçalho UDP é muito simples, contendo apenas os números de porta, comprimento da mensagem e o *checksum*. A figura 3 que pode ser vista logo abaixo ilustra o tema desta seção.

Porta Origem	Porta Destino
Comprimento da Mensagem	Checksum

Figura 3 – Cabeçalho UDP

Os campos em amarelo são opcionais. A porta de origem geralmente especifica a porta desejada de resposta, mas pode ser omitida. Isso tipicamente ocorre em comunicações *broadcast* ou mensagens de pânico, que notificam sobre a queda de um equipamento.

6.3 IP

IP ("*Internet Protocol*" ou Protocolo de Internet) é um protocolo usado entre duas ou mais máquinas em rede para encaminhamento dos dados.

Os dados são enviados em blocos como pacotes ou datagramas. No IP não é necessária nenhuma definição antes do *host* tentar enviar pacotes para um *host* com o qual não comunicou previamente.

O IP oferece um serviço de datagramas não confiável (também chamado de *melhor esforço*), onde o pacote é transportado quase sem garantias, podendo chegar ao destino desordenado, duplicado ou até mesmo não chegar. Os roteadores são usados para reencaminhar datagramas IP através das redes interconectadas na segunda camada. A falta de qualquer garantia de entrega proporciona maior velocidade. A versão 4 (ou IPv4) possui endereçamento de 32 *bits* e a versão 6 (ou IPv6) oferece um endereçamento de 128 *bits*.

6.4 WAN

A WAN (*Wide Area Network*) é a união estabelecida entre duas ou mais LAN's (*Local Area Network*). A Internet é um exemplo de WAN. Em geral, as redes geograficamente distribuídas contêm conjuntos de servidores, que formam sub-redes. Na maioria das WAN's, a sub-rede consiste em dois componentes distintos: linhas de transmissão e elementos de comutação. As linhas de transmissão (circuitos, canais ou troncos), transportam os bits entre os computadores da LAN. Os elementos de comutação são equipamentos usados para conectar duas ou mais linhas de transmissão. Os dispositivos de comutação são os roteadores, que estabelecem comunicação entre si através de um *link*. Um exemplo desta estrutura, conforme Tanenbaum (2003) pode ser vista na figura 4.

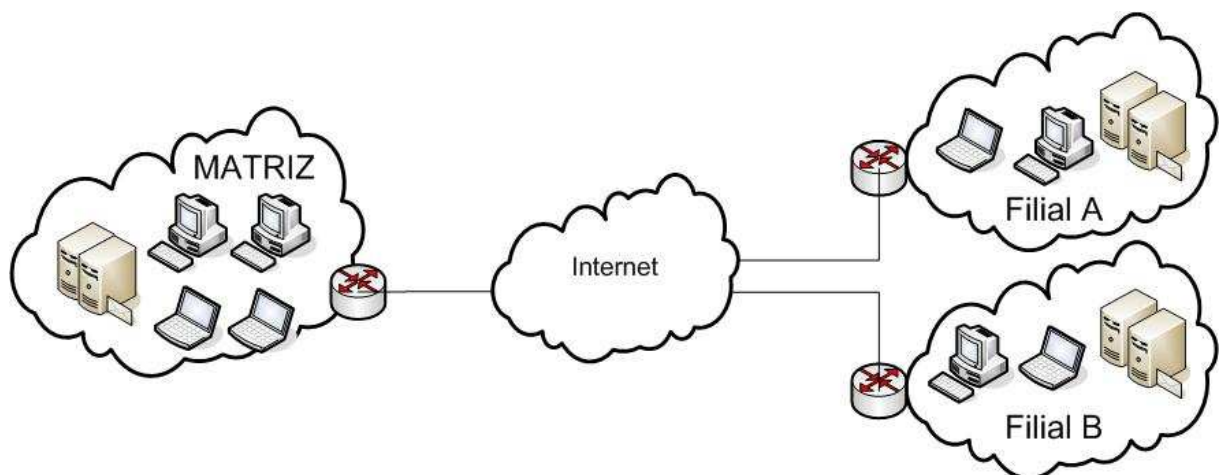


Figura 4 – Exemplo de WAN

6.4.1 Protocolos WAN

Possibilitam a transmissão de dados de uma rede fisicamente distante através de uma infraestrutura de canais de dados de longa distância. São estes protocolos:

- PPP Protocolo ponto-a-ponto (*Point-to-Point Protocol*): protocolo mais comum para acesso à internet. É utilizado tanto em conexões discadas como em conexões dedicadas. Este seja talvez o mais usado atualmente no nível da camada de enlace de dados, tendo em vista que toda a Internet é baseada nele para poder fazer os equipamentos se comunicarem. Ele trabalha também na conexão que há entre sua casa e o seu provedor de acesso à Internet. Basicamente, ele tem recursos que permitem a detecção de erros de transmissão: um subprotocolo chamado LCP (*Link Control Protocol* - Protocolo de Controle de Enlace) que é usado para ativar e desativar linhas de transmissão, testá-las e negociar opções de funcionamento e um subprotocolo chamado NCP (*Network Control Protocol* - Protocolo de Controle de Rede) que é usado para fazer com que a camada de enlace de dados "converse" com a camada de rede. Quanto ao NCP, existem vários tipos dele, cada um sendo usado para trabalhar em conjunto com um determinado protocolo correspondente na camada de rede.
- *Frame Relay*: é uma arquitetura de rede de pacotes de alta velocidade. O *Frame Relay* permite vários tipos de serviço, até altas velocidades de comunicação entre nós da rede, por exemplo, DS3 (45 Mbps). Com a evolução e uso de meios de transmissão confiáveis (por exemplo, cabos óticos), viabilizou a comunicação entre redes locais (LAN) e é um serviço oferecido comumente pelas operadoras.
- Rede ATM (*Asynchronous Transfer Mode*): é uma tecnologia de rede usada para WAN (e também para *backbones* de LAN), suportando a transmissão em tempo real de dados de voz e vídeo. A topologia típica da rede ATM utiliza-se de *switches* que estabelecem um circuito lógico entre o computador de origem e destino, deste modo garantindo alta qualidade de serviço e uma taxa de erros baixa. Diferentemente de uma central telefônica, a rede ATM permite que a banda excedente do circuito lógico estabelecido seja usada por outras aplicações. A tecnologia de transmissão e comutação de dados utiliza a

comutação de células como método básico de transmissão, uma variação da comutação de pacotes onde o pacote possui um tamanho reduzido.

- MPLS (*Multiprotocol Label Switching*): proporciona o encaminhamento e a comutação eficientes de fluxos de tráfegos através da rede. A informação em uma rede MPLS é processada e dividida em classes de serviço, e os dados são encaminhados através de rotas estabelecidas anteriormente por essas classes, sendo feito apenas comutação. O MPLS é uma tecnologia utilizada em *backbones*, e tem o objetivo de aumentar a velocidade nas redes, melhorar a escalabilidade, o gerenciamento do QoS e a engenharia de tráfego. A aplicação mais interessante do MPLS consiste na sua utilização em conjunto com o IP. Esta junção possibilita a interoperabilidade entre o roteamento de pacotes e a comutação de circuitos.
- DSL Linha Digital de Assinante (*Digital Subscriber Line*) XDSL: Permite tráfego de alta capacidade usando o cabo telefônico normal entre a casa ou escritório do assinante e a central telefônica. Possui dois modos básicos:
 - ADSL DSL Assimétrico (*Asymmetric DSL*): ADSL compartilha uma linha de telefone comum, usando uma faixa de frequência de transmissão acima daquelas usadas para a transmissão de voz. É uma variação do protocolo DSL onde a capacidade de transmissão é assimétrica, isto é, a banda do assinante é projetada para receber maior volume de dados do que este pode enviar. Serviço mais adequado ao usuário comum que recebe dados da Internet (usuários domésticos).
 - HDSL DSL (*High-Bit-Rate DSL*): O HDSL fornece um enlace de alta taxa de transmissão de dados, sobre o par trançado comum, exigindo a instalação de pontes e repetidores. É uma variação do protocolo DSL onde a banda do assinante tem a mesma capacidade de envio e recebimento de dados. Serviço mais adequado ao usuário corporativo que disponibiliza dados para outros usuários comuns (pequenas empresas).

6.4.2 Segurança em WAN

Ao pensar em segurança de redes de longa distância, é preciso que se tenha em mente que a segurança na transmissão de dados é necessária e exige certos cuidados. Pela WAN de uma empresa trafegam as mais confidenciais e importantes informações. Na internet, milhares de pessoas navegam e nem todos são bem intencionados. Nesse contexto todos precisam tomar atitudes para aumentar o grau de confiabilidade de sua conexão. Como exemplo pode ser citada a comunicação por e-mail. Embora pareça que tal comunicação é altamente segura, um e-mail pode ser capturado, lido por outros, destruído ou até sofrer modificações de conteúdo. Outro ponto importante é a questão de utilização de senha de acesso, pois é comum que os usuários não dispensem muita atenção a isso. O quesito segurança visa assegurar a integridade, confidencialidade, disponibilidade e autenticidade das informações. Dentre os recursos mais utilizados pode-se citar: IDS (*Intrusion Detection System*), *firewall*, criptografia, PKI (*Public Key Infrastructure*), VPN (*Virtual Private Network*).

6.4.2.1 VPN (*Virtual private network*)

O objetivo principal da VPN (*Virtual Private Network*) é criar um canal de comunicação seguro entre dois pontos de rede, utilizando como meio uma rede pública (por exemplo: a Internet), garantindo a confidencialidade e autenticidades dos dados que por ela trafegam. A figura 5 ilustra um exemplo de VPN.

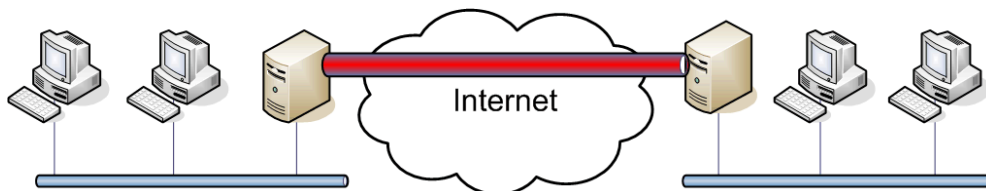


Figura 5 - VPN

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

A VPN pode representar uma redução de custos considerável, pois, por exemplo, elimina a necessidade de contratação de *links* dedicados para a conexão de filiais de uma empresa.

Dentre as aplicações de uma VPN, podem-se citar como principais as duas que seguem abaixo:

- Acesso remoto via Internet: o acesso remoto a redes corporativas através da Internet pode ser viabilizado com a VPN através da ligação local a algum provedor de acesso (*Internet Service Provider* - ISP). A estação remota disca para o provedor de acesso, conectando-se à Internet e o *software* de VPN cria uma rede virtual privada entre o usuário remoto e o servidor de VPN corporativo através da Internet. É uma solução muito útil para que os funcionários de uma empresa possam desempenhar suas funções sem estar no escritório. A figura 6 demonstra de forma mais clara como é estabelecida esta conexão VPN.

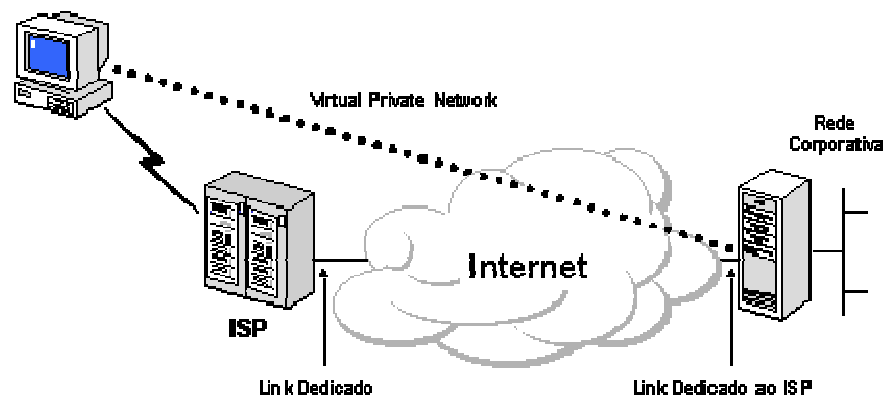


Figura 6 – Acesso remoto via Internet

- Conexão de LAN's via rede pública: uma solução que substitui as conexões entre LAN's através de circuitos dedicados de longa distância é a utilização de circuitos dedicados locais interligando-as à Internet. O *software* de VPN assegura esta interconexão formando a WAN corporativa. Desta forma, filiais podem trabalhar como se estivessem dentro da rede da matriz da empresa. Abaixo segue a figura 7 que exemplifica esta conexão.

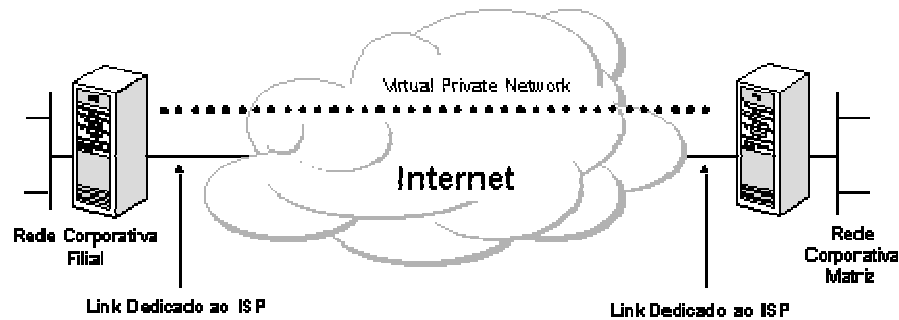


Figura 7 – Conexão de LAN via rede pública

Em qualquer modelo de conexão VPN, o quesito segurança deve ser valorizado. Abaixo seguem os requisitos para garantir uma rede virtual privada segura:

Autenticação de Usuários: Verificação da identidade do usuário, restringindo o acesso às pessoas autorizadas. Deve dispor de mecanismos de auditoria, provendo informações referentes aos acessos efetuados: quem acessou, o quê e quando foi acessado.

Gerenciamento de Endereço: O endereço do cliente na sua rede privada não deve ser divulgado, devendo-se adotar endereços fictícios para o tráfego externo.

Criptografia de Dados: Os dados devem trafegar na rede pública ou privada num formato cifrado e, caso sejam interceptados por usuários não autorizados, não deverão ser decodificados, garantindo a privacidade da informação. O reconhecimento do conteúdo das mensagens deve ser exclusivo dos usuários autorizados.

Gerenciamento de Chaves: O uso de chaves que garantem a segurança das mensagens criptografadas deve funcionar como um segredo compartilhado exclusivamente entre as partes envolvidas. O gerenciamento de chaves deve garantir a troca periódica das mesmas, visando manter a comunicação de forma segura.

Suporte a Múltiplos Protocolos: Com a diversidade de protocolos existentes, torna-se bastante desejável que uma VPN suporte protocolos padrão de fato usadas nas redes públicas, tais como IP (*Internet Protocol*), IPX (*Internetwork Packet Exchange*), etc.

As redes virtuais privadas baseiam-se na tecnologia de tunelamento cuja existência é anterior às VPN's. Ele pode ser definido como processo de encapsular um protocolo dentro de outro. O uso do tunelamento nas VPN's 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.

Este processo de tunelamento envolve encapsulamento, transmissão ao longo da rede intermediária e desencapsulamento do pacote. Na figura 8 pode-se analisar o funcionamento do tunelamento.

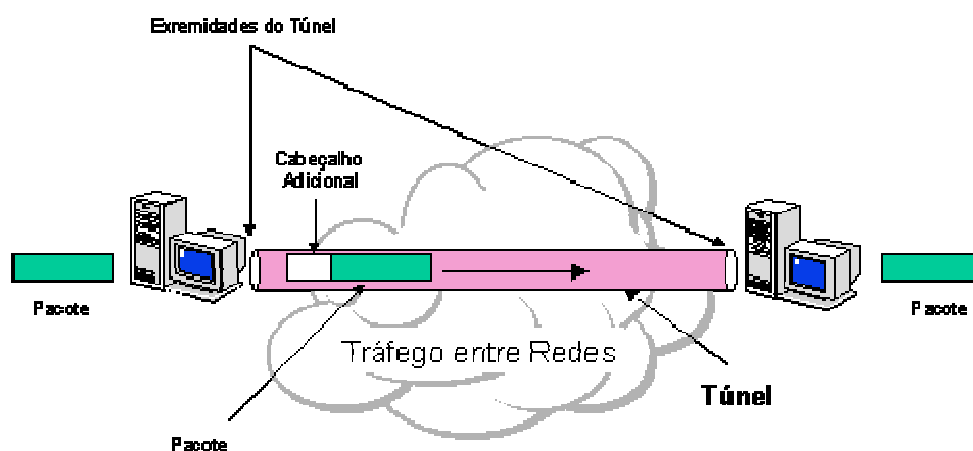


Figura 8 - Tunelamento

Assim como em outros temas, para se estabelecer um túnel é necessário que as suas extremidades utilizem o mesmo protocolo de tunelamento.

O tunelamento pode ocorrer na camada 2 ou 3 (respectivamente enlace e rede) do modelo da tabela OSI.

No tunelamento em nível 2 (enlace), o objetivo é transportar protocolos de nível 3, tais como o IP e IPX na Internet. Os protocolos utilizam quadros como unidade de troca, encapsulando os pacotes da camada 3 (como IP/IPX) em quadros PPP (*Point-to-Point Protocol*). Como exemplos podem ser citados:

- PPTP (*Point-to-Point Tunneling Protocol*) da Microsoft permite que o tráfego IP, IPX e NetBEUI sejam criptografados e encapsulados para serem enviados através de redes IP privadas ou públicas como a Internet.
- L2TP (*Layer 2 Tunneling Protocol*) da IETF (*Internet Engineering Task Force*) permite que o tráfego IP, IPX e NetBEUI sejam criptografados e enviados através de canais de comunicação de datagrama ponto a ponto tais como IP, *Frame Relay* ou ATM.
- L2F (*Layer 2 Forwarding*) da Cisco é utilizada para VPNs discadas.

No tunelamento em nível 3 (rede), os pacotes IP são encapsulados com um cabeçalho adicional deste mesmo protocolo antes de enviá-los através da rede.

- O *IP Security Tunnel Mode* (IPSec) permite que pacotes IP sejam criptografados e encapsulados com cabeçalho adicional deste mesmo protocolo para serem transportados numa rede IP pública ou privada. O IPSec é um protocolo desenvolvido para IPv6, devendo, no futuro, se constituir como padrão para todas as formas de VPN caso o IPv6 venha de fato substituir o IPv4. O IPSec sofreu adaptações possibilitando, também, a sua utilização com o IPv4.

Os protocolos de nível 2, tais como PPTP e L2TP, foram baseados no PPP, e, como consequência, herdaram muito de suas características e funcionalidades. Estas características e suas contrapartes de nível 3 são analisadas juntamente com alguns dos requisitos básicos das VPNs:

- Autenticação de Usuário: Os protocolos de tunelamento da camada 2 herdaram os esquemas de autenticação do PPP e os métodos EAP (*Extensible Authentication*

Protocol). Muitos esquemas de tunelamento da camada 3 assumem que as extremidades do túnel são conhecidas e autenticadas antes mesmo que ele seja estabelecido. Uma exceção é o IPSec que provê a autenticação mútua entre as extremidades do túnel. Na maioria das implementações deste protocolo, a verificação se dá em nível de máquina e não de usuário. Como resultado, qualquer usuário com acesso às máquinas que funcionam como extremidades do túnel podem utilizá-lo. Esta falha de segurança pode ser suprida quando o IPSec é usado junto com um protocolo de camada de enlace como o L2TP.

- Suporte a *Token Card*: Com a utilização do EAP, os protocolos de tunelamento de camada de enlace podem suportar uma variedade de métodos de autenticação, tais como senhas e cartões inteligentes (*smart cards*). Os protocolos de camada 3 também podem usar métodos similares, como, por exemplo, o IPSec que define a autenticação de chave pública durante a negociação de parâmetros feita pelo ISAKMP (*Internet Security Association and Key Management Protocol*).

- Endereçamento dinâmico: O tunelamento na camada 2 suporta alocação dinâmica de endereços baseada nos mecanismos de negociação do NCP (*Network Control Protocol*). Normalmente, esquemas de tunelamento na camada 3 assumem que os endereços foram atribuídos antes da inicialização do túnel.

- Compressão de dados: Os protocolos de tunelamento da camada 2 suportam esquemas de compressão baseados no PPP. O IETF está analisando mecanismos semelhantes, tais como a compressão de IP, para o tunelamento na camada 3.

- Criptografia de dados: Protocolos de tunelamento na camada de enlace suportam mecanismos de criptografia baseados no PPP. Os protocolos de nível 3 também podem usar métodos similares. No caso do IPSec são definidos vários métodos de criptografia de dados que são executados durante o ISAKMP. Algumas implementações do protocolo L2TP utilizam a criptografia provida pelo IPSec para proteger cadeias de dados durante a sua transferência entre as extremidades do túnel.

- Gerenciamento de chaves: O MPPE (*Microsoft Point-to-Point Encryption*), protocolo de nível de enlace, utiliza uma chave gerada durante a autenticação do usuário, atualizando-a periodicamente. O IPSec negocia uma chave comum através do ISAKMP e, também, periodicamente, faz sua atualização.

- Suporte a múltiplos protocolos: O tunelamento na camada de enlace suporta múltiplos protocolos o que facilita o tunelamento de clientes para acesso a redes corporativas utilizando IP, IPX, NetBEUI e outros. Em contraste, os protocolos de tunelamento da camada de rede, tais como o IPSec, suportam apenas redes destino que utilizam o protocolo IP.

6.4.2.2 Criptografia

Trata-se de um conjunto de conceitos e técnicas que visam codificar uma informação de forma que somente o emissor e o receptor possam acessá-la, evitando que um intruso consiga interpretá-la. Para isso, uma série de técnicas são usadas e muitas outras surgem com o passar do tempo.

Conforme Paine e Burnett (2002), na computação, as técnicas mais conhecidas envolvem o conceito de chaves (as chamadas chaves criptográficas). Trata-se de um conjunto de *bits* baseado em um determinado algoritmo capaz de codificar e de decodificar informações. Se o receptor da mensagem usar uma chave incompatível com a chave do emissor, não conseguirá extrair a informação.

Os primeiros métodos criptográficos existentes usavam apenas um algoritmo de codificação. Assim, bastava que o receptor da informação conhecesse esse algoritmo para poder extraí-la. No entanto, se um intruso tivesse posse desse algoritmo, também poderia efetuar um processo de decifragem, caso capturasse os dados criptografados. Há ainda outro problema: imagine que o usuário A tivesse que enviar uma informação criptografada ao usuário B. Esta última teria que conhecer o algoritmo usado. Imagine agora que um usuário C também precisasse receber uma informação do usuário A, porém o usuário C não poderia descobrir qual é a informação a ser enviada ao usuário B. Se o usuário C capturasse a informação enviada ao usuário B, também conseguiria decifrá-la, pois quando o usuário A enviou sua informação, o usuário C também teve que conhecer o algoritmo usado. Para o usuário A evitar esse problema, a única solução seria utilizar um algoritmo diferente para cada receptor.

Com o uso de chaves, um emissor pode usar o mesmo algoritmo (o mesmo método) para vários receptores. Basta que cada um receba uma chave diferente. Além disso, caso um receptor perca ou exponha determinada chave, é possível trocá-la, mantendo-se o mesmo algoritmo.

As chaves podem ser de 64, 128 *bits* e assim por diante. Esses valores expressam o tamanho de uma determinada chave. Quanto mais *bits* forem utilizados, mais segura será a criptografia. Explica-se: caso um algoritmo use chaves de 8 *bits*, por exemplo, apenas 256 chaves poderão ser usadas na decodificação, pois 2 elevado a 8 é 256. Isso deixa claro que 8 *bits* é inseguro, pois até uma pessoa é capaz de gerar as 256 combinações, basta ter tempo. Porém, se forem usados 128 ou mais *bits* para chaves (faça 2 elevado a 128 para ver o que acontece), teremos uma quantidade extremamente grande de combinações, deixando a informação criptografada bem mais segura.

Há dois tipos de chaves criptográficas: chaves simétricas e chaves assimétricas. Ambas são abordadas a seguir:

- Chave simétrica: Esse é um tipo de chave mais simples, onde o emissor e o receptor fazem uso da mesma chave, isto é, uma única chave é usada na codificação e na decodificação da informação. Existem vários algoritmos que usam chaves simétricas, como o DES, o IDEA, e o RC. O uso de chaves simétricas tem algumas desvantagens, fazendo com que sua utilização não seja adequada em situações onde a informação é muito valiosa. Para começar, é necessário usar uma grande quantidade de chaves caso muitas pessoas ou entidades estejam envolvidas. Ainda, há o fato de que tanto o emissor quanto o receptor precisam conhecer a mesma chave. A transmissão dessa chave de um para o outro pode não ser tão segura e cair em "mãos erradas".

- Chave assimétrica: Também conhecida como "chave pública", a chave assimétrica trabalha com duas chaves: uma denominada *privada* e outra denominada *pública*. Neste método, um emissor deve criar uma chave de codificação e enviá-la ao receptor. Essa é a chave pública. Uma outra chave deve ser criada para a decodificação. Esta, a chave privada, é secreta. Por exemplo: a Empresa A criou uma chave pública e a enviou a várias outras empresas. Quando qualquer uma dessas empresas quiser enviar uma informação criptografada à Empresa A deverá utilizar a chave pública desta. Quando o Empresa A receber essa informação, apenas será possível extraí-la com o uso da chave privada, que só a Empresa A tem. Caso a Empresa A queira enviar uma informação criptografada a outro *site*, deverá obter uma chave pública fornecida por este. Entre os algoritmos que usam chaves assimétricas, têm-se o RSA. É um dos algoritmos de chave assimétrica mais

usados. Nele, números primos são utilizados da seguinte forma: dois números primos são multiplicados para se obter um terceiro valor. Porém, descobrir os dois primeiros números a partir do terceiro (ou seja, fazer uma fatoração) é muito trabalhoso. Se dois números primos grandes (realmente grandes) forem usados na multiplicação, será necessário usar muito processamento para descobri-los, tornando essa tarefa praticamente inviável. Basicamente, a chave privada no RSA são os números multiplicados e a chave pública é o valor obtido. Existem ainda outros algoritmos, como o DSA (*Digital Signature Algorithm*), o Schnorr (praticamente usado apenas em assinaturas digitais) e Diffie-Hellman.

Outro recurso muito utilizado com chaves públicas é a certificação digital. Trata-se de um meio que permite, por exemplo, provar que um certo documento eletrônico foi mesmo emitido por uma determinada entidade ou pessoa. O receptor da informação usará a chave pública fornecida pelo emissor para se certificar da origem. Além disso, a chave fica integrada ao documento de forma que qualquer alteração por terceiros imediatamente a invalide.

6.4.3 Otimização de WAN

Existem algumas ferramentas/métodos que auxiliam no desempenho de uma WAN. Logo abaixo serão descritos alguns destes métodos.

6.4.3.1 Proxy

Um *proxy* é um servidor que atende a requisição de conexão, navegação, *download* de arquivos ou qualquer outro recurso disponível em outro servidor. Pode desempenhar a função de *Caching Web Proxy*, que visa manter em uma área de acesso rápido informações já acessadas por algum usuário. Pode ser um *proxy* transparente, que tem a função de obrigar os usuários de uma rede a realizarem os acessos através do *proxy*. Pode também ser um *proxy* aberto, possibilitando que qualquer internauta seja capaz de usar este serviço. Quando a intenção é realizar acessos à Internet ocultando as identificações como computador de origem e *login*, pode ser usado o *proxy* anônimo.

6.4.3.2 QoS (*Quality of Service*)

Normalmente a Internet trabalha com a filosofia do melhor esforço: cada usuário compartilha largura de banda com outros e, portanto, a transmissão de seus

dados concorre com as transmissões dos demais usuários. Os dados empacotados são encaminhados da melhor forma possível, conforme as rotas e banda disponíveis. Quando há congestionamento, os pacotes são descartados sem distinção. Não há garantia de que o serviço será realizado com sucesso. Entretanto, aplicações como voz sobre IP e videoconferência necessitam de tais garantias.

Com a implantação de qualidade de serviço (*quality of service* – QoS), é possível oferecer maior garantia e segurança para aplicações avançadas, uma vez que o tráfego destas aplicações passa a ter prioridade em relação a aplicações tradicionais.

Com uso de QoS os pacotes são marcados para distinguir os tipos de serviços e os roteadores são configurados para criar filas distintas para cada aplicação, de acordo com as prioridades das mesmas. Assim, uma faixa da largura de banda, dentro do canal de comunicação, é reservada para que, no caso de congestionamento, determinados tipos de fluxos de dados ou aplicações tenham prioridade na entrega.

Tal configuração tem que ser aplicada nos roteadores das duas “pontas”, para que os mesmos tenham as mesmas definições de prioridade.

Os principais critérios que permitem apreciar a qualidade de serviço são os seguintes:

- **Débito** (em inglês *bandwidth*): às vezes chamado banda concorrida por abuso de linguagem, define o volume máximo de informação (*bits*) por unidade de tempo.
- **Latência, prazo ou tempo de resposta** (em inglês *delay*): caracteriza o atraso entre a emissão e a recepção de um pacote. *Delay* é o tempo que o pacote demora para ser transmitido de um ponto a outro, conforme ilustra a figura 9.

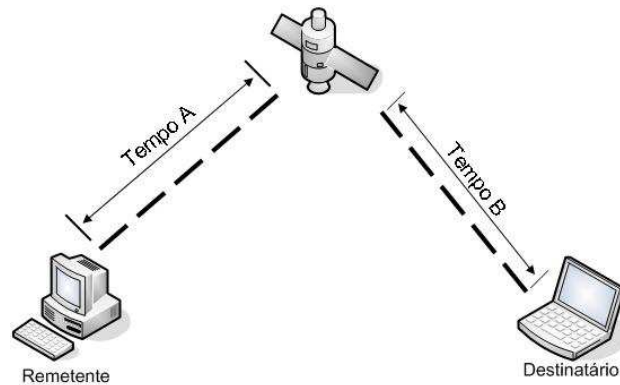


Figura 9 – Delay = Tempo A + Tempo B

- **Flutuação** (em inglês *jitter*): é a variação do *delay*. O lado que envia os pacotes realiza este envio de forma contínua, com um espaçamento semelhante entre os pacotes, porém, devido ao congestionamento da rede ou algum erro de configuração o recebimento destes pacotes pode ocorrer sem o mesmo espaçamento com que foram enviados. Um exemplo pode ser conferido na figura 10. O *Jitter* influencia bastante, pois reduzindo o *delay*, tornando a variação menor possível, o prazo de entrega dos pacotes também reduz. O *Jitter* pode ser suavizado através das técnicas de compressão de dados.

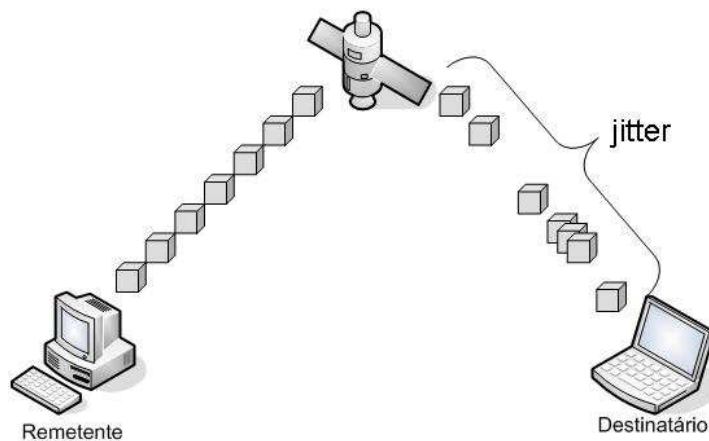


Figura 10 – Exemplo de *jitter*

- **Perda de pacote** (em inglês *packet loss*): corresponde a não entrega de um pacote de dados, a maior parte do tempo devido a uma obstrução na rede;
- **Desequencing**: trata-se de uma modificação na ordem de chegada dos pacotes.

Dentro da qualidade de serviço, existe o conceito de CoS (*Class of Service*). É uma forma de agrupar diversas aplicações com características comuns, permitindo o tratamento diferenciado em relação a outras classes de serviço (ou grupo de aplicações).

Como vários serviços podem ser oferecidos e utilizados simultaneamente, concorrendo assim pelos recursos da rede, temos que garantir, para cada tipo ou classe de serviço, o nível de serviço adequado ao seu funcionamento.

A identificação de diversos fluxos de dados com a mesma característica facilita a construção de políticas específicas para tratamento daquele tráfego de forma individualizada em cada classe de serviço, independentemente de sua origem ou do seu destino.

6.4.3.3 Aceleradores WAN

Com a evolução crescente das aplicações, a demanda por velocidades de transmissão aumenta cada vez mais. Gera-se então uma situação complicada, afinal é necessário melhorar a qualidade e velocidade de transmissão para suportar os novos sistemas, porém projetos como este esbarram nas diretorias das empresas devido aos altos custos com *links* maiores.

As tecnologias existentes para otimização de WAN vieram para solucionar este problema. Aceleradores WAN permitem contornar as limitações mencionadas acima, permitindo transferências mais rápidas e eficientes e uma menor taxa de perda de pacotes sem que seja necessário contratar mais largura de banda. Estes dispositivos trabalham comprimindo e armazenando (*caching*) os dados, otimizando parâmetros do TCP e implementando políticas de qualidade de serviço (QoS) para aumentar a eficiência no processo de transmissão de dados. Alguns fabricantes alegam que seus dispositivos chegam a aumentar a capacidade de transmissão em até 10 vezes.

Não é de hoje que é realizada abordagem sobre este assunto. Modems analógicos já comprimiam dados antes de enviá-los pelas linhas telefônicas há mais de 20 anos atrás (Filippetti, 2009). O que ocorre é que novas técnicas foram sendo criadas e aperfeiçoadas. Basicamente, os aceleradores baseiam-se no armazenamento (*caching*) e fortes algoritmos de compressão para fazer seu trabalho. Quando o TCP foi criado, as necessidades de transmissão e a qualidade

das redes eram muito diferentes das de hoje. Os aceleradores alteram propriedades do TCP para adequá-lo a realidade das redes de hoje. Isso torna o processo de transmissão extremamente eficiente e resulta nos ganhos anteriormente mencionados.

Em uma implementação tradicional, aceleradores WAN são colocados em cada ponta do *link* WAN. São necessários pelo menos dois motores, já que os dados são comprimidos e, sem um dispositivo que possa entender os dados alterados que chegam à outra “ponta”, estes se tornam ilegíveis para equipamentos convencionais (como roteadores ou computadores). Além destes dois motores aceleradores, podemos implementar outro, para que o mesmo faça o gerenciamento da aplicação. Esta estrutura pode variar de acordo com o fabricante.

Podemos encontrar hoje, no mercado, alguns *appliances* pagos para esta finalidade, das seguintes empresas: *A10 Networks, Riverbed Technology, Cisco Systems, Blue Coat Systems, Citrix, Exinda Networks, Intelligent Compression Technologies (ICT), Juniper Networks, Expand Networks, Nortel Networks*.

A grande vantagem de utilizar aceleradores é que em muito pouco tempo temos o retorno do investimento, já que não temos um aumento de custo recorrente (banda). O que dificulta tal implantação, no caso de optar por uma solução paga, é o alto custo dos equipamentos.

Logo abaixo, na figura 11, pode ser visualizada uma forma de aplicar esta tecnologia:

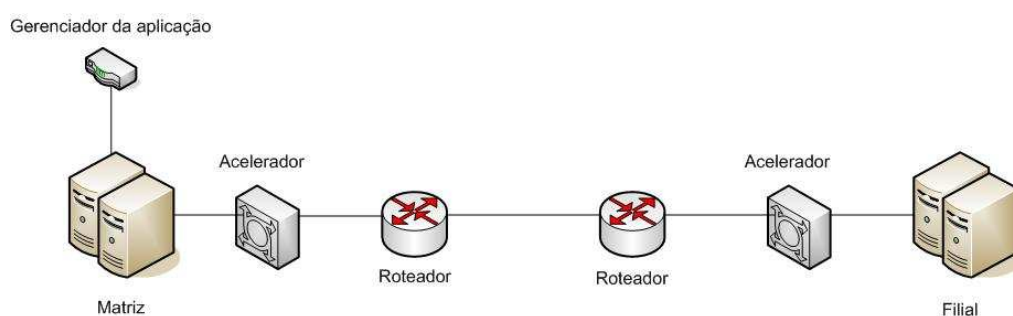


Figura 11 - aplicação de acelerador WAN.

Neste exemplo, pode-se perceber a existência de dois equipamentos diferentes dentro de uma WAN. Estes equipamentos são *appliances* de rede. Os dois *appliances* que estão entre a LAN da matriz e da filial e os roteadores são

motores responsáveis por realizar a compressão e o *caching* (motores da aceleração) e um terceiro *appliance* localizado na matriz é responsável por gerenciar e configurar a forma de aceleração que será aplicada.

6.4.3.4 Compressão de dados

A compressão de dados é o ato de reduzir o espaço ocupado por dados num determinado dispositivo. Este processo é realizado através de algoritmos de compressão, reduzindo a quantidade de *bits* para representar cada dado, sendo esse dado uma imagem, um texto, ou um arquivo qualquer.

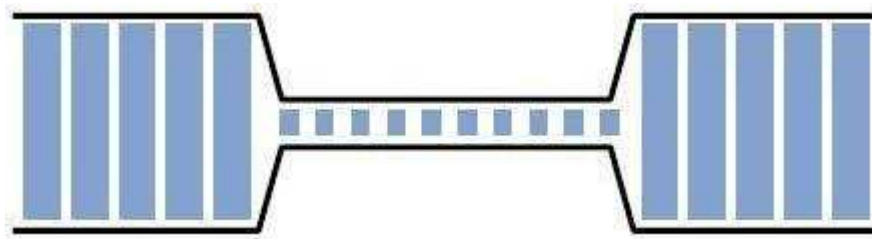


Figura 12 - Exemplo de compressão e descompressão dos pacotes.

A compressão de dados pode também retirar a redundância, levando em conta que muitos arquivos possuem informações redundantes que podem ou precisam ser eliminadas de alguma forma para diminuir o número de *bits*. Essa forma é através de regra, chamada de código ou protocolo, que, quando seguida, elimina os *bits* redundantes de informações, de modo a diminuir seu tamanho nas pastas. Por exemplo, a sequência "CCCCCCCC" que ocupa 8 bytes, poderia ser representada pela sequência "8C", que ocupa 2 bytes, economizando 75% de espaço.

As maiores intenções da compressão de dados são:

- Economizar espaço em dispositivos de armazenamento, como discos rígidos;
- Ganhar desempenho nas transmissões.

Ambas as ações visam à redução de custos, tanto com relação à aquisição de *hardware* quanto com relação à contratação de *links* maiores, por exemplo.

Embora possam parecer sinônimos, compressão e compactação de dados são processos distintos. A compressão, como visto, reduz a quantidade de *bits* para representar algum dado, enquanto a compactação tem a função de unir dados que

estejam desunidos. Um exemplo clássico de compactação de dados é a desfragmentação de discos.

Como exemplo de algoritmo de compressão de dados, pode-se citar o algoritmo *Deflate*. Esse algoritmo tem como princípio encontrar *strings* duplicadas no texto de entrada. Quando uma segunda ocorrência é achada, ela é trocada por um ponteiro para a *string* anterior, contendo os campos distância e comprimento. A distância é limitada por 32 *Kbytes*, e o tamanho por 258 *bytes*. Se uma determinada *string* não for achada em nenhum ponto dos 32 *Kbytes* anteriores, ela é lançada como uma sequência literal de *bytes*.

As *strings* duplicadas são encontradas através de uma *hash table*, que guarda todas as *strings* de comprimento 3 da entrada. Um *index* é criado para os próximos 3 *bytes*, selecionando uma cadeia de *bits*. Enquanto essa cadeia não estiver vazia, ela vai sendo comparada com a parte da entrada que estiver sendo analisada no momento, e o maior trecho que tiver correspondência é selecionado. A procura por essa cadeia é feita sempre começando pelas *strings* mais recentes, a fim de favorecer pequenas distâncias, o que é melhor para se obter um bom desempenho. Assim, o tamanho das sequências que serão analisadas não deve ser muito grande, e, por isso, ele é uma das entradas da função *Deflate*. Dessa forma, essa função não encontra o maior grupo de *bits* que correspondam à entrada que está sendo analisada, mas ela encontra um grupo semelhante que já é suficientemente grande para o bom funcionamento do algoritmo.

Existem ainda mecanismos que visam acelerar o processo de compressão, em níveis de velocidade avaliados em uma escala de 1 a 3. Esses mecanismos, no entanto, reduzem a taxa de compressão. Portanto, antes de usá-los, deve-se observar se o que importa mais é o tamanho final ou tempo que será usado para fazer a compressão.

Para que se possa realizar a descompressão dos dados, é usada a função *Inflate*. Esta função, primeiramente, cria uma tabela contendo certo número de bits da entrada em um tamanho menor do que o código mais longo. Então, ela pega esses bits e extrai informações da tabela dizendo se o tamanho do próximo código é o mesmo ou se é menor, e nesse caso dizendo qual é a diferença. Além disso, ela também descobre qual o próximo código apontado, que servirá para se criar um

novo nível de tabela. Desse modo, a função tenta achar mais bits, para decodificar um código mais longo.

O tamanho real da primeira tabela é decidido equilibrando o tempo para fazer a decodificação e o tempo para construir a tabela. Assim, se construir uma tabela não levasse nenhum tempo e se a memória fosse grande o suficiente, só haveria o primeiro nível de tabela, que cobriria o maior código inteiro. Como isso não acontece, a função escolhe o número de bits para a primeira tabela que irá proporcionar a maior velocidade.

O algoritmo *Deflate* é usado nos utilitários que comprimem ou descomprimem arquivos no padrão ZIP ou no padrão gzip. O formato ZIP se tornou popular através do programa PKZIP e é hoje usado na maioria dos programas de compressão de dados. Esse algoritmo também é usado para comprimir imagens no formato PNG.

Outro exemplo de algoritmo de compressão de dados é o LZ. Trata-se de uma técnica de compressão sem perdas de dados, criada em 1977 por Lempel e Ziv. É baseada no princípio de compressão por substituição. O LZ procura a ocorrência de conjuntos de caracteres repetidos em um arquivo e substitui-os por um código que usa menos *bits* para representar aquele conjunto de caracteres. O algoritmo LZ deu origem ao LZW, um dos mais usados pelos programas de compactação de hoje.

6.5 ACELERADORES WAN

Neste capítulo serão descritos os conceitos de duas tecnologias diferentes de aceleração WAN. A solução criada pela Cisco Systems chamada WAAS e uma solução *Opensource* da plataforma Linux, chamada Vtun.

6.5.1 WAAS

Conforme artigo publicado pela Cisco Systems Inc., em 2007, Cisco *Wide Area Application Services* (WAAS) é uma abrangente solução para otimização WAN (desenvolvida pela Cisco Systems, Inc). Tal solução visa acelerar o tráfego de dados (informações) entre duas LAN's. Cisco WAAS permite que o departamento de TI centralize aplicações e armazenamento de dados, pois oferece um grande ganho de desempenho na comunicação entre as redes locais.

6.5.1.1 Processo de aceleração

A solução Cisco WAAS é capaz de acelerar a WAN através dos seguintes mecanismos:

- *Payload Compression* com RTM (*Router Transparency Mode*): A compressão do pacote IP tem duas fases: compressão de saída ("compressão") e descompressão de entrada ("descompressão"). A compressão é realizada através de:
 - DRE (*Data Redundancy Elimination*, Eliminação de redundância de dados);
 - Algoritmo LZ.

O tratamento é feito com compressão sem perdas, garantindo que o pacote IP, depois de comprimido e descomprimido, seja idêntico ao pacote IP original. Cada pacote IP é comprimido e descomprimido por si só, sem qualquer relação com outros pacotes. Como os pacotes IP's podem chegar fora de ordem ou não chegar, cada pacote IP comprimido encapsula uma única carga. Na compressão e descompressão com o *Router Transparency Mode* o cabeçalho IP, TCP e UDP são preservados, e a rede tem total visibilidade de todos os fluxos de IP. Usando o RTM, todos os atuais e futuros serviços são garantidos para que funcionem corretamente. Os roteadores podem classificar, estruturar e marcar o tráfego IP e fluxos originais de provisionamento sem modificações ou alterações. A criptografia pode ser aplicada ao tráfego sensível a ameaças. O tráfego não autorizado pode ser bloqueado. Se a compressão fosse feita sem o RTM, ocorreria a alteração do cabeçalho, perdendo assim a gerência dos pacotes, bem como a aplicação de QoS.

- Tecnologia de *Caching* (também conhecido como WAFS – *Wide Area File Services*): Esta função é construída através de vários algoritmos:
 - ***Vertical Data Analysis (VDA)***: Divide todo o tráfego em vários cabeçalhos e componentes de dados, reduz o tamanho deles e marca dados que vão para *cache*.

- **Selective Caching (SC):** Opera em um nível de *byte* e dados de *cache* que são transmitidos repetidamente.

- **Adaptive Packet Compression (APC):** Aplicado a dados que não são tratados por *caching* ou otimização de cabeçalho.

Por exemplo, pode-se citar qualquer empresa que trabalhe com um servidor central onde filiais acessem, por exemplo, uma planilha de Excel. No primeiro acesso, esta planilha será acelerada pelo RTM e será salva em um disco ou memória *flash* de um *appliance* acelerador. Na segunda vez, quando o usuário abrir a planilha, ela já estará salva localmente e a única coisa que será acrescentada à planilha são as atualizações feitas por outros usuários.

- Otimização de TCP: através da otimização do TCP, a resposta aos *flags* de SYN é local, mantendo a conexão WAN ativa, não aguardando um ACK para o envio do próximo SYN. Apresenta também uma melhor utilização da banda, não desperdiçando recursos. A retransmissão ocorre somente para pacotes perdidos.
- BIC TCP: *Binary Increase Congestion* TCP é um novo protocolo de transmissão de dados que gerencia o congestionamento e permite que a rede se recupere mais rápido de eventos de perda de pacotes. Alguns estudos apontam que o BIC TCP pode atingir velocidades 6 mil vezes maior que o padrão DSL. O BIC TCP funciona através dos mecanismos: *Binary Search Increase*, *Additive Increase*, *Fast Convergence* e *Slow Start*.
- *Application-Specific Acceleration*: Trata-se da possibilidade de configurar acelerações específicas para determinadas aplicações. O WAAS permite transformar uma sequência de comandos em uma sequência mais curta, a fim de reduzir ida e volta das informações. Além do mais, faz uso do WAFS (*Wide Area File System*), podendo apresentar respostas mais rápidas através da informação contida em *cache*.
- WCCP: é um protocolo de redirecionamento transparente que permite o uso de um *proxy-cache* para manipular o tráfego Web, reduzindo o custo de transmissão e o tempo de *download*. Atualmente, existem duas versões do protocolo do WCCP. A versão 1 (WCCPv1) permite apenas que um roteador participe do serviço de *proxy* transparente, ou seja, só é permitido que um

roteador atue como redirecionador de pacotes, entretanto é possível usar mais de um servidor *proxy-cache* para participar do serviço. A versão 2 (WCCPv2) suporta múltiplos roteadores o que aumenta o desempenho do serviço, pois a tarefa de encapsulamento de pacotes é balanceada entre os mesmos. Além disso, a versão 2 provê recursos de segurança na comunicação entre o roteador e o *proxy-cache*.

6.5.1.2 WAE

O WAE (*Wide-Area Application Engine*) é um *appliance* (equipamento) de rede desenvolvido pela Cisco para prover a aceleração de WAN.



Figura 13 - Cisco WAE 512.

Um WAE pode suporta a instalação diferentes *softwares* que fornecem um conjunto abrangente de serviços para escritórios remotos:

- WAAS (*Cisco Wide Area Application Services*);
- WAFS (*Cisco Wide Area File System*)

Quando o *software* do WAAS é instalado, o WAE funciona como um gerenciador central da solução e/ou como o motor de aceleração propriamente dito (artigo publicado pela Cisco Systems Inc, em 2007).

Além de prover a aceleração, o WAE pode também funcionar como servidor de arquivos (basta estar equipado com um disco rígido) e como servidor de impressão. A grande vantagem destas duas funcionalidades é a eliminação de *hardware*. Uma vez acelerada a comunicação entre matriz e filial e o próprio WAE suportando arquivos e impressoras, elimina-se a necessidade de um servidor em uma filial.

6.5.2 Vtun

A ferramenta Vtun é utilizada para criar VPN's sobre redes TCP/IP suportando uma variedade de tipos de tunelamento e providenciando encriptação, compressão e controle de tráfego. Suporta túneis IP, túneis ponto a ponto, túneis *Ethernet* suportando todos os protocolos que trabalham sobre a ethernet: IP, IPX, Appletalk, túneis serial suportando protocolos que utilizam linhas serias como: PPP, SLIP e túneis *pipes* suportando todos os túneis que trabalham sobre *pipes* UNIX.

Vtun possibilita o uso dos protocolos TCP permitindo, com isso, estabelecer túneis sobre *firewalls* e o protocolo UDP (*User Datagram Protocol*) permitindo trabalhar com um pequeno *overhead* de tunelamento.

A ferramenta permite compressão usando zlib (suportado apenas sobre TCP) e lzo suportado por UDP e TCP.

A forma de encriptação usa: autenticação baseada em desafio permitindo que passwords não passem em claro e uso de chaves BlowFish (algoritmo criptográfico de chave simétrica) de 128 bits com rápida encriptação e chaves *hash* com 128 bits MD5.

Esta solução pode ser utilizada em ambientes Linux (Red Hat, Debian, Corel, Ubuntu), FreeBSD, HPUX e Solaris.

6.5.2.1 Zlib

ZLIB é uma biblioteca multiplataforma de compressão de dados escrita por Jean-Loup Gailly e Mark Adler como uma generalização do algoritmo *deflate* usado em seu programa de compressão de dados gzip (abreviação de gnu zip, um software livre de compressão sem perda de dados, que gera arquivos *.gz).

Hoje, zlib é de fato um padrão para compressão de dados em documentos portáteis (Roelofs, 2005). A biblioteca zlib é usada por centenas de aplicativos dos

sistemas operacionais Unix, como o Linux. Seu uso também é crescente em outros sistemas, como o Microsoft Windows e o Palm OS.

A última versão publicada do Zlib é a 1.2.3, no dia 18 de julho de 2005.

6.5.2.2 LZO

LZO é uma biblioteca de compressão de dados que realiza o processo de compressão e descompressão em tempo real.

LZO é escrito em ANSI C. Tanto o código fonte quanto o formato comprimido de dados são projetados para serem portáteis entre plataformas.

Esta biblioteca implementa uma série de algoritmos com as seguintes características (Oberhumer, 2008):

- A descompressão é simples e muito rápida;
- Não requer memória para descompressão;
- A compressão é muito rápida;
- Requer 64 KB de memória para realizar a compressão;
- Inclui níveis de compressão para a geração de dados pré-comprimido que atingem uma taxa de compressão bastante competitiva;
- Há também um nível de compressão, que precisa de apenas 8 Kb para compactação;
- O algoritmo é seguro e sem perdas;
- Permite compressão em acessos discados, porém o processo de compressão é mais lento.

A última versão do LZO é a 2.03, lançada em 30 de abril de 2008.

Em testes realizados em um computador Pentium 133, alcançaram-se resultados de 5 MB/segundo no processo de compressão e 16 MB/segundo na descompressão dos dados. Obviamente, em computadores mais rápidos, melhores resultados são obtidos.

O LZO foi criado e testado com sucesso nas plataformas Microsoft Windows 9x, NT, Me, 2000, 2003 e XP, Linux e HPUX.

7 METODOLOGIA

Durante a realização do TCC I, foi planejada a implementação das duas soluções. Serão abordados os principais tópicos, custos, estruturas necessárias e orientações fornecidas pelos fabricantes das soluções.

7.1 PLANEJANDO A IMPLEMENTAÇÃO DA SOLUÇÃO VTUN

O primeiro passo para se configurar uma VPN é decidir o sistema operacional utilizado nos *gateways*, bem como a ferramenta que estabelecerá a conexão entre os dois pontos, atendendo aos requisitos já citados anteriormente. Como este trabalho visa descrever uma solução gratuita, o sistema operacional escolhido é o Ubuntu Server 9.04. Conforme o site do fabricante (www.ubuntu.com), esta versão apresenta como benefícios:

- Facilidade na instalação e no gerenciamento;
- Plataforma de computação aberta;
- Facilidade em realizar upgrade de versão;
- Gerenciamento de atualizações de fácil controle;
- Se o assunto for sustentabilidade, é uma excelente solução “Verde”, pois maximiza a eficiência da virtualização;
- Segurança.

Visando atender às necessidades de uma conexão VPN conforme todas as descrições feitas no referencial teórico e a compressão de dados para prover a aceleração, será adotada a ferramenta Vtun.

7.1.1 Pré-requisitos

Tal solução precisa de dois computadores com conexão à Internet para ser implementada, ficando cada *gateway* em uma das extremidades. Para que seja definido o hardware mínimo necessário, será tomada por base a especificação do sistema operacional Ubuntu Server 9.04, que solicita:

- Processador: Intel, AMD x86, AMD 64;

- Memória: mínimo 192Mb de RAM;
- Espaço em Disco: 1Gb;

Os 2 equipamentos utilizados neste trabalho possuirão a seguinte configuração:

- Computador A – Processador AMD Sempron 2400+, 512Mb de memória RAM e um disco rígido de 40Gb;
- Computador B – Processador Intel Pentium IV, 512Mb de memória RAM e um disco rígido de 20Gb.

7.1.2 Instalando o Vtun

Primeiramente é necessário fazer o *download* do Vtun diretamente do site <http://vtun.sourceforge.net/>.

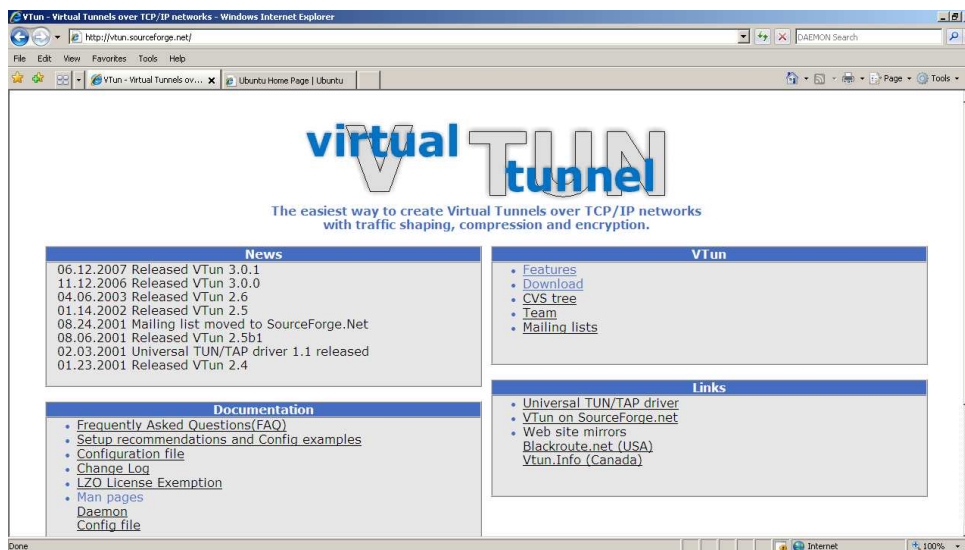


Figura 14 - Página oficial do Vtun

Depois de realizado o *download* e instalação da versão 3.0.1, lançada em 06/12/2007, devem ser seguidos os seguintes passos:

- Descompactar o arquivo vtun-3.0.1.tar.gz baixado, acessando a pasta onde foi feito o *download* e executando o comando `tar -vzxf vtun-3.0.1.tar.gz`;

7.1.3 Configurando o Vtun

Toda a configuração do Vtun fica localizada em `/etc/vtund.conf`. Abaixo segue a configuração do arquivo pelo lado do servidor. Repare que todas as seções começam com “{” e terminam com “}”.

```
options {
port 5000; # Listen on this port.
# Path to various programs
ppp /usr/sbin/pppd;
ifconfig /sbin/ifconfig;
route /sbin/route;
firewall /sbin/ipchains;
}

# Ethernet example. Session 'lion'.
lion {
pass teste; # Password
type ether; # Ethernet tunnel
device tap0; # Device tap0
comp lzo:1; # LZO compression level 1
encr yes; # Encryption
stat yes; # Log connection statistic
keepalive yes; # Keep connection alive
speed 256:128;
up {
# Connection is Up

# Assign IP address
ifconfig "%% 10.1.0.1 netmask 255.255.255.0";

# Add route to net 10.2.0.0/24
route "add -net 10.2.0.0 netmask 255.255.255.0 gw 10.1.0.2";

# Enable masquerading for net 10.2.0.0/24
firewall "-A forward -s 10.2.0.0/24 -d 0.0.0.0/0 -j MASQ";
```

```
};

down {
# Connection is Down

# Shutdown tap device.
ifconfig "%%" down";

# Disable masquerading for net 10.2.0.0/24
firewall "-D forward -s 10.2.0.0/24 -d 0.0.0.0/0 -j MASQ";
};
}
```

Dentro da configuração é importante destacar os seguintes itens:

- “*port 5000;*” – define a porta que será usada;
- “*pass teste;*” – define a senha utilizada na conexão. Neste caso a senha definida foi ‘teste’;
- “*type ether*” – especifica o tipo de conexão. Neste exemplo a conexão é do tipo *Ethernet*;
- “*device tap0*” – este será o dispositivo virtual usado para a comunicação;
- “*comp lzo:1*” – define o nível de compressão que será utilizado;
- “*encr yes*” – define se a seção será encriptada ou não;
- “*speed 256:128*” - Esta é uma das mais importantes, pois nessa linha é possível definir o *traffic shape*. O valor 256 é a velocidade de *download* e 128 é a velocidade de *upload*.
- “*up*” – O Vtun executa um “*up*” quando a conexão é estabelecida, definindo um endereço IP para o tap0:

```
Up{
    ifconfig "%% 10.1.0.1 netmask 255.255.255.0";
    route "add -net 10.2.0.0 netmask 255.255.255.0 gw 10.1.0.2";
    firewall "-A forward -s 10.2.0.0/24 -d 0.0.0.0/0 -j MASQ";
```

```
}

```

A terceira linha define a rota para a rede 10.2.0.0 com o *gateway* e a quarta linha a máscara de rede interna e o *IP forward* pelo *ipchains*.

Do lado do cliente, utiliza-se o exemplo abaixo. É importante notar que o algoritmo cliente possui basicamente as mesmas opções do algoritmo servidor:

```
options {
port 5000; # Connect to this port.
timeout 60; # General timeout

# Path to various programs
ppp /usr/sbin/pppd;
ifconfig /sbin/ifconfig;
route /sbin/route;
firewall /sbin/ipchains;
}

# Ethernet example. Session 'lion'.
lion {
pass tiago; # Password
type ether; # Ethernet tunnel
device tap0; # Device tap1
up {
# Connection is Up

# Assign IP address and netmask.
ifconfig "%%" 10.1.0.2 netmask 255.255.255.0";
};
down {
# Connection is Down

# Shutdown tap device
ifconfig "%%" down";
};

```

}

Para iniciar o Vtun no computador que atua como servidor, deve-se usar o comando “#vtund -s”. Para conectar o cliente deve-se utilizar o comando “#vtund lion”.

Desta forma será estabelecida uma VPN utilizando a ferramenta Vtun.

7.1.4 Ambiente a ser implementado

A solução será aplicada criando uma VPN entre um escritório remoto e a empresa ThyssenKrupp Elevadores. Ambos os sites estão localizados em Guaíba/RS. O escritório remoto possui uma conexão banda larga (ADSL) de 1Mb, provida pela empresa de telefonia Oi.

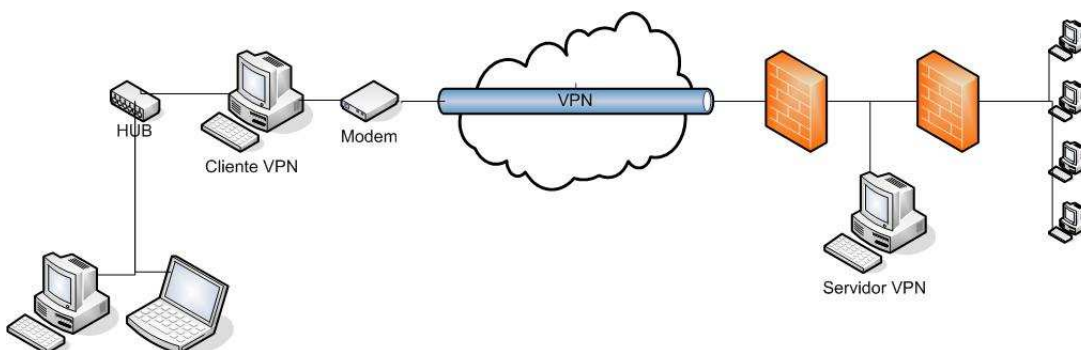


Figura 15 – VPN entre escritório remoto e a ThyssenKrupp

Na figura 15, pode ser visto à esquerda o escritório remoto com o computador que fará o cliente neste conexão e à direita a empresa, onde ficará o servidor que receberá a conexão VPN.

7.2 PLANEJANDO A IMPLEMENTAÇÃO DA SOLUÇÃO WAAS

O planejamento da implementação do Cisco WAAS pode ser dividido em três partes: fase de dimensionamento, análise da otimização desejada, disponibilidade para gestão (gerenciamento) do sistema.

A fase de dimensionamento inclui as definições das tarefas descritas abaixo:

- Determinar onde se faz necessária a aceleração (WAN onde a comunicação é mais lenta, onde o negócio requer mais agilidade, etc.);

- Determinar o modelo de WAE a ser adquirido. Esta determinação se faz levando em conta a largura de banda contratada com a operadora de telefonia, o número de usuários da filial e a expectativa de utilização. Verificar se a filial tem a necessidade de um servidor de arquivos e de impressão;
- Determinar números de licenças a serem adquiridas.

O planejamento em torno da questão de gestão deve ser feito levando em conta os seguintes quesitos:

- Analisar a estrutura que receberá a aceleração antes da alteração na arquitetura;
- Determinar o método de autenticação e autorização de *login* (por exemplo, RADIUS, TACACS+, servidor de domínio Microsoft Windows) e as políticas de conta para manutenção do Cisco WAAS;
- Para fins de segurança, definir políticas de senha, bem como pré-definir a alteração de senha do superusuário assim que a implementação for finalizada;
- Definir a necessidade de criar outras contas de usuário para administrar a solução;
- Determinar se será mais fácil administrar os WAE's reunindo-os em grupos lógicos. Estes grupos facilitam a administração e aplicação de alterações nos WAE's;
- Determinar se o gerenciamento será feito por Telnet ou SSH;

Quando o assunto é determinar a otimização desejada com a implementação da solução, deve-se analisar o seguinte:

- Analisar a funcionalidade do *link* e dos roteadores, para definir se existem problemas de performance, quedas, etc.
- Determinar os tipos de aceleração necessários (por exemplo, WAFS, servidor de impressão, servidor de arquivo, aceleração de aplicações específicas);
- Todas as configurações possíveis estarão disponíveis no decorrer deste trabalho, quando o assunto for o gerenciamento da solução.

7.2.1 Requisitos necessários

Baseando-se em uma WAN já estabelecida, é necessária a aquisição de três WAE's. Dois farão a função de motor da aceleração e um a função de gerenciador. Existem diversos modelos de WAE disponíveis pela Cisco para a implementação do WAAS. Nesta implementação serão usados o Cisco WAE 512 na filial Porto Alegre, o WAE 674 na matriz (trata-se de um equipamento mais robusto, visando suportar depois a aplicação de aceleradores nas demais filiais) e um WAE 474 como gerenciador da aplicação. O 474 será utilizado por fornecer uma maior facilidade na manutenção, pois possui uma estrutura muito semelhante a um *desktop*.

Vale salientar que a aquisição de três *appliances* de rede ocorre somente na primeira implementação. Caso a empresa deseje aplicar o WAAS em outra filial, será necessária a aquisição de somente mais um WAE, pois o motor de aceleração e o gerenciador, ambos localizado na Matriz, serão o mesmo que já vem sendo utilizado na aceleração da primeira filial.

7.2.2 Ambiente a ser implementado

A implementação do WAAS será realizada entre a LAN da Matriz, localizada em Guaíba e a LAN da filial localizada em Porto Alegre. Hoje a comunicação entre essas duas redes conta com um link MPLS de 1,5Mbps, seguindo a figura 16:

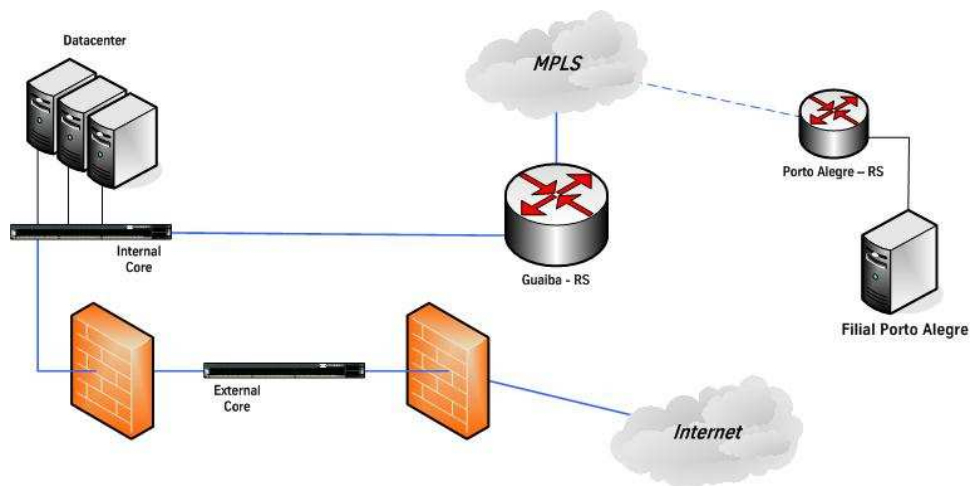


Figura 16 - Desenho da rede antes da implementação do WAAS

Com a implementação dos WAE's, a rede sofrerá as seguintes alterações:

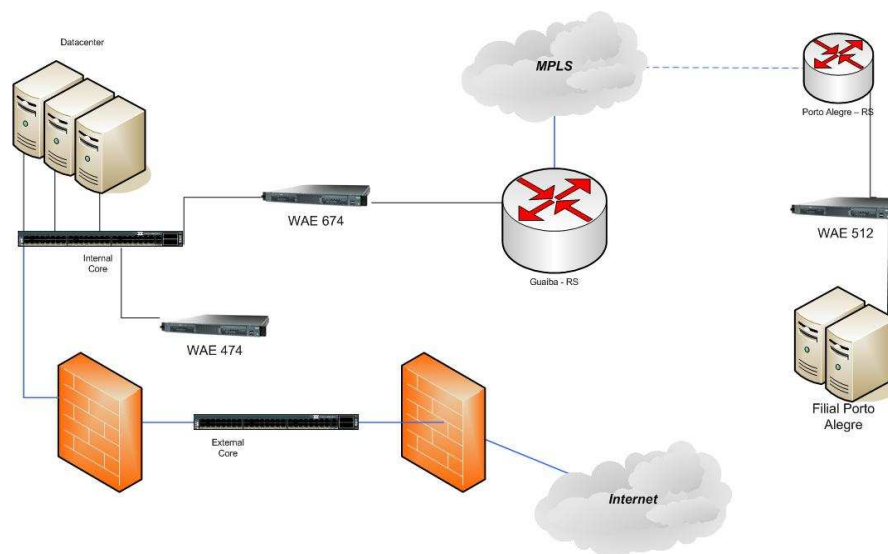


Figura 17 – Cenário após a implementação dos WAE's

Pode-se verificar pelo desenho que dois WAE's ficaram entre a LAN e o roteador, realizando a função de motor da aceleração e um terceiro equipamento ficará localizado na matriz realizando o gerenciamento da aceleração. A vantagem de se ter um terceiro equipamento apenas com a função de gerenciador é a não necessidade de parar a aceleração para gerar relatórios ou realizar alterações no mecanismo.

7.2.3 Monitorando e Configurando o WAAS

Nesta seção serão descritas as principais funções a serem monitoradas e configuradas no WAAS. O monitoramento e as configurações são realizadas no *appliance* gerenciador e o mesmo replica para os motores de aceleração. Neste trabalho será descrita a versão 4.1.5a do *software*.

Ao desembalar e alocar o WAE em um *rack*, deve-se ligar o mesmo na energia elétrica e conectar um monitor e um teclado. Depois de ligado, o WAE solicitará a execução do utilitário de configuração, usado para definir as configurações básicas do *appliance*. Dentro deste utilitário será selecionado se o WAE será um gerenciador central ou o motor da solução. Escolhendo a opção de *Central Manager*, o utilitário solicitará informações sobre DHCP, servidor NTP e fuso-horário, promovendo assim o WAE a gerenciador da solução. Após inserir todas estas informações, é necessário confirmar as alterações o equipamento será reinicializado.

Se na primeira tela do utilitário for selecionada a opção de motor de aceleração (*application-accelerator*), o assistente de configuração solicitará as informações

necessárias para saber quem é o *Central Manager* e receberá as definições realizadas no mesmo. O mesmo processo ocorre no WAE que ficará localizado na outra LAN (filial).

Com as instalações realizadas, o *Central Manager* disponibiliza, via *browser*, o acesso para monitoramento e configurações do WAAS. A primeira tela, que pode ser vista logo abaixo, solicitará o *logon* do usuário.



Figura 18 – Tela de *Logon*

Na tela inicial do gerenciador será apresentada uma visão geral da aplicação. Nesta tela é possível analisar o tráfego ocorrido entre os *appliances*, dividindo e mensurando em percentual o quanto cada tipo de tráfego está consumindo do link, bem como um comparativo entre a utilização do link no seu estado original e com a aplicação desta tecnologia de otimização.

Esta tela exibe também os “*Alarms Informations*”, onde é possível identificar se existe algum dispositivo com alguma não conformidade. A tela indica qual dispositivo está apresentando problema, qual o problema e o seu impacto no bom funcionamento da solução. Nesta tela podem ser identificados problemas onde o WAE não está ligando, está com o horário incorreto, entre outros. A figura 19 exibe a tela descrita.

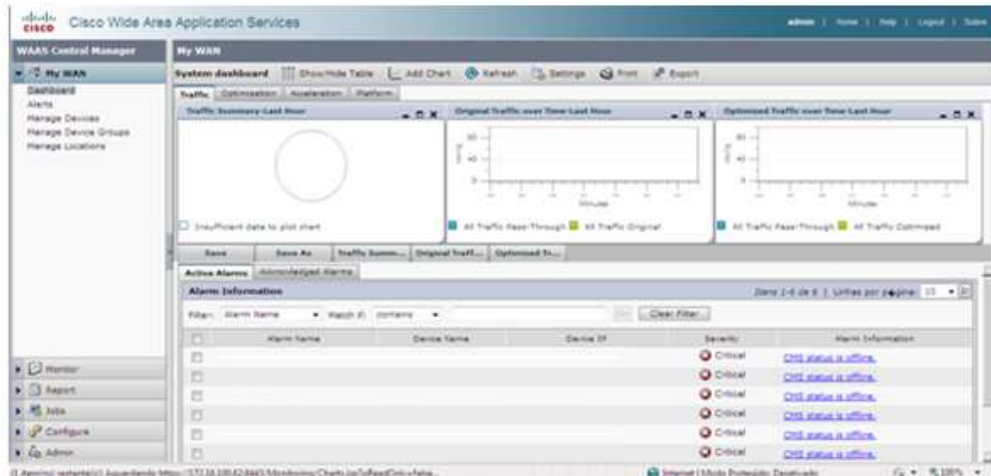


Figura 19 – Tela Inicial

Clicando em “Alerts”, o gerenciador apresenta uma tela de alertas mais detalhada. Nela é possível tomar algumas ações como editar as configurações do *application-accelerator* a fim de resolver problemas, visualizar *logs* e executar comandos de diagnóstico disponibilizados pelo próprio *software*.

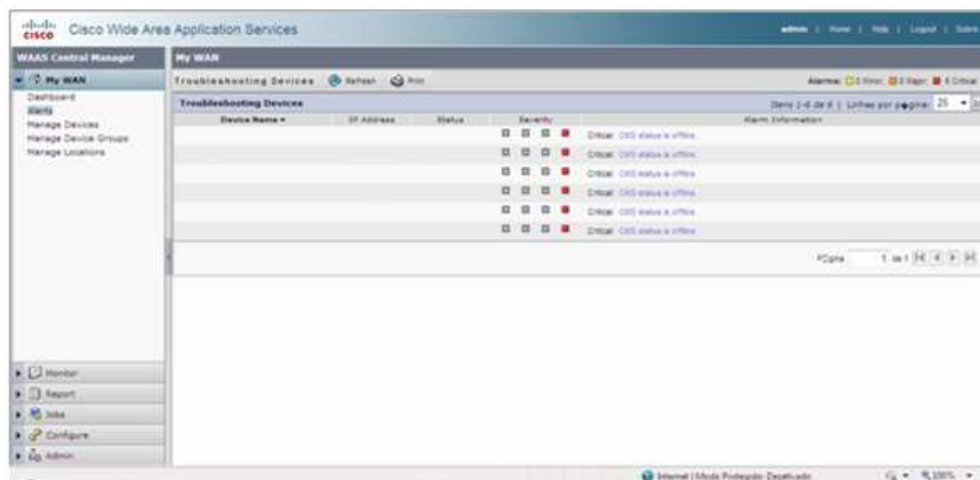


Figura 20 – Tela de alertas

No submenu “*Manage Device*” é possível gerenciar cada WAE da rede, podendo ativá-los, desativá-los, identificar o endereço IP, a localização geográfica, a versão do *software* que está sendo utilizada, o modelo do dispositivo, o *status* (*online* ou *offline*), bem como exportar relatórios e realizar pesquisas avançadas (opção mais utilizada em grandes empresas equipadas com WAAS em todas as suas filiais).

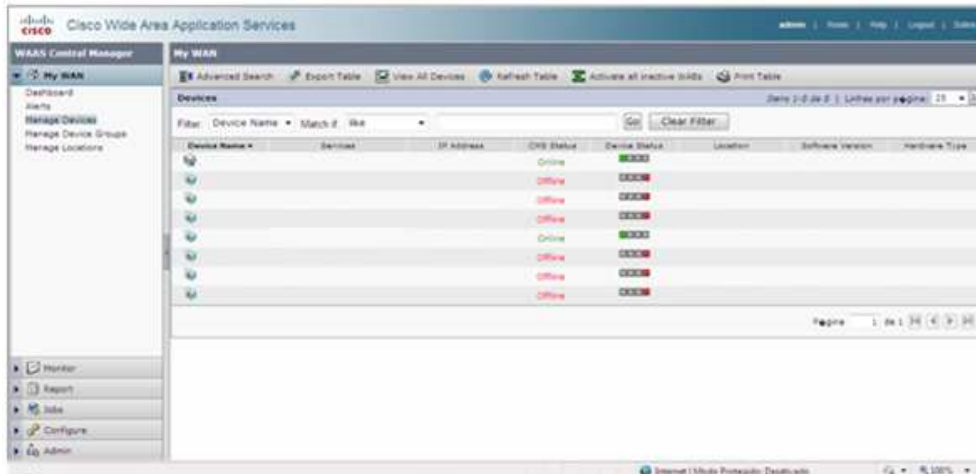


Figura 21 – Tela de gerenciador de dispositivos

No item “*Manage Device Group*” é possível criar grupos para armazenar os dispositivos. Esta opção torna-se importante por facilitar a aplicação de configurações em diversos WAE’s ao mesmo tempo além de facilitar a organização. Um “grupo” aceita os parâmetros de configuração da mesma forma que um dispositivo e todos os dispositivos que estiverem dentro deste grupo herdam as configurações.



Figura 22 – Tela de gerenciamento dos grupos de dispositivos

A opção “*Manage Locations*” visa organizar e indicar a localização de cada equipamento.

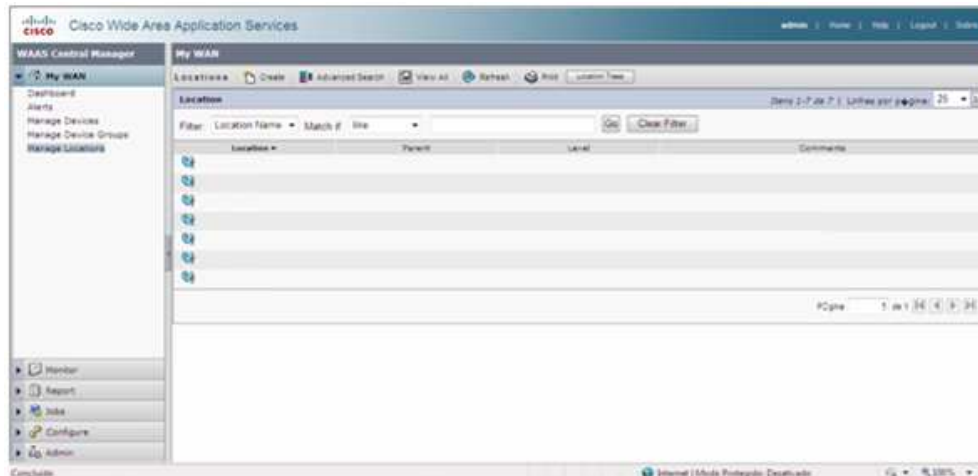


Figura 23 – Tela de gerenciamento de localização dos WAE's

Acessando o menu “Monitor” e o submenu “Optimization” é possível acessar tabelas e gráficos gerais (fornecem percentuais) sobre os resultados que estão sendo obtidos com a implementação do WAAS. É semelhante ao que pode ser visualizado na tela inicial, porém com mais detalhes, permitindo exportar estes resultados a fim de gerar relatórios.

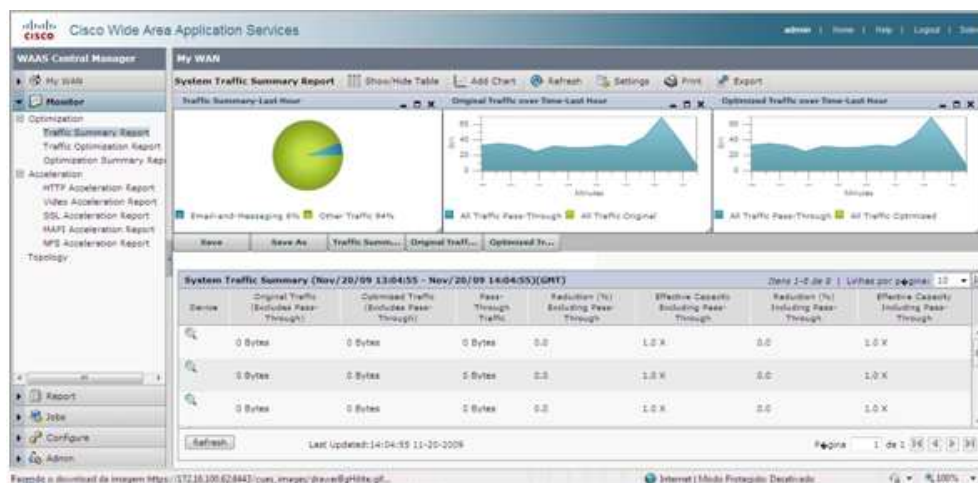


Figura 24 – Tela monitoramento da otimização

No submenu “Acceleration” é possível ter acesso às informações específicas de cada tipo de aceleração. Pode-se visualizar, por exemplo, o número de conexões que foram estabelecidas em cada dispositivo e o comportamento da WAAS com estas conexões. Assim como nos submenus anteriores, é possível exportar estes dados para a geração de relatórios.

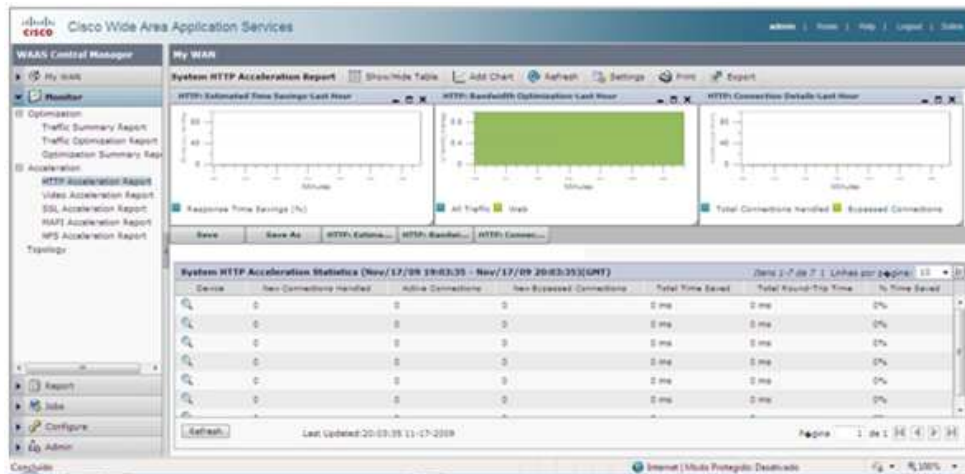


Figura 25 – Tela monitoramento da aceleração

No submenu “*Topology*” identifica-se quais conexões existentes entre os WAE’s. Por padrão, as conexões são estabelecidas entre filial e matriz, mas o WAAS permite que se estabeleça a aceleração entre duas filiais remotas, desde que elas tenham um *link* entre as duas LAN’s.

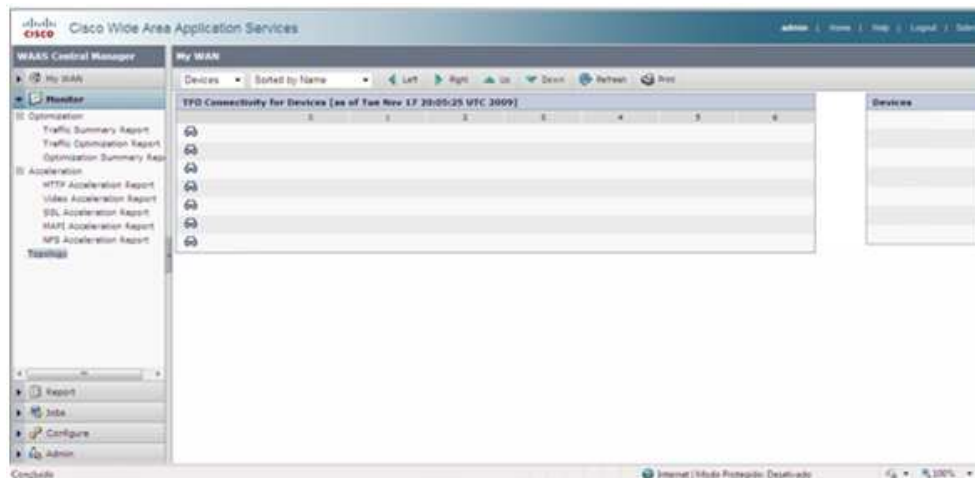


Figura 26 – Tela monitoramento dos WAE’s

O menu “*Report*” exhibe relatórios já prontos e possibilita planejar a geração destes relatórios através da opção “*Scheduled Reports*”

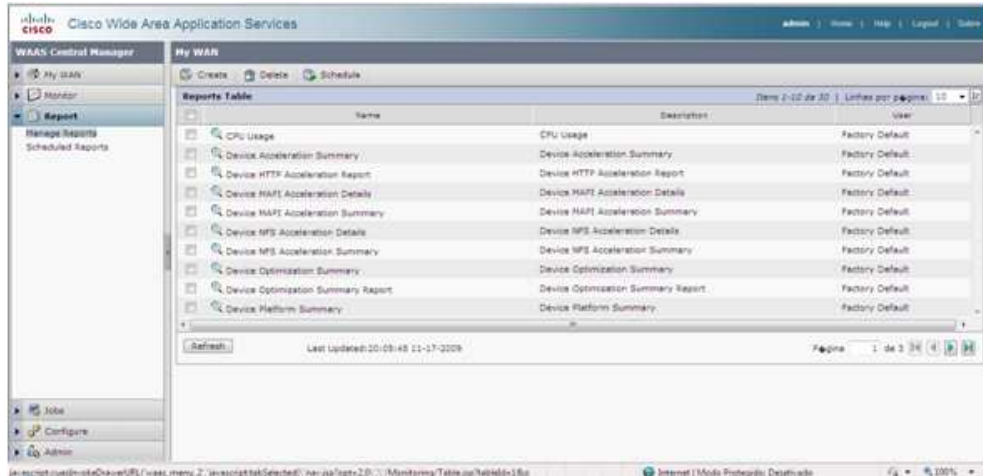


Figura 27 – Tela de relatórios

O menu “Jobs” exibe as atualizações disponíveis para o *software* do WAAS. Estes *Jobs* são criados através de um filtro que solicita origem dos arquivos, versões a serem pesquisadas e tamanho de arquivo.

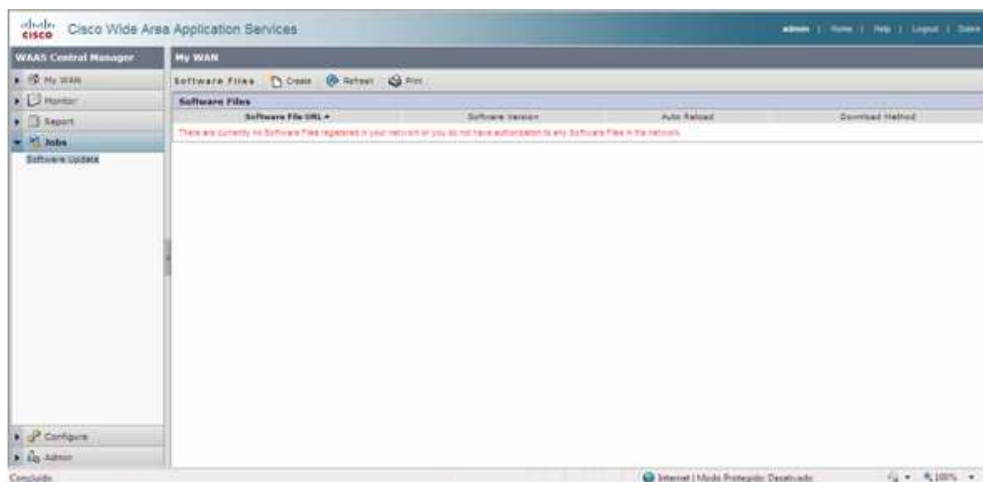


Figura 28 – Tela de gerenciamento de atualizações

O menu “Configure” disponibiliza o acesso a todas as possíveis configurações da solução Cisco WAAS. Serão descritas a seguir as opções disponíveis.



Figura 29 – Tela de configuração do WAAS

- **File Service:** conforme já descrito neste trabalho, é possível que o WAE seja um servidor de arquivos, reduzindo assim custos com servidores.
 - *Preposition* – cria o servidor de arquivo, atribuindo a ele um nome, credenciais de acesso para um administrador, espaço em disco que será utilizado para o armazenamento de arquivos, percentual de espaço em disco que será utilizado para o *caching*, entre outras configurações;
 - *Dynamic Shares* – possibilita configurar o compartilhamento de pastas;
 - *Baseline Group* – possibilita aplicar as configurações de servidor de arquivos à grupos de dispositivos já criados previamente.
- **Acceleration:** configura-se deste submenu a principal ferramenta do WAAS.
 - *Applications* – defini-se que tipos de aplicação terão aceleração. Dentre as diversas aplicações, destaca-se a aceleração de ferramentas de *backup*, ferramentas de conferência, aplicações de e-mail e mensagens instantâneas, acesso e transferência de arquivos, serviços de impressão e acesso remoto;
 - *Policies* – é possível criar políticas de aceleração para cada tipo de aplicação. Com isso é possível concentrar mais ainda os esforços da aceleração em serviços ou aplicações essenciais para o negócio da empresa;
 - *Classifiers* – indica em quais WAE's os serviços estão configurados;

- *Legacy Services*: exibe de forma mais resumida os serviços habilitados, as conexões estabelecidas e as impressoras instaladas (afinal o WAE pode também ser um servidor de impressão).

O menu “Admin” permite configurar e administrar os *logins* de acesso ao sistema. Neste mesmo menu é possível também criar grupo de usuários e auditar *logs* gravados em cada *logon* e *logoff* realizado no sistema.

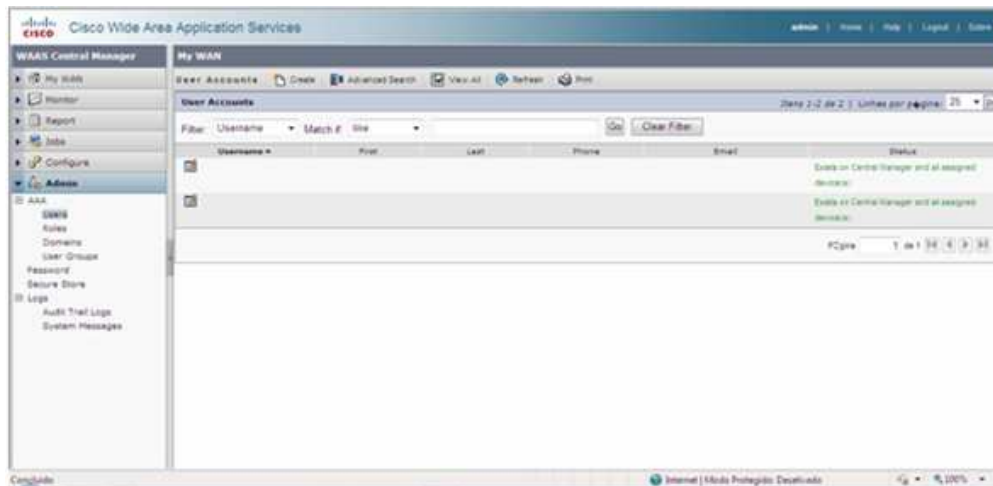


Figura 30 – Tela de administração do software 4.1.5a

7.3 TESTES A SEREM REALIZADOS

Como cada solução será implementada em um ambiente diferente, será realizada uma comparação de performance entre o “antes” e o “depois” da implementação dos aceleradores. No caso do WAAS, a WAN entre a Matriz e a filial Porto Alegre será testada antes da implementação dos WAE’s. Para a efetivação dos testes com o Vtun, primeiramente será criada uma VPN sem a aplicação de compressão. Após realizar diversas transferências de arquivo, serão testados os níveis possíveis de compressão de dados, podendo assim avaliar a otimização proporcionada pelo Vtun.

Os testes serão realizados mediante a transferência de diferentes tipos de arquivos, através da análise em diferentes enlaces (*jitter*, *delay*, *vazão*).

Acessos a sistemas internos, como Intranet, aplicações Metaframe, servidor de arquivos, entre outros, também serão analisados, usando como unidade o tempo.

8 RESULTADOS

O material teórico encontrado indica que a aceleração WAN é uma boa alternativa para a otimização das redes geograficamente distribuídas. Um fator que facilita a implementação é a existência de alternativas gratuitas desta tecnologia. Com esta descrição realizada nos capítulos acima é possível realizar a implementação e testes de performance, para que seja definida a melhor alternativa.

9 TRABALHO DE CONCLUSÃO DE CURSO - II

No Trabalho de Conclusão de Curso II, serão implementadas as duas soluções conforme planejado e descrito no TCC I. Após esta implementação, serão analisados os resultados obtidos pelas duas soluções e definida a melhor relação custo benefício, bem como a apresentação de um ROI (*Return of Investment*) da solução Cisco WAAS.

Vale salientar que este resultado vai depender muito das condições da empresa. Será necessário avaliar a disponibilidade de investimento que a empresa possui, uma análise de utilização do *link*, entre outras opções que serão abordadas no TCC II.

10 REFERÊNCIAS

TANEMBAUM, Andrew. Redes de Computadores, 4a Edição. Ed. Campus. 2003.

CISCO, Documentation - Cisco Systems - Cisco Wide Area Application Services (WAAS), disponível para download em www.cisco.com. Acessado em 18 de Agosto de 2009.

CISCO, Documentation - Cisco Systems - Disponível em http://www.cisco.com/en/US/prod/collateral/contnetw/ps5680/ps6870/solution_overview_gain_strategic_advantages_with_cisco_wide_area_application_services_WAAS.html – Acessado em 13 de abril de 2009.

CISCO, Documentation - Cisco Systems - Disponível em http://www.cisco.com/support/br/public/nav/III_280484571_1_212.shtml. Acessado em 10 de março de 2009.

FILIPPETTI, Marco, WAN Optimization – A bola da vez? – Disponível em <http://blog.ccna.com.br/2009/01/06/wan-optimization-a-bola-da-vez>. Acessado em 10 de abril de 2009.

FILIPPETTI, Marco, Tecnologias utilizadas em aceleradores WAN – Disponível em <http://blog.ccna.com.br/2009/05/08>. Acessado em 12 de maio de 2009.

PAINE, Stephen; BURNETT, Steven. Criptografia e Segurança: o Guia Oficial RSA, 1ª Edição. Ed. Campus. 2002.

Vtun – Disponível em <http://vtun.sourceforge.net/>. Acessado em 15 de Setembro de 2009.

Zlib – disponível em <http://zlib.net/>. Acessado em 20 de Setembro de 2009.

LZO – disponível em <http://www.oberhumer.com/opensource/lzo/lzodoc.php>. Acessado em 25 de Setembro de 2009.