CAROLINE HELENA MIRANDA DA LUZ

CONTROLE DOS ENLACES PARA PROVISIONAMENTO DE QOS EM TÚNEIS VPN

Projeto de Conclusão de Curso apresentado ao Curso de Ciência da Computação, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Universidade do Sul de Santa Catarina

Orientador Prof°. Dr. Fernando Cerutti

Palhoça, 2006

CAROLINE HELENA MIRANDA DA LUZ

CONTROLE DOS ENLACES PARA PROVISIONAMENTO DE QOS EM TÚNEIS VPN

Projeto de Conclusão de Curso apresentado ao Curso de Ciência da Computação, como requisito à obtenção do título de Bacharel em Ciência da Computação.

Universidade do Sul de Santa Catarina

_____, ___de_____de_____.
Local dia mês ano

_____
Prof°. Dr. Fernando Cerutti
Universidade do Sul de Santa Catarina

_____
Profª. Dra. Maria Inés Castiñeira
Universidade do Sul de Santa Catarina

_____
Profª. Msc. Kathia Regina Lemos Jucá
Universidade do Sul de Santa Catarina

DEDICATÓRIAS

Aos meus pais, que me acompanharam durante toda essa trajetória, tornando possível a realização deste trabalho científico.

À minha irmã, por sempre me apoiar e incentivar nas horas difíceis.

AGRADECIMENTOS

Ao meu Orientador, Prof°. Dr. Fernando Cerutti, pelas recomendações, dedicação e acompanhamento integral deste trabalho científico.

Ao Coordenador do Curso de Ciência da Computação Mauro Notarnicola Madeira, pelo excelente acompanhamento ao longo dos cinco anos de estudo.

Ao Prof°. Coordenador dos Trabalhos de Conclusão de Curso Mauro Pacheco Ferreira, pelas orientações prestadas durante o desenvolvimento do projeto.

Às professoras Maria Inês Castiñeira e Kathia Regina Lemos Jucá, por terem aceitado o convite para participação na banca examinadora.

Ao Prof°. Luis Haroldo Mattos, que ministra cursos de Formação Cisco CCNA no Serviço Nacional de Aprendizagem Comercial - SENAC, pela grande disposição em auxiliar-me, acompanhando-me em diversas etapas práticas do projeto.

Ao Sidnei Rodrigo Basei, pela incondicional amizade e contribuição no desenvolvimento prático do projeto.

A todos que, direta ou indiretamente, apoiaram-me durante toda essa trajetória.

RESUMO

Este trabalho é sobre o monitoramento dos enlaces para provisionamento de QoS em túneis VPN. Uma solução de VPN foi implantada com software livre em plataforma Linux, e a utilização da banda nos enlaces WAN foi monitorada através de ferramentas de geração e medição de tráfego. O ambiente foi testado na ausência de QoS e com uso de DiffServ sob várias condições de tráfego para 3 perfis distintos: o túnel da VPN, um fluxo de voz e um tráfego *Best Effort*, todos utilizando UDP na camada de transporte. Os resultados foram medidos em termos de perdas de pacotes percentuais e variação do atraso nos pacotes (*jitter*). Utilizou-se um pacote estatístico para verificação das influências do uso de QoS e de fatores como tamanho dos pacotes e banda usada em cada um dos fluxos. A arquitetura de marcação DiffServ utilizada nos roteadores mostrou-se efetiva para manutenção da qualidade do túnel VPN.

Palavras-chave: segurança**,** rede private virtual, qualidade de serviço, desempenho

ABSTRACT

This work is about the monitoring of the links for provisioning of QoS in tunnels VPN. A VPN solution was implanted with free software in Linux platform, and the use of the band in the WAN links was monitored through the tools of generation and measurement of traffic. The environment was tested in the absence of QoS and with use of DiffServ under several conditions of traffic for 3 distinct profiles: the VPN tunnel, flow of voice, traffic Best Effort, all using UDP in the transport layer. The results had been measured in terms of losses of percentile packages and variation of the delay in the packages (to jitter). It was used a statistical package to check the influences of the use of QoS and factors such as: size of the packages and the band used in each flow. The architecture of the DiffServ marking used in the routers has shown being effective for maintenance of the quality of VPN tunnel.

Key words: security, virtual private network, quality of service, performance

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE SIGLAS

AAA: *Authentication, Authorization and Accounting*

ARP: *Address Resolution Protocol*

BSD: *Berkeley Software Distribution*

CHAP: *Challenge Handshake Authentication Protocol*

CBQ: *Class-Based Queuing*

CL: *Controlled-Load*

CQ: *Custom Queuing*

COS: *Class of Service*

DIFFSERV: *Differentiated Services*

DNS: *Domain Naming System*

DSCP: *Differentiated Service CodePoint*

FIFO: *First-In First-Out*

FTP: *File Transfer Protocol*

GL: *Guaranteed Load*

GRE: *Generic Routing Encapsulation*

HTTP: *Hypertext Transfer Protocol*

ICMP: *Internet Control Message Protocol*

IEEE: *Institute of Electrical and Electronics Engineers*

IETF: *Internet Engineering Task Force*

INTSERV: *Integrated Services*

IP: *Internet Protocol*

IPSEC: *Internet Protocol Security*

IPX: *Internetwork Packet Exchange*

ISAKMP: *Internet Security Association and Key Management Protocol*

ISO: *International Organization for Standardization*

ISP: *Internet Service Provider*

L2F: *Layer 2 Forwarding*

L2TP: *Layer 2 Tunneling Protocol*

LAN: *Local Area Network*

MD5: *Message Digest 5*

MTU: *Maximum Transmission Unit*

NAS: *Network Authentication Service*

OSI: *Open System Interconnection*

PAP: *Password Authentication Protocol*

PHB: *Per Hop Behavior*

PPP: *Point-to-Point Protocol*

PPTP: *Point-to-Point Tunneling Protocol*

PQ: *Priority Queuing*

QOS: *Quality of Service*

RADIUS: *Remote Authentication Dial In User Service*

RARP: *Reverse Address Resolution Protocol*

RFC: *Request For Comments*

RSVP: *Resource Reservation Protocol*

SMTP: *Simple Mail Transfer Protocol*

TCP: *Transmission Control Protocol*

TCP/IP: *Transmission Control Protocol/Internet Protocol*

TFTP: *Trivial File Transfer Protocol*

TOS: *Type of Service*

UDP: *User Datagram Protocol*

UNISUL: Universidade do Sul de Santa Catarina

VPN: *Virtual Private Network*

WAN: *Wide Area Network*

WFQ: *Weighted Fair Queuing*

SUMÁRIO

1 INTRODUÇÃO

Uma *Virtual Private Network* (VPN) ou Rede Privada Virtual tem como objetivo usar uma rede pública para transferência de informações, de modo seguro, entre redes corporativas ou usuários remotos.

Uma das principais funções das VPNs é a segurança. As redes públicas são consideradas não confiáveis, tendo em vista que os dados que nela trafegam estão sujeitos à interceptação e captura. O uso de VPN torna as redes públicas não confiáveis, como a Internet, em meios altamente confiáveis, fornecendo confidencialidade, integridade e autenticidade.

Existe, porém, carência de garantias de qualidade no tráfego. Segundo Chin (1998), em aplicações onde o tempo de transmissão é crítico, podem ocorrer problemas de desempenho e atrasos na transmissão sobre os quais a organização não tem nenhum tipo de gerência ou controle.

Estas garantias podem ser oferecidas através da aplicação de Qualidade de Serviço (QoS) na VPN.

A Internet é uma rede que possui serviço de melhor esforço para a troca de informação, porém, não oferece garantia de que os dados sejam entregues. A QoS deve ser aplicada para diferenciar os diversos fluxos de dados que passam pela Internet, como voz, vídeo e dados. Por exemplo, fluxos de áudio e vídeo têm exigências rígidas quanto ao atraso,

já os de dados não podem tolerar a perda de pacotes. A QoS possibilita tratar diferenciadamente os fluxos.

Existem duas abordagens principais: Serviços Integrados (IntServ) e Serviços Diferenciados (DiffServ). Estas tecnologias estão descritas nas seções 4.3 e 4.4, respectivamente.

Nessa proposta, os pacotes serão marcados em classes, usando-se a abordagem DiffServ.

1.1 Problematização

Atualmente, muitas empresas têm a necessidade de interligar suas filiais à matriz de forma barata e segura. Além disso, gerentes precisam ter acesso do computador de casa ao servidor corporativo de suas empresas para que também possam produzir fora dela.

A maneira mais comum de interligar *Local Area Networks* (LANs) ou Redes de Área Local é através de circuitos dedicados de alta velocidade, como por exemplo, o *Frame Relay*. Porém, o custo é relativamente alto, havendo necessidade da implantação de outra tecnologia mais barata e que continue sendo segura.

É preciso ter a garantia de que essa tecnologia irá funcionar corretamente, ou seja, se os dados alcançarão seu destino com a largura de banda adequada e se, parâmetros como variação do atraso e perda estarão dentro do limite proposto.

Caso o funcionamento da VPN não atenda aos requisitos necessários, poderá gerar sérios transtornos, como prejuízos para as empresas, funcionários e clientes. Além disso, a total integridade dos dados deve ser mantida, para evitar que estes cheguem ao seu destino

com erros. (CHIN, 1998)

O problema pode ser constituído na seguinte situação, como mostrado na Figura 1.

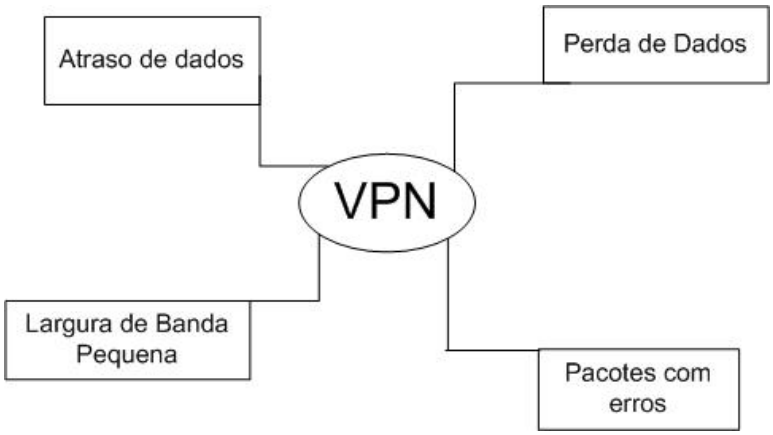

Figura 1 - Problemas relacionados à VPN

A criação de um túnel, através da Internet, para transmissão de dados pode suprir a necessidade atual das empresas por ser uma alternativa de baixo custo em relação aos *links* dedicados ou às redes de pacotes (como *Frame Relay* ou X25).

Para garantir qualidade nos serviços através dos túneis é necessário aplicar QoS nessa solução. O principal problema no provisionamento de QoS nos túneis é como monitorar os enlaces.

1.2 Justificativa

A implementação de uma VPN é uma ótima solução para redução de custos das empresas no que diz respeito ao acesso remoto de dados. Além disso, o uso desta tecnologia é

geralmente fácil, variando de acordo com o negócio de cada organização, e bastante eficiente. (CHIN, 1998).

As VPNs também resolvem os problemas de segurança através do uso de criptografia, fornecendo confidencialidade, integridade e autenticidade.

Embora as empresas utilizem a VPN, não se tem garantia de que ela realmente funcione de maneira adequada. Este projeto propôs uma técnica para que esse tunelamento atendesse aos requisitos de usuário necessários para o devido funcionamento de uma rede privada virtual. Por isso se fez necessário o monitoramento do enlace, oferecendo garantias de banda, perdas e variação do atraso nas comunicações de missão crítica. Diversos autores apresentam estudos nesta área. Referenciando alguns: (STALLINGS, 2002; PAXSON, 1998; FERGUSON; HUSTON, 1998; BRADEN, 1997).

Desse modo, a empresa, funcionários e clientes são beneficiados com o correto funcionamento da rede, evitando, por exemplo, o atraso na entrega de pacotes altamente importantes.

1.3 Objetivos

Os objetivos deste trabalho foram divididos em objetivo geral e objetivos específicos.

1.3.1 Objetivo Geral

Estabelecer uma solução de QoS, com uso da arquitetura *Differentiated Services* (DiffServ), para garantir as métricas de banda, *jitter* e perdas, em túneis VPN.

1.3.2 Objetivos Específicos

- Definir um ambiente de trabalho;
- Configurar VPN;
- Implantar QoS;
- Testar as condições de tráfego a partir dos enlaces que precisam de garantia;
- Monitorar o uso do enlace;
- Verificar limites de interferência dos tráfegos;
- Visar o controle de banda para aplicações;
- Estabelecer casos de variação de QoS.

1.4 Recursos Previstos

Para o desenvolvimento deste projeto foram necessários recursos de *hardware*, *software* e *peopleware*.

Os *hardwares*, componentes responsáveis pela parte física do trabalho, se constituem de três roteadores 1721, dois *switches* 2950, um servidor VPN, um servidor Radius, um servidor Iperf, um cliente VPN e um cliente Iperf.

Na área de *software* foram utilizados o Sistema Operacional Linux Fedora Core, o *Openswan*, o *FreeRadius* e o Iperf.

Do grupo de pessoas envolvidas neste estudo, fizeram parte o coordenador, o orientador, o co-orientador e a orientanda.

1.5 Especificação da Proposta de Solução

Este projeto propõe o uso de protocolos auxiliares ao IP, para garantir QoS nas redes com VPN. O IP é um protocolo que não oferece garantias. Portanto, para que o serviço de VPN seja estabelecido é necessária a utilização de protocolos como *Point to Point Tunneling Protocol* (PPTP), *Layer 2 Tunneling Protocol* (L2TP), *Layer 2 Forwarding* (L2F) ou IPSec, que são protocolos de tunelamento. Neste projeto foi utilizado o IPSec, por oferecer um método de autenticação que garante que os pacotes não sejam alterados durante o caminho.

Outro protocolo auxiliar ao IP é o RSVP, que serve para fazer a reserva de recursos de aplicativos específicos da rede, para garantir a qualidade de serviço na VPN. Embora tenha sido utilizado, não obteve resultados satisfatórios, como será comentado posteriormente na conclusão.

Depois de implantadas as arquiteturas, foram efetuados testes com um software de geração de tráfego e medidas algumas variáveis (perdas, *jitter* e largura de banda) para avaliação do desempenho.

A Figura 2 demonstra a topologia da solução proposta, onde estão representadas duas redes locais, conectadas por três roteadores. O túnel VPN, configurado com o protocolo de tunelamento IPSec, faz conexão de uma rede local com a outra. Cada roteador está configurado com a arquitetura DiffServ, para a implantação de QoS no túnel.

Na rede local da matriz estão localizados três servidores: Radius, VPN e Iperf, onde está o administrador da rede. Na rede local da filial, encontram-se um cliente VPN e um Iperf, onde se situam os usuários da VPN.

As especificações de hardware e software seguem o modelo abaixo:

- **Sistema operacional:** Linux Fedora Core 3, instalado nos servidores e nos clientes, possuindo Kernel versão 2.6.9-1.667.
- **Roteadores:** Todos são Cisco, modelo 1721, possuindo IOS (tm) C1700 *Software* (C1700-IPBASE-M), versão 12.3(6c).
- **Switches:** Todos são Cisco, modelo 2950.
- **Servidor/Cliente VPN:** Software Linux Openswan, versão U2.4.5, disponível em <http://www.openswan.org>.
- **Servidor Radius:** Software FreeRadius, versão 1.1.2, disponível em <http://www.freeradius.org>.
- **Servidor/Cliente Iperf:** Software Iperf, versão 2.0.2, disponível em <http://dast.nlanr.net/Projects/Iperf/>.

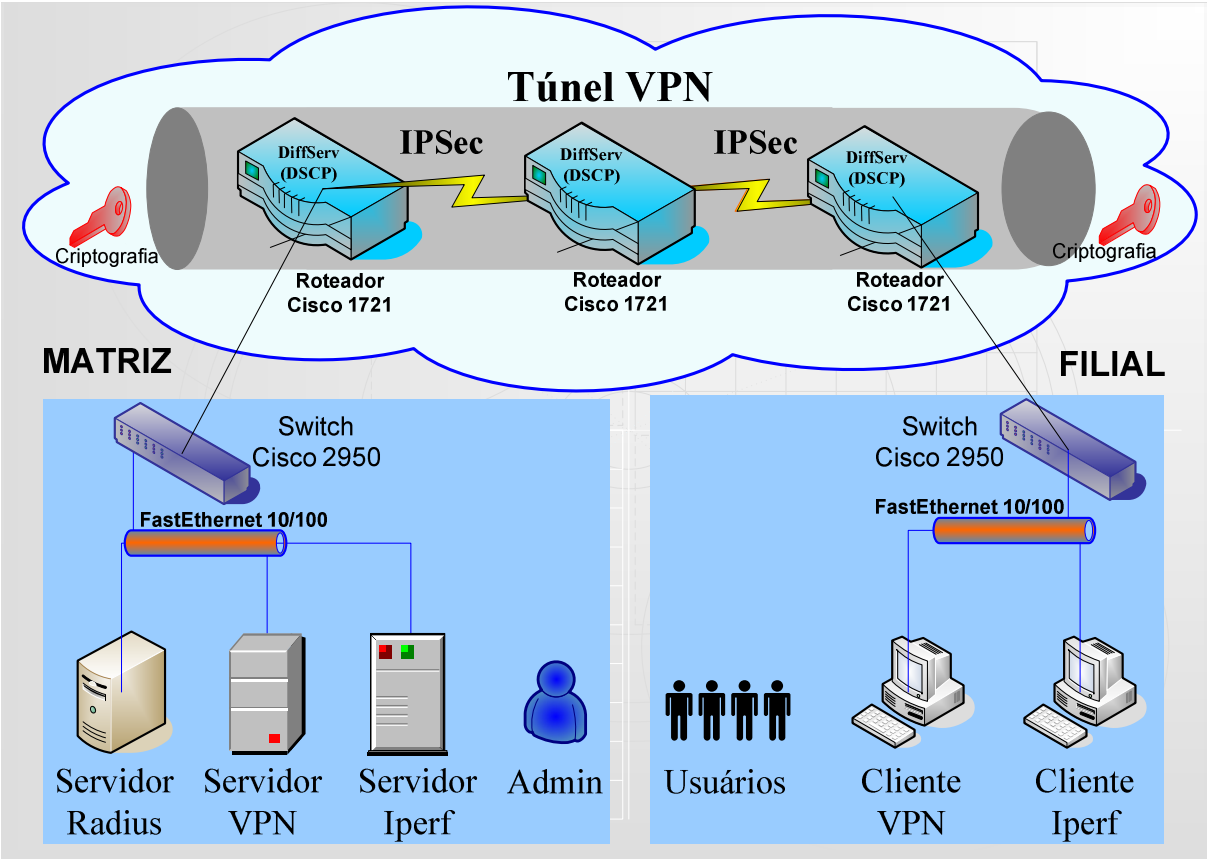

Figura 2 - Topologia da Proposta de Solução

A seguir foi realizada a descrição detalhada dos elementos da Figura 2:

IPSec: É um dos principais protocolos para segurança relacionada com VPN para uma rede IP. Oferece transferência segura de informações fim a fim através de rede pública ou privada. Este protocolo foi definido no RFC 2401 e descrito na seção 3.4.

DiffServ (DSCP): Tecnologia de QoS que utiliza marcação de pacotes para classificar níveis de prioridade. Está descrita na seção 4.4.

Nuvem internet: Internet controlada, ambiente onde foi testado o controle de tráfego.

Roteador 1721: Dispositivo de camada de rede que determina o melhor caminho por onde os dados devem ser transmitidos. Os três roteadores acima estão configurados com a arquitetura de QoS DiffServ.

Switch 2950: Dispositivo da camada de enlace que filtra, encaminha e preenche quadros baseados no endereço de destino de cada quadro. Cada *switch* está conectado a porta *FastEthernet* 10/100 de um roteador, formando uma LAN.

Criptografia: Os dados são cifrados na origem e decifrados no destino, através do método de chave pública/chave privada, que será estudada na seção 3.2.

Servidor Radius: O Radius é um protocolo de autenticação que provê um sistema de segurança Cliente/Servidor. O cliente faz um pedido de autenticação enviando o usuário e uma senha cifrada. Chegando no servidor, a senha é decifrada e se os dados estiverem corretos, a conexão é estabelecida, caso contrário, o usuário é negado. Este protocolo foi definido no RFC 2138 e detalhado na seção 3.5.2.

Servidor VPN: Estabelece uma conexão de VPN;

Cliente VPN: Após ter se autenticado no servidor Radius, o usuário estabelece uma conexão com o servidor VPN, onde pode usufruir os benefícios da VPN;

Servidor/Cliente Iperf: Software de geração de tráfego e medição.

Admin: São os administradores da rede. Responsáveis pelo gerenciamento de toda a rede da empresa, inclusive da VPN com o controle de tráfego.

Usuários: Pessoas que estão na filial e utilizam a VPN da empresa.

1.6 Delimitação do Trabalho

As seguintes delimitações foram propostas para este trabalho:

- QoS na LAN, especificada no IEEE 802.1p, não faz parte da proposta deste projeto. A QoS foi aplicada somente na *Wide Area Network* (WAN) ou Rede de Longa Distância, onde passa o túnel VPN;
- Houve QoS somente na rede privada, ou seja, em todos os roteadores que estão sob controle do ambiente de rede, não havendo testes de controle de tráfego na rede externa ou Internet;
- Esta solução não é genérica, ou seja, não significa que irá funcionar em qualquer ambiente. A topologia proposta foi apresentada na Figura 2;
- Os tráfegos são sintéticos, ou seja, foram gerados sob controle de uma ferramenta de geração de tráfego e medição;
- Todas as ferramentas que foram usadas no desenvolvimento deste projeto já existem. Não foram desenvolvidas interfaces de console para integrar tais ferramentas;

- O protocolo de criptografia IPSec não foi comparado com os demais protocolos, sendo que ele foi escolhido por oferecer um método de autenticação que garante que os pacotes não sejam alterados durante o caminho.

1.7 Metodologia Científica Aplicada ao Trabalho

Conforme Silva e Menezes (2005, p. 9), "A metodologia tem como função mostrar a você como andar no "caminho das pedras" da pesquisa, ajudá-lo a refletir e instigar um novo olhar sobre o mundo: um olhar curioso, indagador e criativo."

Silva e Menezes (2005, p. 20) também argumentam que "pesquisa é um conjunto de ações, propostas para encontrar a solução para um problema, que têm por base procedimentos racionais e sistemáticos. A pesquisa é realizada quando se tem um problema e não se têm informações para solucioná-lo."

"O tipo de pesquisa consiste em informar qual o desenho que a pesquisa terá, ou seja, se a pesquisa será bibliográfica, ou descritiva, ou experimental, ou estudo de caso, ou documental, etc." (UNIVERSIDADE DO SUL DE SANTA CATARINA, 2003, p. 41)

A pesquisa pode ser classificada conforme vários aspectos, listadas a seguir, de acordo com Silva e Menezes (2005, p. 20):

- **Do ponto de vista de sua natureza**: pesquisa básica, pesquisa aplicada.
- **Do ponto de vista da forma de abordagem do problema**: pesquisa quantitativa, pesquisa qualitativa.

- **Do ponto de vista de seus objetivos**: pesquisa exploratória, pesquisa descritiva, pesquisa explicativa.
- **Do ponto de vista dos procedimentos técnicos**: pesquisa bibliográfica, pesquisa documental, pesquisa experimental, levantamento, estudo de caso, pesquisa expost-facto, pesquisa-ação, pesquisa participante.

Os tipos de pesquisas adotadas neste trabalho foram comentadas a seguir, conforme Silva e Menezes (2005, p. 21):

> - Pesquisa aplicada: gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.
> - Pesquisa quantitativa: considera que tudo pode ser quantificável, o que significa traduzir em números opiniões e informações para classificá-las e analisá-las. Requer o uso de recursos e de técnicas estatísticas (percentagem, média, moda, mediana, desvio-padrão, coeficiente de correlação, análise de regressão, etc.).
> - Pesquisa descritiva: visa descrever as características de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis. Envolve o uso de técnicas padronizadas de coleta de dados: questionário e observação sistemática. Assume, em geral, a forma de Levantamento.
> - Pesquisa explicativa: visa identificar os fatores que determinam ou contribuem para a ocorrência dos fenômenos. Aprofunda o conhecimento da realidade porque explica a razão, o "porquê" das coisas. Quando realizada nas ciências naturais, requer o uso do método experimental, e nas ciências sociais requer o uso do método observacional. Assume, em geral, a formas de Pesquisa Experimental e Pesquisa Expost-facto.
> - Pesquisa bibliográfica: quando elaborada a partir de material já publicado, constituído principalmente de livros, artigos de periódicos e atualmente com material disponibilizado na Internet.
> - Pesquisa experimental: quando se determina um objeto de estudo, selecionam-se as variáveis que seriam capazes de influenciá-lo, definem-se as formas de controle e de observação dos efeitos que a variável produz no objeto.

1.8 Estrutura do Trabalho

O trabalho foi dividido em oito capítulos. Neste primeiro é apresentada a introdução, a problematização, a justificativa, o objetivo geral e os específicos, os recursos de *hardware*, *software* e *peopleware* necessários para o desenvolvimento do projeto, a proposta de solução, a delimitação e a metodologia científica aplicada ao trabalho.

No segundo são apresentados conceitos relativos aos protocolos básicos de rede, abordando os modelos *Open System Interconnection* (OSI) e *Transmission Control Protocol/Internet Protocol* (TCP/IP).

No terceiro são estudadas as redes privadas virtuais e seus principais conceitos, como criptografia, protocolos de tunelamento e servidores de autenticação.

No quarto são estudados conceitos relativos à qualidade de serviço, abrangendo os mecanismos de enfileiramento e as duas arquiteturas: IntServ e DiffServ.

No quinto são apresentados conceitos relativos a análise de desempenho em relação à QoS aplicada nos túneis VPN, mostrando o controle de tráfego gerado, bem como as métricas de perda, *jitter* e largura de banda.

No sexto é apresentada a modelagem, onde se explica com detalhes o processo que acontece ao longo do projeto, através de um fluxograma, figuras e textos.

No sétimo são mostrados como foram realizados os testes, assim como também a análise dos resultados através de meios estatísticos.

O capítulo oito diz respeito à conclusão, onde são apontadas a conclusão, as dificuldades do trabalho e apresentadas sugestões para trabalhos futuros.

2 PROTOCOLOS DE REDE

Neste capítulo é apresentado o Modelo de Referência OSI, pois ele é a base para o entendimento de vários tópicos deste trabalho. Porém, o principal enfoque desta área de conhecimento é apresentar os protocolos básicos de rede. Foi analisada a pilha de protocolos TCP/IP, fazendo uma abordagem de seus principais protocolos: IP, TCP e UDP.

2.1 Modelo de Referência *Open System Interconnection* (OSI)

O modelo de referência OSI foi criado pela *International Organization for Standardization* (ISO) com o intuito de garantir a interoperabilidade entre redes. Há algumas décadas atrás, o crescimento de redes aumentou muito e várias delas foram criadas através de implementações diferentes de *hardware* e de *software*, fazendo com que fossem incompatíveis. Assim, a comunicação entre redes com diferentes especificações tornou-se difícil. Com o modelo OSI foi possível estabelecer comunicação entre redes que antes eram impossíveis.

Diógenes (2002, p.24) reforça que “ao contrário do que muitos pensam, o modelo OSI não é um protocolo, nem um conjunto deles. Na realidade ele é um conjunto de normas para permitir que fabricantes diferentes possam criar produtos interoperáveis”.

O modelo de referência OSI dividiu a rede em 7 camadas. Essa divisão possui algumas vantagens porque reduz a complexidade, padroniza as interfaces, facilita a engenharia modular, garante a tecnologia interoperável, acelera a evolução e simplifica o ensino e a aprendizagem.

Diógenes (2002) destaca e explica essas camadas:

- **Camada 7 (Aplicação)** - fornece serviços de rede aos aplicativos do usuário;
- **Camada 6 (Apresentação)** – faz a formatação dos dados, garante que os dados cheguem legíveis no sistema receptor;
- **Camada 5 (Sessão)** – estabelece, gerencia e termina sessões entre aplicativos;
- **Camada 4 (Transporte)** – segmenta os dados do sistema remetente e monta-os em seqüência no sistema receptor;
- **Camada 3 (Rede)** – onde ocorre o roteamento. Seleção do melhor caminho;
- **Camada 2 (Enlace)** - fornece trânsito confiável de dados através de um *link* físico;
- **Camada 1 (Física)** - define as especificações elétricas, mecânicas, funcionais e de procedimentos para ativar, manter e desativar o *link* físico entre sistemas finais.

2.2 *Transmission Control Protocol/Internet Protocol* (TCP/IP)

Durante a Guerra Fria, surgiu a necessidade do Departamento de Defesa dos Estados Unidos terem que criar um protocolo de rede confiável, que fosse capaz de continuar funcionando mesmo se algumas regiões sofressem ataques e fossem destruídas. Inicialmente foi desenvolvido o protótipo (ARPANET). Porém era bastante instável, sofria constantes quedas. Então, foi iniciado um estudo para criar um conjunto de protocolos mais eficiente. Em 1970, foi desenvolvido o TCP/IP.

Kurose e Ross (2003, p. 162) explicam que este conjunto "foi elaborado antes dos PCs e das estações de trabalho, antes da proliferação das *Ethernets* e de outras tecnologias de redes locais, antes da *Web*, da recepção de vídeo e do bate papo virtual".

O TCP/IP possuía algumas vantagens sobre os demais protocolos, entre elas, tinha menor custo e era leve. Em 1983, foi integrado na versão 4.2 do UNIX da *Berkeley Software Distribution* (BSD), logo vindo a se tornar o padrão Internet, permitindo a comunicação de milhões de computadores no mundo todo.

A união dos dois protocolos, TCP e IP, permitiu que eles fornecessem uma quantidade maior de serviços.

Diógenes (2002) acredita que o protocolo TCP/IP foi dividido em um modelo de quatro camadas para melhor ser entendido e estudado. São elas:

- **Aplicação** – Compreende as três últimas camadas do modelo OSI: Aplicação, Apresentação e Sessão. É onde acontecem serviços referentes à transferência de arquivos, correio eletrônico, *logon* remoto, gerenciamento de rede e gerenciamento de nomes;

- **Transporte** – É equivalente à camada de transporte do modelo de referência OSI. Suas principais funções são controlar o fluxo de dados e fornecer confiabilidade. TCP e UDP são os protocolos usados nesta camada.
- **Internet** – Esta camada corresponde à camada de rede do modelo OSI. Abrange protocolos como: *Internet Protocol* (IP), *Internet Control Message Protocol* (ICMP), *Address Resolution Protocol* (ARP) e *Reverse Address Resolution Protocol* (RARP);
- **Acesso à rede** – Compreende as duas últimas camadas do modelo OSI: Enlace e Física. Responsável pelo estabelecimento de um *link* físico entre os dispositivos e também pelos meios, ou seja, cabos, conectores, *hubs*, etc.

Kurose e Ross (2003), por seu lado, afirmam que este modelo foi dividido em cinco camadas: aplicação, transporte, rede, enlace e física, conforme mostrado na Figura 3.

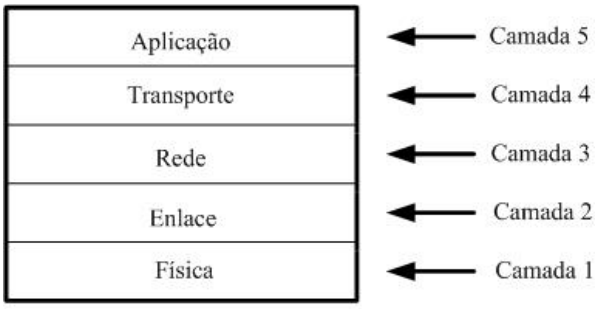

Figura 3 - As cinco camadas do modelo TCP/IP
Fonte: Kurose e Ross (2003, p. 39)

Essas cinco camadas são equivalentes às camadas do modelo OSI. A única diferença é que esse modelo apresentado por Kurose e Ross (2003) não possui as camadas de apresentação e sessão.

A seguir serão estudados alguns dos protocolos que fazem parte da pilha de protocolos TCP/IP.

2.2.1 *Internet Protocol* (IP)

O IP ou Protocolo de Internet é um número de 32 bits usado para identificação de *hosts* na rede. Geralmente é representado na notação decimal para facilitar o entendimento.

Conforme Comer (2001, p.224), "a notação decimal pontilhada é uma forma sintática que o software IP usa para expressar valores binários de 32 bits quando está interagindo com as pessoas. A notação representa cada octeto em decimal e usa um ponto para separar octetos".

Os endereços são divididos em cinco classes, conforme mostra o Quadro 1. Porém, somente as três primeiras (A, B e C) são usadas comercialmente. A classe D é utilizada para endereçamento *multicast* e a classe E foi reservada para o uso futuro.

A classe de um endereço determina a divisão entre a parte de rede e a do *host*. A quantidade de redes e *hosts* que podem ser obtidos em cada classe estão mostradas no Quadro 1.

Quadro 1 – Endereçamento IP

Classe	Faixas de IP	Divisão Rede/Host	Máscara de Sub-Rede Padrão	Número de Sub-Redes	Número de Hosts
A	1 a 126*	R.H.H.H	255.0.0.0	126	16.777.214
B	128 a 191	R.R.H.H	255.255.0.0	16.382	65.534
C	192 a 223	R.R.R.H	255.255.255.0	2.097.150	254
D	224 a 239	endereço multicast			
E	240 a 254	reservado para uso futuro			

Fonte: Adaptado de Diógenes (2002, p. 98)

A função da máscara de sub-rede é informar ao sistema operacional se sub-redes estão sendo utilizadas ou não. Se a máscara de sub-rede for padrão, conforme especificado no Quadro 1, significa que elas não estão sendo usadas.

Se uma organização usa um endereço de rede de classe A, significa que terá disponível até 16.777.214 de *hosts* para definir. Isso só será possível através do uso de roteadores e sub-redes para dividi-los.

> É possível dividir as redes para abrigar um número diferente de *hosts* de acordo com a necessidade de cada instituição. A divisão é feita utilizando máscara de sub-rede. A máscara informa qual parte do endereço IP é referente à rede e ao *host*. Os roteadores realizam uma operação de E Lógico (*and*) entre a máscara de rede e o endereço IP para obter o endereço de rede. (ORTIZ; FERREIRA, 2003, p. 38).

Desse modo, é possível que um computador, situado numa rede, possa trocar informações com outro, localizado numa rede diferente.

O IP é um protocolo da camada de rede que não possui serviço orientado à conexão. Ele usa o serviço de melhor esforço para entrega de dados. Como o pacote vem quase sem garantias, ele pode chegar desordenado, duplicado, ou até mesmo perdido por inteiro.

> Como IP foi projetado para operar através de todos os tipos de *hardware* de rede, o *hardware* subjacente pode se portar mal. Como resultado, os datagramas IP podem ser perdidos, duplicados, atrasados, entregues fora de ordem ou entregues com dados adulterados. É necessário que as camadas mais altas de *software* de protocolo tratem de cada um destes erros. (COMER, 2001, p. 252).

A Figura 4 representa o formato do cabeçalho de um datagrama IP.

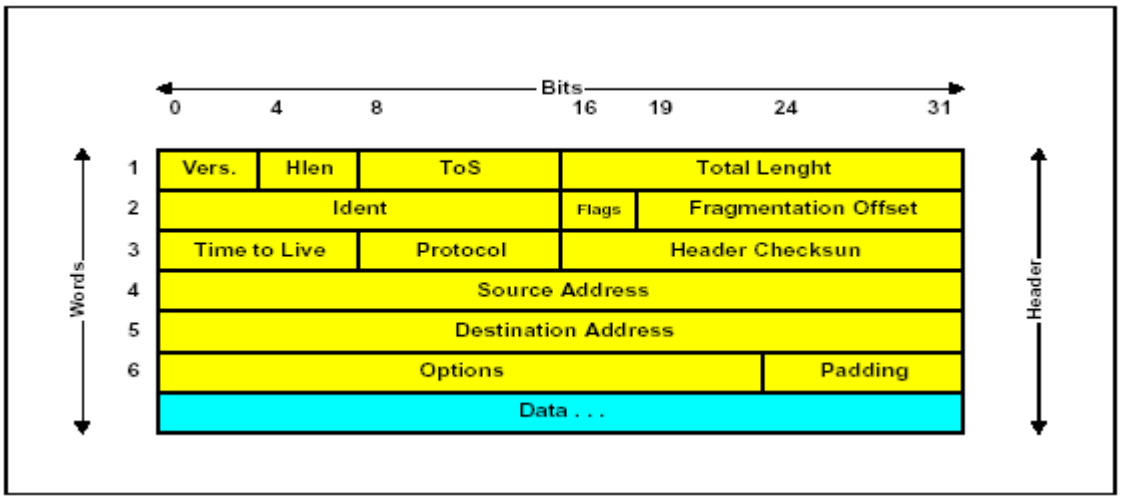

Figura 4 - Formato de Datagrama IP
Fonte: Cirne (2003)

Cirne (2003) explica os campos neste datagrama:

Version: versão do IP (atualmente 4);
Hlen: tamanho do cabeçalho do datagrama;
ToS: tipo do serviço (precedência normal/controle, baixo retardo, alta eficiência, alta confiabilidade), sem garantia de cumprimento;
Total Length: tamanho total do datagrama (máximo de 64 Kbytes);
Ident: identificação do datagrama (único para cada datagrama);
Flags: controle de fragmentação (habilita/desabilita fragmentação, marca de fim do datagrama original);
Fragmentation Offset: deslocamento do fragmento;
TTL: tempo de vida do datagrama (inicia em N, decrementa a cada passagem por um roteador; chegando em 0 (zero), o datagrama é descartado e é gerada uma mensagem de erro);
Protocol: protocolo de nível superior (TCP, UDP);
Header checkksun: verificação de integridade do datagrama;
Source Address: endereço origem (máquina emissora);
Dest Address: endereço destino (máquina receptora);
Options: opções de teste e depuração:
. *Record Route Option*: datagrama guarda endereços de roteadores intermediários por onde passou;
. *Source Route Option*: sistema origem define rota que um datagrama deve seguir;
. *Timestamp Option*: datagrama guarda informação sobre data e hora que chegou aos roteadores intermediários;
Padding: preenchimento;
Data: dados transportados.

2.2.2 *Transmission Control Protocol* (TCP)

"O *Transmission Control Protocol* (TCP) ou protocolo de controle de transmissão é um protocolo confiável, orientado à conexão, fim-a-fim, projetado para encaixar em uma camada hierárquica de protocolos que suportam aplicações multi-redes" (INFORMATION, 1981).

O TCP é um protocolo da camada de transporte, com serviço orientado à conexão, ou seja, é estabelecida uma conexão entre o remetente e o destinatário antes do envio de dados. Ele fornece aos programas aplicativos um serviço de transporte *full-duplex*, com controle de fluxo e confiabilidade. Além disso, ainda possui outras características como: divisão das mensagens enviadas em segmentos, reagrupamento das mensagens na estação de destino, reenviação de tudo o que não foi recebido e reagrupamento das mensagens a partir de segmentos recebidos. Ele foi definido no RFC 793.

Comer (2001) aborda essa questão da confiabilidade fornecida pelos protocolos de transporte, dizendo que o TCP é o protocolo em nível de transporte que fornece confiabilidade no suíte de protocolos TCP/IP.

File Transfer Protocol (FTP), *Hypertext Transfer Protocol* (HTTP), *Simple Mail Transfer Protocol* (SMTP) e *Domain Naming System* (DNS) são exemplos de protocolos que usam o TCP.

O TCP oferece um serviço fim-a-fim, ou seja, transporta mensagens de um computador origem para um destino remoto. Ele usa o IP para fazer a transferência dos dados. As mensagens TCP são encapsuladas em um datagrama IP e enviadas pela rede. O IP não intercepta as mensagens, apenas as trata como dados a serem transferidos.

A Figura 5 ilustra a relação entre o TCP e o IP, num exemplo com dois *hosts* e um roteador. Comer (2001, p. 287) explica que "o TCP vê o IP como um mecanismo que permite ao software de TCP em um *host* trocar mensagens com o software de TCP em um *host* remoto".

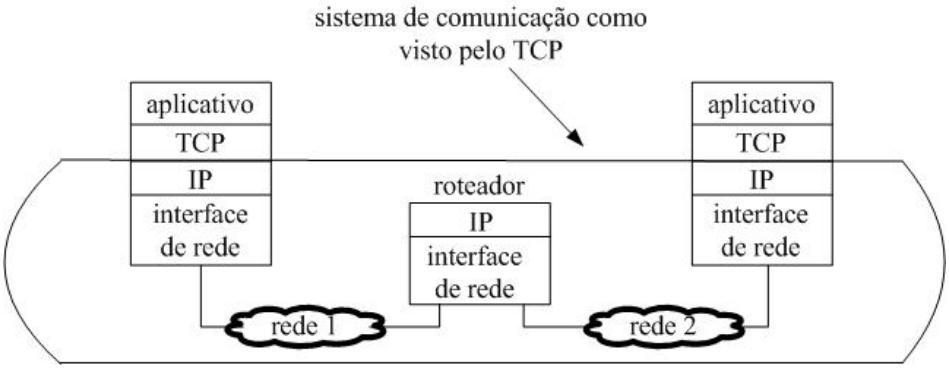

Figura 5 – Ilustração do TCP como protocolo de transporte fim-a-fim
Fonte: Comer (2001, p. 287)

Na fase de transmissão de dados, é estabelecida uma sincronização entre o receptor e o destinatário.

O TCP usa um mecanismo para controle de fluxo chamado janelamento. Esta técnica garante que os dados enviados alcancem o destino.

> Os hospedeiros de cada lado de uma conexão TCP reservam um *buffer* de recepção para a conexão. Quando a conexão TCP recebe *bytes* que estão corretos e em seqüência, ela coloca os dados no *buffer* de recepção. O processo de aplicação associado vai ler os dados a partir desse *buffer*, mas não necessariamente no momento em que os dados são recebidos. Na verdade, a aplicação receptora pode estar ocupada com alguma outra tarefa e pode nem ao menos tentar ler os dados até muito após a chegada deles. Se a aplicação for lenta na leitura dos dados, o remetente pode muito facilmente saturar o *buffer* de recepção da conexão por enviar demasiados dados muito rapidamente. Assim, o TCP fornece um serviço de controle de fluxo para suas aplicações, a fim de eliminar a possibilidade de o remetente saturar o *buffer* do destinatário. (KUROSE; ROSS, 2003, p. 173).

A Figura 6, usando um tamanho de janela 1, demonstra esse processo. O *Host* 1 envia uma mensagem para o *Host* 2. O *Host* 2 confirma o recebimento enviando um *ack* 1 para o *Host* 1. Enquanto o *host* destino não fizer a confirmação, o *host* origem não poderá

enviar novas mensagens. O *Host* 1 retransmitirá os dados que conterem erro com uma taxa de transmissão menor, como é o caso do envio da mensagem 3.

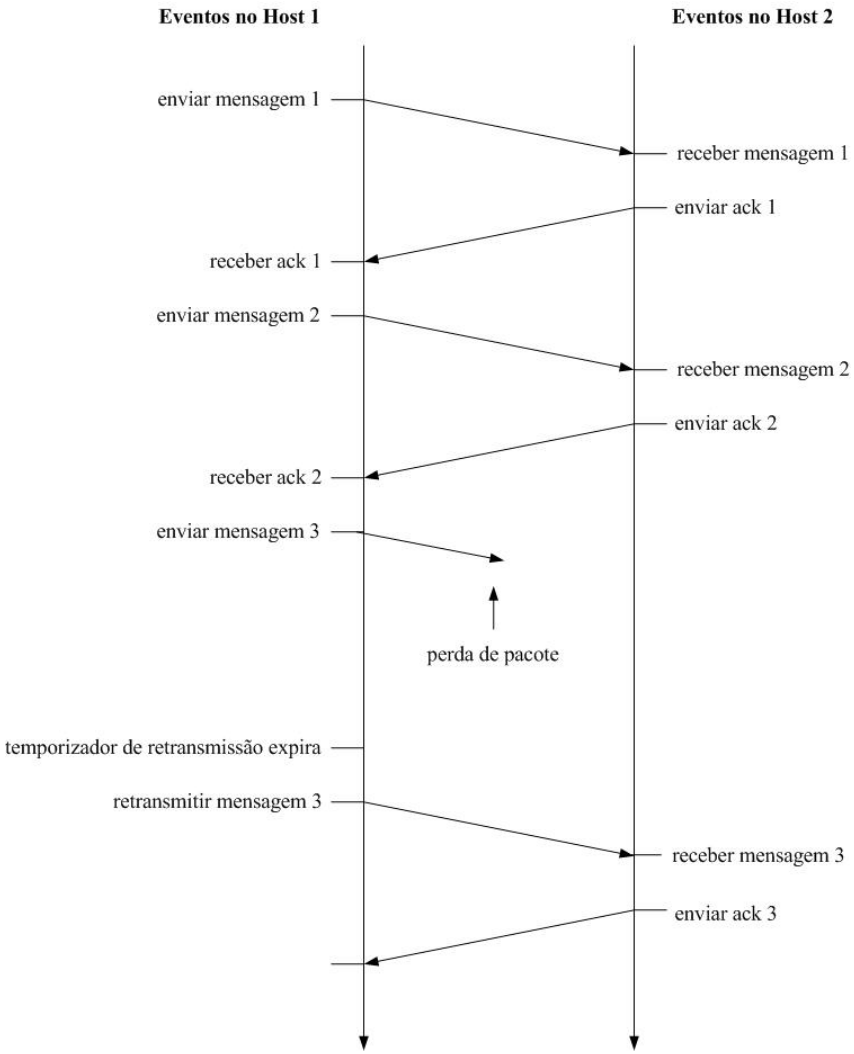

Figura 6 - Exemplo de retransmissão
Fonte: Comer (2001, p. 289)

O tamanho da janela indica a quantidade de dados que podem ser transmitidos até que a origem receba uma confirmação. Por exemplo, com um tamanho de janela 3, um *host* origem poderá enviar três segmentos, e só após receber confirmação de que os dados foram recebidos, ele poderá enviar mais informações. Quanto maior o tamanho da janela, maior a quantidade de *bytes* que podem ser enviados de cada vez.

A Figura 7 ilustra os campos do segmento TCP, seguida de suas explicações, conforme Cirne (2003).

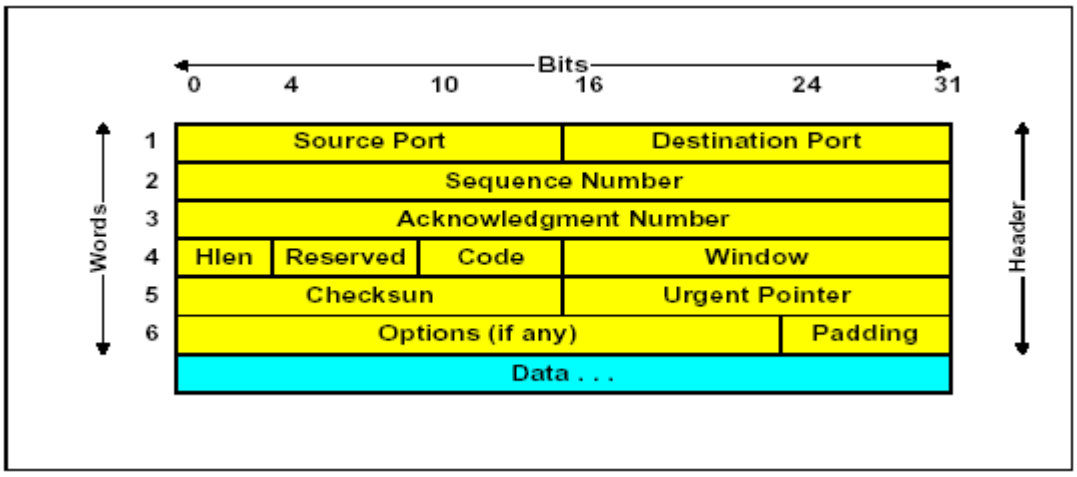

Figura 7 - Formato de Segmento TCP
Fonte: Cirne (2003)

Source/Destination Port: porta de origem/destino (identificam os processos envolvidos na conexão);
Sequence Number: posição do segmento dentro da seqüência de *bytes* da/para a aplicação;
Acknowledgment Number: número do próximo *byte* esperado no destino;
Hlen: tamanho do cabeçalho;
Reserved: reservado;
Code bits: finalidade do segmento:
. URG: campo *Urgent Pointer* é válido;
. ACK: campo *Ack number* é válido;
. PSH: segmento requer um *push*;
. RST: *reset* a conexão;
. SYN: sincronize números de seqüência;
. FIN: origem terminou sua seqüência de *bytes*;
Window: tamanho da janela deslizante (número de segmentos enviados em seqüência antes de receber reconhecimento);
Checksun: verificação de erro no segmento;
Urgent Pointer: posição onde os dados urgentes se encontram no segmento;
Options: tamanho (opcional) máximo de segmento ("*Maximum Segment Size*");
Padding: preenchimento;
Data: dados do segmento.

2.2.3 *User Datagram Protocol* (UDP)

O UDP, protocolo de datagrama do usuário, foi definido no RFC 768. Oferece um serviço sem conexão, não é confiável, não reagrupa as mensagens de entrada e não fornece controle de fluxo. É um protocolo simples que troca datagramas sem confirmações ou entrega garantida. O processamento de erros e a retransmissão devem ser tratados por outros protocolos. É ideal para aplicações que não precisam juntar seqüências de segmentos.

Postel (1980) afirma que este protocolo fornece um procedimento para programas de aplicação para enviar mensagens para outros programas com um mínimo de mecanismo de protocolo.

Alguns exemplos de protocolos que usam o UDP são o DNS e o *Trivial File Transfer Protocol* (TFTP).

Kurose e Ross (2003, p. 138) oferecem uma boa explicação do funcionamento deste protocolo:

> Se o criador da aplicação escolher o UDP em vez do TCP, a aplicação estará 'falando' quase diretamente com o IP. O UDP pega as mensagens do processo de aplicação, anexa os campos de número de porta da fonte e do destino para o serviço de multiplexação/demultiplexação, adiciona dois outros pequenos campos e passa o segmento resultante à camada de rede. A camada de rede encapsula o segmento dentro de um datagrama IP e, em seguida, faz uma tentativa de melhor esforço para entregar o segmento ao hospedeiro receptor. Se o segmento chega ao hospedeiro receptor, o UDP usa os números de porta para entregar os dados do segmento ao processo de aplicação correto.

Na Figura 8 está ilustrado o formato do segmento UDP. Abaixo, de acordo com Cirne (2003), estão as explicações de cada campo.

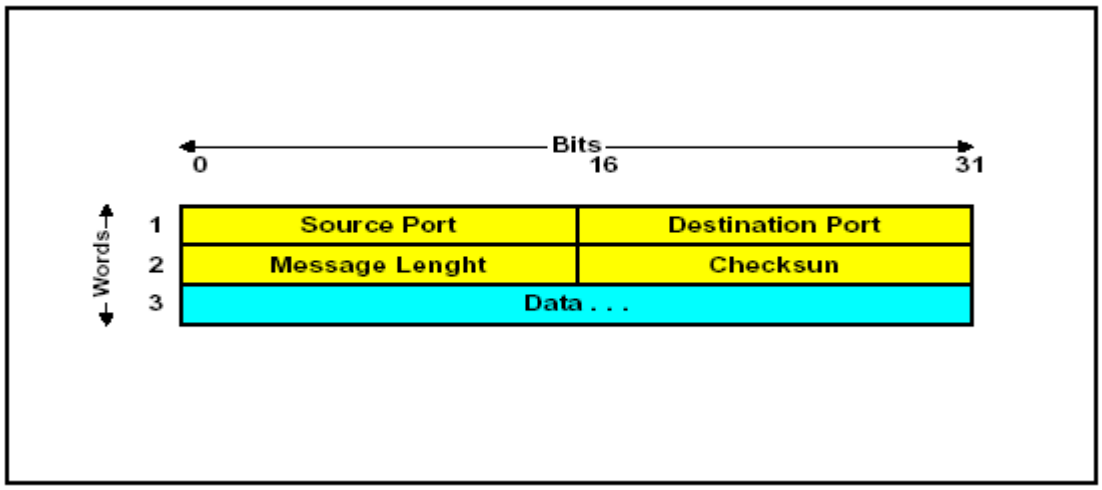

Figura 8 - Formato de Segmento UDP
Fonte: Cirne (2003)

Source/Destination Port: porta de origem/destino (identificam os processos envolvidos na conexão);
Message Lenght: tamanho do segmento;
Checksun: verificação de erro;
Data: início dos dados

2.3 Considerações finais

Este capítulo teve como objetivo estudar os principais protocolos da pilha TCP/IP. O IP, protocolo da camada de rede do modelo OSI, identifica os *hosts* na rede. Seu uso foi imprescindível no desenvolvimento deste projeto, pois ele é responsável por transportar as informações de um local ao outro. O TCP e o UDP são protocolos da camada de transporte que oferecem serviços para que a transferência dos dados possa ser realizada.

3 REDE PRIVADA VIRTUAL (VPN)

Neste capítulo são estudadas criptografia e principais conceitos dos protocolos de tunelamento, onde é feito um estudo mais aprofundado do protocolo de segurança IPSec; e para finalizar é apresentado o processo de autenticação, autorização e contabilização, abordando os principais métodos de autenticação usados pelo Linux e posteriormente, explicando o funcionamento do servidor de autenticação Radius.

3.1 Conceitos Básicos

A VPN ou Rede Privada Virtual consiste na criação de um túnel seguro, através de uma rede pública, geralmente a Internet, que permite a transferência de informações de um ponto ao outro.

Chin (1998) oferece uma boa explicação a respeito do conceito de VPN:

> A idéia 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. 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.

Amâncio (2004, p. 28) também destaca essa idéia comentando que “as VPNs são redes privativas que utilizam a rede pública, geralmente a Internet, integrando ambientes fechados com uma arquitetura aberta”.

O túnel é dito seguro, pois possui uma série de requisitos de segurança. Entre os mais importantes estão: a autenticidade, que verifica a identidade dos usuários, permitindo a entrada somente dos que são autorizados; a integridade, que assegura que os dados não serão alterados durante a transferência; e a confidencialidade, que mantém o conteúdo criptografado, de forma que, mesmo interceptado, não possa ser entendido.

Existem várias aplicações para VPN. As mais importantes são para acesso remoto e conexões de LANs. Ambas são realizadas via Internet.

Na rede privada virtual para acesso remoto, o *host* remoto deve se conectar à Internet, através de um *Internet Service Provider* (ISP) ou Provedor de Serviço de Internet. Após ter estabelecido essa conexão, o *software* de VPN cria uma rede privada virtual entre o *host* remoto e o servidor de VPN através da Internet (CHIN, 1998).

O processo está descrito na Figura 9.

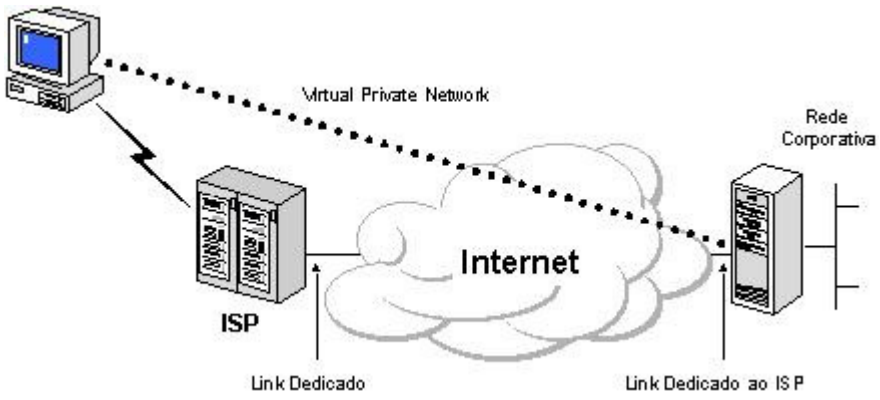

Figura 9 - Acesso Remoto Via Internet
Fonte: Chin (1998)

A conexão entre LANs ou corporações usa circuitos dedicados locais, para interligar à Internet, ao invés de usar circuitos dedicados de longa distância. Chin (1998) argumenta que, dependendo das aplicações, pode-se fazer a utilização de circuitos discados em uma das portas, deixando a LAN corporativa conectada 24h via circuito dedicado local. A Figura 10 ilustra o processo.

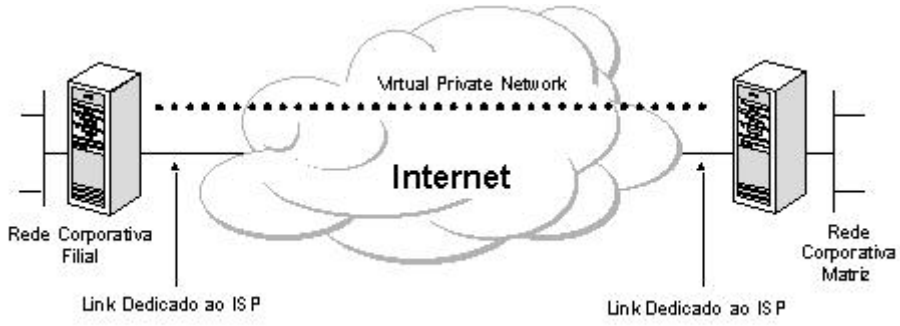

Figura 10 - Conexão de LANs Via Internet
Fonte: Chin (1998)

A grande vantagem da utilização de VPNs é o baixo custo que ela oferece, em relação aos *links* dedicados de longa distância, como o X.25 e o *Frame Relay*. Amâncio (2004) aponta que a redução do custo pode chegar a 30%.

3.2 Criptografia

Para tornar seguro um túnel que passa por uma rede insegura de comunicação de dados, como a Internet, é necessário utilizar a técnica de criptografia.

A criptografia pode ser definida como o estudo de técnicas matemáticas, capaz de transformar uma mensagem de texto cifrado, mantendo assim, o sigilo de seu conteúdo.

Dentre os principais objetivos estão a confidencialidade dos dados, o que garante que ninguém irá ter acesso aos mesmos; a integridade, ou seja, os dados não serão alterados; e a autenticidade, que está relacionada à prova de identidade do remetente e destinatário das informações.

Carvalho (2000, p. 1) oferece uma boa explicação de como ocorre o processo de criptografia:

> Na criptografia existem dois tipos de textos (ou mensagens). O primeiro é a mensagem a ser transmitida, na sua forma original. Este será chamado de texto puro. O texto puro passa por um processo que se denomina encriptação, e assumirá uma nova forma. Esta nova forma, chamada de texto cifrado, é a que será transmitida, e que, quando interceptada por um terceiro, deverá permanecer ininteligível. Ao receber o texto cifrado, o destinatário usará um processo que se chama desencriptação para recuperar o texto puro.

A Figura 11 representa um sistema criptográfico, onde uma mensagem de texto puro (M) é cifrada com a chave de cifragem (K_e) e logo após é decifrada com a chave de decifragem (K_d), voltando a sua forma original.

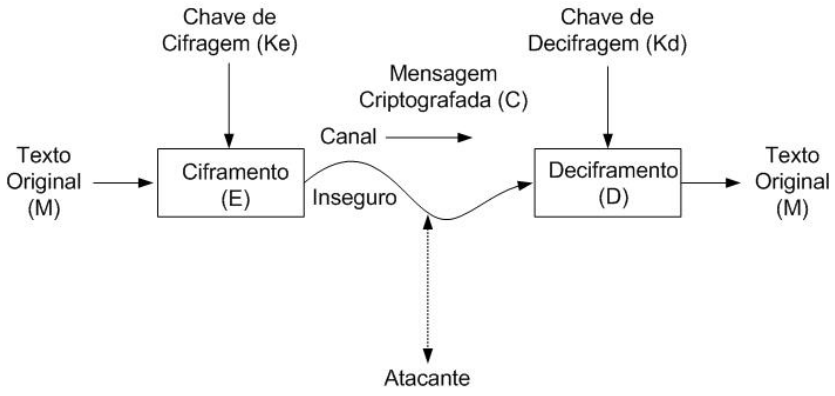

Figura 11 - Modelo de um sistema criptográfico
Fonte: Ortiz; Ferreira (2003, p. 96)

Esse processo, o de cifrar e decifrar, é conseguido através do uso de algoritmos criptográficos. Conforme Ribeiro (2003, p. 13), “os algoritmos criptográficos, com base em

substituições, funções e operadores matemáticos, transformam textos na sua forma original, também conhecido como planos ou claros, em textos cifrados".

Existem dois tipos de sistemas para algoritmos criptográficos. O primeiro é conhecido como criptografia simétrica ou de chaves secretas e o outro, como criptografia assimétrica ou de chaves públicas.

O sistema de criptografia de chaves secretas faz referência ao uso de um par de chaves. Uma serve para cifrar e a outra para decifrar. Geralmente, as chaves usadas para cifrar e decifrar são as mesmas, ou seja, a partir da chave que cifra, pode-se obter a que decifra, e vice-versa. (RIBEIRO, 2003, p. 14).

Carvalho (2000) sustenta essa idéia afirmando que os algoritmos são simétricos porque a chave que cifra é a mesma que decifra.

Triple DES, IDEA, RC5, CAST e RC2 são exemplos de algoritmos de criptografia simétricos.

A criptografia assimétrica faz uso de chave privada e de chave pública. De maneira diferente dos algoritmos simétricos, uma mensagem cifrada com a chave K_e, só pode ser decifrada com a chave K_d, e vice-versa, pois as duas chaves são diferentes, uma depende da outra.

Carvalho (2000) considera que os algoritmos são assimétricos porque possuem chaves diferentes para cifrar e decifrar.

O processo da criptografia assimétrica acontece da seguinte forma:

> Se um usuário **A** deseja receber mensagens criptografadas, primeiro ele calcula as chaves K_{eA} e K_{dA}, mantém secreta K_{dA} e divulga K_{eA}. Quando um segundo usuário **B** enviar uma mensagem criptografada para **A**, ele utiliza a chave pública K_{eA}, e assim somente quem conhecer a chave secreta K_{dA} poderá ler a mensagem original, nesse caso o usuário **A**. (ORTIZ; FERREIRA, 2003, p. 99)

O principal exemplo de algoritmo criptográfico assimétrico é o RSA, usado pela ferramenta *Openswan*..

Com relação à Figura 11, pode-se observar que se $K_e = K_d$, o sistema é chamado de criptografia simétrica; se K_e for diferente de K_d, de criptografia assimétrica.

3.3 Protocolos de Tunelamento

Os protocolos de tunelamento têm como função transmitir os dados através da VPN, garantindo a segurança e a confiabilidade desses dados.

Ortiz e Ferreira (2003, p. 43) afirmam que, "para executar esta tarefa, esses protocolos usam o tunelamento, um método que utiliza as redes públicas para transportar dados de uma rede a outra".

O tunelamento encapsula um protocolo dentro de outro. Os protocolos podem ser os mesmos ou completamente diferentes. A VPN permite a conexão de uma rede remota através da Internet, que é uma rede IP. Porém, nem todas as LANs trabalham com esse protocolo. O tunelamento faz a adaptação de protocolos diferentes, por exemplo, transportando o *Internetwork Packet Exchange* (IPX) através de uma rede IP.

Scott (1999) aborda essa questão, explicando que servidores Windows NT podem usar o NetBEUI e servidores *Novell* usam o IPX. Entretanto, as redes tendem futuramente a usar somente o IP.

Antes de serem enviados pelo túnel, os dados são encapsulados. É criado um cabeçalho do protocolo de tunelamento, contendo informações de roteamento, como o endereço de destino. Após o encapsulamento, os dados são transmitidos e ao chegarem na rede final, são desencapsulados e enviados ao *host* destino.

Tanto Ortiz e Ferreira (2003) quanto Chin (1998) citam alguns requisitos básicos que devem ser seguidos para que uma VPN possa ser implementada:

- **Autenticação de usuários:** sua função é restringir o acesso não autorizado e permitir o acesso de usuários autorizados;
- **Suporte a cartão *Token*:** suportar *Smart Cards* (cartões inteligentes) aumenta a segurança da VPN. Somente quem tiver seu cartão conseguirá utilizar a rede privada.
- **Endereçamento dinâmico:** os endereços devem ser entregues às redes privadas, de modo que eles também continuem privados. No IPSec, os endereços são atribuídos antes da inicialização do túnel.
- **Compressão de dados:** o *Internet Engineering Task Force* (IETF) está estudando possibilidades de compressão de dados para o IPSec, tal como compressão de IP.
- **Criptografia de dados:** o objetivo é proteger os dados de maneira que, mesmo que alguém consiga interceptar os dados, não conseguirá entendê-los, pois estarão incompreensíveis. A criptografia é aplicada antes dos dados serem encapsulados.
- **Gerenciamento de chaves:** responsável por atualizar as chaves de cada lado do túnel. É através delas que os dados podem ser decriptografados. O IPSec utiliza o *Internet Security Association and Key Management Protocol* (ISAKMP), que é um protocolo de gerenciamento de chaves, para negociar uma chave comum.

- **Suporte a multiprotocolo:** o ideal é que pelo menos o IP e o IPX sejam suportados. Protocolos da camada 2, como PPTP e L2TP, suportam mais de um protocolo. Já o de camada 3, como o IPSec, só suporta o IP.

Dentre os protocolos de tunelamento mais utilizados estão o PPTP, o L2F, o L2TP e o IPSec. Os três primeiros são da camada 2 ou camada de enlace do Modelo de Referência OSI e o último é da camada de rede.

No próximo item será abordado o IPSec, protocolo escolhido para implementar a VPN deste projeto, por ser o que melhor atende às necessidades deste cenário.

3.4 *Internet Protocol Security* (IPSec)

O IPSec ou Segurança do Protocolo da Internet é um protocolo de tunelamento, usado para a transmissão de dados de um ponto a outro. É de camada 3 ou camada de rede do modelo OSI. Foi desenvolvido pelo IETF, com a finalidade de proteger os pacotes IP.

Ortiz e Ferreira (2003, p. 53) sustentam essa idéia conceituando o protocolo:

> O IPSec nada mais é que uma série de diretrizes de segurança que protege o tráfego de dados IP, provendo privacidade, integridade e autenticidade. Foi desenvolvido pelo IETF e além de proteger o tráfego dos pacotes IP, contém um protocolo que dita as metodologias de gerenciamento de chaves.

Kent e Atkinson (1998) também reforçam esse conceito dizendo que a meta da arquitetura IPSec é fornecer vários serviços de segurança de tráfego na camada IP, tanto em ambientes IPv4 (o padrão atual) quanto em IPv6 (o IP do futuro).

Surgiu em resposta de por que o TCP/IP não era atualizado para suportar autenticação e criptografia, sendo que dessa forma, todas as redes TCP/IP seriam seguras.

Scott (1999) salienta que desde que o protocolo TCP/IP se tornou tão onipresente, a produção de um sistema de rede seguro é uma evolução natural, desenvolvido quase em paralelo com o sistema existente, de maneira que, fazendo a atualização para produtos e serviços IPSec, só aumentará a segurança.

As principais áreas que o IPSec trata são os algoritmos criptográficos, os algoritmos de autenticação e o gerenciamento de chaves. Estes três componentes são indispensáveis para fornecer métodos de segurança mais completos para um sistema. Os principais algoritmos que estão incluídos no IPSec serão mostrados a seguir:

> - Criptografia: DES, Blowfish, 3-DES, CAST, TWOFISH, entre outros.
> - Autenticação: HMAC, MD5, SHA1, SHA2, entre outros.
> - Gerência de chaves: *Diffie-Hellman-Key-exchanges* para entregar chaves criptográficas entre as partes na rede pública e *Public-key-criptography* para sinalizar trocas do tipo *Diffie-Hellman* e garantir a identificação das duas partes, evitando assim ataques de intrusos no meio do caminho. (AMÂNCIO, 2004, p. 36).

Existem duas maneiras de enviar dados de um ponto a outro. O modo de transporte e o modo túnel.

No primeiro, o *host* possui suporte a IPSec, pois o encapsulamento é feito no próprio *host*. Os dados são enviados de *host* para *host*. Ortiz e Ferreira (2003, p. 53) afirmam que "...o IPSec é incorporado pelo sistema da mesma forma que é feito o empilhamento dos dados TCP/IP. Desta forma, o próprio *host* é o responsável pela segurança...". Somente a carga útil, também conhecida como *payload*, é criptografada, deixando os demais campos IP desprotegidos. A Figura 12 representa como é realizada a transferência de dados no modo de transporte.

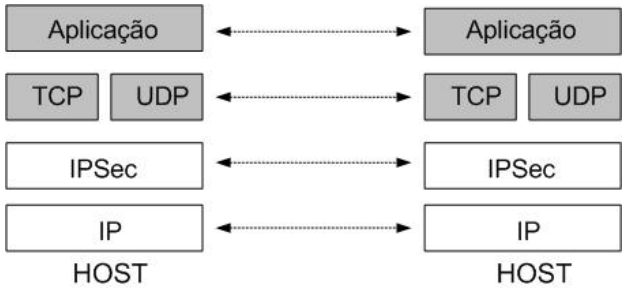

Figura 12 - IPSec no modelo TCP/IP
Fonte: Ortiz; Ferreira (2003, p. 53)

No modo túnel, além da carga útil, o cabeçalho e outros campos intermediários do IP são cifrados. Isso torna este modo mais confiável e seguro que o modo transporte, pois esconde os endereços do remetente e do destinatário. Outra característica importante do último modo é que os *hosts* não possuem suporte a IPSec. Os dispositivos que fazem o encapsulamento e o desencapsulamento são os *gateways*, que estão localizados um em cada ponta da VPN. Quando os dados chegam no *gateway* destino, são desencapsulados e liberados para o *host* final. Ortiz e Ferreira (2003) ressaltam que os responsáveis pela segurança agora são os *gateways* e não mais os *hosts*. A Figura 13 ilustra esse processo.

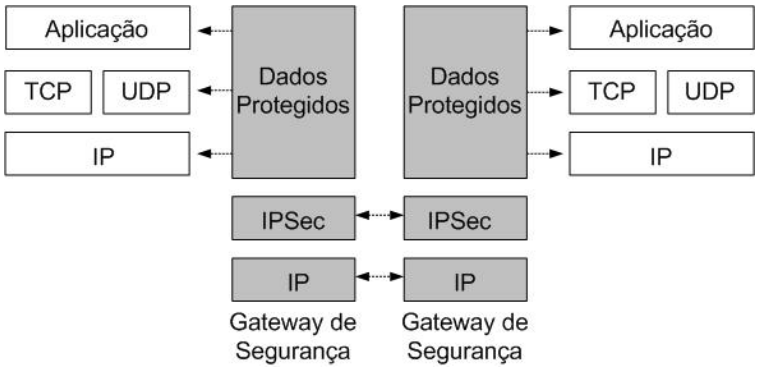

Figura 13 - IPSec e modo de túnel
Fonte: Ortiz; Ferreira (2003, p. 54)

Dentre as vantagens do IPSec sobre outros protocolos de tunelamento estão o suporte aos protocolos criptográficos e o gerenciamento de chaves. Os outros protocolos,

como o PPTP, o L2TP e o L2F não possuem suporte a esses dois importantes componentes, que são essenciais à segurança da rede VPN.

O IPSec também possui algumas desvantagens. Uma delas é o não oferecimento de suporte a multiprotocolo. Ele só suporta pacotes IP. Um outro fator que deve ser levado em consideração é no caso de se usar o modo túnel, os dois *gateways* devem ter políticas de segurança similares. Scott (1999) acredita que arquiteturas diferentes poderiam deixar uma rede sendo menos segura que a outra, de forma que deixaria o sistema bastante suscetível a ataques *hacker*.

3.5 *Authentication, Authorization and Accounting* (AAA)

O *Authentication, Authorization and Accounting* (AAA) é um processo que realiza funções como autorização, autenticação e contabilização de usuários na rede. É ele quem decide se um usuário pode ou não ter acesso à rede.

De acordo com Ortiz e Ferreira (2003, p. 57), "o AAA nada mais é que um processo que identifica e autoriza usuários a ingressarem em uma rede ou em um *site*".

O processo AAA em um servidor Linux pode ser representado pelo fluxograma da Figura 14, seguido das explicações de Ortiz e Ferreira (2003).

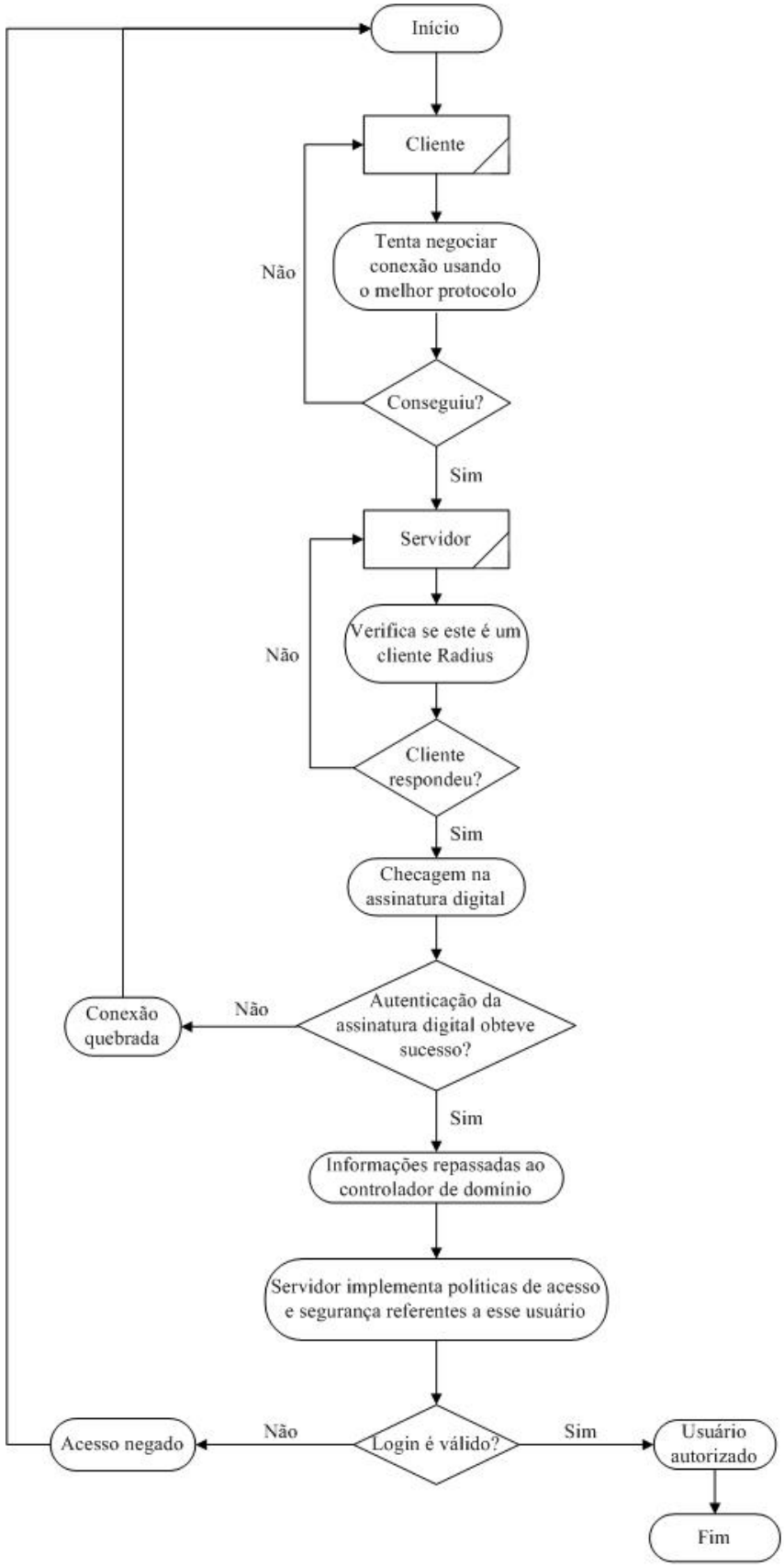

Figura 14 - Processo AAA em servidor Linux com Radius *Server*
Fonte: Adaptada de Ortiz; Ferreira (2003, p. 57)

Na primeira etapa, o servidor tenta negociar conexão com o cliente, usando o melhor protocolo. Caso não seja este o protocolo que o cliente estiver usando, ele tentará conexão com o próximo protocolo mais seguro. Esta rotina será repetida até que o servidor estabeleça uma conexão com o cliente.

Depois de realizada a conexão, o servidor utiliza o *Remote Authentication Dial In User Service* (Radius) para ter garantia de que este é um cliente Radius. Esta verificação é feita por meio do número IP do cliente. Se o cliente responder, ou seja, se for um cliente válido, o servidor irá fazer uma checagem na assinatura digital do pacote.

Se a autenticação da assinatura digital não obtiver sucesso, a conexão será quebrada. Caso contrário, as informações serão repassadas ao controlador de domínio, que é o responsável pela validação das informações de *login* do cliente.

Após isso, o servidor irá implementar políticas de acesso e segurança referentes a esse usuário. Caso as informações de *login* forem aceitas, este usuário é autorizado. De outra forma, o acesso é negado.

A utilização do AAA no Linux oferece várias vantagens à rede. Ortiz e Ferreira (2003, p. 60) apontam algumas delas: “administração centralizada para conexões VPN e *Dial-up*; conta de usuários e auditoria centralizada; fácil administração; e escalabilidade.”

A seguir serão vistos alguns dos principais métodos de autenticação utilizados pelo Linux. Também será decrito o servidor de autenticação Radius.

3.5.1 Métodos de Autenticação

O *Point-to-Point Protocol* (PPP) ou Protocolo Ponto-a-Ponto oferece suporte a autenticação por meio de dois métodos de autenticação: *Password Authentication Protocol* (PAP) e *Challenge Handshake Authentication Protocol* (CHAP).

O PAP é um protocolo simples. Sua senha não é criptografada. O cliente envia uma senha em texto claro para o *Network Authentication Service* (NAS) ou Serviço de Autenticação de Rede. Após isto, a senha é enviada ao servidor Radius, somente então que ela é criptografada. Ortiz e Ferreira (2003) salientam este problema, explicando que a falta de segurança está entre o cliente e o NAS.

O CHAP é um protocolo mais seguro que o anterior, pois a senha que o cliente envia ao NAS é criptografada. O algoritmo de criptografia utilizado é o *Message Digest* 5 (MD5). Outra vantagem dele sobre o PAP é que mesmo após a conexão do cliente com o NAS ter se estabelecido, ele envia verificações periódicas para assegurar que a comunicação continua sendo feita pelo mesmo dispositivo.

Diógenes (2002, p. 310) ressalta essa idéia:

> A autenticação CHAP é mais segura que a PAP, não só pela forma dela trabalhar, mas também por enviar verificações periódicas durante a sessão com o roteador ao qual estabeleceu a conexão, de forma a certificar-se de que ele continua se comunicando com o mesmo roteador.

Além da autenticação PPP, existem outras como o *Extensible Authentication Protocol* (EAP), o *Dialed Number Identification Service* (DNIS), o *Automatic Number Identification/Calling Line Identification* (ANI/CLI) e a autenticação *GUEST*.

3.5.2 Autenticação Radius

Remote Authentication Dial In User Service (Radius) é um sistema de autenticação. Está descrito no RFC 2138. Sua principal função é aceitar ou rejeitar um usuário na rede VPN, ou seja, verificar se o *login* e senha são válidos.

Segundo Ortiz e Ferreira (2003, p. 66), "Radius é um servidor de autenticação de acesso, autorização e protocolo de contabilidade desenvolvido pela *Livingston Enterprises*".

Ele suporta vários protocolos de autenticação. Entre os principais estão a autenticação PPP, incluindo os métodos de autenticação PAP e CHAP; e o *login* Unix.

O Radius é um protocolo cliente/servidor. O protocolo usado na transferência de pacotes Radius entre cliente e servidor é o UDP. A porta usada pelo Radius é 1812.

Rigney (1997) cita algumas das vantagens do uso do UDP sobre o TCP: as exigências do sincronismo deste protocolo são significativamente diferentes das que o TCP fornece; a natureza simples deste protocolo simplifica o uso do UDP; UDP simplifica a implementação do servidor.

A segurança é bastante visível quando as informações são enviadas entre cliente e servidor, pois seus dados são protegidos e a senha é criptografada. Ortiz e Ferreira (2003, p. 72) comentam esse aspecto dizendo que:

> Transações entre o cliente e o servidor Radius são autenticadas pelo uso de um segredo compartilhado, que nunca é enviado pela rede. Em adição, as senhas de usuários são enviadas criptografadas entre o cliente e servidor Radius a fim de eliminar a possibilidade de que alguém que esteja invadindo uma rede sem segurança possa determinar a senha do usuário.

O processo de autenticação Radius acontece de forma simples. Primeiro, o cliente que quer se autenticar num servidor Radius cria um "*Access-Request*" contendo atributos como *login*, senha e porta. O "*Access-Request*" é enviado para o servidor Radius através da

rede. Se algum dos atributos enviado for desconhecido, o servidor envia ao cliente uma resposta "*Access-Reject*", informando que este é um usuário inválido. Se todos os atributos forem aceitos, o servidor envia uma resposta "*Access-Accept*", informando que o usuário é válido. (RIGNEY, 1997)

A Figura 15 mostra a estrutura do pacote Radius.

Código	Identificador	Tamanho	Autenticador	Atributos
1 byte	1 byte	2 bytes	16 bytes	Variável

Figura 15 - Estrutura do pacote Radius
Fonte: Ortiz; Ferreira (2003, p. 69)

Ortiz e Ferreira (2003, p. 69) oferecem uma boa compreensão de seus campos:

- **Código** – É um campo de 1 *byte* que indica o tipo de pacote Radius. Para consultar os tipos de pacotes Radius existentes.
- **Identificador** – Esse campo tem 1 *byte* de tamanho e é utilizado para combinar as solicitações enviadas com as suas respectivas respostas;
- **Tamanho** – Determina o tamanho dos pacotes Radius e o tamanho da mensagem enviada;
- **Autenticador** – Responsável por transportar as informações de autenticação que são responsáveis pela validação do cliente e servidor Radius;
- **Atributos** – O Atributo é um campo de tamanho que armazena todos os detalhes de autenticação, autorização, informação e configuração dos pacotes Radius.

3.6 Considerações Finais

Este capítulo teve como objetivo explicar os principais processos que envolvem a construção de um túnel VPN. Foram revisados conceitos e técnicas de criptografia, a qual é responsável pela segurança do túnel. O IPSec, protocolo de tunelamento, é usado para transmitir os dados de um ponto ao outro. Foi escolhido, entre outros, por oferecer suporte aos

protocolos criptográficos e o gerenciamento de chaves. Por fim, foi estudado o método de autenticação Radius, responsável por aceitar ou rejeitar um usuário na rede VPN.

4 QUALIDADE DE SERVIÇO (QoS)

Este capítulo tem como base explicar os principais conceitos de QoS. São estudados alguns mecanismos de enfileiramento e as duas grandes arquiteturas: IntServ, que usa o RSVP para reservar recursos para aplicações específicas na rede; e DiffServ, que utiliza marcação de pacotes para classificar níveis de prioridade.

4.1 Conceitos Básicos

Ultimamente, as redes de computadores têm se tornado cada vez mais complexas. O protocolo predominante da Internet é o IP, oferecendo o *best-effort service* ou serviço de melhor esforço. Segundo essa abordagem, existe o máximo esforço para fazer a entrega dos dados, porém, sem garantias.

As aplicações atuais estão exigindo mais qualidade no serviço na rede. Aplicações pesadas ou aplicações de tempo real, como voz e vídeo necessitam de uma maior largura de banda para alcançarem seu destino, sem perder a integridade. Stallings (2002) comenta que

elas são sensíveis ao atraso, variações de *throughput* (largura de banda real, medida a uma determinada hora do dia) e perda de pacote.

Tais aplicações não podem mais competir com as triviais, como transferências de dados ou envio de e-mail. Para contemplar essas necessidades foram propostas arquiteturas que têm como principal função alocar largura de banda para aplicações específicas na rede.

Chowdhury (2002, p. 172) afirma que "um QoS pode ser visto como oferecendo algum serviço especial para uma classe de tráfego específica por várias tecnologias básicas. É o serviço da rede que especifica o desempenho do tráfego através de uma ou mais redes".

Ferguson e Huston (1998), por sua vez, consideram que a melhor definição para QoS é aquela que define métodos para diferenciação de tráfego e serviços.

Muitas pessoas confundem o conceito de QoS com *Class of Service* (CoS), acreditando que são sinônimos. Para Ferguson e Huston (1998), QoS tem uma conotação ampla e ambígua, enquanto CoS conclui que os serviços podem ser classificados em classes separadas, que podem, em ordem, serem tratados individualmente.

É muito difícil encontrar a utilização de QoS na Internet. Sua principal atuação está nas redes privadas ou LANs. De acordo com Ferguson e Huston (1998), mecanismos de QoS estão em maior demanda em corporações, universidades e em *intranets* do que na Internet global e na comunidade ISP.

Stallings (2002) cita algumas exigências de projeto para internet baseada em IP, para que possam oferecer um serviço de qualidade que trate dos altos volumes de tráfego:

- **Controle de congestionamento:** *switches* e roteadores congestionados podem fazer com que a rede não trabalhe de maneira satisfatória, fazendo com que pacotes sejam descartados;

- **Fornecer baixo *delay*:** *delay* ou atraso é minimizado quando o congestionamento é ausente e o comprimento da fila é muito pequeno;
- **Fornecer alto *throughput*:** alta vazão pode ser alcançada por capacidade dedicada;
- **Suporte a QoS:** a provisão de diferentes níveis de QoS para diferentes fluxos de tráfego exigem tratamento inteligente de pacotes quando eles fluem através da rede;
- **Fornecer serviço justo:** se refere ao provisionamento de uma quantidade aproximadamente igual de capacidade para todos os fluxos de tráfego que estão competindo na mesma QoS.

Para que a QoS seja implantada numa rede, é necessário que existam mecanismos de enfileiramento, Integrated Service (IntServ) e/ou Differentiated Service (DiffServ), que serão descritos nos itens a seguir.

4.2 Mecanismos de Enfileiramento

Algumas filas são baseadas no best-effort service. Os serviços de melhor esforço, na Internet, não garantem QoS para nenhuma aplicação. São úteis para o envio de arquivos em geral e e-mails. Porém, aplicações mais recentes e pesadas necessitam de novos métodos para que possam ser enviadas.

Sendo assim, para Chowdhury (2002, p. 192), "o *Best-effort Service* é um modelo de serviço único, em que uma aplicação envia dados sempre que precisar, em qualquer quantidade, e sem solicitar permissão ou informar primeiro à rede".

Por outro lado, alguns tipos de filas oferecem mecanismos de priorização de tráfego.

A seguir serão apresentados os principais tipos de filas: *First-In First-Out* (FIFO), que é baseada no serviço de melhor esforço, *Weighted Fair Queuing* (WFQ), *Priority Queuing* (PQ) e *Class-Based Queuing* (CBQ), que oferecem serviço de priorização.

4.2.1 *First-In First-Out* (FIFO)

Como o nome já sugere, FIFO significa que o primeiro pacote a chegar é o primeiro a sair ou a ser atendido. Ferguson e Huston (1998) sugerem que a fila FIFO é considerada o método padrão para a operação de armazenamento e envio de tráfego de uma interface de entrada para uma interface de saída.

Este algoritmo de fila não dá importância nenhuma à prioridade. Assim, pacotes com maior prioridade não passam na frente de outros com prioridade menor. Chowdhury (2002, p. 175) acredita que "problemas poderão surgir quando uma fonte não comportada começar a transmitir continuamente, já que tal fonte pode consumir toda a largura de banda".

É vantajoso usar FIFO quando a rede opera em níveis adequados de capacidade de transmissão e comutação. Desse modo, como a fila é pequena, o tempo de atraso do pacote é insignificante. Porém, se a carga na rede for grande, as filas ficarão totalmente cheias, fazendo com que os pacotes posteriores sejam descartados, degenerando assim, o nível de serviço.

(FERGUSON; HUSTON, 1998, p. 58). A Figura 16 mostra o procedimento do algoritmo FIFO.

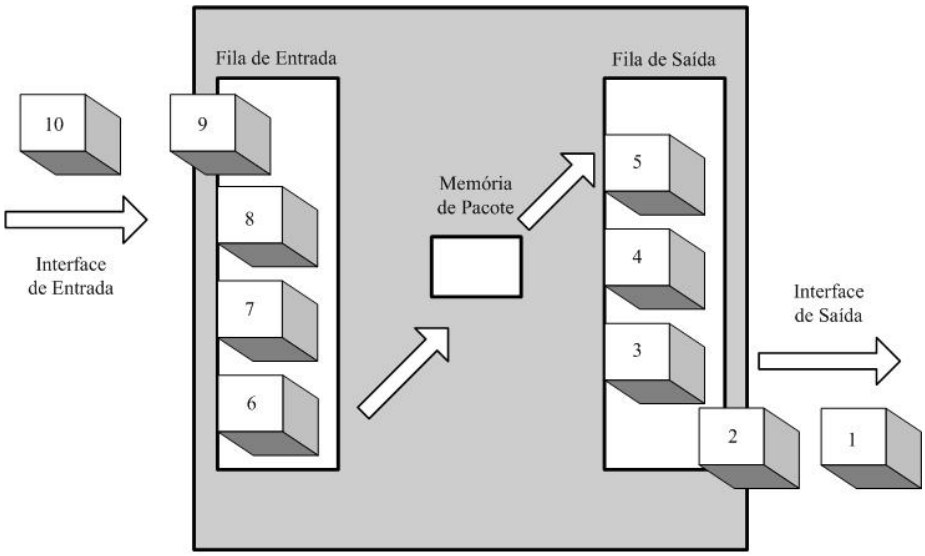

Figura 16 - Fila FIFO
Fonte: Ferguson; Huston (1998, p. 57)

Existe uma variação do FIFO. É o algoritmo FIFO++. Este possui vantagens sobre o FIFO, pois conforme Chowdhury (2002, p. 175), "ele utiliza técnicas de diferenciação de classe e oferece filas por fluxo com prioridade". O sistema FIFO++ é analisado na Figura 17.

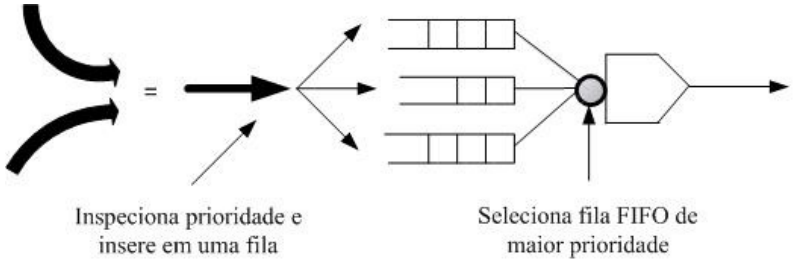

Figura 17 - Exemplo do algoritmo de fila FIFO++
Fonte: Chowdhury (2002, p. 175)

4.2.2 *Weighted Fair Queuing* (WFQ)

Primeiramente foi proposto por John Nagle, um modelo de fila chamado de *Fair Queuing* (FQ). Chowdhury (2002, p. 175) entende que ele foi desenvolvido "...para solucionar o problema de alocação imparcial de largura de banda". Surgiu da necessidade de criar vários fluxos de saídas. Um fluxo para cada fila.

Antes, se um roteador fosse sobrecarregado com quatro fontes, todas enviando pacotes juntas, uma delas teria que recuar, e conseqüentemente seria prejudicada.

Com este modelo, cada fila é tratada de maneira individual. Segundo Chowdhury (2002, p. 175), "em FQ, cada fila recebe uma fatia imparcial de largura de banda, cada fluxo possui sua própria fila e não pode afetar os outros fluxos".

Mais tarde, este modelo foi alterado por Demers e outros, passando a se chamar WFQ ou Enfileiramento Baseado por Tamanho. Além de continuar fornecendo a alocação por largura de banda imparcial, também oferece prioridade ou peso ao tráfego. A preferência deste método de fila é dar prioridade aos tráfegos menores.

O algoritmo WFQ impede que grandes fluxos consumam os recursos de rede, que poderiam tomar o lugar dos fluxos pequenos. Sua função é assegurar que fluxos de tráfegos maiores não tirem a vez dos menores. (FERGUSON; HUSTON, 1998, p. 62).

4.2.3 *Priority Queuing* (PQ)

O PQ ou Enfileiramento por Prioridade é caracterizado por aplicar uma prioridade a um determinado tráfego. Os tráfegos são tratados de acordo com o nível de prioridade que lhe foi atribuído. Os que possuem maior prioridade sempre são tratados primeiro. Em seguida são tratados os outros, de modo que o nível de prioridade é sempre respeitado.

Chowdhury (2002), Ferguson e Huston (1998) ressaltam que as prioridades disponíveis com PQ são alta, média, normal e baixa.

Para que esse processo funcione, as prioridades precisam ser atribuídas aos diferentes tipos de tráfegos. Como vantagem, pode-se dar preferência ao IPX em vez do IP, TCP em vez do UDP, etc.

Contudo, este processo apresenta várias desvantagens. O PQ tem sido usado por vários anos como um método de tráfego diferenciado primitivo em várias classes de serviço. Com o tempo, foi descoberto que este mecanismo simples não foi projetado para fornecer desempenho para altas velocidades. (FERGUSON; HUSTON, 1998, p. 60).

Chowdhury (2002) explica que ele apresenta melhor desempenho quando aplicado em enlaces com pouca largura de banda.

Outra vulnerabilidade do PQ é que processos com baixa prioridade podem nunca ser transmitidos. Isso pode ocorrer devido a grande carga de pacotes de alta prioridade. Além disso, Ferguson e Huston (1998) citam ainda a ocorrência de latência nos roteadores, devido ao processador do roteador precisar analisar cada pacote para conhecer seu nível de prioridade.

4.2.4 *Class-Based Queuing* (CBQ)

O CBQ ou Enfileiramento Baseado em Classe é um método de enfileiramento onde várias filas podem ser definidas. Também é conhecido como *Custom Queuing* (CQ) ou Enfileiramento Customizado.

Os protocolos são atribuídos às filas e a largura de banda é dividida entre as filas de maneira heterogênea.

Conforme Ferguson e Huston (1998), CBQ é uma variação do PQ. Porém, no PQ, cada protocolo recebe um nível de prioridade. No CBQ acontece diferente. O nível de prioridade é atribuído a cada fila, com seus respectivos protocolos.

Chowdhury (2002, p. 177) explica que "pode-se definir o quanto será transmitido de cada fila por vez, para que algumas filas possam transferir mais do que outras filas...".

Uma desvantagem do CBQ foi não ter sido projetado para fornecer desempenho desejado em alguns casos. Um exemplo é que o gerenciamento intensivo de filas não combina com redes que possuem enlaces de alta velocidade.

Ferguson e Huston (1998) comentam que apesar do CBQ fornecer os mecanismos básicos para tratar das classes de serviços diferenciadas, ele é apropriado somente para enlaces de baixa velocidade.

4.3 *Integrated Services* (IntServ)

A arquitetura IntServ ou Serviços Integrados não exige uma mudança na estrutura da Internet. Em vez disso foram acrescentadas extensões no serviço de melhor esforço para dar suporte aos protocolos para provisionamento de QoS numa rede IP.

O conceito da estrutura do IntServ começa com a sugestão de que a arquitetura de Internet básica não precisa ser modificada para fornecer suporte customizado para diferentes aplicações. Ao invés disso é sugerido que uma série de extensões possam ser desenvolvidas para fornecer serviços além do serviço tradicional de melhor esforço. (FERGUSON; HUSTON, 1998, p. 122).

Chowdhury (2002, p. 193) concorda com esta idéia fazendo a seguinte afirmação:

> A arquitetura da Internet é estendida para dar suporte à provisão de multiserviços, que é o tráfego classe de dados e vídeo/áudio. Um desenvolvedor componente de sistema interligado é conhecido como IntServ. Ele especifica uma natureza de serviço de entrega de pacotes que é descrito por parâmetros como retardo de pacote, perda de pacote e largura de banda alcançada.

Stallings (2002) também relata que a exigência fundamental é adicionar novas funcionalidades aos roteadores e achar um meio de solicitar das Internets, serviços baseados em QoS.

A seguir será analisado o controle de tráfego no roteador, como também serão apresentados os componentes necessários para o funcionamento desta arquitetura.

4.3.1 Controle de Tráfego no Roteador

Existem alguns mecanismos que determinam as funções de controle de tráfego na camada 3 do modelo OSI (camada em que opera o roteador). Ferguson e Huston (1998), assim como também Stallings (2002) apresentam os principais:

- **Escalonador de pacote:** esta função gerencia uma ou mais filas para cada porta de saída. Determina a ordem em que os pacotes na fila são transmitidos e a seleção de pacotes para descartes, se necessário. É responsável também por decidir se um fluxo particular pode ser admitido na entrada da rede;
- **Classificador de pacote:** para as finalidades de encaminhamento e controle de tráfego, os pacotes de entrada devem ser traçados em classes. Uma classe pode corresponder a um fluxo simples ou uma série de fluxos com as mesmas exigências de QoS. Por exemplo, os pacotes de todos os fluxos de vídeo ou os pacotes de todos os fluxos atribuíveis a uma organização particular podem ser tratados identicamente para finalidades de alocação de recursos e disciplina de filas. A seleção de classe é baseada em campos do cabeçalho IP. Baseada nas classes de pacotes e no seu endereço IP de destino, esta função determina o endereço do próximo salto para este pacote;
- **Controle de admissão:** quando um novo fluxo é solicitado, o protocolo de reserva chama a função de controle de admissão. Esta função determina se recursos suficientes estão disponíveis para este fluxo no QoS solicitado.

Esta determinação é baseada no nível atual de compromisso com outras reservas e/ou na carga atual da rede;

- **Protocolo de reserva de recurso:** este protocolo é usado entre roteadores e entre roteadores e sistemas finais para reservar recursos para um fluxo novo em um nível dado de QoS. Este protocolo de reserva é responsável por manter informações do estado de um fluxo específico nos sistemas finais e nos roteadores ao longo do fluxo. Este tipo de protocolo será analisado com detalhes no item 4.3.2.3.

4.3.2 Componentes da arquitetura IntServ

De acordo com Ferguson e Huston (1998), a arquitetura IntServ foi dividida em cinco componentes: exigências de QoS, exigências de compartilhamento de recursos, permissões de descarte de pacotes, provisionamento para *feedback* de uso, e um protocolo de reserva de recursos (RSVP).

No escopo deste trabalho serão discutidas as exigências de QoS, as exigências de compartilhamento de recursos e o protocolo de reserva de recursos.

4.3.2.1 Exigências de QoS

Quanto às exigências de QoS, é importante entender como funcionam as aplicações de tempo real ou inflexíveis e as aplicações de tempo não real ou flexíveis. O modelo IntServ está mais centrado em aplicações de tempo real. Essas duas classes foram avaliadas a seguir.

a) Aplicação de Tempo Real

As aplicações de tempo real podem ser divididas em tolerantes ou intolerantes quanto à influência do *jitter* e ao atraso do pacote.

As aplicações tolerantes, mesmo induzidas pelo *jitter*, podem produzir uma boa qualidade quando executadas. Por outro lado, as intolerantes, quando expostas à indução do *jitter* ou à perda de pacotes, degeneram a qualidade da aplicação, podendo levá-la a ficar infuncional. O modelo IntServ recomenda o uso da *Controlled Load* (CL) ou Carga Controlada para as aplicações tolerantes, e para as intolerantes, o uso da *Guaranteed Load* (GL) ou Carga Garantida. (FERGUSON; HUSTON, 1998, p. 124).

O serviço de classe de carga controlada se aproxima do serviço de melhor esforço. Segundo esta afirmação, não controla parâmetros como atraso ou perda de pacote, apenas tem o compromisso de fornecer um serviço melhor do que aquele que é oferecido pela Internet.

Chowdhury (2002) argumenta que se a rede estiver funcionando corretamente, a aplicação pode assumir a entrega de pacote bem sucedida em uma porcentagem muito alta e também, minimizar o retardo no trânsito do pacote.

Stallings (2002) explica que a CL garante que a rede reserve recursos suficientes para uma aplicação, e faz parecer como se esta aplicação de tempo real não estivesse presente e competindo por recursos.

O serviço de classe de carga garantida controla alguns parâmetros, como largura de banda e atraso do pacote. É responsável por garantir que pacotes não sejam descartados mesmo com um estouro de fila.

Braden (1994) considera que como o limite de atraso é fixo, o atraso deve receber um valor maior do que o atraso máximo absoluto, para evitar a possibilidade de perda de pacotes.

A GL não controla o atraso mínimo ou médio de tráfego, e não controla ou minimiza o *jitter*. Somente controla o atraso máximo de fila. (FERGUSON; HUSTON, 1998, p. 129).

b) Aplicação de Tempo Não Real

As aplicações de tempo não real possuem as mesmas propriedades que o serviço de melhor esforço possui. Geralmente elas usam os protocolos TCP e UDP para transportar dados.

Stallings (2002) acredita que o tráfego das aplicações flexíveis é o tipo tradicional de tráfego suportado nas internets baseadas em IP e é o tipo de tráfego em que as internets foram projetadas.

Conforme Braden (1994), em contraste às aplicações intolerantes, as tolerantes não precisam receber um valor maior do que o valor máximo absoluto, já que elas podem tolerar algumas perdas de pacotes. Alguns exemplos deste tipo de aplicação são Telnet, FTP, HTTP, SNMP e SMTP.

Como numa internet baseada em IP, tais aplicações devem possuir processos para tratar os erros.

Segundo Ferguson e Huston (2002), aplicações flexíveis usam mecanismos de detecção de erro e recuperação de retransmissão na camada de aplicação para recuperar erros de transmissão de dados.

4.3.2.2 Exigências de Compartilhamento de Recursos

Exigências de compartilhamento de recursos estão diretamente ligadas ao conceito de *link sharing*, também conhecido como compartilhamento de enlace.

O *Link sharing* acontece quando, embora cada fluxo esteja submetido a um critério de controle de admissão, muitos fluxos compartilham os recursos disponíveis na rede. Com o compartilhamento de enlace, a largura de banda total na rede é compartilhada por vários tipos de tráfego. O modelo IntServ dá enfoque também ao *link sharing* com uma função adicional de controle de admissão: um mecanismo de enfileiramento justo, como por exemplo o WFQ, que fornece alocação proporcional de recursos de rede. (FERGUSON; HUSTON, 1998, p. 130)

4.3.2.3 *Resource Reservation Protocol* (RSVP)

O *Resource Reservation Protocol* (RSVP) ou Protocolo de Reserva de Recursos é um protocolo que reserva recursos, tal como largura de banda de enlace e *buffers* de roteador, para aplicações específicas na rede.

Chowdhury (2002, p. 195) relata que o "RSVP oferece a um *host* a capacidade de solicitar QoS específico da rede para uma aplicação de fluxo de dados. Os roteadores também usam RSVP para oferecer QoS solicitado por um *host*".

Ferguson e Huston (1998) também abordam este conceito dizendo que o RSVP é simplesmente um mecanismo de sinalização, que foi projetado para ser usado com uma variedade de serviços QoS.

Este protocolo atua na camada de transporte, porém, não faz o transporte de dados. É executado em segundo plano. De acordo com Ferguson e Huston (1998), o RSVP é análogo a outros protocolos de controle IP, como o *Internet Control Message Protocol* (ICMP), ou um dos muitos protocolos de roteamento IP.

As aplicações são responsáveis por sinalizar ou reservar os recursos necessários na rede, através do RSVP. Depois que as reservas estejam concluídas, os roteadores, por sua vez, têm como função oferecer tais recursos às aplicações.

A Figura 18 ilustra como ocorre o processo do RSVP entre *hosts* e roteadores.

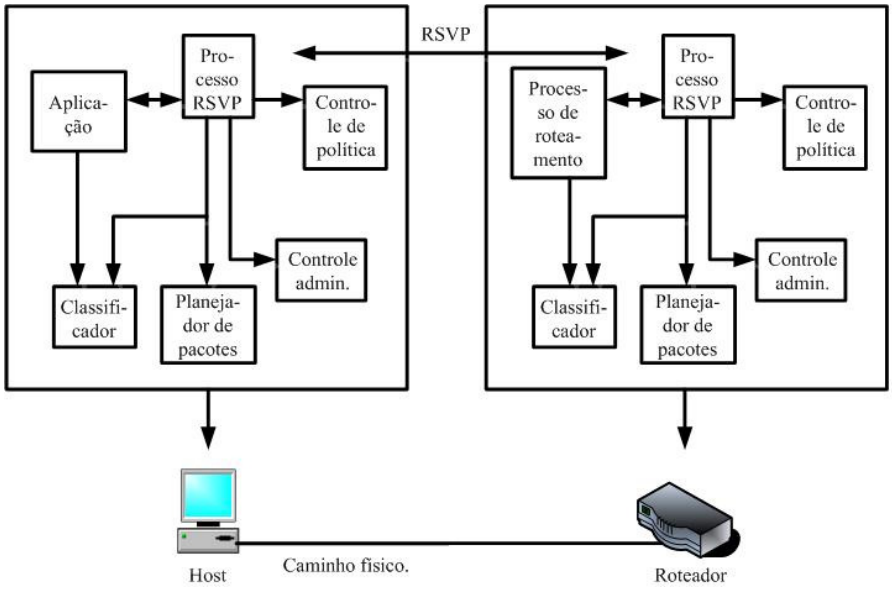

Figura 18 - *Host* e roteador usando negociação QoS
Fonte: Chowdhury (2002, p. 195)

Braden (1997) comenta algumas características importantes referentes ao RSVP:

- Faz reserva de recursos para aplicações *unicast* e *multicast*;
- É *simplex*, ou seja, ele faz as reservas para os fluxos de dados de maneira unidirecional;
- É orientado a receptor, ou seja, o receptor de um fluxo de dados inicia e mantém a reserva de recurso usado para este fluxo;
- Não é um protocolo de roteamento, mas depende dos protocolos de roteamento atuais e futuros;
- Fornecem vários modelos ou estilos de reservas que cabem em uma variedade de aplicações;
- Suporta Ipv4 e Ipv6.

4.4 *Differentiated Service*s (DiffServ)

Segundo Chowdhury (2002, p. 203), "...o modelo DiffServ foi desenvolvido para oferecer melhor serviço a algum tráfego, à custa de oferecer pior serviço para outro".

É através do campo *Type of Service* (TOS) ou Tipo de Serviço do datagrama IP que é possível a solicitação de um serviço específico pela rede. Ferguson e Huston (1998) comentam que este campo faz parte do pacote IP desde o começo, porém, foi muito pouco usado no passado. Atualmente, seu uso é indispensável para fornecer um serviço de qualidade para as redes. Este campo será analisado a seguir.

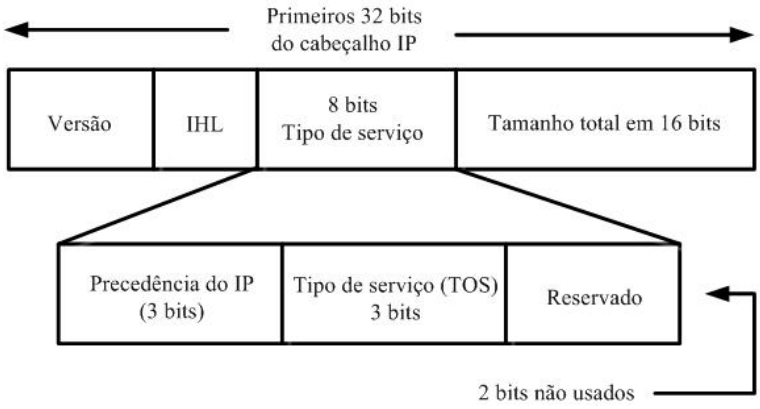

Figura 19 - O *byte* TOS e a precedência do IP
Fonte: Chowdhury (2002, p. 180)

Conforme mostra a Figura 19, pode-se observar que o campo TOS possui 8 bits, sendo subdividido em três campos: Precedência do IP (3 bits), Tipo de Serviço (3 bits) e Reservado (2 bits). Conforme Chowdhury (2002, p. 188), "a precedência IP é usada para alocar recursos na rede com base na importância relativa dos diferentes fluxos de tráfego". Quanto mais alta for a precedência de um tráfego, mais importante ele é.

A solicitação de um serviço específico na rede é realizado através da marcação do campo DiffServ (DS), com um valor específico. Este campo DS é o equivalente ao campo de *byte* TOS do pacote IP. Os seis primeiros *bits* deste campo são conhecidos como *Differentiated Service CodePoint* (DSCP) ou Ponto de Código para Serviços Diferenciados. Ele é usado como ponto de código para selecionar experiências do pacote *Per Hop Behavior* (PHB) ou Comportamento Por Salto. (CHOWDHURY, 2002, p. 204)

A Figura 20 ilustra a diferença entre o campo TOS de 8 bits do datagrama IP e do campo DSCP.

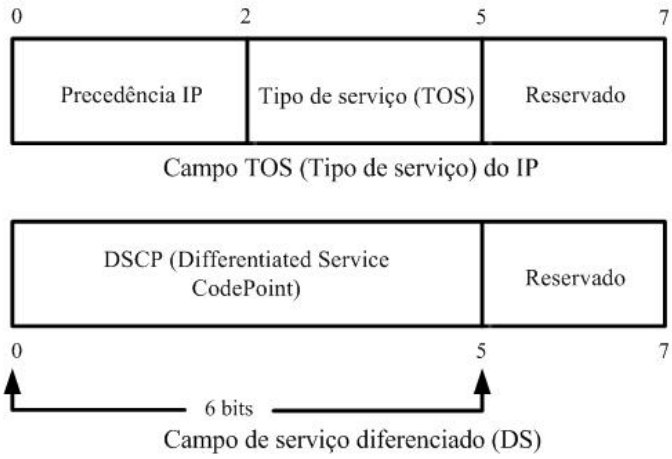

Figura 20 - Uma visão comparativa do campo DS e do byte TOS de um pacote IP
Fonte: Chowdhury (2002, p. 204)

O PHB, mecanismo implementado nos roteadores, tem como função oferecer tratamento de encaminhamento aos pacotes no momento em que os mesmos estão saindo do nó.

Chowdhury (2002, p. 204) afirma que "...os PHBs são o tratamento de encaminhamento de pacote que oferece o DiffServ aos pacotes na saída do nó da rede e incluem política, modelagem, possível remarcação do DSCP, tratamento de fila e planejamento."

Blake (1998) por seu lado, afirma que os PHBs são definidos para permitir meios razoavelmente granular de recursos para alocação de *buffer* e largura de banda para cada nó entre a competição de fluxos de tráfego.

Esta arquitetura fornece serviço diferenciado somente em uma direção de fluxo de tráfego, sendo por esse motivo, assimétrica. Está sendo estudado o desenvolvimento de uma arquitetura simétrica. (BLAKE, 1998)

4.5 Considerações Finais

Este capítulo teve como objetivo estudar técnicas necessárias responsáveis pelo funcionamento da QoS, tais como, mecanismos de enfileiramento, serviços integrados e serviços diferenciados.

No escopo deste projeto foi escolhida a tecnologia de serviços diferenciados para a implantação da QoS, pois dessa forma, é possível oferecer prioridade a um determinado serviço.

O DiffServ foi a opção viável, pois a ferramenta de geração de tráfego possibilita marcar o campo DSCP.

5 DESEMPENHO

Este capítulo se dedica ao estudo da análise de desempenho em relação à QoS aplicada nos túneis VPN. São estudadas as técnicas de avaliação, para decisão de qual técnica se aplica melhor ao trabalho; definem-se também as métricas, que abrangem perdas, atraso, *jitter* ou variação do atraso e vazão ou largura de banda.

Outro fator importante analisado neste capítulo é o estudo do software Iperf, responsável por gerar o tráfego, que será usado para realizar as medições.

5.1 Conceitos Básicos

Segundo Jain (1991), usuários, administradores, e projetistas de sistemas computacionais estão todos interessados na avaliação de desempenho, desde que seu objetivo é obter ou fornecer o mais alto desempenho pelo menor custo.

Para avaliar o desempenho de um sistema é necessário que seja feita a escolha do melhor método de avaliação e da melhor ferramenta, que serão discutidos adiante.

Para Jain (1991), cada avaliação requer um conhecimento profundo do sistema que está sendo modelado e de uma seleção cuidadosa da metodologia, da carga de trabalho, e das ferramentas.

Muitos erros podem ocorrer durante o processo da avaliação de desempenho, sendo que a maioria acontece devido aos descuidos, à má interpretação, e à simples falta do conhecimento sobre técnicas da avaliação de desempenho. A seguir foram identificados alguns desses erros: nenhum objetivo definido, aproximação assistemática, analisar sem entender o problema, utilizar métricas incorretas de desempenho, utilizar técnica de avaliação errada, ignorar fatores significantes. (JAIN, 1991, p. 14)

É de grande importância testar os conceitos em sistemas reais, pois existe bastante diferença entre projetar a avaliação do desempenho de algum sistema e testá-la.

A melhor maneira de aprender um assunto é aplicar os conceitos em um sistema real. Isto é especialmente verdadeiro na avaliação de desempenho dos sistemas computacionais, porque mesmo que as técnicas pareçam simples na superfície, suas aplicações a sistemas reais oferecem uma experiência diferente, desde que os sistemas reais não se comportam de uma maneira simples. (JAIN, 1991, p. 11)

A seguir serão discutidas as técnicas de avaliação de desempenho, como também apresentados conceitos para determinar qual deve ser utilizada em um determinado sistema. Também será escrito sobre as métricas existentes para a avaliação do desempenho.

5.2 Seleção de técnicas de avaliação e de métricas

Selecionar uma técnica de avaliação e uma métrica são dois fatores imprescindíveis para se alcançar um resultado desejável na avaliação do desempenho de um sistema.

Jain (1991) explica que a seleção de uma técnica de avaliação e de uma métrica são as duas etapas chaves em todo o projeto de avaliação de desempenho, de modo que, há muitas considerações que estão envolvidas na seleção correta.

5.2.1 Técnicas de avaliação

Jain (1991) afirma que as três técnicas para a avaliação de desempenho são modelagem analítica, simulação e medição. Conforme explicação deste autor, foi elaborado o Quadro 2, que explica qual técnica deve ser utilizada em determinado sistema.

Quadro 2 – Técnicas de Modelagem

	Ciclo de Vida do Sistema	Tempo Disponível	Disponibilidade das Ferramentas	Nível de Exatidão	Custo
Modelagem Analítica	Sistema novo	Gasta pouco tempo	Ideal para analistas que são hábeis em modelar	Baixo	Baixo
Simulação	Sistema novo	Gasta muito tempo	Conhecimento em linguagens de simulação	Médio	Médio
Medição	Somente se algo similar ao sistema proposto já existir	Gasta tempo médio	Ideal para analistas que preferem lidar com sistemas reais	Alto	Alto

Fonte: Adaptado de Jain (1991, p. 31)

No escopo deste projeto foi utilizada a técnica de medição, num ambiente de laboratório (real), com uma ferramenta de geração de tráfego e medições.

5.2.2 Métricas

Existem quatro métricas para medir o desempenho de um sistema baseado no IP: perdas, definida na RFC 2680; atraso, definida na RFC 2679; *jitter* ou variação do atraso, definida na RFC 3393 e vazão ou largura de banda.

De acordo com Paxson (1998), foram desenvolvidos alguns critérios para métricas de desempenho e confiabilidade:

- Devem ser concretas e bem definidas;
- Devem ser reprodutíveis: se forem usadas múltiplas vezes sob circunstâncias idênticas, devem apresentar os mesmos resultados;

- Não devem exibir nenhuma tendência para as nuvens IP implementadas com tecnologia idêntica;
- Devem exibir tendências compreensíveis e justas para as nuvens IP implementadas com tecnologia não-idêntica;
- Devem ser úteis para usuários e provedores compreender o desempenho que estão experimentando ou fornecendo;
- Devem evitar a indução de objetivos de desempenho artificial.

A seguir serão avaliadas as métricas de perda, *jitter* e largura de banda. O atraso não foi analisado por exigir técnicas de medição mais apuradas, como o sincronismo dos relógios através de sistemas GPS. Essa métrica também exigiria mais uma ferramenta de software, uma vez que o Iperf não fornece essa medida.

5.2.2.1 Métricas de Perda do Pacote

Perda de pacote é uma métrica para análise de desempenho que possui três parâmetros: endereço IP da origem, endereço IP do destino e um tempo. Suas unidades compreendem ou o valor 0 (zero), que significa que o pacote foi transmitido com sucesso; ou o 1 (um), que significa perda. (ALMES, 1999).

Almes (1999) explica que quando a métrica de perda de pacote da origem para o destino, num determinado tempo, for igual a zero, significa que a origem enviou o primeiro *bit* do pacote para o destino no tempo determinado e o destino recebeu o pacote; porém, quando o valor for um, significa que o destino não recebeu o pacote.

Almes (1999) oferece uma metodologia para estas métricas:

- No *host* de origem, selecionar os endereços IP de origem e destino, e definir um pacote para testes com estes endereços;
- Determinar o *host* de destino para receber um pacote;
- No *host* de origem, colocar um sinalizador de tempo ou *timestamp* no pacote preparado, e enviá-lo para o destino;
- Se o pacote chegar dentro de um período de tempo razoável, o valor da métrica é dado como zero;
- Se o pacote não chegar dentro de um período de tempo razoável, o valor da métrica é dado como um.

5.2.2.2 Métrica de Variação do Atraso ou *Jitter*

Esta métrica é válida para medidas entre dois *hosts*, tanto no caso de ambos terem o relógio sincronizado, quanto no caso de não terem. No último caso, as indicações do desvio recíproco dos relógios podem ser derivadas da medida e as correções são possíveis. A precisão relacionada é frequentemente comparável com uma que pode ser conseguida com relógios sincronizados, sendo da mesma ordem de valor de erros da sincronização. (DEMICHELIS, 2002)

Neste trabalho, se aplica o caso em que os relógios não estão sincronizados.

Um uso importante da variação de atraso é a medida dos *buffers* de saída para aplicações exigindo a devolução regular de pacotes (por exemplo, saída de voz e vídeo). O

que é normalmente importante neste caso é a variação máxima de atraso, que é usada para medir *buffers* de saída para tais aplicações. Outros usos da métrica de variação de atraso são, por exemplo, para determinar a dinâmica das filas dentro de uma rede (ou roteador), onde as mudanças na variação de atraso podem ser ligadas às mudanças no processo do comprimento da fila, num dado *link* ou numa combinação de *links*. (DEMICHELIS, 2002)

5.2.2.3 Métrica de Largura de Banda

Peterson (1999) sustenta a idéia de que a largura de banda de um enlace pode ser definida de duas formas:

- Uma delas é que, fisicamente, pode-se dizer que é a faixa de freqüências que pode passar pelo enlace com perdas mínimas. Por exemplo, para transmitir um sinal de voz na linha telefônica precisamos de uma banda de 3000 Hz, pois a voz humana usa freqüências de 300 a 3300 Hz.
- A outra forma, mais prática, pode determinar a largura de banda como sendo a quantidade de sinal, em bits, que uma interface pode inserir em um enlace em um segundo (taxa de transmissão da interface).

Sob essa óptica, uma interface *Fast Ethernet* (padrão IEEE 802.3u), teria uma banda de 100 Mbps. Essa quantidade, normalmente, não é alcançada na prática, devido aos problemas de implementação das tecnologias. A palavra *throughput* ou vazão, geralmente é usada para definir o desempenho que um enlace fornece entre duas interfaces. Por exemplo, um enlace de 10 Mbps poderia fornecer uma vazão de 4 Mbps, devido às deficiências de implementação (PETERSON, 1999).

5.3 Ferramenta Iperf

O Iperf é uma ferramenta de medição e geração de tráfego, atualmente mantido na Universidade de *Illinois*. De acordo com Blum (2003), o Iperf fornece diferentes tipos de testes de comunicação TCP e UDP entre dois *hosts* em uma rede. E pode ser usado tanto em ambientes *Unix* quanto em *Windows*.

O aplicativo Iperf foi projetado para trabalhar como uma aplicação simples e interativa, que permitisse que os administradores de redes e sistemas pudessem ver como os parâmetros de soquete TCP usados em aplicações e em configurações de *hosts* podem afetar o desempenho da rede. Consiste de um arquivo executável, usado como uma simples aplicação tanto para funções cliente quanto para servidor; um aplicativo Java que fornece uma interface gráfica para o aplicativo Iperf; e uma série de arquivos de biblioteca usados para fornecer funções adicionais para o programa. (BLUM, 2003, p. 99)

Somente um simples arquivo (iperf) precisa ser carregado nos dispositivos de teste na rede, e qualquer dispositivo pode ser usado tanto como cliente quanto servidor. O Iperf também contém uma distribuição para o ambiente *Windows*, permitindo usar estações de trabalho ou servidores baseados em *Windows* como dispositivos de teste na rede. Isto pode expandir muito o número de dispositivos de teste da rede. (BLUM, 2003, p. 100)

Blum (2003) afirma que o aplicativo Iperf fornece vários testes de desempenho de rede tanto para ambientes TCP quanto para UDP.

Cerutti (2006) salienta que, embora existam muitas alternativas de ferramentas de medição, a única capaz de medir a variação do atraso é o Iperf. A Figura 21 ilustra o funcionamento desta ferramenta.

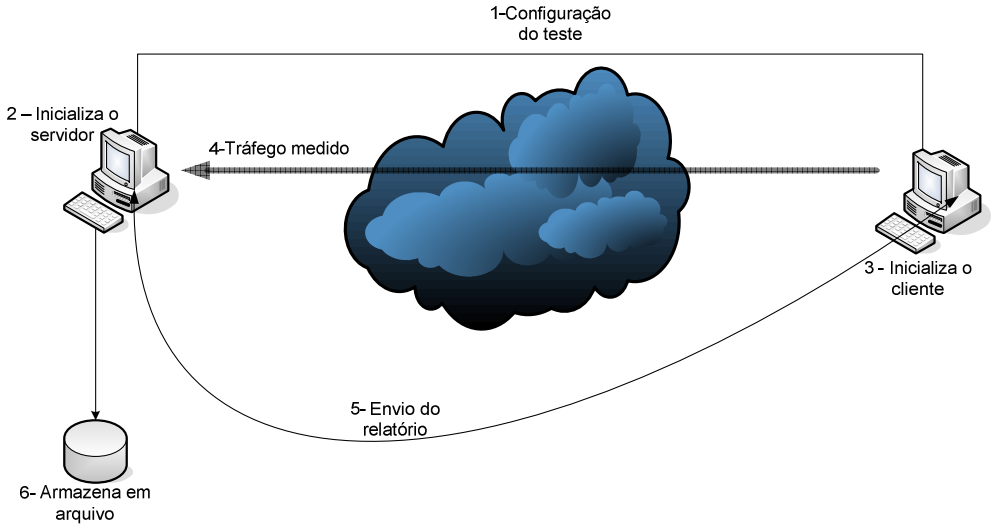

Figura 21 – Uso da ferramenta Iperf
Fonte: Cerutti (2006, p. 126)

5.4 Considerações Finais

Este capítulo teve como intuito realizar a análise de desempenho em relação à QoS aplicada nos túneis VPN, mostrando o controle de tráfego gerado, bem como as métricas de perda, *jitter* e largura de banda. Foi utilizado o software de geração de tráfego Iperf para realizar os experimentos.

6 DEFINIÇÃO DOS CENÁRIOS DE ESTUDO

A modelagem deste projeto seguiu o modelo de camadas TCP/IP apresentado por Kurose e Ross (2003), na seção 2.2, Figura 3. As camadas abrangidas foram: aplicação, transporte, rede, enlace e física. Na camada de aplicação, ocorrem serviços referentes à transferência de arquivos, correio eletrônico, *logon* remoto, gerenciamento de rede e gerenciamento de nomes; a camada de transporte tem como função controlar o fluxo de dados e fornecer confiabilidade; a camada de rede é responsável pelo endereçamento IP, roteamento e mensagens de controle; a camada de enlace é responsável pelo estabelecimento de um *link* físico entre os dispositivos; e por fim, a camada física trata dos meios e da sinalização, ou seja, conversão dos dados em sinais adequados ao tipo do enlace.

Foi necessária a identificação de alguns requisitos para que este modelo fosse implementado, entre eles:

- Rede deve estar funcionando corretamente;
- Estrutura do cabeamento deve estar correta;
- VPN deve funcionar;
- Usuário deve estar cadastrado no servidor Radius;

A seguir será apresentado um fluxograma que explica o processo realizado nesta pesquisa, bem como o detalhamento das cinco camadas ilustradas pelos desenhos abaixo.

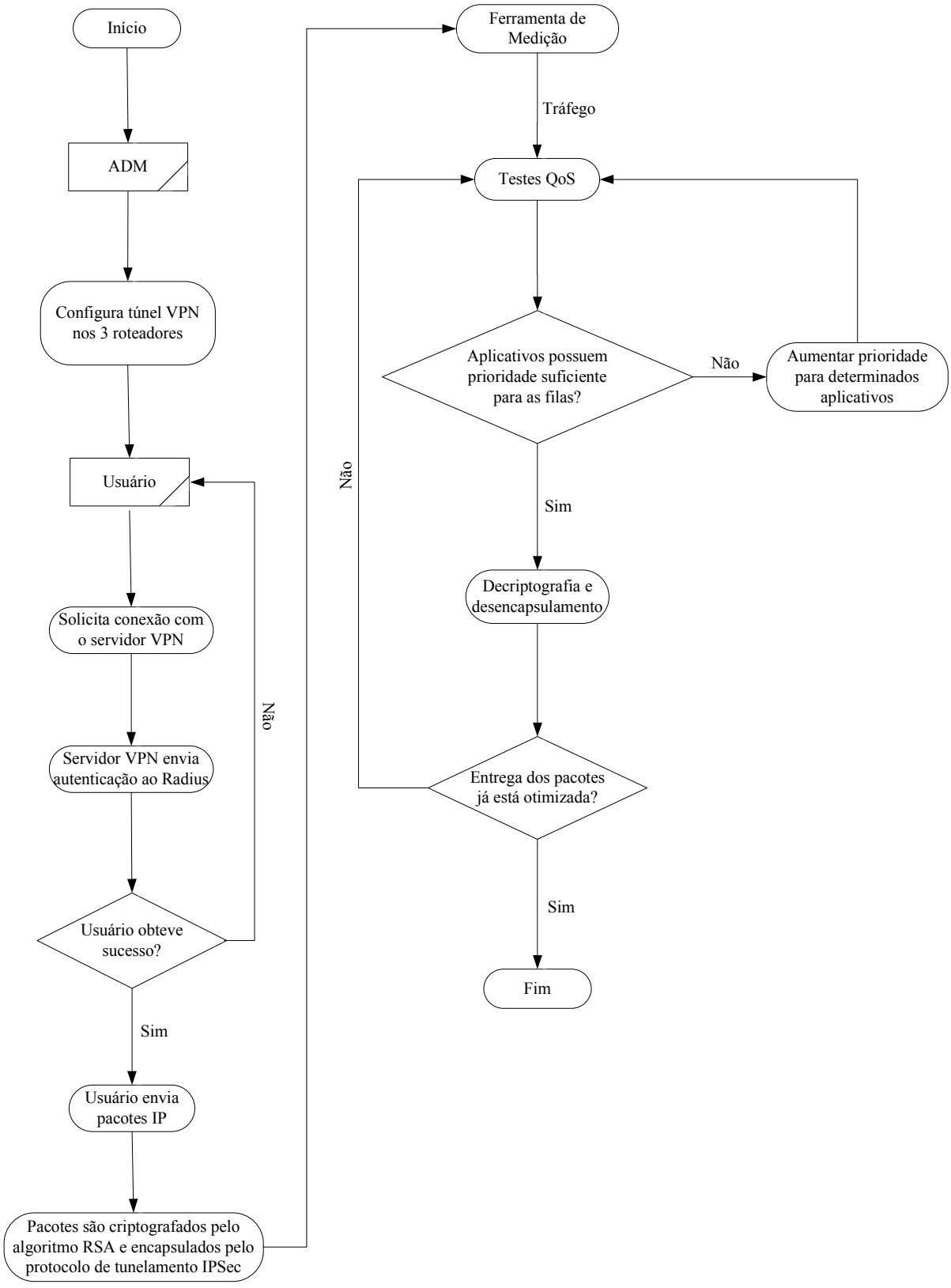

Figura 22 - Processos envolvidos no projeto

O modelo acima descreve um fluxograma que ilustra os processos que aconteceram no decorrer do desenvolvimento deste projeto.

O administrador configurou um túnel privado virtual na parte WAN da rede, ou seja, nos três roteadores, tornando esta rede segura. Em seguida, um usuário previamente cadastrado no servidor Radius, se autenticou no servidor VPN para enviar informações através do túnel. Essas informações, antes de entrarem no túnel, foram cifradas pelo algoritmo de criptografia RSA e encapsuladas pelo protocolo de tunelamento IPSec.

A próxima etapa foi instalar um software de medição e geração de tráfego. Foi usado o Iperf. Esta ferramenta foi responsável por medir algumas variáveis como largura de banda, *jitter* e atraso. Foram realizados alguns testes e colhidos os resultados.

Após, foi implantado QoS nos três roteadores com a tecnologia DiffServ, que ofereceu prioridade ao tráfego de VPN.

Depois da configuração dos serviços diferenciados, foram realizados mais testes com o Iperf para verificar o desempenho da rede com QoS. Se for obtido sucesso em todos os testes, significa que a rede está funcionando de acordo com o esperado.

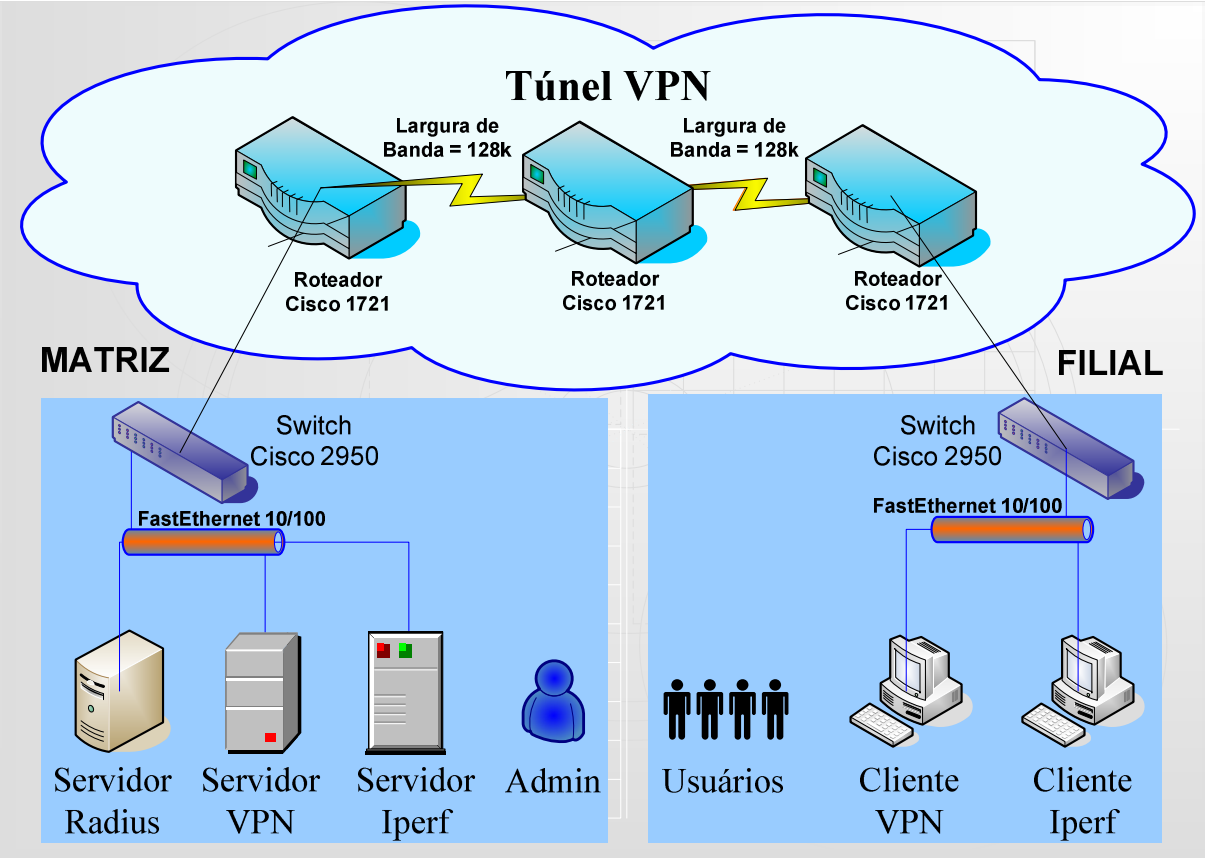

Figura 23 - Camada 1 ou Física

Na Figura 23 estão demonstradas as conexões entre os dispositivos e as tecnologias empregadas.

A conexão WAN entre os roteadores é feita através de cabos V.35, protocolo síncrono da camada física usado para comunicações entre um dispositivo de acesso à rede; os *switches* são conectados aos computadores por cabos *Unshielded twisted-pair* (UTP) ou Par Trançado Não Blindado, Categoria 5e.

Para a terminação dos cabos é usado o conector RJ-45, seguindo o padrão EIA/TIA 568b.

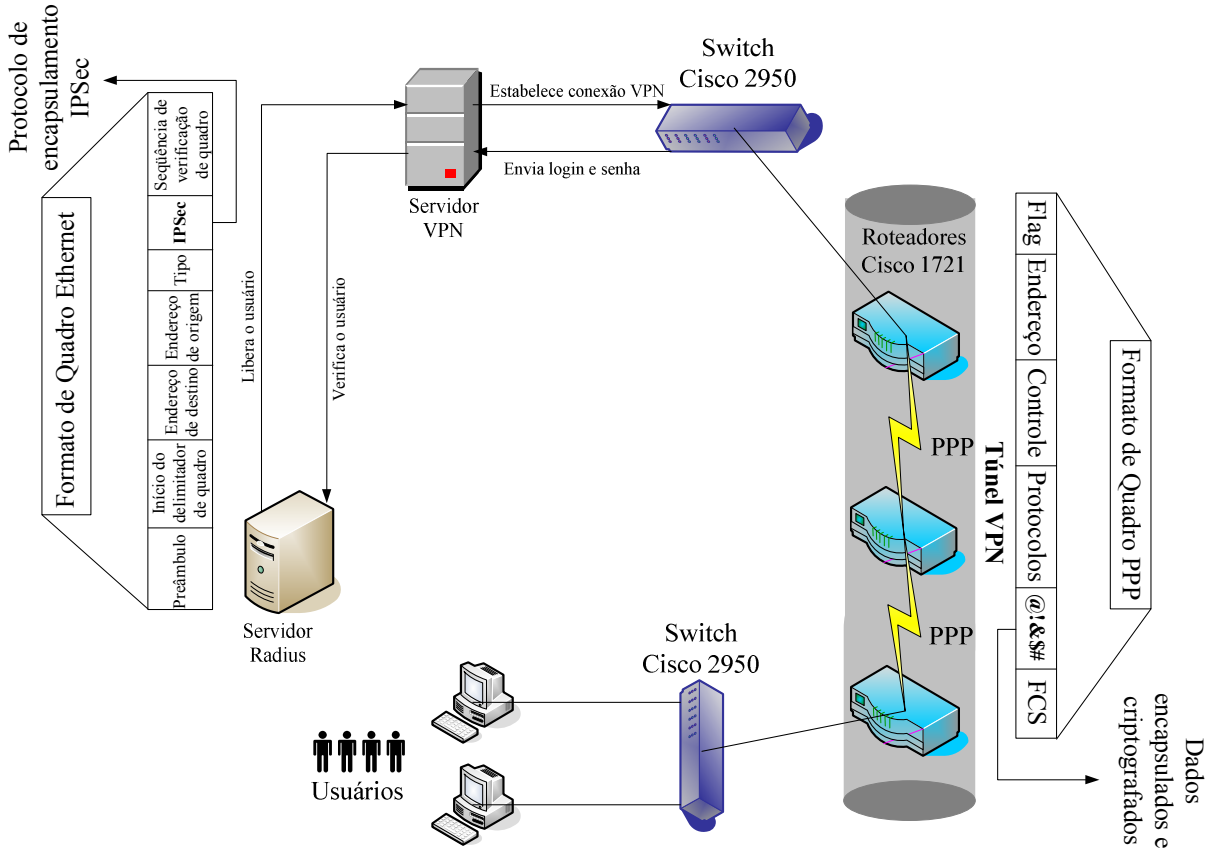

Figura 24 - Camada 2 ou Enlace

A Figura 24 apresenta a tecnologia usada para interligar as redes, que é a *Fast Ethernet* 10/100, uma variação da *Ethernet*, onde o formato do quadro *Ethernet* é mostrado. O tipo de topologia utilizado na parte LAN da rede é estrela, onde existe um concentrador, nesse caso um *switch*, e vários computadores ligados diretamente a ele.

O servidor Radius recebe o *login* e a senha do servidor VPN, faz a validação e retorna o resultado para o servidor VPN, sendo que este último pode aceitar ou negar a conexão.

O protocolo usado para o enlace entre os roteadores é o PPP. A Figura 24 ilustra o formato do quadro PPP, junto com os dados já encapsulados e criptografados enviados pelo servidor VPN. Esses dados, ao saírem do túnel, são decriptografados e desencapsulados, e entregues ao destino em texto claro.

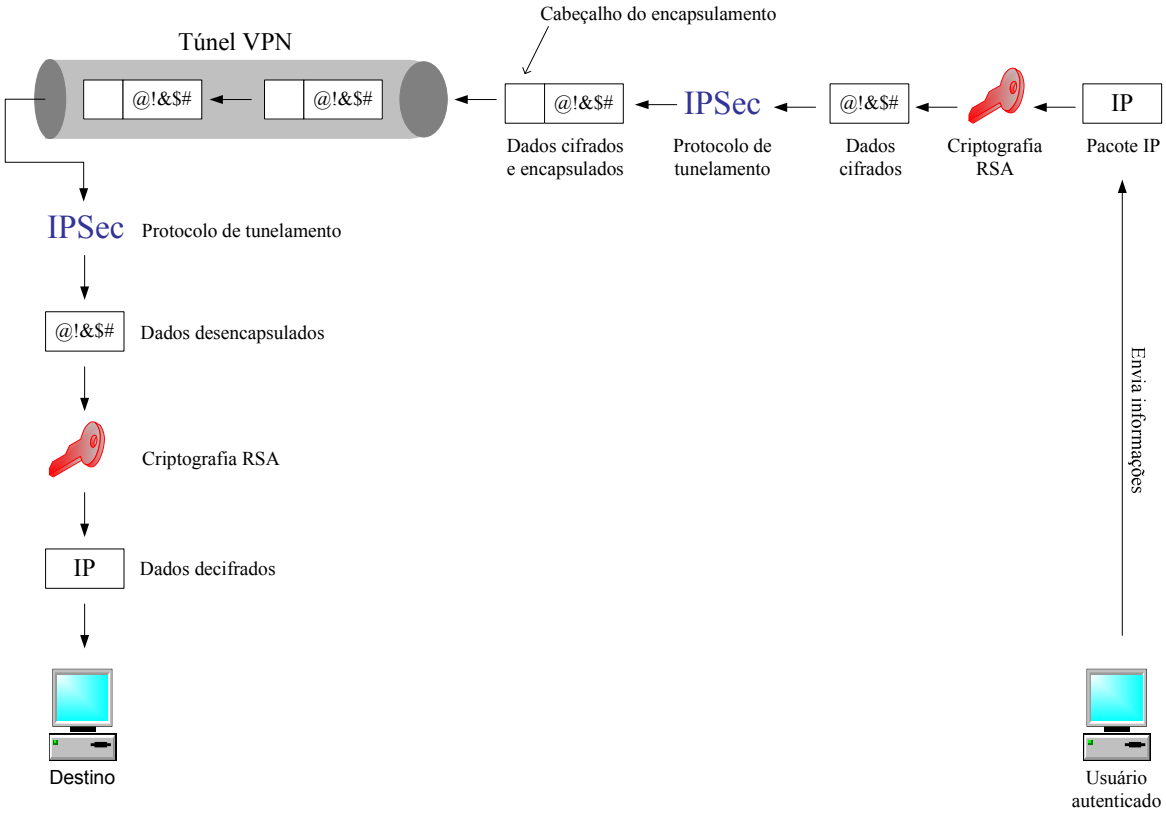

Figura 25 - Camada 3 ou Rede

Nessa camada é onde ocorre a criptografia e o encapsulamento dos dados. A Figura 25 ilustra o processo, mostrando um usuário autenticado enviando dados para um destino.

Os dados, antes de entrarem no túnel, são cifrados pelo algoritmo de criptografia RSA, e logo após, encapsulados pelo protocolo de tunelamento IPSec. Ao saírem do túnel, são decriptografados pelo algoritmo RSA e desencapsulados pelo IPSec, podendo ser entregues ao destino.

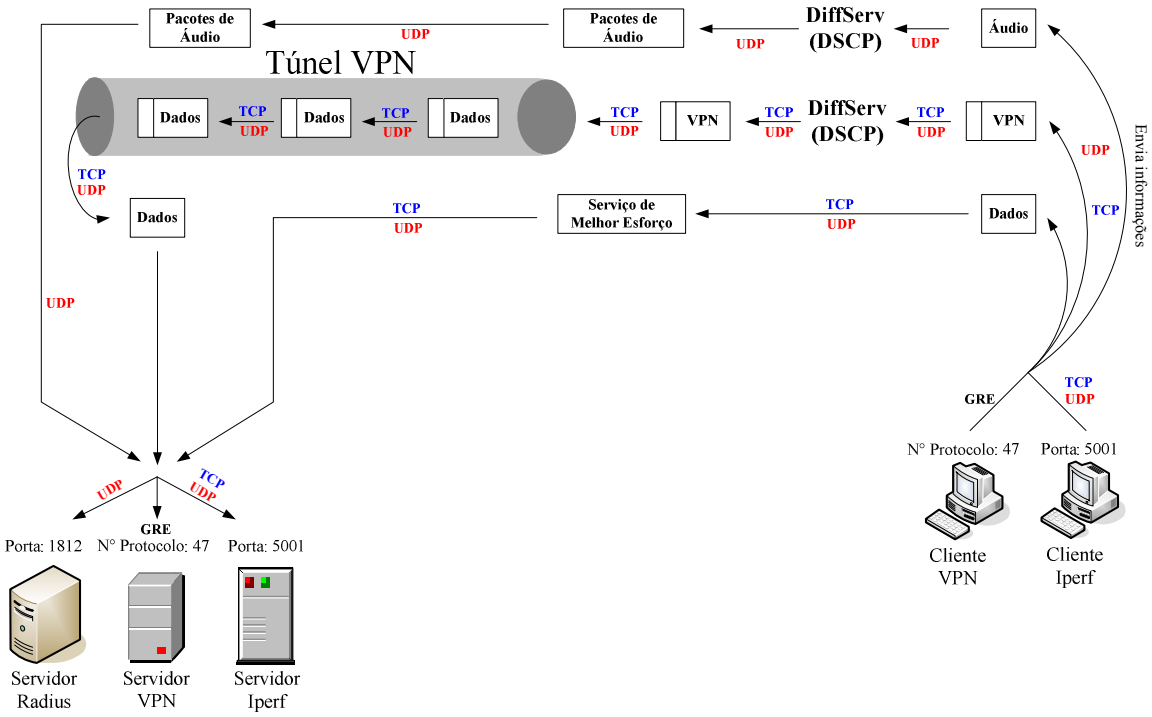

Figura 26 - Camada 4 ou Transporte

A Figura 26 mostra a importância do DiffServ, que prioriza aplicações na rede, e também dos protocolos básicos de rede.

O usuário autenticado pode usar serviços de melhor esforço (FTP, HTTP, etc), serviços de áudio e utilizar a VPN. Foi dada prioridade máxima (DSCP = 0xb8, em hexadecimal ou 46, em decimal) ao tráfego da VPN, para garantir que outros serviços não atrapalhem o desempenho da mesma. O áudio também não pode sofrer muitas perdas, portanto, recebeu uma prioridade que está classificada num nível de médio a alto (DSCP = 0x30, em hexadecimal ou 12 , em decimal). Os diferentes níveis de prioridade estão ilustrados nas Quadros 3 e 4.

Quadro 3 – Classes do DiffServ

DS-class	DS-name	DS codepoint (decimal)
EF	Expedited Forwarding	46
AF1x	Assure Forwarding class 1 (x=1, 2, or 3)	10, 12, 14
AF2x	Assure Forwarding class 2	18, 20, 22
AF3x	Assure Forwarding class 3	26, 28, 30
BE	Best-effort	0

Fonte: Balliache (1998)

O Quadro 3 mostra as classes do serviço diferenciado e seus respectivos DSCP em decimal.

Quadro 4 – Códigos DSCP em hexa para as classes do Quadro 3

CLASS	DP	DSCP	b-DSCP	x-DCSP	b-DS	x-DS
AF1	1	001010	0000-1010	0xa	0010-1000	0x28
	2	001100	0000-1100	0xc	0011-0000	0x30
	3	001110	0000-1110	0xe	0011-1000	0x38
AF2	1	010010	0001-0010	0x12	0100-1000	0x48
	2	010100	0001-0100	0x14	0101-0000	0x50
	3	010110	0001-0110	0x16	0101-1000	0x58
AF3	1	011010	0001-1010	0x1a	0110-1000	0x68
	2	011100	0001-1100	0x1c	0111-0000	0x70
	3	011110	0001-1110	0x1e	0111-1000	0x78
AF4	1	100010	0010-0010	0x22	1000-1000	0x88
	2	100100	0010-0100	0x24	1001-0000	0x90
	3	100110	0010-0110	0x26	1001-1000	0x98
EF		101110	0010-1110	0x2e	1011-1000	0xb8

Fonte: Balliache (1998)

O Quadro 4 mostra também as classes, com seus respectivos DSCP em binário e hexadecimal.

Nesta camada é onde atuam os protocolos TCP e UDP, que são protocolos de transporte. O servidor Radius trabalha somente com o protocolo UDP, enquanto que o servidor/cliente Iperf trabalham com ambos. O servidor/cliente VPN utilizam o *Generic Routing Encapsulation* (GRE), responsável pela formação do túnel que, em conjunto com o IPSec, encapsulam os dados, completando as conexões seguras entre os dois sistemas finais.

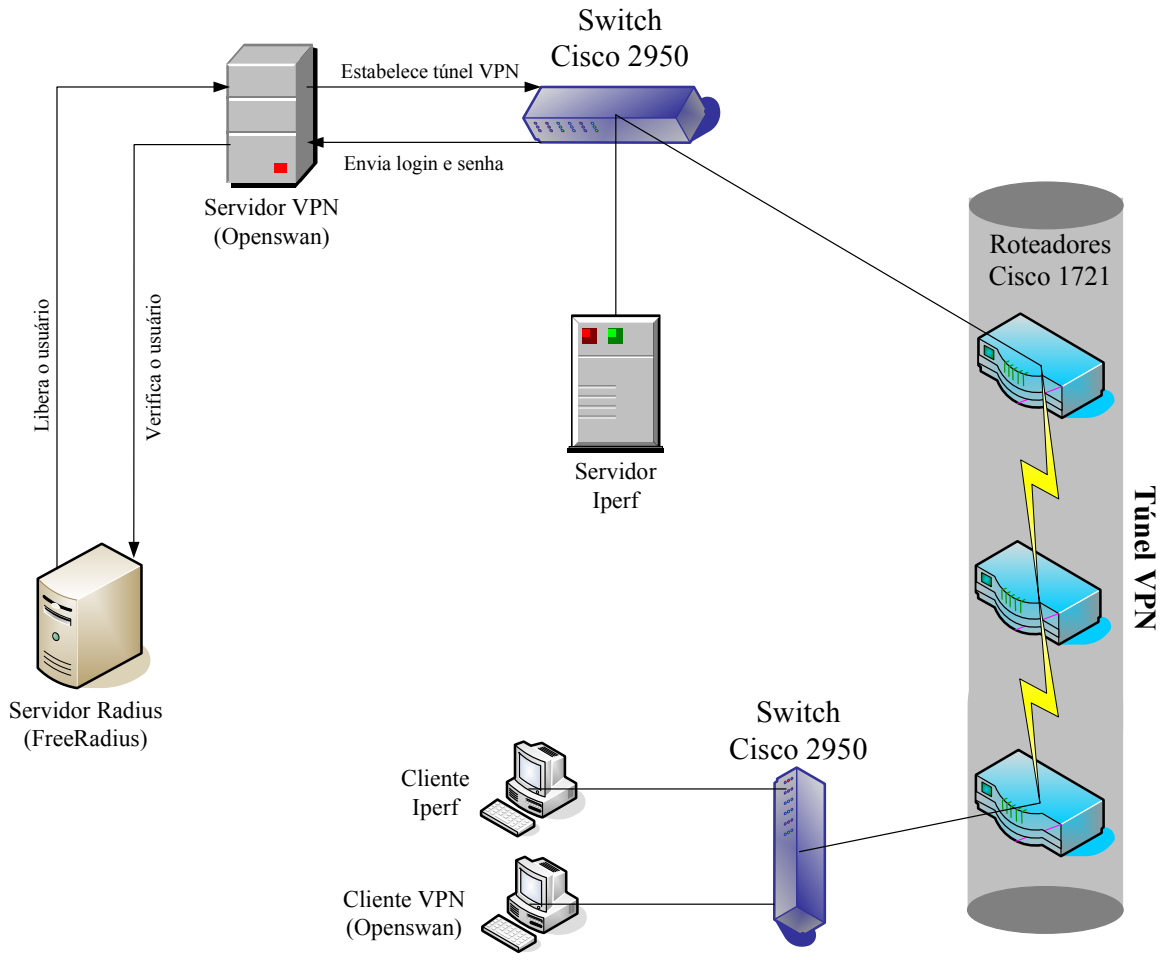

Figura 27 - Camada 5 ou Aplicação

A camada 5 é responsável por oferecer serviços de rede aos aplicativos de usuário. A Figura 27 apresenta o servidor e cliente Iperf, responsáveis pela geração de tráfego na rede; servidor FreeRadius, que autentica os usuários; e servidor e cliente VPN, representados pelo software Openswan.

6.2 Problemas/Soluções

Durante a fase de execução da proposta, alguns problemas foram ocorrendo. A seguir, estão descritos os principais problemas e a abordagem adotada para solucioná-los e contorná-los.

Dentre a escolha das ferramentas, o servidor de VPN mostrou-se uma decisão crítica, em função de alguns fatores, como:

- **Protocolos disponíveis (IPSec, L2TP, PPTP, L2F)**: necessitava-se de um que trabalhasse com IPSec;
- **Compatibilidade com Radius**: nem todos os servidores de VPN são compatíveis com um servidor radius.

Para solucionar esta questão, foram testados três servidores diferentes: Freeswan, OpenVPN e por último, o Openswan, que foi o que melhor se adequou.

Outra ferramenta que apresentou problema foi o servidor de autenticação. Devido aos diversos tipos de *softwares* e versões, é complicado fazer a integração entre os *softwares*, neste caso, entre o Openswan e o FreeRadius.

O protocolo RSVP, inicialmente proposto, não atendeu aos objetivos do trabalho. Apesar de a configuração nos roteadores ser simples, faltaram aplicações que fizessem a reserva de banda para testar este protocolo. Foi então, utilizada outra tecnologia, o DiffServ, que garante a QoS no túnel. Esta arquitetura é suportada pela ferramenta escolhida, o Iperf.

6.3 Desenho da Tecnologia

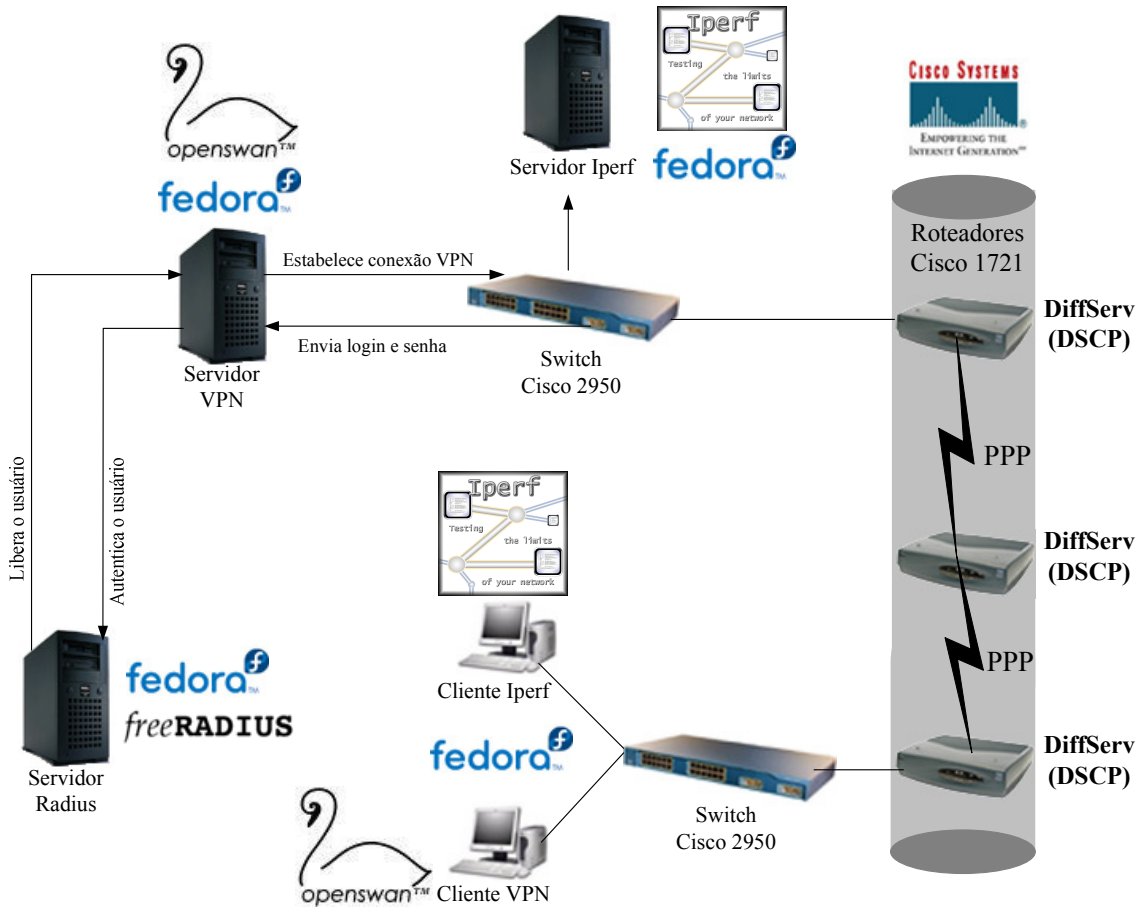

Figura 28 – Tecnologias envolvidas no projeto

Estão representados o hardware e as ferramentas de softwares usados na solução.

7 TESTES E RESULTADOS

A validação deste projeto foi realizada através de testes estatísticos. O *software* utilizado foi o Statistica versão 6.0, disponível em <http://www.statsoft.com>.

Num ambiente real de laboratório, onde foi desenvolvido o trabalho, foram feitos os experimentos e coletados seus resultados. Após isso, os valores foram analisados pelo *software* de estatística, pela metodologia ANOVA. De acordo com o manual do *software*, a finalidade do *analysis of variance* (ANOVA) ou análise de variança é testar as diferenças significativas entre os meios, comparando as variações.

7.1 Configuração dos cenários de testes

Para a realização dos testes, foram coletados dados usando como fatores de variação a banda e o tamanho dos pacotes dos tráfegos de VPN, áudio e *Best Effort*, conforme Figura 29. Para cada situação, foram aplicados três experimentos com duração de um minuto cada. O protocolo da camada de transporte utilizado foi o UDP, que é usado pelas aplicações de áudio, possui os resultados mais detalhados pela ferramenta Iperf e é mais simples.

Cada ambiente recebeu dois tipos de testes: um sem QoS e outro onde o tráfego de VPN e o de áudio estavam configurados com DiffServ, sendo que a VPN tinha prioridade máxima e áudio tinha prioridade de média a alta, com classes DiffServ 46 e 12, respectivamente, conforme Quadro 3.

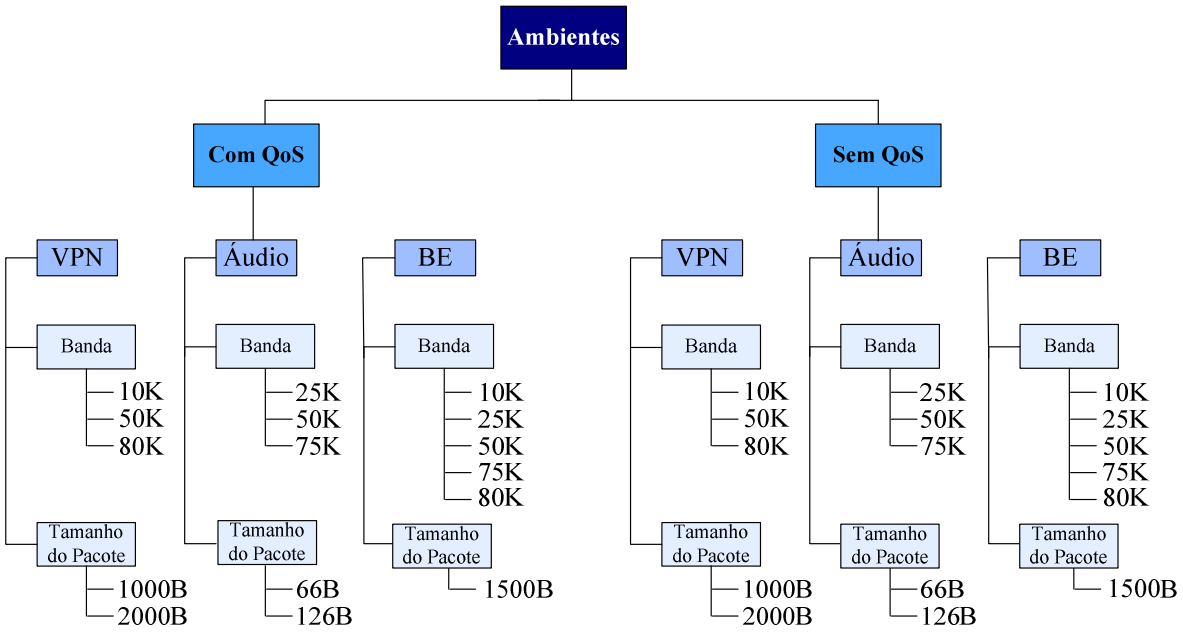

Figura 29 – Cenários e fatores utilizados nos testes

Segundo Cerutti (2006), o tamanho dos pacotes de áudio corresponde à soma da carga gerada pelo CODEC (Codificador/Decodificador) e a sobrecarga dos cabeçalhos.

Cerutti (2006, p. 45), oferece uma boa explicação sobre a obtenção do tamanho do pacote de 66 bytes:

> Quando a camada de enlace utiliza PPP como protocolo, as taxas necessárias devem considerar o cabeçalho de 6 octetos, totalizando (para CODEC G.729a):
> Carga=8000/50 = 160 bits (20 octetos)
> Cabeçalhos RTP/UDP/IP = 40 octetos
> Cabeçalho PPP = 6 octetos
> Total = 66 octetos

Cerutti (2006) aponta que, para os pacotes de áudio de 126 *bytes*, é utilizado o CODEC G.726.

Os motivos pelos quais foram escolhidos tais CODECS é que o G.729a é o mais usado atualmente e o G.726, por apresentar tamanho maior.

O tamanho do pacote do BE foi de 1500 *bytes*, pois de acordo com Cerutti (2006), a *maximum transmission unit* (MTU) ou unidade de transmissão máxima da camada de enlace (normalmente *ethernet*) é igual a 1500 *bytes*.

O tamanho do pacote da VPN foi de 1000 e 2000 *bytes* para fazer testes com valores abaixo e acima da MTU.

As análises estão apresentadas a seguir, através de figuras e tabelas.

7.2 Resultados para a variável *jitter*

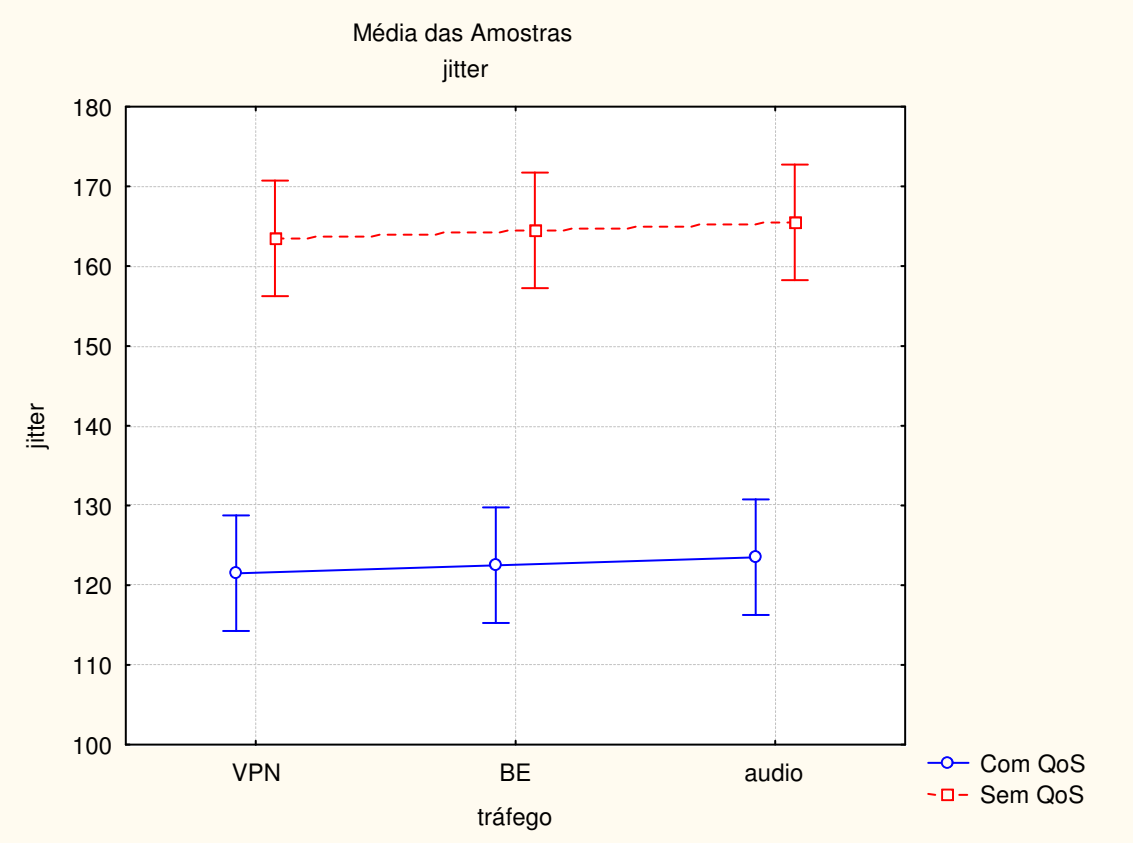

Figura 30 - Variação do *jitter* em relação ao uso de QoS

A Figura 30 mostra a variação do *jitter* em relação ao uso de QoS para os tráfegos de VPN, *best-effort service* (BE) ou serviço de melhor esforço e áudio. Nota-se que com a aplicação de QoS, o *jitter* possui valores menores do que sem QoS, para os três tipos de tráfegos analisados. Apesar de o áudio ainda não ter obtido um valor de *jitter* desejável, observa-se que o teste feito com QoS resultou num desempenho bem melhor do que o teste sem QoS.

A Tabela 1 exibe os valores do *jitter* para os três tipos de tráfego analisados. Observa-se que no ambiente com QoS, o tráfego de VPN possui um *jitter* de 121 ms, enquanto que sem QoS, o *jitter* passa a ser de 163 ms. O tráfego de áudio possui um *jitter* de 123 ms, na presença de QoS, e 165 ms sem QoS.

Tabela 1 - Variação do *jitter* em relação ao uso de QoS

Tráfego	QoS VPN/Áudio	Jitter	Nº de Amostras
VPN	Sim	121,5000	14
VPN	Não	163,5000	14
BE	Sim	122,5000	14
BE	Não	164,5000	14
Áudio	Sim	123,5000	14
Áudio	Não	165,5000	14
Todos os Grupos		143,5000	84

7.3 Resultados para a variável perda

Esta seção está subdividida em variação da perda em relação ao uso de QoS; variação da perda em relação ao tamanho do pacote; variação da perda em relação ao tamanho do pacote e ao uso de QoS; e variação da perda em relação à largura de banda e ao uso de QoS.

7.3.1 Variação da perda em relação ao uso de QoS

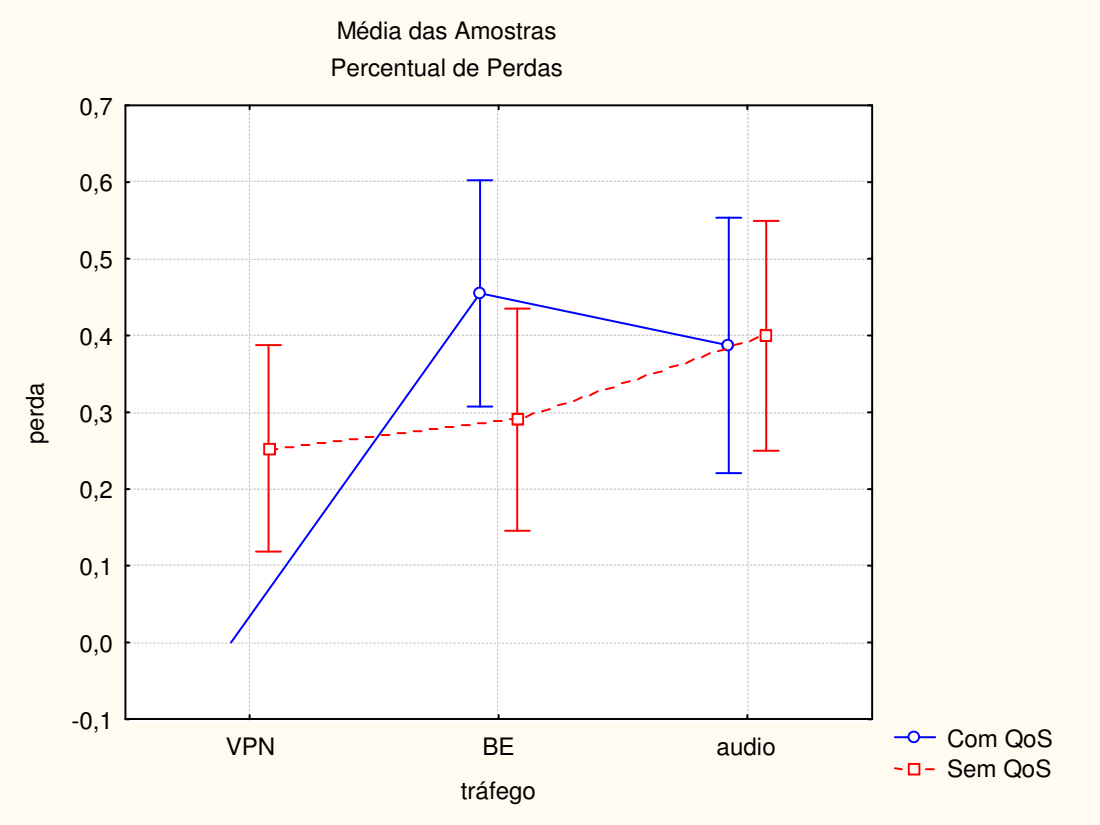

Figura 31 - Variação da perda em relação ao uso de QoS

Constata-se na Figura 31, que a aplicação de QoS exerceu influência positiva no tráfego de VPN, que não sofreu perda em presença do DiffServ. A percentagem de perda do tráfego de áudio se manteve constante, com uma média de 40%. O BE obteve um percentual de perda maior com QoS (aproximadamente 45%) do que sem (aproximadamente 30%). Esse resultado era esperado porque quando é aplicado QoS para VPN e áudio, os pacotes do BE são os últimos no tratamento da política de filas. A banda que sobra para BE é muito estreita, fazendo com que a perda de pacotes aumente para este tráfego.

A Tabela 2 mostra os valores da variação da perda para os tráfegos de VPN, BE e áudio. Nota-se que no ambiente sem QoS, o tráfego de VPN sofreu uma perda de 25%.

Tabela 2 - Variação da perda em relação ao uso de QoS

Tráfego	QoS VPN/Áudio	Perda	N° de Amostras	Desvio Padrão
VPN	Sim	0,000000	14	0,000000
VPN	Não	0,253071	14	0,232851
BE	Sim	0,455000	14	0,255396
BE	Não	0,290286	14	0,250620
Áudio	Sim	0,387143	14	0,288509
Áudio	Não	0,399429	14	0,259268
Todos os Grupos		0,297488	84	0,273266

7.3.2 Variação da perda em relação ao tamanho do pacote

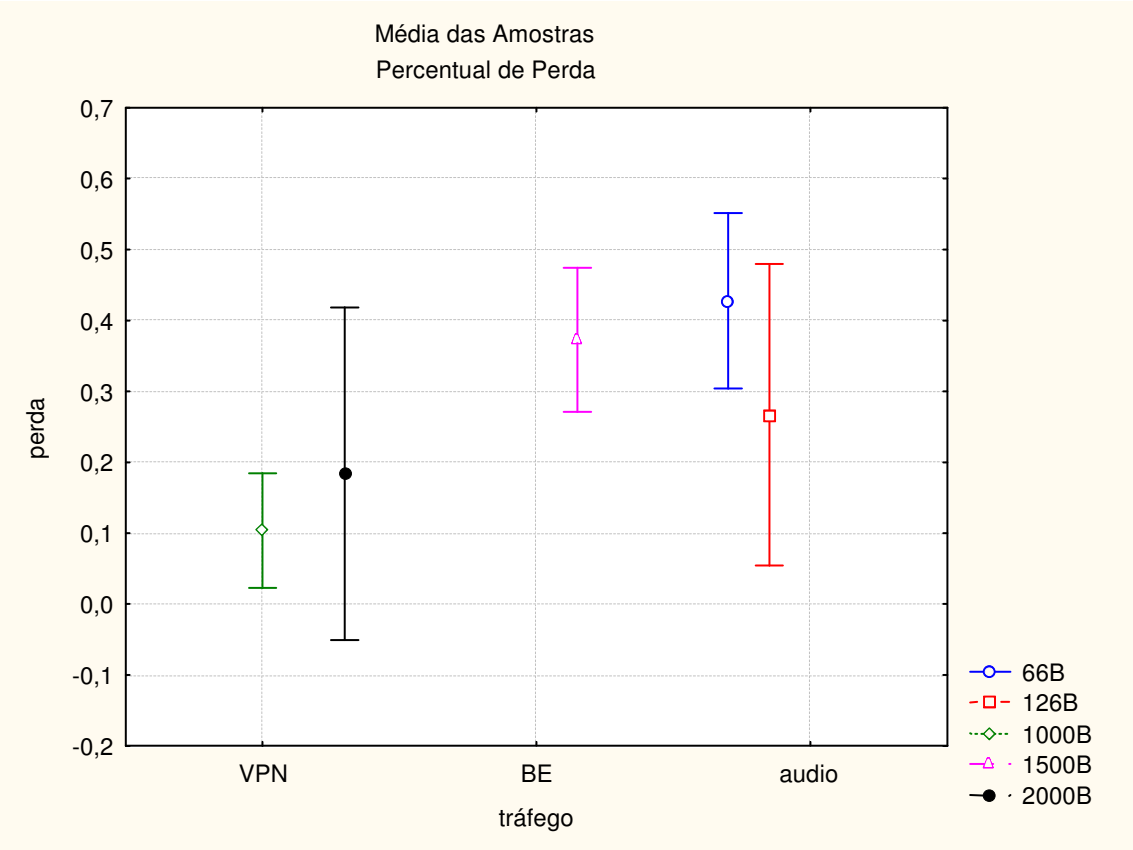

Figura 32 - Variação da perda em relação ao tamanho do pacote

A Figura 32 mostra a variação da perda em relação ao tamanho do pacote para os tráfegos de VPN, BE e áudio. Observou-se que para o tráfego de VPN, os pacotes menores (1000 *bytes*) tiveram menos perda do que os maiores (2000 *bytes*), nos experimentos com e sem QoS. Os pacotes de 1000 *bytes* tiveram uma perda de 10%, enquanto que os de 2000 *bytes* perderam quase 20%.

Para o tráfego de áudio, os pacotes menores (66 *bytes*) tiveram mais perda do que os maiores (126 *bytes*). Esse resultado era esperado, pois segundo Cerutti (2006), normalmente as perdas de pacotes podem se dar por congestionamentos (*buffers* cheios) ou

erros (nas interfaces, nos cabeçalhos), e como tais *buffer*s medem a capacidade em *bytes*, e não em número de pacotes, espera-se que os pacotes menores tenham taxas de perda maiores. Os pacotes de 66 *bytes* tiveram uma perda de aproximadamente 40%, enquanto que a perda dos de 126 *bytes* chegou perto dos 30%.

O tamanho do pacote de BE se manteve o mesmo (1500 *bytes*), tendo uma percentagem de perda de aproximadamente 40%.

A Tabela 3 apresenta os valores de perda para os três tipos de tráfego, com seus variados tamanhos de pacotes.

Tabela 3 - Variação da perda em relação ao tamanho do pacote

Tráfego	Tamanho	Perda	N° de Amostras	Desvio Padrão
VPN	1000	0,103650	20	0,172432
VPN	2000	0,183750	8	0,280710
BE	1500	0,372643	28	0,262072
Áudio	66	0,427727	22	0,278567
Áudio	126	0,267000	6	0,202795
Todos os Grupos		0,297488	84	0,273266

7.3.3 Variação da perda em relação ao tamanho do pacote e ao uso de QoS

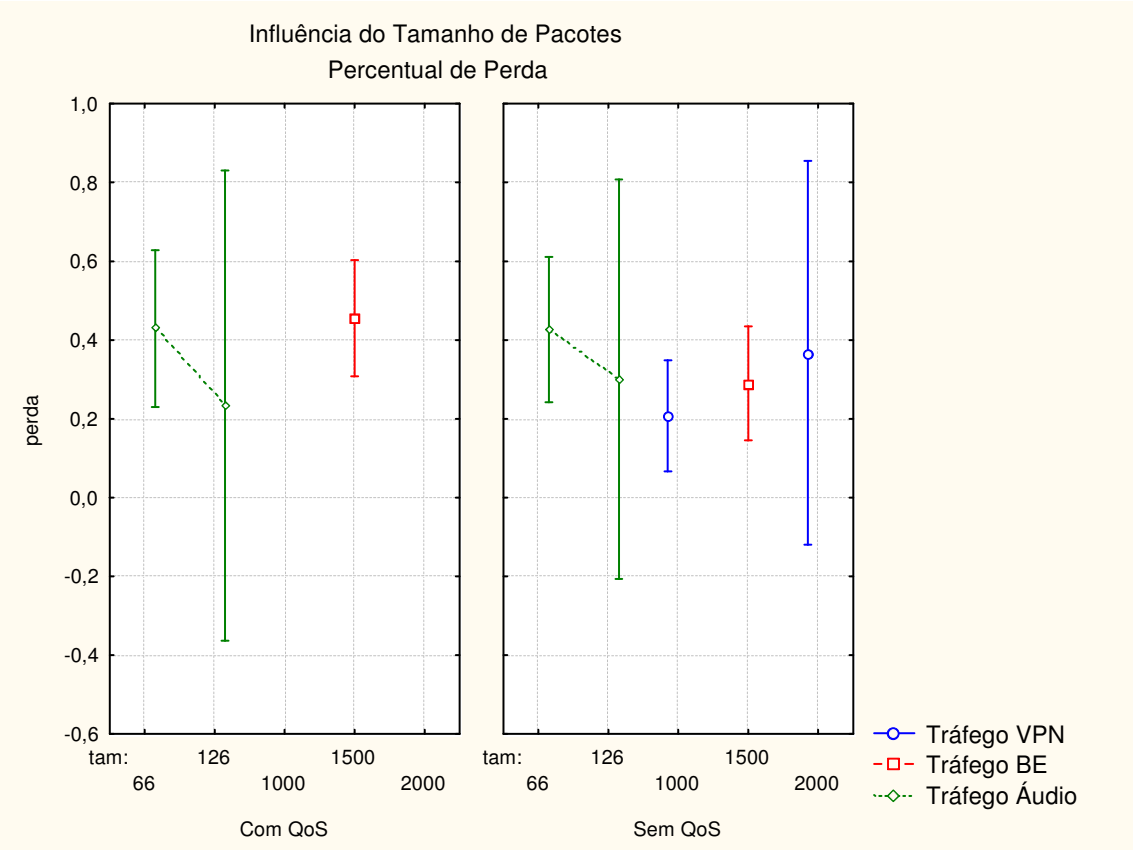

Figura 33 - Variação da perda em relação ao tamanho do pacote e ao uso de QoS

A Figura 33 mostra a variação da perda em relação ao tamanho do pacote e ao uso de QoS. Nota-se que com a aplicação de QoS, a VPN não sofre nenhuma perda, independente do tamanho do pacote. O áudio sofre uma perda de aproximadamente 40%, com um pacote de tamanho 66 *bytes*. Com um pacote de tamanho 126 *bytes*, a percentagem de perda cai para 20%, aproximadamente. O BE tem uma perda de aproximadamente 40%.

Sem a aplicação de QoS, a VPN tem uma percentagem de perda considerável, de 20% para pacotes com tamanho de 1000B e 36% para pacotes com tamanho de 2000B.

A Tabela 4 exibe os valores de perda para os tráfegos de VPN, BE e áudio.

Tabela 4 - Variação da perda em relação ao tamanho do pacote e ao uso de QoS

Tráfego	Tamanho	QoS VPN/Áudio	Perda	N° de Amostras	Desvio Padrão
VPN	1000	Sim	0,000000	10	0,000000
VPN	1000	Não	0,207300	10	0,197218
VPN	2000	Sim	0,000000	4	0,000000
VPN	2000	Não	0,367500	4	0,306309
BE	1500	Sim	0,455000	14	0,255396
BE	1500	Não	0,290286	14	0,250620
Áudio	66	Sim	0,429091	11	0,296022
Áudio	66	Não	0,426364	11	0,274455
Áudio	126	Sim	0,233333	3	0,240278
Áudio	126	Não	0,300667	3	0,204160
Todos os Grupos			0,297488	84	0,273266

7.3.4 Variação da perda em relação à largura de banda e ao uso de QoS

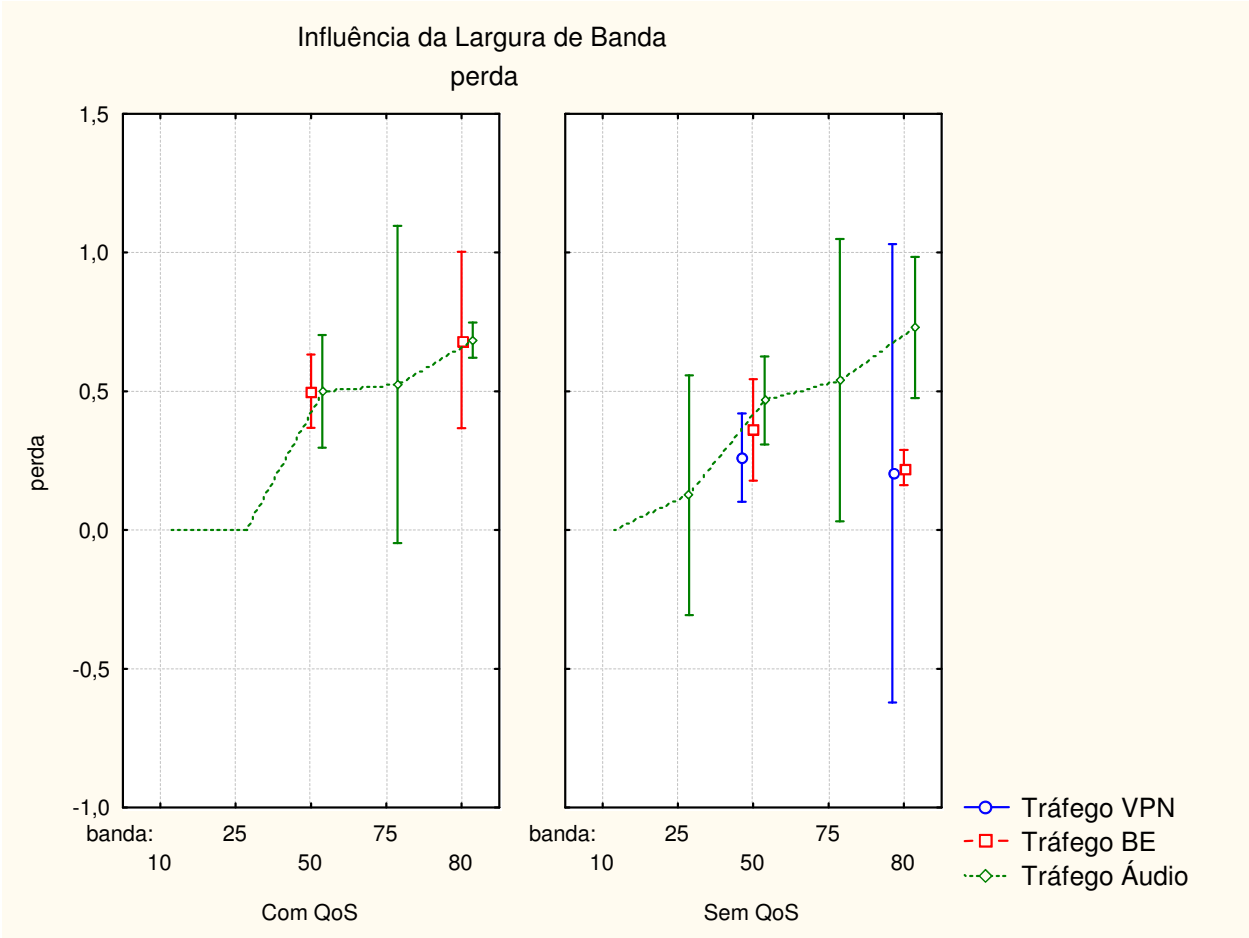

Figura 34 - Variação da perda em relação à largura de banda e ao uso de QoS

A Figura 34 mostra a variação da perda em relação à largura de banda e ao uso de QoS. Nota-se que com a aplicação de QoS, a VPN não sofre nenhuma perda. Para os demais tráfegos, quanto maior for a banda utilizada, maior será a percentagem de perda de pacotes.

Sem a aplicação de QoS, a VPN passa a ter uma percentagem de perda, de modo que, quanto maior for a banda utilizada, maior o desvio padrão das perdas. O tráfego de BE depende dos demais tráfegos. Não seguiu nenhum padrão, sendo que com 10K não teve perda, com 50K teve uma perda de 36% e com 80K perdeu 22%. Já com o áudio, as perdas continuaram aumentando conforme se aumentou a banda.

A Tabela 5 exibe os valores de perda para os tráfegos de VPN, BE e áudio.

Tabela 5 - Variação da perda em relação à largura de banda e ao uso de QoS

Tráfego	Banda	QoS VPN/Áudio	Perda	N° de Amostras	Desvio Padrão
VPN	50	Sim	0,000000	12	0,000000
VPN	50	Não	0,261083	12	0,250638
VPN	80	Sim	0,000000	2	0,000000
VPN	80	Não	0,205000	2	0,091924
BE	10	Sim	0,000000	2	0,000000
BE	10	Não	0,000000	2	0,000000
BE	50	Sim	0,500000	10	0,184572
BE	50	Não	0,361400	10	0,255790
BE	80	Sim	0,685000	2	0,035355
BE	80	Não	0,225000	2	0,007071
Áudio	10	Sim	0,000000	2	0,000000
Áudio	10	Não	0,000000	2	0,000000
Áudio	25	Sim	0,000000	2	0,000000
Áudio	25	Não	0,126000	2	0,048083
Áudio	50	Sim	0,500000	6	0,193184
Áudio	50	Não	0,466667	6	0,150820
Áudio	75	Sim	0,525000	2	0,063640
Áudio	75	Não	0,540000	2	0,056569
Áudio	80	Sim	0,685000	2	0,007071
Áudio	80	Não	0,730000	2	0,028284

8 CONCLUSÕES

Com o rápido crescimento da Internet e a vasta extensão das redes de computadores, foi necessária a utilização de tecnologias além daquelas que existiam até então, para assegurar a transmissão das informações. Antigamente, como as redes não englobavam grandes áreas e existiam poucas aplicações, não havia necessidade de usar tecnologias que pudessem melhorar o desempenho das mesmas. A quantidade de arquivos transmitidos era muito pequena. Os pacotes também eram pequenos.

A realidade atual exige cuidados especiais com uma rede de computadores. Na maioria das vezes são extensas e recebem uma carga de pacotes muito grande. Precisa-se garantir que os dados enviados alcancem seu destino sem perda de pacotes e atraso, especialmente aqueles de tempo real, como áudio e vídeo.

Neste projeto foi apresentada uma solução para garantir largura de banda suficiente, para que aplicativos passem através de um túnel VPN com segurança e sem perder a integridade. A solução propôs o uso de protocolos auxiliares ao IP, como o IPSec, para garantir QoS aos aplicativos da rede.

Este trabalho foi dividido em duas etapas. A primeira etapa do projeto apresentou uma revisão bibliográfica com informações sobre os protocolos básicos de rede, VPN, segurança de rede, QoS e desempenho. Ainda nesta etapa foi desenvolvida a modelagem,

onde todas as camadas do modelo TCP/IP foram detalhadas, através de figuras e textos, explicando o processo que acontece no desenvolvimento da solução.

A segunda etapa do trabalho foi configurar a VPN, o servidor Radius e fazer o provisionamento de QoS nos túneis. Foi analisado o desempenho em um ambiente que consiste de três roteadores, nos quais a VPN foi configurada. Foram realizados experimentos nesse túnel com o *software* Iperf, sem a implantação de QoS, e depois, com o QoS implantado.

Primeiramente, a tecnologia utilizada para configuração de QoS, o RSVP, não atendeu aos objetivos esperados. Apesar de possuir configuração simples, não foi possível encontrar aplicações que fizessem a reserva de recursos. Os principais motivos foram a escassez desses tipos de aplicações na Internet e os prazos para entrega do trabalho que eram curtos.

Optou-se então, em configurar a QoS com o uso de outra tecnologia. O DiffServ, arquitetura adotada, que trabalha com níveis de prioridade, tornou possível marcar cada tipo de tráfego com uma prioridade diferente.

A solução mostrou-se válida após as baterias de testes, que demonstraram que as perdas no túnel da VPN baixaram de aproximadamente 25% para 0%, com a aplicação da solução. O *jitter* passou de 163ms para 121ms, para o tráfego de VPN. Para os pacotes de áudio e BE, o *jitter* também diminuiu significativamente com a QoS. O *jitter* do áudio passou de 165ms para 123ms e o do BE, de 164ms para 122ms.

Notou-se, através dos dados coletados, que o tamanho do pacote não segue um padrão de comportamento, pois para o tráfego de VPN, os pacotes maiores (2000B) perdem mais, provavelmente por excederem o tamanho da MTU (1500B), o que gera necessariamente uma fragmentação. E com o tráfego de áudio, acontece o oposto, os pacotes menores perdem mais, pois os *buffer*s medem a capacidade em *bytes*, e não em número de pacotes,.

Para a variável banda, no geral, o percentual de perda de pacote aumenta conforme o aumento da banda utilizada.

8.1 Dificuldades

Nota-se que a Qualidade de Serviço numa rede, onde vários dados críticos devem ser transmitidos, é imprescindível para se alcançar um bom desempenho. Porém, ainda existem carências de estudo nesta área.

Além da literatura de QoS e análise de desempenho ser escassa, os autores discordam uns dos outros em alguns conceitos.

A QoS, por ser uma tecnologia emergente, não apresenta um serviço bem definido. É diferente da telefonia, onde existem várias operadoras, e todas elas oferecem serviços com os mesmos parâmetros de qualidade. Na Internet, cada provedor adota uma postura diferente, por este motivo não existe motivação para implantar QoS na mesma. A Internet é um sistema aberto, onde novas aplicações surgem diariamente, necessitando parâmetros novos de QoS. Os padrões para serviços são raros ou inexistentes.

8.2 Sugestão para Terceiros

A falta de pesquisas anteriores sobre quais *softwares* seriam mais apropriados para este trabalho causou problemas extras. Foram testados vários programas até se optar pelos

mais compatíveis com a proposta desse trabalho, e também com as versões do sistema operacional.

Uma recomendação é estudar as ferramentas a serem usadas antes de iniciar a configuração. Verificar o suporte aos protocolos que foram propostos para o projeto, como também analisar as dificuldades da mesma. Deve-se adquirir um conhecimento geral de todas as tecnologias que pretendem ser usadas para ver se são compatíveis. Outro fator importante é entender como funciona o SO a ser usado, neste caso, o Linux.

8.3 Perspectivas Futuras

Inúmeros fatores podem tornar diferente o resultado da análise de desempenho. Alguns deles são o algoritmo criptográfico e o protocolo de tunelamento. Uma idéia é trocar o RSA e o IPSec, que foram, respectivamente, o algoritmo de criptografia e o protocolo de tunelamento usado para a configuração da VPN neste projeto, por outros. E após feita a troca, fazer a análise de desempenho e comparar com as já existentes.

Outras sugestões compreendem comparar as duas arquiteturas de QoS, que são IntServ e DiffServ; implantar QoS na parte LAN do ambiente estudado; analisar outras métricas, como o atraso; utilizar aplicações reais para fazer os testes e utilizar ferramentas que reservem recursos para o RSVP.

REFERÊNCIAS

ALMES, G.; KALIDINDI, S.; ZEKAUSKAS, M. **A One-way Packet Loss Metric for IPPM.** RFC 2680, September 1999, IETF.

AMÂNCIO, A. Utilização da VPN na interligação de redes corporativas. **RTI** – Redes, Telecom e Instalações. São Paulo, SP, ano 5, n. 52, p. 28-43, set. 2004.

BALLIACHE, L. **Portal de Software Opal.** Practical QoS. Informações sobre Qualidade de Serviço (QoS) na rede, focando problemas do ponto de vista prático. 1998. Disponível em: <http://www.opalsoft.net/qos/CDS-21.htm>. Acesso em: 2 jun. 2006.

BLAKE, S. et al. **An Architecture for Differentiated Services.** RFC 2475, December 1998, IETF.

BLUM, R. **Network Performance Open Source Toolkit**: Using Netperf, tcptrace, NIST Net, and SSFNet. Indianápolis: Wiley Publishing, Inc., 2003. 405 p.

BRADEN, R. et al. **Resource Reservation Protocol (RSVP).** RFC 2205, September 1997, IETF.

_____; CLARK, D.; SHENKER, S. **Integrated Services in the Internet Architecture: An Overview.** RFC 1633, June 1994, IETF.

CARVALHO, D. B. **Segurança de Dados com Criptografia**: Métodos e Algoritmos. Rio de Janeiro: Book Express, 2000. 218 p.

CERUTTI, F. A. Plano de controle para QoS dinâmica em fluxos de voz em redes IP. 2006. 234p. Tese (Doutorado em Engenharia de Produção) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

CHIN, L. K. **Rede Privada Virtual - VPN.** Produzido e publicado pela RNP - Rede Nacional de Ensino e Pesquisa, em 13 nov. 1998, v 2, n. 8. Promove o uso inovador de redes avançadas no Brasil. Disponível em: <http://www.rnp.br/newsgen/9811/vpn.html#ng-introducao>. Acesso em: 31 mai. 2005.

CHOWDHURY, D. D. **Projetos avançados de redes IP**: roteamento, qualidade de serviço e voz sobre IP. Rio de Janeiro: Campus, 2002. 380 p.

CIRNE, W. **Redes de Computadores.** Desenvolvido pelo professor do Departamento de Sistemas e Computação da Universidade Federal de Campina Grande. Apresenta textos produzidos pelo professor do Departamento de Sistemas e Computação. Paraíba, 2003. Disponível em: <http://walfredo.dsc.ufcg.edu.br/cursos/2003/redes20031/p4a.pdf>. Acesso em: 30 aug. 2005.

COMER**,** D. E. **Redes de computadores e internet**: abrange transmissão de dados, ligação inter-redes e web. 2. ed. Porto Alegre: Bookman, 2001. 522 p.

DEMICHELIS, C.; CHIMENTO, P. **IP Packet Delay Variation Metric for IP Performance Metrics (IPPM).** RFC 3393, November 2002, IETF.

DIÓGENES, Y. **Certificação Cisco**: CCNA 3.0: guia de certificação para o exame 640-607. 2. ed. Rio de Janeiro: Axcel Books, 2002. 411 p.

FERGUSON, P.; HUSTON, G. **Quality of Service**: Delivering QoS on the Internet and in Corporate Networks. New York: John Wiley & Sons, 1998. 288 p.

INFORMATION Sciences Institute. **Transmission Control Protocol.** RFC 793, September 1981, IETF.

JAIN, R. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. New York: John Wiley, 1991. 720 p.

KENT, S.; ATKINSON, R. **Security Architecture for the Internet Protocol.** RFC 2401, November 1998, IETF.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet**: uma nova abordagem. São Paulo: Addison-Wesley, 2003. 548 p.

ORTIZ, E. B.; FERREIRA, E. T. **VPN**: Virtual Private Network: implementando soluções com linux. São Paulo: Érica, 2003. 188 p.

PAXSON, V. et al. **Framework for IP Performance Metrics.** RFC 2330, May 1998, IETF.

PETERSON, L.; DAVIES, B. **Computer Networks: A Systems Approach.** San Francisco: Morgan Kaufman, 1999.

POSTEL, J. **User Datagram Protocol.** RFC 768, August 1980, IETF.

RIBEIRO, P. S. **Um Protocolo Criptográfico Para Comunicação Anônima Segura em Grupo.** 2003. 108 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis.

RIGNEY, C. et al. **Remote Authentication Dial In User Service (RADIUS).** RFC 2138, April 1997, IETF.

SCOTT, C.; WOLFE, P.; ERWIN, M. **Virtual Private Networks.** 2nd ed. Beijing; Sebastopol, CA: O'Reilly, 1999. 211 p.

SILVA, E. L. da; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração de Dissertação.** Universidade Federal de Santa Catarina, 2002. Programa de Pós-Graduação em Engenharia de Produção.

STALLINGS, William. **High-Speed Networks and Internets**: performance and quality of service. 2nd ed. New Jersey: Prentice Hall, 2002. 715 p.

UNIVERSIDADE DO SUL DE SANTA CATARINA. Grupo de Metodologia Científica. **Caderno de metodologia**: diretrizes para a elaboração e apresentação de trabalhos acadêmicos. 2. ed. rev. Tubarão, 2003. 96 p.