Análise Comparativa de um Roteador e Serviços de Segurança Virtualizados

Thiago Antonio Marques[1], **Sérgio M. Trad Júnior**[1]

[1]Departamento de Ciência da Computação – Universidade Presidente Antônio Carlos (UNIPAC)
Rua Palma Bageto Viol S/N – Barbacena – MG – Brasil

thmarqs@gmail.com, smtrad@gmail.com

Abstract. *The studies proposed in this paper have the intention to test the performance and the use of virtualization on routers against dedicated equipment. We've selected two models of routers, a Cisco, with a solid reputation in the market and a Vyatta router, a company that has a different approach by using virtualization in addition to having a completely open source version. Virtualizing a equipment like a firewall comes with many questions about the performance and quality of service before dedicated devices, and along with the results of some tests we should be able to conclude if a virtualized device is better or at least match the performance of a traditional router.*

Resumo. *Os estudos propostos neste trabalho possuem o intuito de testar o desempenho e o uso da virtualização de roteadores contra equipamentos dedicados. Foram escolhidos dois modelos de roteadores, um Cisco, com uma sólida reputação no mercado e o roteador da Vyatta, empresa que apresenta uma abordagem diferenciada utilizando a virtualização além de possuir uma versão totalmente livre. Ao virtualizar um equipamento tal como um firewall surgem vários questionamentos sobre o desempenho e qualidade do serviço perante dispositivos dedicados, e com o resultado de alguns testes podemos concluir se um dispositivo virtualizado é capaz de superar ou ao menos igualar o desempenho de um roteador tradicional.*

1. Introdução

A grande maioria das empresas, sendo de pequenas a grandes, apresenta em seu núcleo de rede, equipamentos dedicados e especializados em comutações na camada três e segurança. A camada três, também chamada de Camada de Rede no modelo OSI[1], segundo [Comer 2007], define como são encaminhados os pacotes de um lado a outro em uma rede.

As soluções líderes no mercado de firewall e roteadores costumam ser dispendiosas, segundo [Modine 2011] a diferença de preço entre soluções de roteamento chegam a $27,804 em um único equipamento. Como o principal objetivo do setor de TI [2] é a valorização dos negócios da companhia e a redução de gastos com equipamentos e sistemas, ao adquirir um equipamento dedicado, boa parte dos seus recursos pode não

[1]Open Systems Interconnect: Um conjunto de protocolos desenvolvidos pela Organização Internacional para Padronização (International Organization for Standardization, ISO) no qual se baseia toda a internet.

[2]Tecnologia da Informação: Define-se por um coletivo de atividades e soluções providas por recursos computacionais.

ser devidamente aplicado, portanto, cabe ao profissional de TI impedir que sejam feitas aquisições de soluções complexas e dispendiosas, correndo risco de se tornarem inviáveis ou subutilizados.

Segundo [Kurose and Ross 2006] a arquitetura de um roteador, nada mais é do que um computador, com um sistema próprio para roteamento e filtragem de pacotes. A partir deste princípio, podemos, instalar em um único equipamento variados sistemas, como Roteadores e Firewalls e adequar estes sistemas ao parque de TI de qualquer organização, sem a real necessidade de novas aquisições de equipamentos.

[Blunden 2002] diz que a utilização de máquinas virtuais, atualmente é uma alternativa viável para vários sistemas de computação, pelos inúmeros ganhos com redução de custos e flexibilização de implementações e manutenção. A virtualização de computadores e sistemas permite que em vez da utilização de inúmeros equipamentos físicos, cada um utilizando apenas um único sistema operacional, utiliza-se somente um hardware, com inúmeras máquinas virtuais rodando vários sistemas operacionais com aplicações e serviços distintos.

1.1. Objetivos

A realização deste trabalho tem como base fundamental, explorar a virtualização de roteadores e firewalls especializados em filtragem de pacotes, como forma de obter um melhor aproveitamento de hardware e economia com custos de licenças de sistemas proprietários, comparando os recursos de sistemas livres virtualizados com os sistemas proprietários.

Neste comparativo entre alguns sistemas especializados em roteamentos e filtragem de pacotes,o principal foco será no sistema IOS[3] da Cisco, líder de mercado em equipamentos de roteamento. Segundo [Santana 2011], a empresa detém em torno 50% da fatia de mercado de roteadores e no Vyatta OS, alternativa livre e totalmente direcionada para virtualização. O Vyatta, conforme diz [Modine 2011], é totalmente compatível com arquiteturas x86, que é a aquitetura mais utilizada em computadores, não dependendo de hardware específico, facilitando a sua virtualização.

Deste modo, o estudo proposto visa permitir:

- Economia de custos com novos equipamentos;
- Agregação de funções em um único sistema (firewall e roteador) facilitando a manutenção e a documentação de uma rede;
- Ajudar na escolha de um sistema de firewall+roteador para implementação em uma rede acadêmica;

1.2. Motivação

A organização alvo deste estudo é uma Universidade Pública Federal que utiliza como principal ferramenta de roteamento e filtragem de pacotes, uma solução proprietária. Por ser uma universidade pública, é importante que custos com softwares proprietários sejam evitados. Foi levantado um questionamento sobre qual seria o melhor sistema para utilização em um parque que comporta cerca de 400 computadores e que seja gratuito. Atualmente o existe uma solução proprietária e fechada, que exerce o papel de firewall e

[3]Internetwork Operating System(IOS): Sistema operacional utilizado por vários equipamentos de rede da Cisco Systems

roteador, para sua substituição foi sugerido um sistema livre e focado em virtualização, o Vyatta.

Tornou-se necessário um estudo para garantir que se o Vyatta atenderá os requisitos necessários para uma operação otimizada em uma rede acadêmica, respeitando algumas peculiaridades da instituição.

Em uma rede universitária, segurança e acesso remoto tornam-se prioritários, devido à necessidade de constantes acessos fora da instituição e por ser foco de ataques. Segundo [Dreher 2011], o crescimento de uma universidade leva a maior exposição e consequentemente a maiores riscos de segurança, justificando assim, uma avaliação sistêmica sobre os recursos de filtragem de pacotes de um roteador, e principalmente, se a virtualização poderá ocasionar uma redução de sua eficiência.

1.3. Metodologia

A pesquisa visa ser um instrumento de orientação com o intuito de mostrar de forma sucinta, uma sugestão de configurações, técnicas e metodologias, de como implementar um sistema de roteamento e filtragem de pacotes, utilizando uma tecnologia diferenciada, que é a virtualização em um ambiente acadêmico. O estudo será desenvolvido através de pesquisas bibliográficas, levantamento de publicações técnicas especializadas e boas práticas em segurança da informação.

2. Virtualização

Segundo [Matthews et al. 2008] o Xen é um monitor de máquina virtual (hipervisor) que possibilita um único computador físico executar inúmeros computadores virtuais. Com a virtualização é permitido conceber e executar vários sistemas operacionais distintos como Windows e um firewall Linux com controle de pacotes e banda, ambos funcionando simultaneamente em um mesmo hardware. O Xen é executado em arquitetura x86, portanto compatível com a maioria das computadores, desde que respeite os requisitos mínimos, e sua codificação é aberta, seu uso não demanda nenhum tipo de gasto com licençaw. O Xen é rápido e não deixa nada a desejar a outros sistemas voltados para servidores, como por exemplo migração ativa.

2.1. Benefícios da Virtualização

Monitores de Máquinas Virtuais são uma método prático de utilizar o mesmo computador físico para executar muitas tarefas diferentes. Em um único sistema de virtualização é possível executar vários sistemas operacionais que possibilita o uso de vários servidores, como servidores de email, de arquivos, web, ambos funcionando em máquinas virtuais diferentes, porém em um mesmo hardware. Não obstante, limitar um sistema operacional por hardware, não aproveita complementamente todos os recursos presentes na máquina, e cria dificuldades ao realizar testes e manutenções. Conforme comenta [Golden and Scheffy 2008] se o seu sistema está totalmente atualizado, pode ser difícil reproduzir problemas de clientes executando versões antigas. Sem máquinas virtuais, seria necessário implementar vários hardwares, cada uma com uma configuração única, mesmo se os recursos computacionais em um hardware forem suficientes para executar,ao mesmo tempo, todos os aplicativos.

Segundo [Golden and Scheffy 2008], a virtualização altera a visão “uma máquina, uma aplicação”, pois foi criada para suportar inúmeras aplicações em um único sistema

físico. A virtualização em computadores é uma maneira eficiente de utilização de equipamentos, aproveitando servidores sem um propósito específico para fornecer uma estrutura compartilhada de hardware oferecendo mobilidade e liberdade na escolha dos sistemas que irão ser utilizados em ambientes de segurança, como firewalls e servidores.

Máquina Virtual, em um conceito básico, nada mais é que um software gerencial chamado hypervisor (hipervisor). [Matthews et al. 2008] comenta que o hipervisor comanda o hardware da máquina, possibilitando que ele seja utilizado por muitos sistemas hóspedes simultaneamente e, desta forma, passando ao *guest* (hóspede) a idéia de exclusividade do hardware. *Guests* utilizam o hardware virtual como se fosse real, o hipervisor assegura que esta impressão seja eficaz. A figura 1 mostra a relação entre o sistema hospedeiro e o hipervisor.

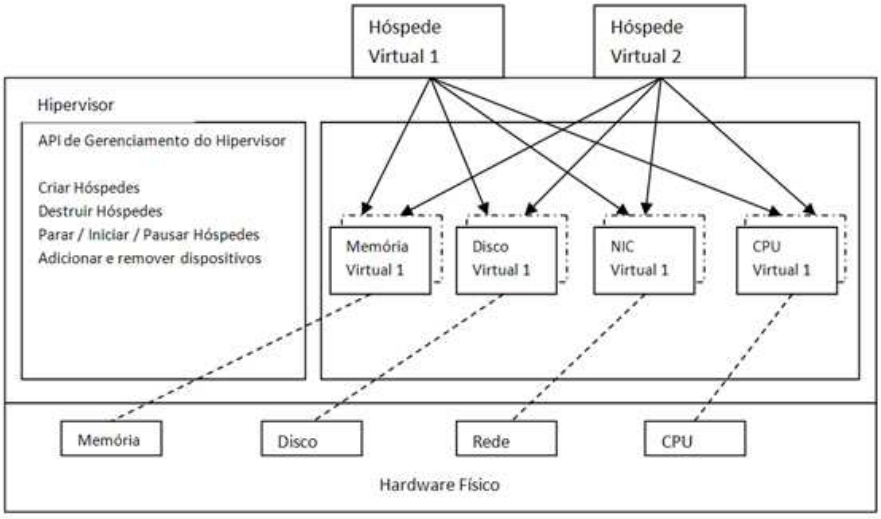

Figura 1. Funcionamento do Hipervisor
Fonte: [Matthews et al. 2008]

Um dos principais benefícios da virtualização, e uma das principais vantagens em utilizar esta tecnologia para empregabilidade em sistemas de segurança é justamente a sua capacidade de recuperar rapidamente de problemas causados por softwares maliciosos, ou por falhas críticas causadas por um mau funcionamento de um sistema. Ao guardar uma cópia utilizável de um *guest*, recuperar-se de uma falha é tão simples como reiniciar um computador.

2.2. Tipos de Virtualização

A virtualização de sistemas hospedeiros pode ser feita de várias maneiras distintas, ambas com benefícios e prejuízos para os *guests*. Segundo [Matthews et al. 2008], dentre os vários métodos, as três principais arquiteturas para virtualização, permitem a ilusão de que vários sistemas funcionem isoladamente em um mesmo hardware, são: emuladores, virtualização completa e paravirtualização.

2.2.1. Emuladores

O autor [Matthews et al. 2008] expõe que nos emuladores, a máquina virtual simula todo o conjunto de hardware necessário para executar os hóspedes através de um software, ou seja, até o processador da máquina virtual precisa ser criado via software no sistema hospedeiro. Isso é ilustrado na tabela 2. O autor [Golden and Scheffy 2008] explica que como a técnica é muito complexa, o desempenho dos hospedeiros são prejudicados severamente. E, segundo [Jones 2011], a emulação do hardware tem algumas vantagens, usando esta técnica é possível criar um sistema hospedeiro sem nenhuma modificação ou adaptação no hardware base. São exemplos de emuladores: PearPC, Bochs e o QEMU.

Figura 2. Máquina Virtual Emulada.
Fonte: [Matthews et al. 2008]

2.2.2. Virtualização Completa

A virtualização completa é parecida com os emuladores. O hardware dos hóspedes são totalmente emulados via software, porém a principal diferença, é que os sistemas e aplicativos são projetados para executar na mesma arquitetura de hardware presente na máquina física, onde os sistemas hospedeiros estão sendo executados. Isso permite que um hóspede, realize várias instruções diretamente no hardware bruto.

Segundo [Matthews et al. 2008], o hipervisor, na virtualização completa, possibilita aos sistemas hospedeiros uma ilusão de terem um único hardware por máquina virtual. São exemplos de virtualização completa os produtos da VMWare, o Win4Lin e o z/VM. O Xen suporta virtualização completa em arquiteturas básicas, isto é, em computadores comuns utilizados diariamente em empresas e residências. [Jones 2011], descreve que, a virtualização completa é mais rápida que a emulação de hardware, mas a performance continua sofrível. A grande vantagem é que se vários sistemas hóspedes, de uma mesma arquitetura, forem criados em um hardware base, os acessos serão otimizados. Isso é ilustrado na tabela 3.

2.2.3. Paravirtualização

A paravirtualização é bem semelhante à técnica da Virtualização Completa, porém o hipervisor fornece aos *guests* uma versão otimizada do hardware físico base, modificações são implementadas para facilitar e acelerar o suporte aos sistemas operacionais hóspedes, possibilitando que os tornem mais adequados a virtualização, segundo [Matthews et al. 2008], ao contrário dos outros métodos de virtualização, na

Figura 3. Máquina Virtual Completa.
Fonte: [Matthews et al. 2008]

paravirtualização o guest sabe que está funcionando sobre uma máquina virtual e coopera com o hipervisor. Hipervisores de paravirtualização são bem parecidos com os da virtualização completa, mas necessitam de alterações nos sistemas operacionais hóspedes, a tabela 4 ilustra o conceito.

Para que a paravirtualização funcione, segundo [Siqueira and Brendel 2007], é necessário que sejam feitas pequenas modificações nos sistemas operacionais hóspedes, o que torna complexo dar suporte em sistemas operacionais com o código fechado[4] como o Windows, porém facilmente praticável em sistemas com o código aberto, como o Linux.

Segundo [Jones 2011], após as adaptações nos sistemas operacionais a interação entre as máquinas virtuais e o hardware base é otimizado substancialmente, aumentando o desempenho e confiabilidade. A grande desvantagem é a dependência da alteração nos sistemas operacionais *guests*. O principal sistema que utiliza a paravirtualização é o Xen.

Figura 4. Máquina Virtual Paravirtualizada.
Fonte: [Matthews et al. 2008]

2.3. Xen e a Paravirtualização

Xen é um sistema de virtualização desenvolvido pela XenSource e a primeira versão veio a público em 2003, mas a instituição foi incorporada pela Citrix[5] em 2007. Segundo [Matthews et al. 2008], pode-se utilizar o Xen sobre uma distribuição Linux qualquer, mas também pode-se simplesmente escolher o XCP (Xen Cloud Platform), um sistema Linux já com o hipervisor integrado, de fácil operação e instalação. O XCP é a versão gratuita do XenServer da Citrix. Segundo [Golden and Scheffy 2008] não conta com todos os recursos da versão comercial, mas, como boa parte dos recursos avançados só é útil para implementações de grande porte, ele se mostra adequado para aplicações de menor

[4]código fechado: Programas que possuem o código fonte bloqueado para verificação ou estudo, ao contrário do código aberto, que possibilita seu estudo e difusão.

[5]Citrix: Empresa fundada em 1989 especializada em virtualização de computadores.

porte. O Xen utiliza técnicas de paravirtualização para alcançar níveis impressionantes de performance.

Após a apresentação de algumas formas de virtualização, conclui-se que a mais adequada é a paravirtualização e seu principal sistema de virtualização é o Xen, com sua versão de fácil utilização Xen Cloud Platform. Os serviços desempenhados por um Roteador demandam grande desempenho do hardware utilizado, [Matthews et al. 2008] diz que, a paravirtualização é o método que possibilita o maior desempenho dentre os demais. Apesar de seus benefícios, a paravirtualização exige que os sistemas possuam alterações necessárias para sua completa virtualização. A principal justificativa, ao selecionar como objetivo de estudo o roteador Vyatta, é sua compatibilidade com sistemas paravirtualizados. Segundo [Modine 2011], o Vyatta possui uma versão modificada, pronta para paravirtualização, utilizando desta forma todo o seu potencial.

3. Máquinas Virtuais e aplicações Críticas

Uma das preocupações de qualquer organização que se utiliza de um parque computacional conectado a Rede Mundial de Computadores, consiste na segurança das informações que são trocadas entre a rede interna e a rede mundial de computadores. A obtenção de informações cruciais por terceiros podem trazem grandes prejuízos as empresas, [Ulbrich 2004] comenta que um computador nunca está totalmente seguro.

Temendo invasões e prejuízos financeiros, as empresas utilizam sistemas de segurança críticos para a privacidade dos dados da empresa, geralmente estas empresas, como cita [Morimoto 2008], em um mesmo servidor instalam vários serviços. Ao utilizar muitos serviços em uma única máquina a possibilidade de falhas é muito maior, de acordo com [Ulbrich 2004], uma simples brecha em um aplicativo, pode corromper todos os serviços em um computador. A solução, portanto, é utilizar um método que permita restringir as aplicações de forma que seu comprometimento não possa se alastrar para os demais serviços. É importante frisar que, mesmo com esta técnica, segundo [Loscocco et al. 1998] as vulnerabilidades de algumas aplicações ainda continuarão a existir, sendo que poderão ser exploradas por um atacante, porém com implicações bem inferiores a segurança do sistema.

A virtualização de computadores encapsula cada serviço em uma máquina virtual distinta, como diz [Matthews et al. 2008], uma máquina virtualizada não pode acessar os domínios de outra, portanto ao utilizar o mesmo hardware, é possível utilizar vários serviços, ambos isolados e blindados. Uma falha de uma determinada aplicação, não irá corromper todos os sistemas de uma organização.

É fácil perceber que quanto mais aplicações servidoras estiverem sendo executadas em uma mesma máquina, maior o número de falhas potenciais a serem exploradas, consequentemente menor a segurança dos sistemas como um todo. Desta forma o ideal seria disponibilizar uma máquina para cada serviço, fazendo com que a possibilidade de invasão seja diminuída drasticamente. Porém, se forem utilizadas vários hardwares bases, para tal o custo seria exorbitante, tanto em hardware, como em manutenção administrativa. O baixo custo das máquinas virtuais torna a possibilidade interessante.

Existem vários questionamentos, no que tange a segurança de dados, ao utilizar várias máquinas virtuais em um único computador, alguns benefícios são bem visíveis

como o isolamento, e a facilidade de criação, após falhas críticas de novos sistemas virtualizados.

3.1. Firewall

O autor [Ulbrich 2004] expõe que, entende-se por firewall, um conjunto de serviços que provém segurança e interligam, como um único ponto de acesso, várias redes distintas, por análise do tráfego entre as regiões internas e externas da rede. A função deste equipamento é controlar o tráfego, permitindo ou bloqueando informações de acordo com regras pré-estabelecidas. [Ulbrich 2004] cita dois tipos de firewalls, são eles: os filtros de pacotes, e os proxies também chamados filtros de controle de aplicação.

3.1.1. Filtragem de Pacotes

De acordo com [Ulbrich 2004], por meio de um conjunto de regras estabelecidas, esse tipo de firewall determina que endereços IPs e dados podem estabelecer comunicação e/ou transmitir/receber dados. [Morimoto 2008] comenta que alguns sistemas ou serviços podem ser liberados completamente, por exemplo, o serviço de e-mail corporativo, enquanto outros são bloqueados por padrão, por terem riscos elevados como softwares de mensagens instantâneas. A principal desvantagem desse tipo de firewall são as regras que podem tornar-se bastante complexas, e como cada pacote deve ser analisado pelo firewall, pode causar perda de desempenho da rede. Este tipo, se restringe a trabalhar nas camadas 2 (enlace de dados) e 3 (rede) do modelo OSI, decidindo quais pacotes de dados podem passar e quais não. Os pacotes são selecionados baseadas nas informações endereço IP remoto, do destinatário, além da porta TCP usada.

O Firewall de filtragem de pacotes possibilita que apenas computadores com permissões explícitas conversem entre si e tenham acesso a recursos privativos na rede interna de uma organização. Um firewall, também possibilita a análise de informações sobre a conexão e observa sensíveis mudanças que podem ser suspeitas, além de ter a capacidade de analisar o conteúdo dos pacotes, o que permite um controle ainda maior do que pode ou não ser acessível.

3.1.2. Proxy Firewall

Firewalls de controle de aplicação, também conhecidos como Proxy, segundo [Ulbrich 2004] não permite a comunicação direta entre a rede e a Internet. Tudo deve passar pelo firewall que atua como uma espécie de intermediador. O Proxy efetua a comunicação entre a rede interna e externa por meio da avaliação do número da sessão TCP dos pacotes. Diferentemente do filtro de pacotes o Proxy trabalha na camada de aplicação do modelo OSI. É um tipo de firewall mais complexo, porém mais seguro.

O firewall de aplicação permite um acompanhamento mais preciso do tráfego entre a rede e a Internet ou entre a rede e outra rede. É possível, inclusive, contar com recursos de registro de eventos e ferramentas de auditoria, ilustra [Morimoto 2008].

3.2. Virtualização de Firewalls

Antes de propor sistemas de segurança baseados em virtualização, faz-se necessário compreender a principal falha que se pretende cobrir, presentes em sistemas não virtualizados. Utilizando virtualização cada usuário administrador possui controle total apenas em um único domínio, não podendo alterar ou influenciar em outros domínios, isto é, em outros sistemas virtualizados. Exemplificando, um servidor de arquivos comprometido, não influenciará no sistema de firewall, pois ambos estão em domínios de virtualização diferenciados.

Entretanto, os sistemas não virtualizados, por funcionarem em apenas um único sistema operacional, apresentam um superusuário que possui acesso irrestrito a qualquer aplicação e arquivo, é também o único com acesso previlegiado a operações envolvendo o hardware. Caso houver comprometimento de qualquer aplicação, todo o sistema operacional poderá ser seriamente afetado.

A virtualização cria uma camada extra de isolamento entre um sistema de Firewall e outras sistemas, como eles estarão isolados de outras máquinas virtuais, ao ocorrer o comprometimento de uma aplicação, a brecha de segurança se restringirá às aplicações contidas na máquina virtual da aplicação insegura. O atacante não poderá, a partir da máquina virtual infectada, coletar dados de qualquer outra máquina virtual, ou mesmo danificar de alguma forma a máquina real.

4. Roteamento de Dados

Dentro de uma rede ou sub-rede, os hosts segundo [Nakamura and Geus 2002] se comunicam uns com os outros sem necessidade de qualquer dispositivo intermediário da camada de rede. Quando um host precisa se comunicar com outra rede, um dispositivo intermediário ou roteador atua como gateway para a outra rede. Como parte de sua configuração, um host[6] possui um gateway[7] padrão definido.

[Morimoto 2008] explica que não é possível para um host qualquer conhecer o endereço de todos os dispositivos da Internet com o qual ele poderá ter que se comunicar. Para comunicar-se com um dispositivo em outra rede, o host usa o endereço deste gateway para encaminhar um pacote para fora de sua rede local. O roteador também precisa de uma rota que define para onde encaminhar o pacote logo em seguida. Isso é chamado de endereço de próximo salto. Se uma rota estiver disponível para o roteador, ele encaminhará o pacote para o roteador de próximo salto que oferece o caminho para a rede de destino.

4.1. Gateway

Conforme [Ulbrich 2004] o gateway é necessário para enviar um pacote para fora da rede local. Se a porção de rede do endereço de destino do pacote for diferente da rede do host de origem, o pacote terá que ser roteado para fora da rede original, para que isso ocorra, o pacote é enviado para o gateway.

[6]Host é qualquer dispositivo ou computador conectado a uma rede.

[7]O gateway é um computador ou dispositivo dedicado, utilizado para unir dois segmentos de redes distintos.

O gateway é a interface de um roteador conectado à rede local. Esta interface possui um endereço da camada de rede que corresponde ao endereço de rede dos hosts. Os hosts são configurados para reconhecer este endereço como o gateway.

4.2. A Ferramenta Vyatta

O Vyatta é uma solução facilitada de roteamento, firewall e diversas outros sistemas, visando a melhor segurança e confiabilidade. Possui código aberto, o que facilita novas implementações e desenvolvimento de módulos que adaptem as necessidades de uma organização. A ferramenta apresenta suporte comercial, foi inicialmente concebida para concorrer com soluções proprietárias e inflexíveis como as ferramentas desenvolvidas pela Cisco.

O Vyatta nada mais que, que uma distribuição Linux baseado no Debian[8] e construída para desempenhar o papel de roteador. Apresenta uma interface de gerenciamento gráfica, o que facilita a gestão das regras e serviços disponibilidades pelo sistema. Segundo o Manual de referência rápida do Vyatta [System 2011b], o gerenciamento também pode ser feito através de linha de comando.

O Vyatta é facilmente adaptável em hardware convencionais, e principalmente, pode ser virtualizado, aumentando ainda mais a gama de benefícios, como a instalação e configuração de serviços que inicialmente não são suportados pelo Vyatta, já que é um sistema Linux, suporta a adição deste tipo de programa. Além de ser de código livre, ainda apresenta suporte comercial, para quem deseja um atendimento mais especializado e customizado, oferecido pela Vyatta Systems além de treinamentos e appliances[9] , que são dispositivos especialmente projetados para utilizarem o Vyatta.

5. A instalação e configuração do ambiente de testes

Como citado anteriormente, para que seja feita a virtualização de um roteador é imprescindível a presença de um sistema hospedeiro. Este será responsável pela gestão e manutenção de todas as máquinas virtuais instaladas. O sistema hospedeiro utilizado neste trabalho é o Xen Cloud Platform - XCP. A versão escolhida do XCP foi a 1.1 Release Candidate, o software pode ser baixado no site do projeto Xen [10]. A instalação e configuração do servidor XCP podem ser visualizadas no Anexo 3 deste trabalho.

O computador utilizado para a instalação do XCP possui seguinte configuração básica:

- Processador Intel I3 540 3,06 GHz
- Memória 4 GB DDR3 800 MHz
- SATA 3 500 GB
- 3 placas de Rede Ethernet 100 Megabit

[8]Debian é uma distribuição do Linux, o que significa que ela é responsável por organizar todos os programas de computador do Linux, de forma que fique mais fácil para que os usuários instalem, utilizem e mantenham um sistema Linux funcionando.

[9]Appliances: São computadores pré-configurados para executar uma tarefa específica, como servir para compartilhar a conexão com a Web ou como um firewall para a rede, ou mesmo roteadores.

[10]http://xen.org/products/cloudxen.html/

5.1. Instalando e configurando o Vyatta OS

Utilizando o OXC é possível que seja feita a instalação do Vyatta OS utilizando sua interface Web. Após a instalação, o Vyatta estará apto a funcionar como um roteador e firewall, totalmente virtualizado.

Os procedimentos de instalação são simples. A própria empresa Vyatta Systems fornece um template, que segundo [Golden and Scheffy 2008] é um conjunto de pré-configurações que podem conter programas já instalados, como por exemplo, servidores de e-mail ou arquivos de configurações já otimizados a funcionar com determinados hipervisores. O uso destes templates automatiza a criação de máquinas virtuais. Este template, em formato ISO[11] pode ser baixado do site da Vyatta Systems.[12]. O procedimento de instalção do Vyatta OS pode ser conferido no Anexo 4 deste trabalho.

Após a instalação do Vyatta no XCP o roteador está apto para ser configurado de acordo com o planejamento de rede presente na figura 5. Através deste projeto que as configurações do Vyatta irão ser baseadas. Todas os parâmetros podem ser feitos através de interface Web ou por linha de comando. Segundo [System 2011b] as configurações por linha de comando são mais ágeis de serem efetuadas, no final deste trabalho está anexado (Anexo 1) o arquivo de configuração do Vyatta contendo todos os passos necessários para a montagem da rede proposta no projeto.

5.2. Instalando e configurando o IOS da Cisco

O roteador 1841 da Cisco com o IOS versão 12.4 foi o equipamento utilizado na confecção do laboratório mostrado na figura 5, ele possui 256 Megabytes de memória e um processador embargado de 600 MHz, com três placas de rede Megabit Ethernet. O computador que hospeda o XCP e virtualiza o Vyatta possui especificações técnicas superiores ao Cisco, porém o IOS foi desenvolvido para o uso em um hardware otimizado para funcionar como roteador sendo seu projeto, específico neste objetivo, bem diferente do Vyatta OS que é basicamente um computador comum com um sistema próprio para roteamento de dados.

Apesar da Vyatta System também possuir equipamentos projetados para uso exclusivo como roteadores ambos os modelos, computadores ou appliances, possuem a mesma versão de software, a 6.2, dados obtidos consultado o manual do Vyatta [System 2011b].

Os passos necessários na configuração proposta para a concepção do laboratório de testes foi seguida rigorosamente igual tanto no Vyatta, virtualizado, como no roteador da Cisco respeitando as particularidades dos dois modelos. No final deste trabalho, está anexo os arquivos de configuração do Cisco e do Vyatta onde é possível visualizar todos os comandos utilizados na configuração dos equipamentos.

5.3. Simulação de tráfego de dados

Neste trabalho foram utilizadas tecnologias que simulam o tráfego de dados em uma rede, desta forma é possível mensurar com maior precisão o desempenho de ambos os roteadores, como diz [Almesberger 2003] a simulação de tráfego é mais prática e confiável do

[11]ISO é um arquivo conhecido como ”imagem de CD”, é uma cópia da matriz original podendo ser alterada ou não.

[12]www.vyatta.org/downloads

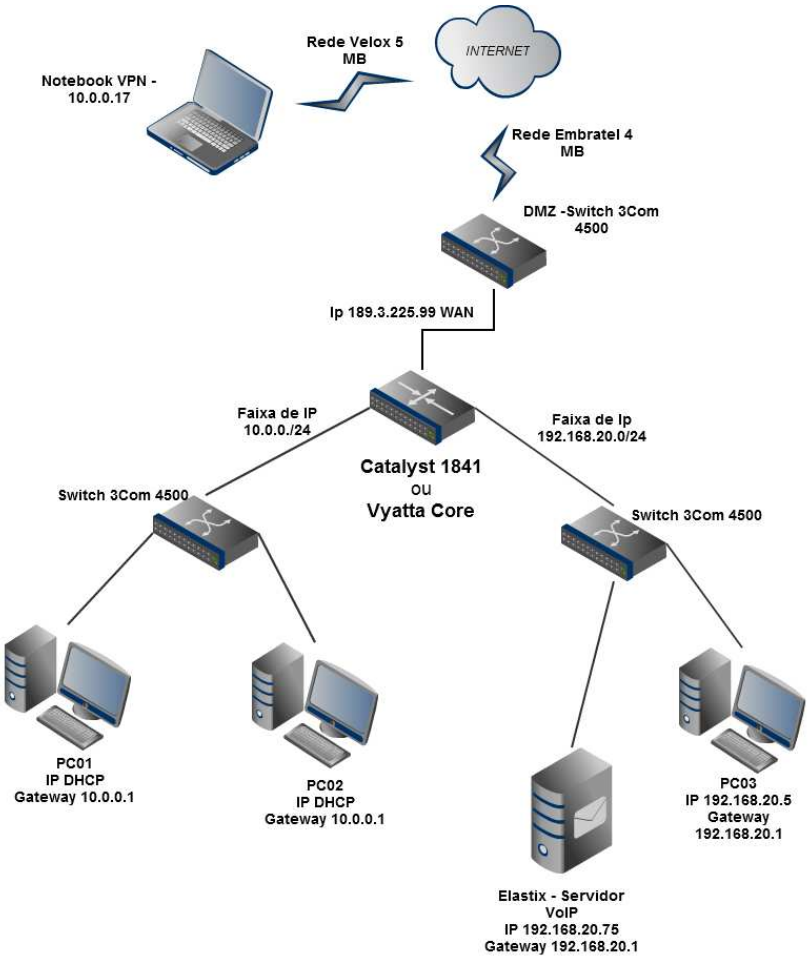

Figura 5. Projeto de Rede

que testar equipamentos diretamente em redes de produção, pois os resultados podem ser inesperados gerando prejuízos aos usuários da rede, além de que, utilizando simuladores de tráfego é possível testar completamente os equipamentos fazendo com que sofram intenso estresse, dificilmente atingido em condições normais de tráfego.

5.3.1. A ferramenta Iperf

A ferramenta utilizada neste trabalho é open source e possui versões em diferentes sistemas operacionais, foi desenvolvida pela Universidade de Illinois, com a finalidade de mensurar e testar a qualidade de um link. O Iperf pode ser utilizado para gerar tráfego em uma rede testando o comportamento de roteadores e repetidores além de avaliar regras de Firewalls. O funcionamento completo da ferramenta pode ser encontrado no site do projeto que também possui [13] o manual [Illinois 2011] que contém explicações detalhadas sobre os comandos que foram utilizados ao longo dos testes apresentados. No anexo 5 é possível ver uma tela de uso do software iperf.

O Iperf gera tráfego através de dois protocolos, O TCP e o UDP, que conforme

[13] http://sourceforge.net/projects/iperf/

[Kurose and Ross 2006] são os mais utilizados na transmissão de dados em redes Ethernet. O manual do Iperf, [Illinois 2011], mostra que cada protocolo pode ser testado separadamente e possui vários opções de ajustes, tornando a simulação do tráfego ainda mais realista. Através das linhas de comando do Iperf é possível alterar a largura da banda, tamanho do MTU[14], tamanho da janela TCP, dentre diversas outras opções.

Os testes realizados utilizaram o protocolo TCP, para medir a largura de banda e como o Roteador se comporta com uma grande quantidade de tráfego, como também a taxa efetiva de dados entregues. O protocolo UDP foi utilizado para simular a perda de pacotes em situações de grande estresse na rede, além de comprovar o funcionamento de regras de contenção e filtragem de pacotes, criadas no Firewall para beneficiar o uso de voz sobre IP.

As métricas utilizadas neste trabalho foram escolhidas visando comparar com maior precisão a performance de cada roteador, utilizando de dados e gráficos. Cada teste foi realizado de forma igualitária nos equipamentos, os ajustes de cada protocolo foram os mesmos para ambos.

6. Cenário de testes

O tráfego de dados deve ser medido entre a transmissão de dois dispositivos conectados através de um comutador, os cenários foram desenvolvidos de acordo com o projeto de rede apresentado na figura 5. O software iperf foi instalado em todas os três computadores, ambos possuem configurações, tanto de software como de hardware idênticas, para que se evite irregularidades nos resultados. Cada um destes computadores possui a seguinte configuração:

- Processador Intel I3 540 3,06 GHz
- Memória 4 GB DDR3 800 MHz
- Hard Drive SATA 500 GB
- Rede Megabit Ethernet
- Sistema operacional Windows XP

Os cenários propostos neste trabalho visam a transmissão e a coleta de dados dentro da rede e principalmente para fora da rede local, para isso foram utilizados os computadores PC01, PC02 e PC03, que podem ser vistos no figura 5 que ilustra o projeto de redes utilizado neste trabalho. O PC01 e o PC03, participaram de situações onde o Roteador tem que gerenciar a transmissão de pacotes de dentro da intranet (Rede 10.0.0.0/24) para uma rede externa (Rede 192.168.20.0/24). O PC02 e o PC03 foram utilizados na coleta de dados em situações onde o tráfego gerado sofreria contenção do Firewall.

Foram criadas três situações para serem testadas nestes cenários, estas situações foram desenvolvidas visando à simulação de ambientes reais observadas no dia a dia de uma organização.

6.1. Testando a largura de banda e vazão dos dados.

A primeira situação é o teste de transmissão de pacotes TCP com o objetivo de medir a largura de banda suportada na rede determinando a quantidade de dados transmitidos em

[14]MTU é o acrônimo para a expressão inglesa Maximum Transmission Unit, que em português significa Unidade Máxima de Transmissão, e refere-se ao tamanho do maior datagrama que uma camada de um protocolo de comunicação pode transmitir.

um tempo fixo. Este teste visa verificar qual o limite de carga de trabalho de cada roteador observando a vazão dos dados, isto é, a diferença entre a taxa de dados que é enviada e a que realmente é entregue pelo roteador.

6.2. Testando a filtragem de pacotes pelo firewall e transmissão de dados através do protocolo UDP.

A segunda situação é a transmissão de pacotes UDP, onde o objetivo é verificar a perda de pacotes e o Jitter [15] gerado na transmissão, sendo que um Jitter excessivamente alto, conforme [Morimoto 2008] esclarece, atrapalha ligações via VoIP. A estabilidade na transmissão do protocolo UDP é deveras importante, pois como comenta [Ulbrich 2004] a perda dos pacotes sendo maiores que 1% indicam problemas na transmissão de dados e os clientes teriam dificuldades na utilização de serviços como, por exemplo, uma teleconferência via rede.

Os pacotes UDP também foram utilizados com o objetivo de testarem a capacidade de cada roteador em filtrar pacotes aplicando uma regra de traffic shaping [16].

Para a simulação deste cenário, foi necessário implementar um servidor VOIP (voz sobre protocolo de internet), virtualizado no XCP utilizando a distribuição Linux de PBX Elastix, tanto a instalação como o uso desta ferramenta pode ser encontrada no site [17] do projeto. Com o Elastix, foi possível fazer ligações com voz e vídeo, e desta forma testar o comportamento dos roteadores ao tratar este tipo de tráfego.

6.3. Testando a qualidade na transmissão de pacotes através de um VPN

A terceira situação foi criada para testar e comparar a capacidade de cada roteador em criar e manter a qualidade na transmissão de pacotes através de conexões VPN. A utilização de conexões VPN atualmente é crucial para vários negócios, exemplo, em um universidade um Professor pode facilmente acessar seus projetos de pesquisa através de uma VPN, trabalhando em casa, mas conectado na intranet da instituição com total segurança.

Para este teste cada roteador foi configurado com acesso PPTP[18] que garante facilidade de implementação e segurança dos dados transmitidos. O objetivo desta comparação é verificar a qualidade e a linearidade do acesso.

7. Resultados obtidos

Os resultados são visualizados através de relatórios gerados pela ferramenta Iperf, acompanhados de gráficos. Os relatório são impressos na tela do software a cada segundo, porém neste trabalho foram suprimidos dos relatórios alguns segundos em cada teste, não sofrendo nenhum prejuízo na interpretação dos dados, já que no final existe o somatório total dos dados enviados e recebidos, além dos erros presentes na transmissão.

[15] Jitter é uma variação estatística do atraso na entrega de dados em uma rede, ou seja, pode ser definida como a medida de variação do atraso entre os pacotes sucessivos de dados.

[16] Traffic shaping é um termo da língua inglesa, utilizado para definir a prática de priorização do tráfego de dados, através do condicionamento do débito de redes, a fim de otimizar o uso da largura de banda disponível.

[17] www.elastix.org

[18] PPTP é um protocolo de rede que adiciona uma infra-estrutura de segurança para garantir a transferência de dados de um cliente remoto para um servidor corporativo particular, criando assim uma rede virtual privada (VPN) usando como base outro protocolo, o TCP/IP

Para uma melhor distinção dos equipamentos e o sentido que o tráfego irá percorrer na rede, em cada teste um computador será comumente chamado de cliente o outro de servidor. A distinção pode ser feita facilmente pelos parâmetros utilizados no iperf:

```
Servidor: iperf -s
Cliente: iperf -c 192.168.20.5 -d
```

O servidor utiliza o parâmetro -s e o cliente o parâmetro -c acompanhado do IP do servidor. Os dados são medidos no sentido cliente para o servidor, porém pode-se utilizar o parâmetro -d, que faz com que o tráfego seja feito também no sentido servidor para o cliente, obtendo assim o transporte simultâneo, próximo do que seria uma comunicação full duplex. Com o parâmetro -d o gráfico possui duas linhas. A linha verde seria o tráfego que está sendo enviado e a linha vermelha tráfego que está retornando.

7.1. Medindo o tráfego entre os computadores da Rede Local através do protocolo TCP.

Os dados foram gerados entre os computadores PC01 e PC03, onde o cliente seria o PC01, também foi utilizado o transporte dual, onde o tráfego retorna a interface após o envio. A ferramenta Iperf gera relatórios completos dos testes que estão presentes no anexo 7 deste artigo.

Resultados obtidos utilizando o Vyatta como roteador:

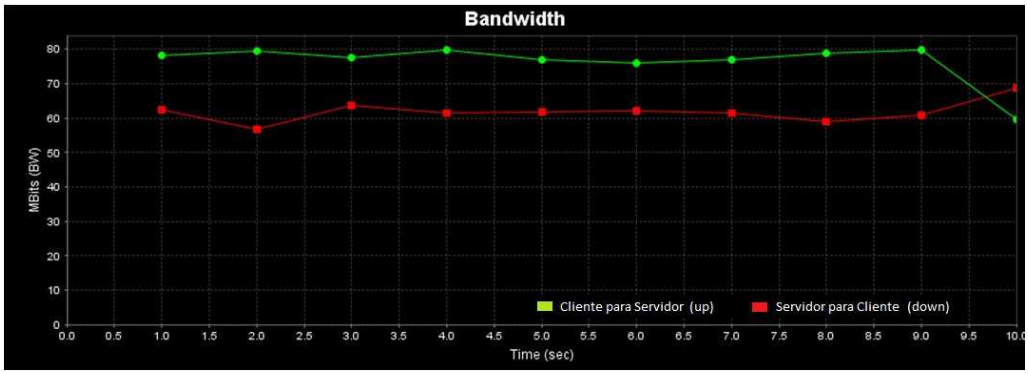

Figura 6. Gráfico da largura de banda gerada pelo cliente - Vyatta.

Resultados obtidos utilizando o Cisco como roteador:

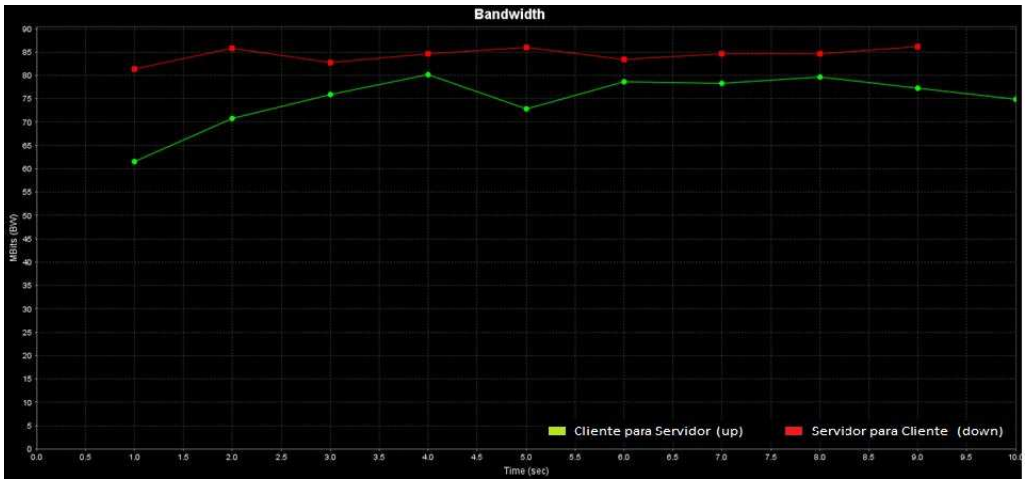

Figura 7. Gráfico da largura de banda gerada pelo cliente - Cisco.

Analisando os resultados obtidos nos relatórios gerados pelo iperf, concluí-se que, a largura de banda máxima obtida na transferência de dados entre os computadores PC01 e PC03, em ambos os roteadores, foram muito semelhantes sendo que o Cisco apresentou desempenho ligeiramente superior, como podemos observar no gráfico 8.

Outro resultado surpreendente foi a taxa de transferência de dados do retorno ao cliente, presente na imagem 7 do gráfico de largura de banda gerada pelo cliente utilizando o roteador Cisco, isso mostra uma melhor velocidade ao tratar os pacotes e reencaminhá-los ao seu correto destino. O buffer de pacotes que o Cisco apresenta se mostrou superior ao Vyatta justificando o resultado.

Comparando os gráficos 6 e 7, a linearidade na transmissão é notável na ferramenta Vyatta, que apesar de entregar um tráfego menor, se mostrou mais estável que o Cisco.

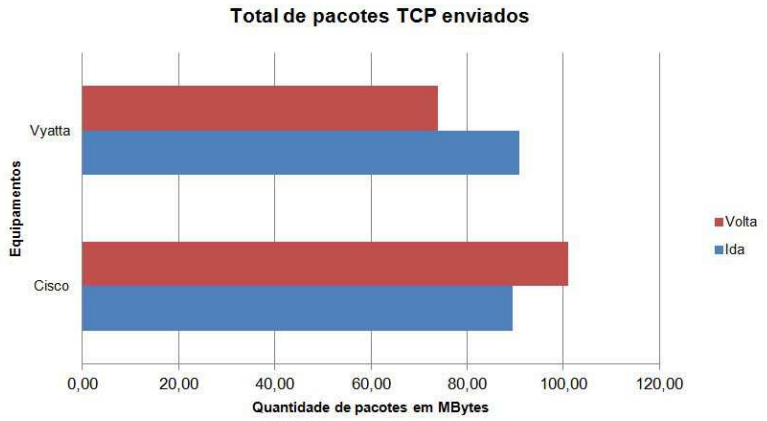

Figura 8. Total de pacotes TCP enviados.

7.2. Medindo a perda de pacotes e o Jitter através do protocolo UDP.

Para testar a perda de pacotes, os dados foram transmitidos entre os computadores PC01 e PC03. O cliente é o PC01, e o servidor o PC03.

Nesta situação o protocolo utilizado foi o UDP, parâmetro -u no Iperf e com uma banda de 12 Mbytes por segundo valor escolhido após seguidas tentativas, no início foram escolhidos valores inferiores, mas que não apresentavam perdas em nenhuma circunstância.

Resultados obtidos utilizando o Cisco como roteador:

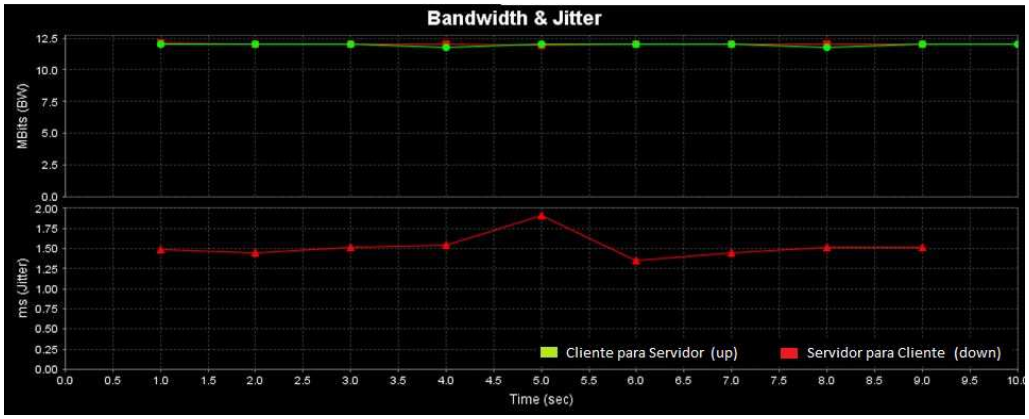

Figura 9. Gráfico do tráfego UDP gerado pelo cliente - Cisco.

Resultados obtidos utilizando o Vyatta como roteador:

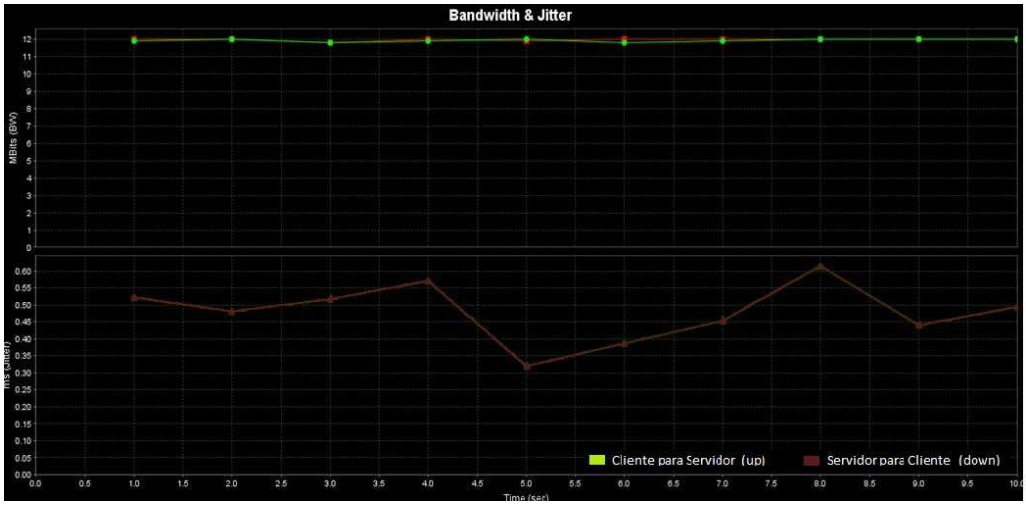

Figura 10. Gráfico do tráfego UDP gerado pelo cliente - Vyatta.

Os testes geraram resultados interessantes, ambos os roteadores tiveram relativo sucesso em obter uma quantidade de perda de pacotes muito irrisória, o Jitter, essencial que seja baixo para um uso otimizado de serviços que demandam grande quantidade de tráfego, ofereceu grande variação entre o Cisco e o Vyatta como podemos ver no gráfico comparativo do Jitter entre roteadores 11, porém ainda em condições razoáveis para uso. O roteador Cisco obteve uma variância de latência inferior ao Vyatta e a perda de pacotes

foi menor, de acordo com o gráfico que compara a perda de pacotes entre os roteadores 12. O Vyatta ainda apresentou uma perda de pacotes ao enviar o tráfego de retorno do cliente (sentido servidor para cliente) bem superior ao Cisco. O ideal é que perda de pacotes nunca seja superior a 1% de todo o tráfego gerado, e em nenhum dos dois roteadores houve esta ocorrência.

Comparando os gráficos 9 e 10, o tráfego gerado pela iperf de 12 MBytes por segundo foi satisfatoriamente entregue em ambos os roteadores, com resultados semelhantes, já o jitter obteve uma variação bem grande no Vyatta, e o Cisco se mostrou mais linear.

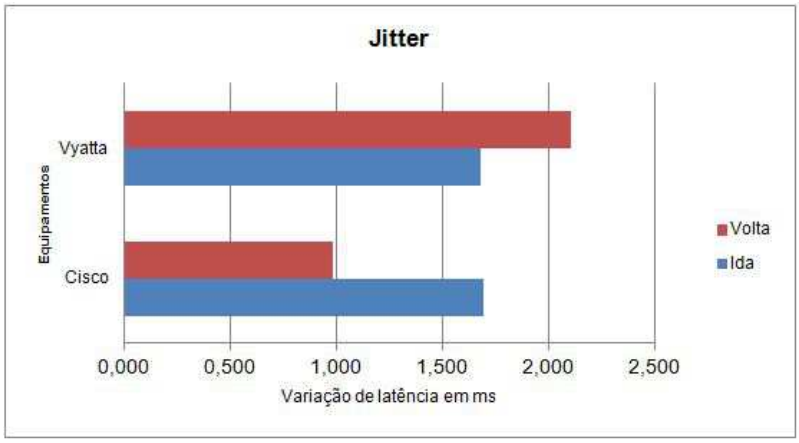

Figura 11. Comparando o Jitter entre os roteadores.

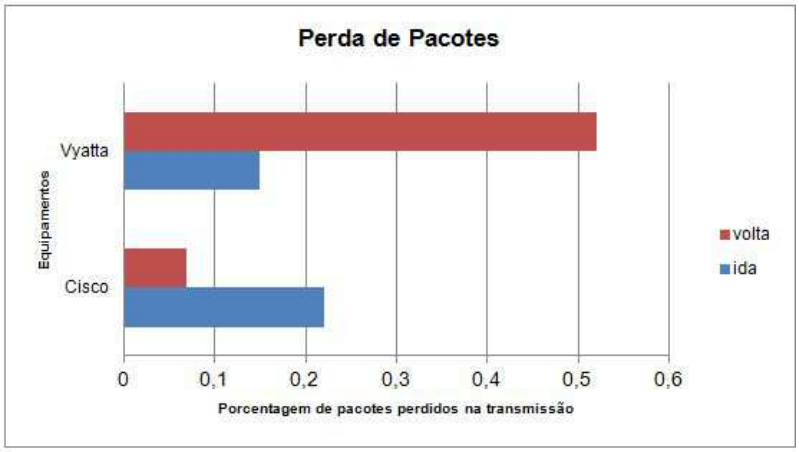

Figura 12. Comparando a perda de pacotes entre os roteadores.

7.3. Medindo o controle de banda através do protocolo UDP.

Para o teste de filtragem de pacotes com traffic shapping, a ligação foi feita utilizando o softfone Ekiga [19], sua operação e configuração podem ser obtidas no site do projeto. Este software terá o SIP provido pelo Elastix, onde será cadastrado dois usuários cada um com um respectivo SIP. [20].

Os SIPs criados foram o 9090 configurado no PC02 e o SIP 100 configurado no PC03. A ligação será feita através do Ekiga discando do PC02 para o PC03 que está fora da Rede Local, desta forma será possível observar a qualidade do sinal na ligação, colocando a regra de controle de banda na saída da interface WAN de cada Roteador. No anexo 6 deste trabalho está presente uma figura com a tela do software Elastix mostrando os SIPs criados.

O controle de tráfego foi feito preservando 60% da banda para ligações VoIP e o teste de envio de pacotes UDP foi executado pelo iperf no sentido do cliente para o servidor no mesmo instante que a ligação estava sendo feita entre os computadores. As imagens com os relatório gerados pelo Iperf estão presentes no anexo 9.

Após a conclusão dos testes o Cisco e o Vyatta tiveram comportamentos bem distintos, durante a ligação entre o PC02 e o PC03 o sinal continuou perfeito, tanto o áudio como o vídeo obtiveram clareza na transmissão em ambos os roteadores, porém ao remover o QoS o Cisco apresentou alguns atrasos durante a ligação, mas não caindo nenhuma vez, já o Vyatta teve duas quedas, sendo necessário que se fosse feita a discagem novamente para continuação dos testes.

Observando os resultados gerados pelo iperf o Cisco obteve uma incrível vantagem ao realizar a contenção do tráfego em favor dos pacotes provenientes do softfone, os relatórios gerados pelo Iperf, presentes no anexo 9 e no gráfico número 13, que mostra os pacotes perdidos com o QoS é possível notar a grande quantidade de perdas de pacotes no servidor, graças a regra de traffic-shapping do Firewall. Com o QoS ligado boa parte dos pacotes UDP foram descartados favorecendo uma ligação clara e sem quedas, porém ao desligar o QoS a perda de pacotes praticamente não aconteceu ficando abaixo de 1%, como mostra a figura do gráfico de envio de pacotes número 16 o que justifica os atrasos na ligação VoIP.

Na figura 13 do gráfico de pacotes perdidos com QoS, é fácil notar que 80% dos pacotes foram descartados pelo Roteador Cisco, porém ao desligar o QoS a figura 14 do gráfico de pacotes perdidos sem QoS mostra claramente que quase todos os pacotes chegam ao servidor mantendo um índice abaixo dos 0,5%.

O Roteador Vyatta obteve resultados insatisfatórios, na figura refimg:relatorioqosvyatta do relatório de envio de pacotes com QoS do Vyatta, presente no anexo 9 e no gráfico 15 o servidor no computador PC03 gerou vários erros impossibilitando que se mostrasse a quantidade de pacotes que foi realmente recebido. Em uma Rede de Produção, isto é um sério problema, pois caso o QoS esteja habilitado outros serviços que utilizem o protocolo UDP podem parar. Ao desligar o QoS no Vyatta o

[19] http://ekiga.org/

[20] SIP, que significa em inglês Session Initiation Protocol (Protocolo de Inicialização de Sessão), é um protocolo de sinalização de telefonia IP usado para estabelecer, modificar e finalizar chamadas telefônicas VoIP

envio de pacotes UDP foi normalizado, o que podemos comprovar na imagem 14 do gráfico de pacotes perdidos sem QoS, porém ainda com uma perda razoável que de acordo com [Ulbrich 2004] é um índice alto para ligações VoIP justificando as duas quedas de conexão no teste entre os computadores PC02 e PC03.

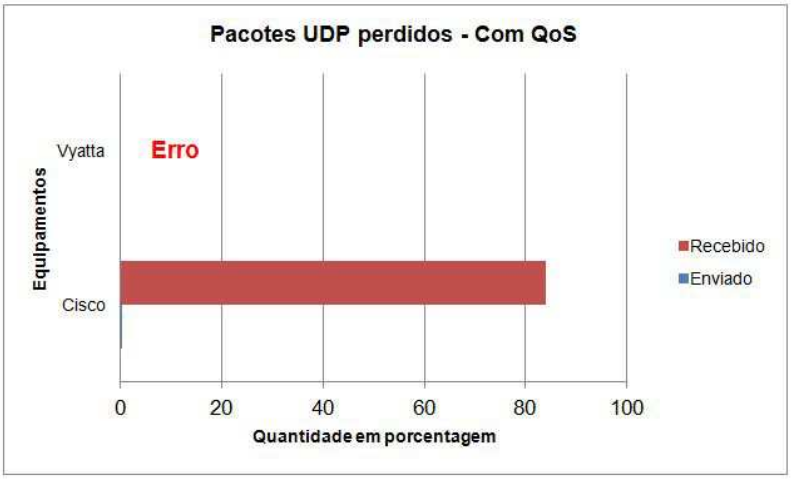

Figura 13. Comparando a perda de pacotes entre os roteadores - Com QoS.

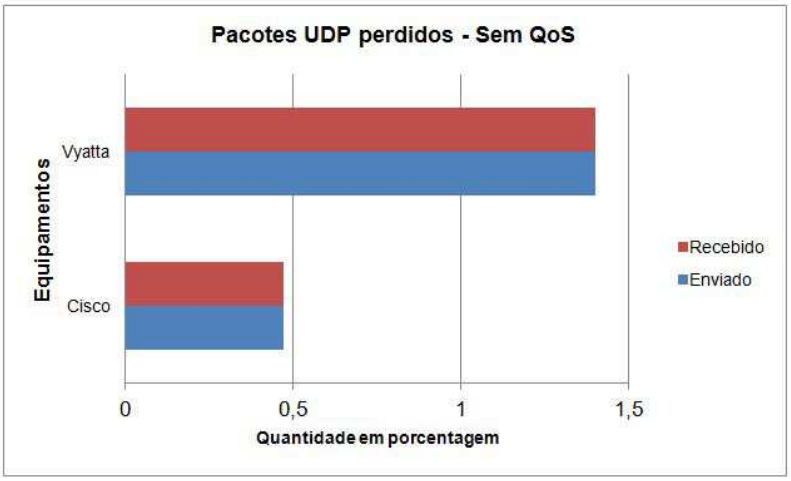

Figura 14. Comparando a perda de pacotes entre os roteadores - Sem QoS.

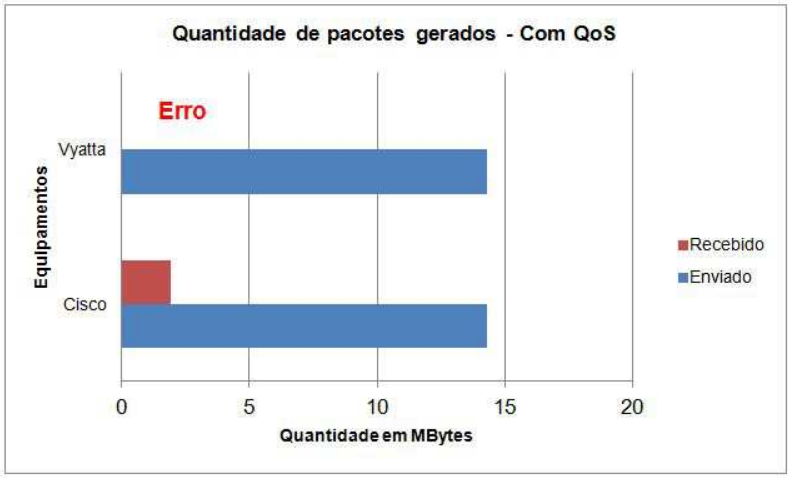

Figura 15. Comparando o envio de pacotes entre os roteadores - Com QoS.

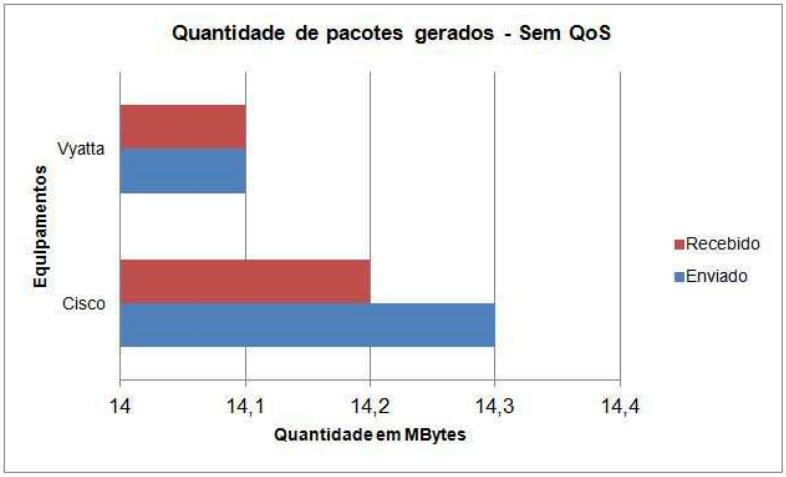

Figura 16. Comparando o envio de pacotes entre os roteadores - Sem QoS.

7.4. Medindo a qualidade na transmissão de pacotes através de uma VPN

Para testar a conexão VPN nos roteadores, foi necessário que um dos computadores esteja fora da rede em um ambiente similar ao utilizado normalmente em uma conexão VPN. Um notebook irá realizar a discagem para um IP real hospedado pela Embratel que irá encaminhar a solicitação aos roteadores testados neste trabalho, de acordo com a figura 5 do projeto de redes. A internet em que o notebook utilizará é uma link ADSL da Velox com uma velocidade contratada de 5 Mbps. Abaixo segue as configurações do notebook:

- Processador Intel Dual Core T6570 1,80 GHz
- Memória 4 GB DDR2 800 MHz
- Hard Drive SATA 2 80 GB 5200 RPM
- Rede Megabit Ethernet
- Sistema operacional Windows Seven

Os testes foram efetuados utilizando o protocolo TCP, onde a largura de banda e o tempo necessário para o envio dos dados, foram avaliados. Os relatórios do Iperf podem ser conferidos no anexo 10 no final deste artigo.

Resultados obtidos utilizando o Cisco como roteador:

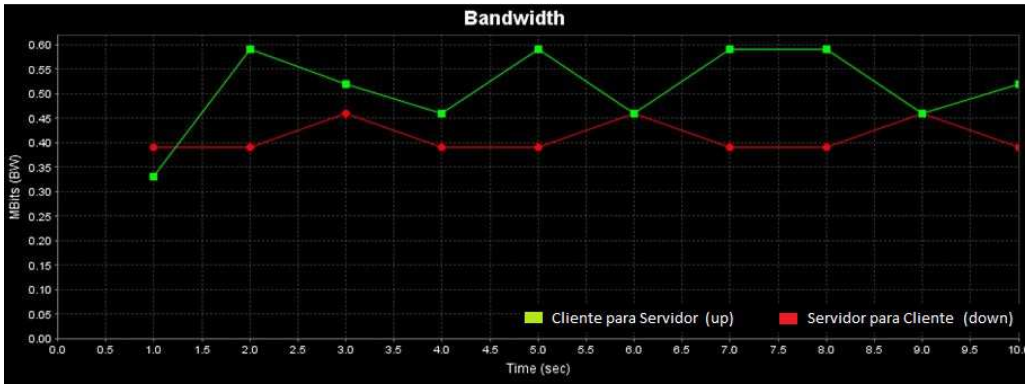

Figura 17. Gráfico do tráfego TCP gerado pelo cliente através de uma VPN - Cisco.

Resultados obtidos utilizando o Vyatta como roteador:

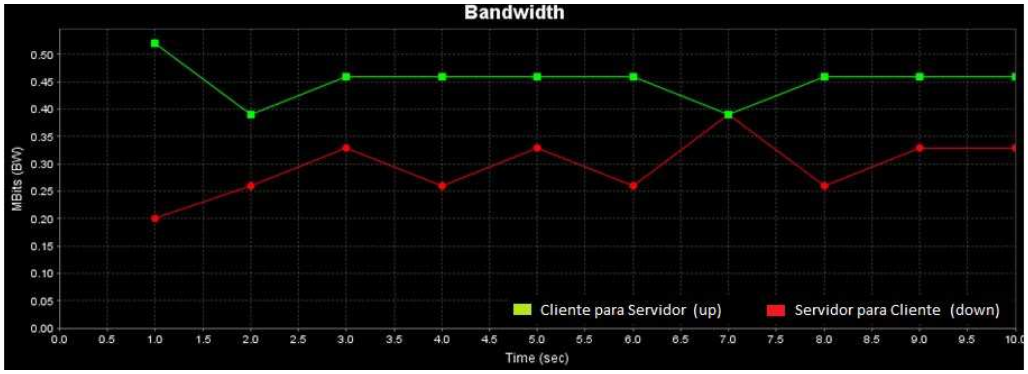

Figura 18. Gráfico do tráfego TCP gerado pelo cliente através de uma VPN - Vyatta.

A facilidade na implementação ao acesso VPN utilizando o protocolo PPTP em ambos os roteadores é louvável, estavam aptos a receberem conexões seguras com apenas algumas linhas de configuração, que podem ser conferidas no Anexo 1 e 2 deste trabalho. Ao verificarmos a qualidade da transmissão, o resultado foi semelhante em ambos os roteadores, na figura 17 do gráfico de tráfico TCP gerado pelo cliente utilizando o roteador da Cisco, vemos que a transmissão de dados terminou com uma velocidade aquém do esperado, porém o resultado foi praticamente igual no Vyatta, também abaixo das expectativas.

O resultado do Vyatta pode ser conferido na figura 18 do gráfico de tráfico TCP gerado pelo cliente utilizando o roteador Vyatta. O Vyatta levou ligeira vantagem ao desenvolver uma velocidade superior ao Cisco, na figura 19 do gráfico comparativo da velocidade de pacotes através de uma VPN, mostra claramente esse resultado. A velocidade de transmissão foi maior que a de recepção o que já era esperado já que o Link da Embratel possui somente 4 Mbps. Em relação a facilidade de acesso, em nenhum momento a conexão ficou lenta ou caiu,a autenticação nos roteadores foi rápida e transparente.

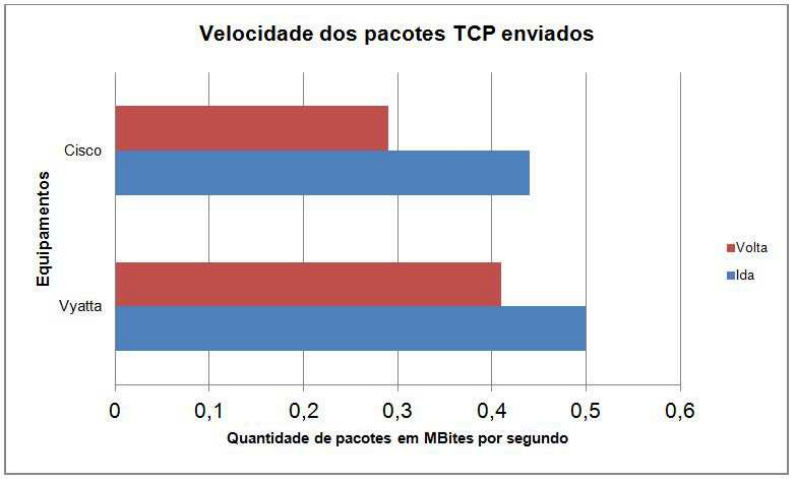

Figura 19. Comparando a velocidade de pacotes entre os roteadores através de uma VPN.

		Cisco	Vyatta	Vantagem do Cisco em relação ao Vyatta
Tráfego TCP em Mbytes	Upload	89,40	90,90	-1,50
	Download	101,00	73,90	27,10
Perda de pacotes UDP em %	Upload	0,069%	0,52%	0,45%
	Download	0,22%	0,15%	-0,07%
Jitter em MS	Upload	1,694	1,68	-0,01
	Download	0,982	2,103	1,12
Perda de Pacotes UDP em % - QoS Ativado	Upload	0,27%	ERRO	0,27%
	Download	84%	ERRO	84%
Perda de Pacotes UDP em % - QoS desativado	Upload	0,47%	1,40%	1%
	Download	0,47%	1,40%	1%
Tráfego UDP em Mbytes - QoS Ativado	Upload	14,3	14,3	0,00
	Download	2,32	0	2,32
Tráfgo UDP em Mbytes - QoS Desativado	Upload	14,3	14,1	0,20
	Download	14,2	14,1	0,10
Tráfego TCP em Mbytes pela VPN	Upload	0,44	0,5	-0,06
	Download	0,29	0,41	-0,12

Tabela 1. Comparativo final com os resultados adquiridos ao longo de todos os testes.

8. Conclusão

Este trabalho teve como objetivo apresentar um comparativo entre dois roteadores propondo ser referência na escolha de uma solução simples e barata que possa substituir equipamentos proprietários, atendendo as expectativas de uma organização universitária.

Visando atingir estes objetivos foi escolhida uma abordagem diferenciada, sendo um destes equipamentos virtualizados através de uma ferramenta de código livre, o XCP, além dos testes escolhidos serem voltados para situações que seriam extremamente utilizadas em ambiente acadêmicos, como conexões remotas e ligações utilizando tecnologias voz sobre ip.

Para comparar cada equipamento foram desenvolvidos alguns cenários com diferentes situações de transmissão de pacotes entre dois computadores, passando pelo roteador, como transmissão de dados através de uma VPN e filtragem de pacotes com QoS. Ao longo dos testes desenvolvidos houve uma preocupação constante em manter soluções livres como ferramentas, exemplo disso foi o Iperf como gerador de tráfego e o Elastix como provedor de VoIP, e o Ekiga como softfone.

Ambos os comparativos tiveram o finalidade de simular situações bem próximas das que encontramos em ambientes de produção, como um roteador é responsável por lidar com pacotes advindos de dentro e fora da rede é crucial que ele seja de qualidade para o bom desempenho geral de todos os seus integrantes. Existe uma resistência em virtualizar equipamentos, mas [Jones 2011] dentre vários outros autores, diz que será inevitável o uso desta tecnologia nos próximos anos.

Apesar do desempenho, de modo geral do roteador Vyatta ter sido inferior ao roteador Cisco, mesmo que por alguns décimos percentuais, a tabela 1 mostra que houve erro ao gerar os relatórios de perda de pacotes UDP com o QoS ativado, ainda sim, obteve resultados satisfatórios, seu principal problema foi com as regras de traffic shapping, com elevada perda de pacotes, o que resultaria em lentidão em toda a rede. Sem as regras de traffic shapping, a situação ainda não foi boa, a ligação VoIP obteve duas quedas durante os testes de ligações.

O Vyatta se mostrou mais estável durante a conexão VPN pelo protocolo PPTP alcançando uma velocidade de transmissão de pacotes TCP maiores que o do Cisco, mas como podemos ver na tabela 1 com uma diferença bem pequena. A configuração do serviço em ambos os roteadores é semelhante gerando uma curva de aprendizado baixa, além de que o Vyatta, em sua versão 6.2, apresenta uma interface Web, não muito completa, mas que pode ajudar em situações mais simples. O Cisco também possui acesso a algumas configurações utilizando de interfaces gráficas, porém necessitam de licenças adicionais. Um dos objetivos deste trabalho é evitar custo com softwares.

O roteador Cisco é conhecido no mercado, líder de vendas, e extremamente caro segundo [Santana 2011]. Superar um roteador com uma reputação tão sólida é um grande desafio, porém o sistema Vyatta que possui uma versão completamente livre se saiu bem nos testes realizados, ficando um pouco abaixo aos resultados que o Cisco apresentou, levando em consideração que o Cisco é pago e o Vyatta completamente gratuito, virtualizado e funcionando em um computador comum, a conclusão é que o Vyatta alcançou seu objetivo e pode sim, ser utilizado em uma rede de médio porte com até 400 computadores e substituindo, sem maiores problemas, soluções proprietárias.

9. Anexos

9.1. Anexo 1 - Configuração do Vyatta

```
interfaces {
        ethernet eth0 {
            address 192.168.20.1/24
            description RedeExterna
            duplex auto
            smp_affinity auto
            speed auto
        }
        ethernet eth1 {
            address 189.3.225.100/28
            description RedeWan
            duplex auto
            smp_affinity auto
            speed auto
            traffic-policy {
                out dscp
            }
        }
        ethernet eth2 {
            address 10.0.0.1/24
            duplex auto
            smp_affinity auto
            speed auto
        }
        loopback lo {
        }
    }
    service {
        dhcp-server {
            disabled false
            shared-network-name redeLocal {
                authoritative disable
                subnet 10.0.0.0/24 {
                    default-router 10.0.0.1
                    dns-server 8.8.8.8
                    lease 86400
                    server-identifier 10.0.0.1
                    start 10.0.0.11 {
                        stop 10.0.0.100
                    }
                }
            }
        }
        https {
```

```
    }
    nat {
        rule 1 {
            outbound-interface eth1
            type masquerade
        }
    }
    ssh {
        port 22
        protocol-version v2
    }
    webproxy {
        cache-size 1024
        default-port 3128
        listen-address 10.0.0.1 {
        }
    }
}
system {
    config-management {
        commit-revisions 20
    }
    console {
        device ttyS0 {
            speed 9600
        }
    }
    gateway-address 189.3.225.97
    host-name vyatta62
    login {
        user vyatta {
            authentication {
                encrypted-password $1$SBeidwYr$y0jCrLjF4CxRTkmiDroR
            }
            level admin
        }
    }
    name-server 8.8.8.8
    name-server 8.8.4.4
    ntp {
        server 0.vyatta.pool.ntp.org {
        }
        server 1.vyatta.pool.ntp.org {
        }
        server 2.vyatta.pool.ntp.org {
        }
```

```
    }
    package {
        auto-sync 1
        repository community {
            components main
            distribution stable
            password ""
            url http://packages.vyatta.com/vyatta
            username ""
        }
        repository lenny {
            components main
            distribution lenny
            password ""
            url http://http.us.debian.org/debian
            username ""
        }
    }
    syslog {
        global {
            facility all {
                level notice
            }
            facility protocols {
                level debug
            }
        }
    }
    time-zone GMT
}
traffic-policy {
    shaper dscp {
        bandwidth 6mbit
        class 10 {
            bandwidth 60%
            burst 15k
            description "DSCP - 46 usado no RTP e RTCP"
            match dscp-46 {
                ip {
                    dscp 46
                }
            }
            queue-type fair-queue
        }
        default {
            bandwidth 5%
```

```
                queue-type fair-queue
            }
            description dscpbase
        }
    }
    vpn {
        pptp {
            remote-access {
                authentication {
                    local-users {
                        username vyatta {
                            password vyatta
                        }
                    }
                    mode local
                }
                client-ip-pool {
                    start 10.0.0.100
                    stop 10.0.0.120
                }
                dns-servers {
                    server-1 8.8.8.8
                }
                outside-address 189.3.225.100
            }
        }
```

9.2. Anexo 2 - Configuração do Cisco

```
CiscoRouter#show running-config
Building configuration...

Current configuration : 1782 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname CiscoRouter
!
!
!
!
!
ip dhcp pool RedeLocal
```

```
 network 10.0.0.0 255.255.255.0
 default-router 10.0.0.1
 dns-server 8.8.8.8
!
!
!
vpdn enable
vpdn-group 1
! Default L2TP VPDN group
 accept-dialin
  protocol pptp
  virtual-template 1
username cisco password 7 0822455D0A16
!
!
!
!
!
no ip domain-lookup
!
!
!
class-map match-any VOIP-CONTROL-CLASS
  match access-group name VOIP-CONTROL-ACL
class-map match-any VOIP-RTP-CLASS
  match access-group name VOIP-RTP-ACL
!
policy-map VOIP
 class VOIP-RTP-CLASS
priority percent 60
 class VOIP-CONTROL-CLASS
 bandwidth percent 5
 class class-default
 fair-queue
!
!
!
bba-group pptp global
  virtual-template 1
!
interface virtual-template 1
 ip unnumbered FastEthernet0/0
 peer default ip address pool PPTP-Pool
 no keepalive
 ppp encrypt mppe auto
 ppp authentication pap chap ms-chap
```

```
!
interface FastEthernet0/0
 description InterfaceLocal1
 ip address 10.0.0.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
!
interface FastEthernet0/1
 description InterfaceLocal2
 ip address 192.168.20.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
!
interface Ethernet0/0/0
 description InterfaceWAN
 ip address 189.3.225.100 255.255.255.240
 ip nat outside
 service-policy output VOIP
 duplex auto
 speed auto
!
interface Vlan1
 no ip address
 shutdown
!
ip local pool PPTP-Pool 10.0.0.100 10.0.0.150
ip nat inside source list 1 interface Ethernet0/0/0 overload
ip classless
ip route 10.0.0.0 255.0.0.0 10.0.0.1
!
!
access-list 1 permit any
ip access-list extended VOIP-CONTROL-ACL
 permit tcp any any eq 5060
 permit tcp any eq 5060 any
 permit tcp any any eq 6970
 permit tcp any eq 6970 any
ip access-list extended VOIP-RTP-ACL
 permit udp any any eq 5060
 permit udp any eq 5060 any
 permit udp any any range 16384 32767
 permit ip any any dscp ef
!
!
```

```
!
!
!
line con 0
line vty 0 4
 login
!
!
!
end
```

9.3. Anexo 3 - Instalação e Configuração de um ambiente virtual XCP

Este servidor possui suporte a virtualização por hardware, necessário para hospedar sistemas Windows, segundo [System 2011a] para hospedar apenas sistemas Linux, caso do Vyatta OS, não seria necessária a presença destas tecnologias já que utilizam a paravirtualização, recurso já comentado anteriormente neste trabalho.

A instalação é facilitada e bem ágil, podendo ser feita por um CD comum ou um Live CD[21] com telas de configuração bem simples, a figura 20 mostra um dos passos da instalação do XCP.

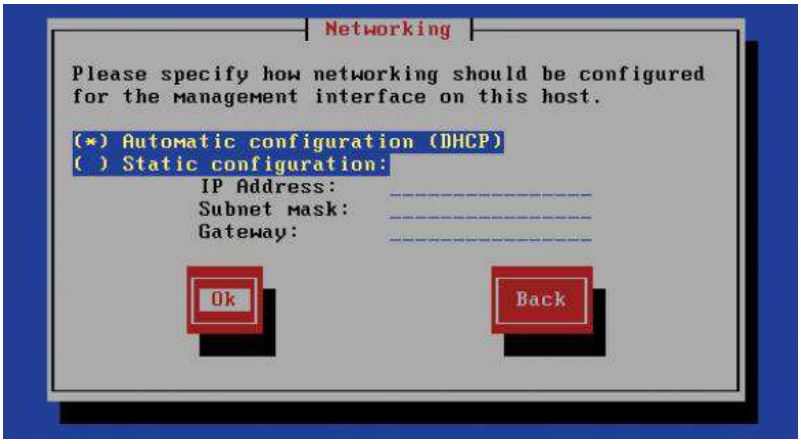

Figura 20. Tela de instalação do XCP - Configurando o endereço IP

O instalador do XCP possui perguntas típicas de qualquer instalação de sistemas baseados em Linux, sendo a configuração do kernel[22] e funcionamento interno do sistema totalmente automatizado, ao contrário de outros sistemas hospedeiros, onde edição de arquivos textos e alterações no kernel são freqüentemente necessárias, como mostra [Matthews et al. 2008] na instalação do hipervisor Xen. Após o término da instalação o XCP está apto a receber máquinas virtuais sem que sejam necessárias configurações extras estando pronto para criação e manutenção das VM's.

[21] Live CD é um CD que contém um sistema operacional que não necessita ser instalada no hard drive do computador uma vez que é executado diretamente a partir do CD e da memória RAM.

[22] Kernel é o componente principal do sistema operacional da maioria dos computadores, ele serve de ponte entre aplicativos e o processamento feito a nível de hardware

9.3.1. A configuração de máquinas virtuais em um ambiente XCP

Para criação de máquinas virtuais o XCP permite que sejam feitas diretamente pelo console mostrado na figura 21 ou utilizando qualquer outro computador através de uma interface WEB, permitindo que, através de uma estação de trabalho com qualquer sistema operacional possa fazer o monitoramento das VM's contidas no domínio.

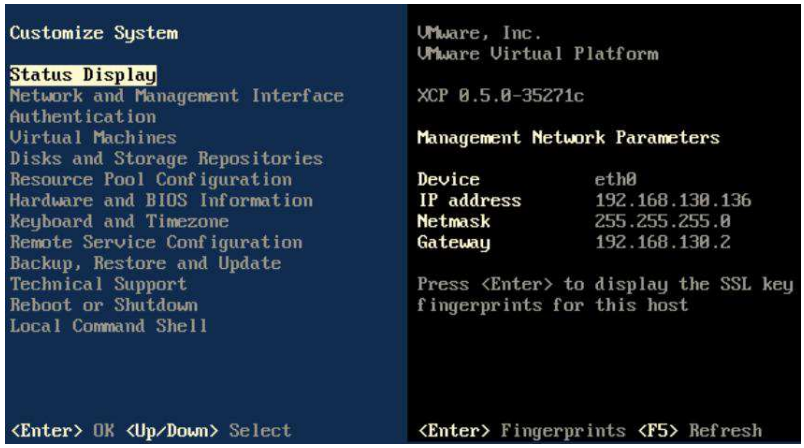

Figura 21. Console principal de configuralção do XCP

A administração do XCP pela interface Web, possui diversos sistemas compatíveis. Segundo [Matthews et al. 2008] a principal finalidade destas ferramentas é fornecer uma interface GUI[23] para simplificar o processo da criação e do gerenciamento de máquinas virtuais. A maioria das ferramentas também fornece consoles para controlar e acessar as VM's hospedeiras em execução, fornecendo uma interface de controle.

Existem diversas opções no mercado, inclusive pagas, que são compatíveis com a versão do XCP utilizado neste trabalho, [Jones 2011] comenta que o Open Xen Center que seria a versão de código aberto de uma ferramenta proprietária da Citrix, com o nome de Xen Center, é uma ferramenta que possui facilidade e agilidade na sua usualidade.

Open Xen Center, possui codificação basicamente em Python e é completamente livre, segundo [Matthews et al. 2008] é um excelente aplicativo. O Open Xen Center - OXC pode ser baixado no site do projeto[24] e sua instalação deve ser feita no servidor que contém o XCP. Na figura 22 visualizamos uma das telas de gerenciamento do OXC, nesta tela podemos observar o console de um Debian, distribuição Linux, completamente virtualizada.

9.4. Anexo 4 - Instalando e configurando o Vyatta OS

Durante a instalação do Vyatta, o OXC irá fazer uma série de solicitações, onde serão necessário selecionar quais as configurações que irão ser utilizadas durante a execução do sistema, [Golden and Scheffy 2008] dizem que é importante respeitar o limite de memória e de espaço em disco, além das interfaces de rede que seu servidor dispõe. Segundo o

[23]GUI: Interface gráfica do usuário,em português, é um tipo de interface do usuário que permite a interação com dispositivos digitais através de elementos gráficos como ícones e outros indicadores visuais, em contraste a interface de linha de comando.

[24]http://sourceforge.net/projects/openxencenter/

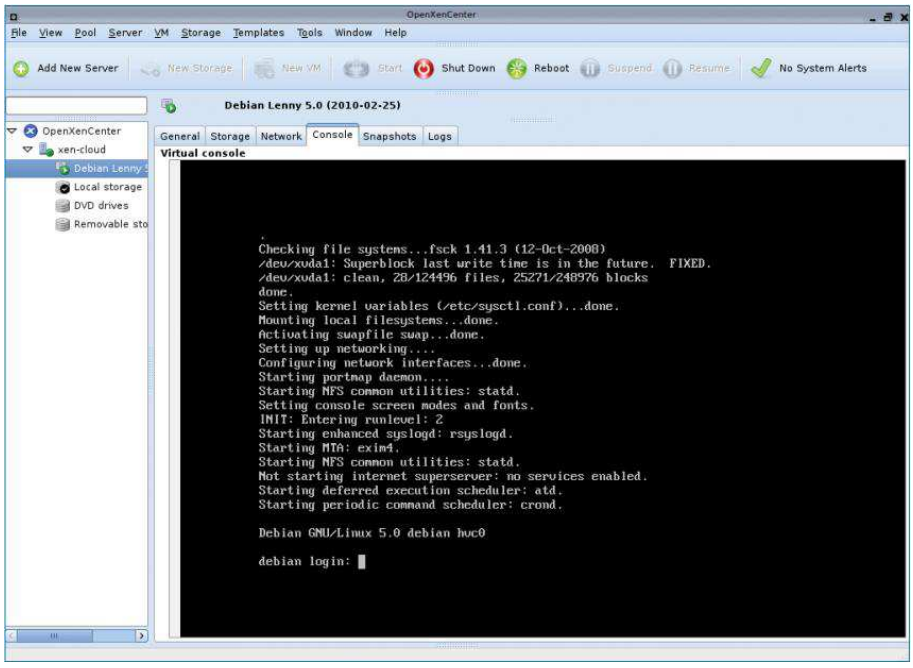

Figura 22. Console de uma máquina virtual pelo OXC

[System 2011a] o Vyatta necessita de apenas 2 gb de espaço e 1 gb de memória para a instalação em ambiente de produção. [Paquet and Teare 2002] diz que, por exemplo, um Roteador CISCO 1841 que possua 256 MB de memória consegue gerir uma rede de tamanho médio sem maiores problemas, porém upgrades de memória em dispositivos CISCO, caso seja necessário, são complexos e custosos.

O Vyatta por utilizar um simples computador virtualizado que pode ser definido inicialmente com o mínimo de memória e na medida que for sendo necessário a sua expansão, o OXC poderá efetuar este ajuste com maior facilidade e transparência, respeitando o limite de um servidor x86, consequentemente maior que um dispositivo CISCO.

O Vyatta irá ser configurado, através do OXC para que utilize o mínimo de recursos possíveis. Na figura 23 é possível ver umas das telas de configuração do OXC onde é selecionado o tanto de memória e o número de CPUs que o Vyatta irá utilizar em sua operação.

Um passo importante na configuração de uma máquina virtual é a definição das placas de redes virtuais, pela qual o Roteador+Firewall irá receber e enviar os pacotes de toda a rede. Segundo [System 2011b] o OXC suporta no máximo seis interfaces virtuais associadas às interfaces do servidor real. As definições de IP e máscara de rede deverão ser feitas individualmente pelo sistema operacional de cada VM.

Após o hospedeiro reconhecer seu novo guest, basta utilizar o OXC para que a instalação do Vyatta, utilizando uma imagem ISO, possa ser efetuada. Através do console no OXC, podemos definir os primeiros parâmetros de utilização do Vyatta, seu login e senha, definição do IP para que o Vyatta possa ser reconhecido na rede e iniciar o serviço HTTP, que possibilita a configuração do roteador por uma interface GUI. A figura 24 mostra como definir um endereço IP em uma interface no Vyatta.

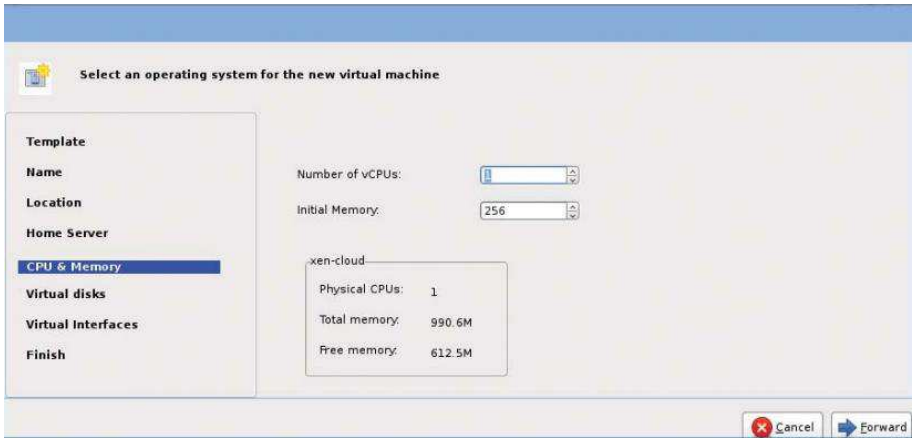

Figura 23. Instalação do Vyatta utilizando o OXC

Figura 24. Definindo um endereço IP para o Vyatta OS

A configuração para que o Vyatta possa ser acessar através de uma interface Web é:

```
vyatta@vyatta:~$ set service https
vyatta@vyatta:~$ commit
```

Ativando do serviço, basta digitar o endereço IP definido anteriormente para que a configuração seja feita através de uma interface simples e intuitiva. A figura 25 mostra a interface Web.

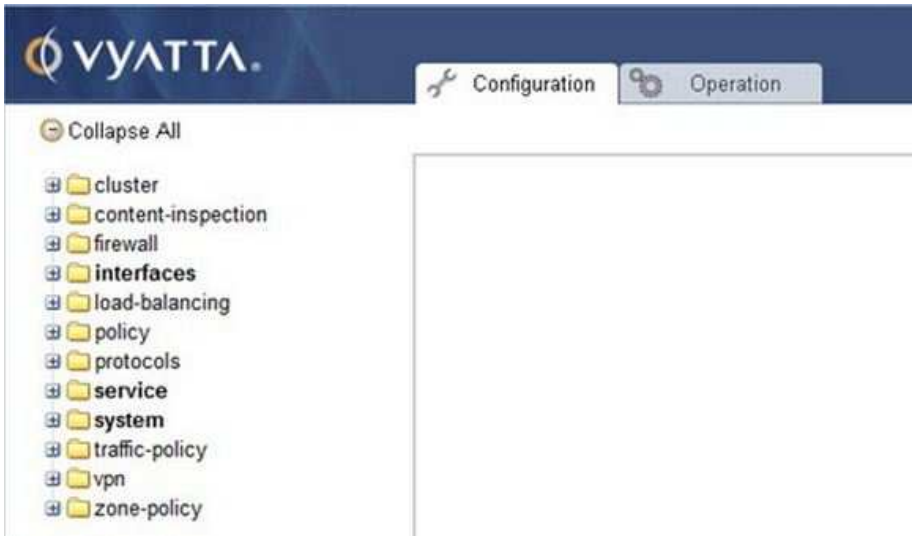

Figura 25. Interface Web do Vyatta OS

9.5. Anexo 5 - A ferramenta Iperf

A figura 26 mostra a tela de uso de um resultado obtido com o Iperf.

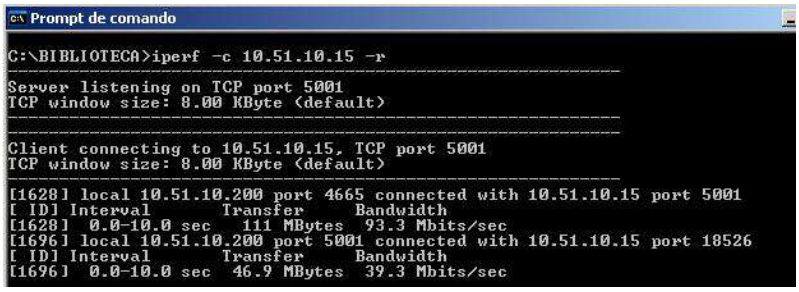

Figura 26. Tela de uso do software Iperf

9.6. Anexo 6 - A ferramenta Elastix

A figura 27, mostra os SIPs criados para o uso no teste de VoIP.

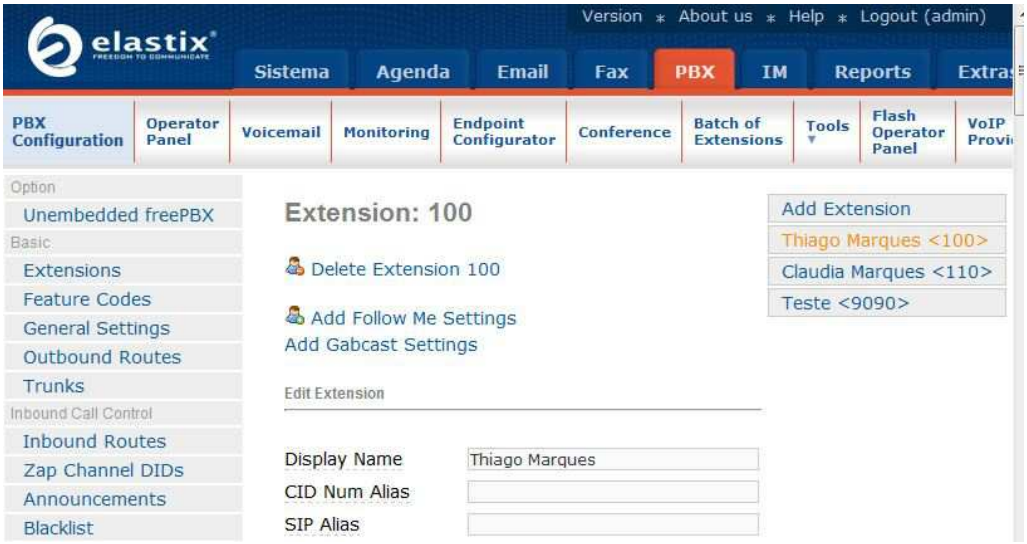

Figura 27. PBX virtual Elastix

9.7. Anexo 7 - Medindo o tráfego entre os computadores da Rede Local através do protocolo TCP.

```
bin/iperf.exe -c 192.168.20.5  -P 1 -i 1 -p 5001 -f m -t 10 -d -L 5001
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 192.168.20.5, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[156] local 10.0.0.14 port 32851 connected with 192.168.20.5 port 5001
[180] local 10.0.0.14 port 5001 connected with 192.168.20.5 port 4206
[ ID] Interval       Transfer     Bandwidth
[156]  0.0- 1.0 sec  9.32 MBytes  78.2 Mbits/sec
[180]  0.0- 1.0 sec  7.44 MBytes  62.4 Mbits/sec
[156]  1.0- 2.0 sec  9.46 MBytes  79.4 Mbits/sec
                                  ...
[156]  9.0-10.0 sec  7.10 MBytes  59.6 Mbits/sec
[180]  9.0-10.0 sec  8.17 MBytes  68.6 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[156]  0.0-10.0 sec  90.9 MBytes  76.2 Mbits/sec
[180]  0.0-10.0 sec  73.9 MBytes  61.9 Mbits/sec
Done.
```

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1856] local 193.168.20.5 port 5001 connected with 10.0.0.14 port 32851
------------------------------------------------------------
Client connecting to 10.0.0.14, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1800] local 192.168.20.5 port 4206 connected with 10.0.0.14 port 5001
[ ID] Interval       Transfer     Bandwidth
[1856]  0.0- 1.0 sec  9.33 MBytes  78.2 Mbits/sec
[1800]  0.0- 1.0 sec  7.53 MBytes  63.2 Mbits/sec
[1856]  1.0- 2.0 sec  9.46 MBytes  79.3 Mbits/sec
                                  ...
[1856]  9.0-10.0 sec  7.11 MBytes  59.6 Mbits/sec
[1800]  9.0-10.0 sec  8.21 MBytes  68.9 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[1856]  0.0-10.0 sec  90.9 MBytes  76.1 Mbits/sec
[1800]  0.0-10.0 sec  73.9 MBytes  61.9 Mbits/sec
```

Figura 28. Relatório Cliente e Servidor do tráfego TCP gerado pelo Iperf - Roteador Vyatta

```
bin/iperf.exe -c 192.168.20.5  -P 1 -i 1 -p 5001 -f m -t 10 -d -L 5001 -T 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 192.168.20.5, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1844] local 10.0.0.15 port 1046 connected with 192.168.20.5 port 5001
[1820] local 10.0.0.15 port 5001 connected with 192.168.20.5 port 1484
[ ID] Interval       Transfer     Bandwidth
[1844]  0.0- 1.0 sec  7.33 MBytes  61.5 Mbits/sec
[1820]  0.0- 1.0 sec  9.70 MBytes  81.4 Mbits/sec
[1820]  1.0- 2.0 sec  10.2 MBytes  85.8 Mbits/sec
[1844]  1.0- 2.0 sec  8.44 MBytes  70.8 Mbits/sec
                                  ...
[1820]  8.0- 9.0 sec  10.3 MBytes  86.1 Mbits/sec
[1844]  9.0-10.0 sec  8.92 MBytes  74.8 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[1844]  0.0-10.0 sec  89.4 MBytes  74.9 Mbits/sec
[1820]  0.0-10.0 sec   101 MBytes  84.4 Mbits/sec

Done.
```

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[232] local 192.168.20.5 port 5001 connected with 10.0.0.15 port 1046
------------------------------------------------------------
Client connecting to 192.168.20.5, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[284] local 192.168.20.5 port 1484 connected with 10.0.0.15 port 5001
[ ID] Interval       Transfer     Bandwidth
[232]  0.0- 1.0 sec  7.23 MBytes  60.7 Mbits/sec
[284]  0.0- 1.0 sec  9.63 MBytes  80.8 Mbits/sec
[232]  1.0- 2.0 sec  8.43 MBytes  70.7 Mbits/sec
[284]  1.0- 2.0 sec  10.2 MBytes  85.7 Mbits/sec
                                  ...
[232]  9.0-10.0 sec  8.89 MBytes  74.6 Mbits/sec
[284]  9.0-10.0 sec  10.2 MBytes  85.4 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[284]  0.0-10.0 sec   101 MBytes  84.4 Mbits/sec
[232]  0.0-10.0 sec  89.4 MBytes  74.9 Mbits/sec
```

Figura 29. Relatório Cliente e Servidor do tráfego TCP gerado pelo Iperf - Roteador Cisco

9.8. Anexo 8 - Medindo a perda de pacotes e o Jitter através do protocolo UDP.

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 12.0M -t 10 -d -L 5001 -T 1
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[160] local 10.0.0.14 port 49593 connected with 192.168.20.5 port 5001
[148] local 10.0.0.14 port 5001 connected with 192.168.20.5 port 4238
[ ID] Interval       Transfer     Bandwidth
[160]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[148]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec  0.518 ms    0/ 1020 (0%)
[160]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
...
[160]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[148]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec  0.529 ms    3/ 1021 (0.29%)
[160] Server Report:
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[160]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec  1.694 ms    7/10205 (0.069%)
[160] Sent 10205 datagrams
[148]  9.0-10.0 sec  1 datagrams received out-of-order
[148]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  0.982 ms   22/10176 (0.22%)
[148]  0.0-10.0 sec  2 datagrams received out-of-order
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1928] local 192.168.20.5 port 5001 connected with 10.0.0.14 port 49593
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1836] local 192.168.20.5 port 4238 connected with 10.0.0.14 port 5001
[ ID] Interval       Transfer     Bandwidth
[1836]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1928]  0.0- 1.0 sec  1.44 MBytes  12.1 Mbits/sec  1.485 ms 1330794496/ 1027
(1.3e+008%)
[1928]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec  1.448 ms    0/ 1020 (0%)
[1836]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
[1928]  2.0- 3.0 sec  1.43 MBytes  12.0 Mbits/sec  1.515 ms    0/ 1020 (0%)
...
[1928]  8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec  1.518 ms    0/ 1021 (0%)
[1928]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec  1.695 ms    7/10205 (0.069%)
[1836]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[1836]  0.0-10.0 sec  14.3 MBytes  11.9 Mbits/sec
[1836] Server Report:
[1836]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  0.982 ms   22/10176 (0.22%)
[1836]  0.0-10.0 sec  2 datagrams received out-of-order
```

Figura 30. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Roteador Cisco

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 12.0M -t 10 -d -L 5001 -T 1
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1840] local 10.0.0.14 port 1055 connected with 192.168.20.5 port 5001
[1912] local 10.0.0.14 port 5001 connected with 192.168.20.5 port 53225
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  0.0- 1.0 sec  1.44 MBytes  12.1 Mbits/sec  1.485 ms    1/ 1031 (0.097%)
[1840]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
...
[1840]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[1840]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[1912]  9.0-10.0 sec  1.42 MBytes  11.9 Mbits/sec  2.002 ms    0/ 1010 (0%)
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[1912]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  2.103 ms   15/10172 (0.15%)
[1840] Server Report:
[1840]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  1.680 ms   53/10206 (0.52%)
[1840]  0.0-10.0 sec  1 datagrams received out-of-order
[1840] Sent 10206 datagrams
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[132] local 192.168.20.5 port 5001 connected with 10.0.0.14 port 1055
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[252] local 192.168.20.5 port 53225 connected with 10.0.0.14 port 5001
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[132]  0.0- 1.0 sec  1.42 MBytes  11.9 Mbits/sec  0.523 ms    0/ 1013 (0%)
[252]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[132]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec  0.481 ms    0/ 1019 (0%)
...
[252]  8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec
[132]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec  0.495 ms    0/ 1022 (0%)
[252]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[252]  0.0-10.0 sec  14.3 MBytes  11.9 Mbits/sec
[132]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  1.681 ms   53/10206 (0.52%)
[132]  0.0-10.0 sec  1 datagrams received out-of-order
[252] Server Report:
[252]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  2.103 ms   15/10172 (0.15%)
```

Figura 31. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Roteador Vyatta

9.9. Anexo 9 - Medindo o controle de banda através do protocolo UDP.

Resultados obtidos utilizando o Cisco como roteador - Com QoS

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 12.0M -t 10 -T 1
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1912] local 10.0.0.15 port 1106 connected with 192.168.20.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  2.0- 3.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  3.0- 4.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  4.0- 5.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  5.0- 6.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  6.0- 7.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  7.0- 8.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[1912] Server Report:
[1912]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec  0.559 ms   125/10206 (0.27%)
[1912] Sent 10206 datagrams
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[132] local 192.168.20.5 port 5001 connected with 10.0.0.15 port 19690
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[132]  0.0- 1.0 sec  0.24 MBytes  2.01 Mbits/sec  8.519 ms 1937331686/  913
(2.1e+008%)
[132]  1.0- 2.0 sec  0.23 MBytes  1.93 Mbits/sec  9.537 ms  872/ 1036 (84%)
[132]  2.0- 3.0 sec  0.23 MBytes  1.92 Mbits/sec  10.831 ms  864/ 1027 (84%)
[132]  3.0- 4.0 sec  0.23 MBytes  1.92 Mbits/sec  9.259 ms  858/ 1021 (84%)
[132]  4.0- 5.0 sec  0.23 MBytes  1.92 Mbits/sec  9.446 ms  860/ 1023 (84%)
[132]  5.0- 6.0 sec  0.23 MBytes  1.94 Mbits/sec  9.082 ms  843/ 1008 (84%)
[132]  6.0- 7.0 sec  0.23 MBytes  1.93 Mbits/sec  9.150 ms  870/ 1034 (84%)
[132]  7.0- 8.0 sec  0.23 MBytes  1.93 Mbits/sec  8.528 ms  835/  999 (84%)
[132]  8.0- 9.0 sec  0.23 MBytes  1.92 Mbits/sec  9.498 ms  869/ 1032 (84%)
[132]  9.0-10.0 sec  0.23 MBytes  1.93 Mbits/sec  9.167 ms  858/ 1022 (84%)
[132]  0.0-10.1 sec  2.32 MBytes  1.93 Mbits/sec  9.413 ms 8548/10206 (84%)
```

Figura 32. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Com QoS Cisco

Resultados obtidos utilizando o Cisco como roteador - Sem QoS

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 2048.0K -t 10 -T 1
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1912] local 10.0.0.15 port 3861 connected with 192.168.20.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  2.0- 3.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  3.0- 4.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  4.0- 5.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  5.0- 6.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  6.0- 7.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  7.0- 8.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[1912] Server Report:
[1912]  0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec  0.559 ms   48/10206 (0.47%)
[1912] Sent 10206 datagrams
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[132] local 192.168.20.5 port 5001 connected with 10.0.0.15 port 3861
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[132]  0.0- 1.0 sec  1.41 MBytes  11.8 Mbits/sec  0.592 ms 1937330944/ 1007
(1.9e+008%)
[132]  1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec  0.611 ms    0/ 1020 (0%)
[132]  2.0- 3.0 sec  1.42 MBytes  11.9 Mbits/sec  0.517 ms    7/ 1021 (0.69%)
[132]  3.0- 4.0 sec  1.42 MBytes  11.9 Mbits/sec  0.594 ms    5/ 1020 (0.49%)
[132]  4.0- 5.0 sec  1.43 MBytes  12.0 Mbits/sec  0.328 ms    3/ 1021 (0.29%)
[132]  5.0- 6.0 sec  1.42 MBytes  11.9 Mbits/sec  0.604 ms    8/ 1020 (0.78%)
[132]  6.0- 7.0 sec  1.42 MBytes  11.9 Mbits/sec  0.556 ms    4/ 1020 (0.39%)
[132]  7.0- 8.0 sec  1.42 MBytes  11.9 Mbits/sec  0.581 ms    7/ 1022 (0.68%)
[132]  8.0- 9.0 sec  1.42 MBytes  11.9 Mbits/sec  0.568 ms    5/ 1020 (0.49%)
[132]  9.0-10.0 sec  1.42 MBytes  11.9 Mbits/sec  0.633 ms    9/ 1022 (0.88%)
[132]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec  0.809 ms   48/10206 (0.47%)
```

Figura 33. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Sem QoS Cisco

Resultados obtidos utilizando o Vyatta como roteador - Com QoS

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 12.0M -t 10 -T 1
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1912] local 10.0.0.15 port 1064 connected with 192.168.20.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912] 0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 2.0- 3.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 3.0- 4.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 4.0- 5.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 5.0- 6.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 6.0- 7.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 7.0- 8.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[1912] WARNING: did not receive ack of last datagram after 10 tries.
[1912] Sent 10206 datagrams
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[132] local 192.168.20.5 port 5001 connected with 10.0.0.15 port 1100
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[132] 0.0- 1.0 sec  0.04 MBytes  0.29 Mbits/sec  31.115 ms 1937330944/   25
(7.7e+009%)
[132] 1.0- 2.0 sec  0.04 MBytes  0.29 Mbits/sec  37.344 ms    0/   25 (0%)
[132] 2.0- 3.0 sec  0.04 MBytes  0.29 Mbits/sec  39.259 ms    0/   25 (0%)
[132] 3.0- 4.0 sec  0.04 MBytes  0.29 Mbits/sec  39.111 ms    0/   25 (0%)
[132] 4.0- 5.0 sec  0.04 MBytes  0.29 Mbits/sec  39.620 ms    0/   25 (0%)
[132] 5.0- 6.0 sec  0.03 MBytes  0.28 Mbits/sec  17.504 ms  712/  736 (97%)
                                ...
[132] 10.0-11.0 sec  0.03 MBytes  0.28 Mbits/sec  7.397 ms  965/  989 (98%)
[132] 14.0-15.0 sec  0.04 MBytes  0.29 Mbits/sec  7.664 ms 1001/ 1026 (98%)
[132] 0.0-15.2 sec  0.53 MBytes  0.29 Mbits/sec  9.593 ms 9829/10206 (96%)
[132] WARNING: ack of last datagram failed after 10 tries.
read failed: Connection reset by peer
read failed: Connection reset by peer
read failed: Connection reset by peer
read failed: Connection reset by peer
read failed: Connection reset by peer
read failed: Connection reset by peer
recvfrom failed: Connection reset by peer
```

Figura 34. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Com QoS Vyatta

Resultados obtidos utilizando o Vyatta como roteador - Sem QoS

```
bin/iperf.exe -c 192.168.20.5 -u -P 1 -i 1 -p 5001 -f m -b 12.0M -t 10 -T 1
------------------------------------------------------------
Client connecting to 192.168.20.5, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1912] local 10.0.0.15 port 1069 connected with 192.168.20.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912] 0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 1.0- 2.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 2.0- 3.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 3.0- 4.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 4.0- 5.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 5.0- 6.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 6.0- 7.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 7.0- 8.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec
[1912] 0.0-10.0 sec  14.3 MBytes  12.0 Mbits/sec
[1912] Server Report:
[1912] 0.0-10.0 sec  14.1 MBytes  11.8 Mbits/sec  0.395 ms  138/10206 (1.4%)
[1912] Sent 10206 datagrams
Done.
```

```
bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[132] local 192.168.20.5 port 5001 connected with 10.0.0.15 port 1069
[ ID] Interval       Transfer     Bandwidth       Jitter   Lost/Total Datagrams
[132] 0.0- 1.0 sec  1.43 MBytes  12.0 Mbits/sec  0.427 ms 1937330944/ 1020
(1.9e+008%)
[132] 1.0- 2.0 sec  1.42 MBytes  11.9 Mbits/sec  0.395 ms    4/ 1020 (0.39%)
[132] 2.0- 3.0 sec  1.42 MBytes  11.9 Mbits/sec  0.305 ms    8/ 1021 (0.78%)
[132] 3.0- 4.0 sec  1.41 MBytes  11.9 Mbits/sec  0.421 ms   11/ 1020 (1.1%)
[132] 4.0- 5.0 sec  1.41 MBytes  11.8 Mbits/sec  0.356 ms   17/ 1021 (1.7%)
[132] 5.0- 6.0 sec  1.43 MBytes  12.0 Mbits/sec  0.417 ms    0/ 1020 (0%)
[132] 6.0- 7.0 sec  1.43 MBytes  12.0 Mbits/sec  0.384 ms    2/ 1020 (0.2%)
[132] 7.0- 8.0 sec  1.30 MBytes  10.9 Mbits/sec  0.380 ms   91/ 1021 (8.9%)
[132] 8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec  0.283 ms    3/ 1020 (0.29%)
[132] 9.0-10.0 sec  1.43 MBytes  12.0 Mbits/sec  0.275 ms    2/ 1021 (0.2%)
[132] 0.0-10.0 sec  14.1 MBytes  11.8 Mbits/sec  0.395 ms  138/10206 (1.4%)
Done.
```

Figura 35. Relatório Cliente e Servidor do tráfego UDP gerado pelo Iperf - Sem QoS Vyatta

9.10. Anexo 10 - Medindo a qualidade na transmissão de pacotes através de uma VPN

```
bin/iperf.exe -c 10.0.0.100 -P 1 -i 1 -p 5001 -f m -t 10 -d -L 5001 -T 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 10.0.0.100, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1840] local 10.0.0.17 port 2836 connected with 10.0.0.100 port 5001
[1816] local 10.0.0.17 port 5001 connected with 10.0.0.100 port 55554
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0- 1.0 sec  0.04 MBytes  0.33 Mbits/sec
[1816]  0.0- 1.0 sec  0.05 MBytes  0.39 Mbits/sec
[1816]  1.0- 2.0 sec  0.05 MBytes  0.39 Mbits/sec
                                   ...
[1816]  8.0- 9.0 sec  0.05 MBytes  0.46 Mbits/sec
[1816]  9.0-10.0 sec  0.05 MBytes  0.39 Mbits/sec
[1840]  9.0-10.0 sec  0.06 MBytes  0.52 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[1840]  0.0-10.4 sec  0.62 MBytes  0.50 Mbits/sec
[1816]  0.0-10.3 sec  0.51 MBytes  0.41 Mbits/sec
Done.
```

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[240] local 10.0.0.100 port 5001 connected with 10.0.0.17 port 2836
------------------------------------------------------------
Client connecting to 10.0.0.17, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[292] local 10.0.0.100 port 55554 connected with 10.0.0.17 port 5001
[ ID] Interval       Transfer     Bandwidth
[292]  0.0- 1.0 sec  0.05 MBytes  0.39 Mbits/sec
[240]  0.0- 1.0 sec  0.04 MBytes  0.34 Mbits/sec
[240]  1.0- 2.0 sec  0.06 MBytes  0.52 Mbits/sec
                                   ...
[240]  8.0- 9.0 sec  0.07 MBytes  0.59 Mbits/sec
[240]  9.0-10.0 sec  0.06 MBytes  0.52 Mbits/sec
[240]  0.0-10.2 sec  0.62 MBytes  0.51 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[292]  9.0-10.0 sec  0.05 MBytes  0.46 Mbits/sec
[292]  0.0-10.3 sec  0.51 MBytes  0.41 Mbits/sec
```

Figura 36. Relatório Cliente e Servidor do tráfego TCP gerado pelo Iperf através de uma VPN - Cisco

```
bin/iperf.exe -c 10.0.0.17 -P 1 -i 1 -p 5001 -f m -t 10 -d -L 5001 -T 1
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
------------------------------------------------------------
Client connecting to 10.0.0.17, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[172] local 10.0.0.100 port 63400 connected with 10.0.0.17 port 5001
[188] local 10.0.0.100 port 5001 connected with 10.0.0.17 port 1737
[ ID] Interval       Transfer     Bandwidth
[172]  0.0- 1.0 sec  0.06 MBytes  0.52 Mbits/sec
[188]  0.0- 1.0 sec  0.02 MBytes  0.20 Mbits/sec
[172]  1.0- 2.0 sec  0.05 MBytes  0.39 Mbits/sec
                                   ...
[188]  8.0- 9.0 sec  0.04 MBytes  0.33 Mbits/sec
[172]  9.0-10.0 sec  0.05 MBytes  0.46 Mbits/sec
[188]  9.0-10.0 sec  0.04 MBytes  0.33 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[188]  0.0-10.3 sec  0.35 MBytes  0.29 Mbits/sec
[172]  0.0-10.4 sec  0.55 MBytes  0.44 Mbits/sec
Done.
```

```
bin/iperf.exe -s -P 0 -i 1 -p 5001 -f m
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1856] local 10.0.0.17 port 5001 connected with 10.0.0.100 port 63400
------------------------------------------------------------
Client connecting to 10.0.0.100, TCP port 5001
TCP window size: 0.01 MByte (default)
------------------------------------------------------------
[1800] local 10.0.0.17 port 1737 connected with 10.0.0.100 port 5001
[ ID] Interval       Transfer     Bandwidth
[1856]  0.0- 1.0 sec  0.05 MBytes  0.39 Mbits/sec
[1800]  0.0- 1.0 sec  0.02 MBytes  0.20 Mbits/sec
[1856]  1.0- 2.0 sec  0.05 MBytes  0.46 Mbits/sec
                                   ...
[1800]  8.0- 9.0 sec  0.05 MBytes  0.39 Mbits/sec
[1856]  9.0-10.0 sec  0.05 MBytes  0.46 Mbits/sec
[1800]  9.0-10.0 sec  0.03 MBytes  0.26 Mbits/sec
[ ID] Interval       Transfer     Bandwidth
[1856]  0.0-10.4 sec  0.55 MBytes  0.44 Mbits/sec
[1800]  0.0-10.4 sec  0.35 MBytes  0.28 Mbits/sec
```

Figura 37. Relatório Cliente e Servidor do tráfego TCP gerado pelo Iperf através de uma VPN - Vyatta

Referências

Almesberger, W. (2003). *Traffic Control Next Generation Reference Manual*.

Blunden, B. (2002). *Virtual Machine Design and Implementation in C/C++*. Wordware Publishing.

Comer, D. E. (2007). *Redes de Computadores e Internet*. Bookman.

Dreher, F. (2011). Universidade reformula estrutura de segurança.

Golden, B. and Scheffy, C. (2008). *Virtualization for Dummies, Sun AMD Special Edition*. Wiley Publishing INC.

Illinois, U. (2011). Iperf manual.

Jones, T. M. (2011). Virtual linux.

Kurose, J. F. and Ross, K. W. (2006). *Redes de Computadores e a Internet: Uma abordagem top-down*. Pearson Addison Wesley.

Loscocco, P., Smalle, S., Muckelbauer, P., Taylor, R., Turner, J. S., and Farrel, J. (1998). The inevitability of failure: The flawed assumption of security in modern computing environments. In *In 21st National Information Systems Security Converence*.

Matthews, J. N., Dow, E. M., Deshane, T., Hu, W., Bongio, J., F.W., P., and Johnson, B. (2008). *Running Xen*. Prentice Hal.

Modine, A. (2011). Vyatta blows out cisco routers with study.

Morimoto, C. E. (2008). *Redes e Servidores Linux: Guia prático*. GDH Press.

Nakamura, E. and Geus, P. (2002). *Segurança de Redes em ambientes cooperativos*. Berkeley.

Paquet, C. and Teare, D. (2002). *Construindo Redes Cisco Escaláveis*. Makron Books.

Santana, I. (2011). Esquenta disputa entre fabricantes.

Siqueira, L. and Brendel, J. C. (2007). *Linux Pocket Pro - Virtualização*. Linux New Media.

System, V. (2011a). *Vyatta Basic System Reference Guide*. Vyatta System, http://www.vyatta.com/downloads/documentation/VC6.3/VyattaBasicSystemR6.3v01.pdf, r6.3.0 edition.

System, V. (2011b). *Vyatta System Quick Start Guide*. Vyatta System, http://www.vyatta.com/downloads/documentation/VC62/VyattaQuickStart.pdf, r6 2 v01 edition.

Ulbrich, C. H. (2004). *Universidade Hacker*. Digerrati.