

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIA  
ESPECIALIZAÇÃO EM GERENCIAMENTO DE REDES E SERVIDORES**

**LUCIANO BUENO**

**ROTEADOR LINUX UTILIZADO COMO GATEWAY DE  
BACKUP PARA ALTA DISPONIBILIDADE**

**MONOGRAFIA DE ESPECIALIZAÇÃO**

**CURITIBA**

**2017**

**LUCIANO BUENO**

**ROTEADOR LINUX UTILIZADO COMO GATEWAY DE BACKUP  
PARA ALTA DISPONIBILIDADE**

Monografia de especialização apresentada como requisito parcial para obtenção do Grau de especialista em gerenciamento de redes e servidores do programa de pós-graduação em tecnologia, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Juliano Mello

**CURITIBA**

**2017**



---

## **TERMO DE APROVAÇÃO**

### **ROTEADOR LINUX UTILIZADO COMO GATEWAY DE BACKUP PARA ALTA DISPONIBILIDADE**

por

**LUCIANO BUENO**

Esta Monografia foi apresentada em 27 de novembro de 2017 como requisito parcial para a obtenção do título de Especialista em Gerenciamento de Servidores e Equipamentos de Rede. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Augusto Foronda  
Prof. Coordenador do Curso

---

Juliano de Mello Pedroso  
Prof. Orientador

---

Kleber Kendy Horikawa Nabas  
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

## RESUMO

BUENO, Luciano. **ROTEADOR LINUX UTILIZADO COMO GATEWAY DE BACKUP PARA ALTA DISPONIBILIDADE.** 2017. 52f. Monografia de especialização (obtenção de título de especialista em gerenciamento de redes e servidores) – Curso de pós-graduação *latu sensu*, Universidade Tecnológica Federal do Paraná, Curitiba, 2017.

Este projeto pretende simular um sistema de alta disponibilidade entre dois roteadores, sendo o *link* principal através de um roteador Cisco e o *link* de *backup* através de um computador com sistema operacional Linux Ubuntu 17.04, carregado com o pacote de roteamento *Quagga* e utilizando o protocolo VRRP para alta disponibilidade.

**Palavras chave:** VRRP. Roteador Linux. Alta disponibilidade. GNS3. Roteador Cisco. *Quagga*. OSPF.

## ABSTRACT

BUENO, Luciano. **LINUX ROUTER USED AS *BACKUP GATEWAY* FOR *HIGH AVAILABILITY***. 2017. 52p. Monografia de especialização (obtenção de título de especialista em gerenciamento de redes e servidores) – Programa de pós-graduação *latu sensu*, Universidade Tecnológica Federal do Paraná, Curitiba, 2017.

This project intends to simulate a high availability system between two routers, the main *link* being through a Cisco router and the *backup link* through a computer with Linux operating system Ubuntu 17.04, loaded with the *Quagga* routing package and using the VRRP protocol for High Availability.

**Keywords:** VRRP. Linux Router. High Availability, GNS3, Cisco Router, *Quagga*, OSPF.

## LISTA DE FIGURAS

Figura 1 - Cenário de simulação dentro da plataforma GNS3 .....	13
Figura 2: Redundância de roteadores para alta disponibilidade .....	22
Figura 3: Troca de roteador de encaminhamento após falha.....	22
Figura 4: Sistema operacional Windows 8 virtualizado em um computador com MacOS.....	26
Figura 5: Oracle VirtualBox com máquinas virtuais instaladas .....	30
Figura 6: Detalhes das configurações das placas de rede da máquina virtual ubuntuserver .....	31
Figura 7: Adição de um roteador através de uma imagem do Cisco IOS no GNS3 .....	33
Figura 8: GNS3 com topologia de teste do ambiente de alta disponibilidade com roteador Cisco e Linux .....	34
Figura 9: Detalhes da configuração da placa de rede da máquina virtual Clonedeubuntu .....	35
Figura 10: detalhes da configuração da placa de rede da máquina virtual ubuntu .....	36
Figura 11: Saída do comando <i>show vrrp</i> , que mostra o roteador R2 como <i>master</i> .....	38
Figura 12: Configuração das interfaces de rede na máquina ubuntuserver .....	39
Figura 13: Configuração dos <i>daemons</i> do Quagga .....	41

<b>Figura 14: Configuração do OSPF dentro do <i>Quagga</i> .....</b>	<b>42</b>
<b>Figura 15: Tabela de roteamento exibida pelo Quagga no roteador ubuntuserver .....</b>	<b>42</b>
<b>Figura 16: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota) .....</b>	<b>44</b>
<b>Figura 17: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota) após desabilitar F0/0 de R2 .....</b>	<b>45</b>
<b>Figura 18: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota), antes e após desabilitar F0/0 de R2.....</b>	<b>45</b>
<b>Figura 19: Traceroute entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota), antes e após desabilitar F0/0 de R2.....</b>	<b>46</b>
<b>Figura 20: Tráfego redirecionado para interface enp0s3 de ubuntuserver após simulação de falha em F1/0 de R2.....</b>	<b>47</b>
<b>Figura 21: Script de inicialização do VRRP, dentro do diretório init.d .....</b>	<b>48</b>

## LISTA DE ABREVIATURAS E SIGLAS

ARP	<i>Address Resolution Protocol</i>
AS	<i>Autonomous System</i>
BGP	<i>Border gateway protocol</i>
CPU	Unidade Central de Processamento
GLBP	<i>Gateway Load Balancing Protocol</i>
EGP	<i>Exterior Gateway Protocol</i>
EIGRP	<i>Enhanced Interior Gateway Routing Protocol</i>
FHRP	<i>First Hop Redundancy Protocols</i>
GLBP	<i>Gateway Load Balancing Protocol</i>
GNS3	<i>Graphical network simulator</i>
GPL	<i>General Public License</i>
HSRP	<i>Hot Standby Router Protocol</i>
IGP	<i>Interior Gateway Protocol</i>
IOS	<i>Internetwork operating system</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol Version 4</i>
LAN	<i>Local Area Network</i>



LSP	<i>Link-state packet</i>
MAC	<i>Media Access Control</i>
OSPF	<i>Open Shortest Path First</i>
PC	<i>Personal Computer</i>
RAM	<i>Random Access Memory</i>
RIP	<i>Routing Information Protocol</i>
TCP	<i>Transmission Control Protocol</i>
VRRP	<i>Virtual Router Redundancy Protocol</i>

## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>6</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>8</b>
<b>1 INTRODUÇÃO .....</b>	<b>10</b>
<b>1.1 PROBLEMA .....</b>	<b>11</b>
<b>1.2 OBJETIVOS.....</b>	<b>12</b>
<b>1.2.1 Objetivo Geral .....</b>	<b>12</b>
<b>1.2.1 Objetivos Específicos .....</b>	<b>12</b>
<b>1.3 JUSTIFICATIVA .....</b>	<b>14</b>
<b>1.4 ESTRUTURA DO TRABALHO .....</b>	<b>14</b>
<b>1.5 MÉTODO DE PESQUISA.....</b>	<b>15</b>
<b>1.6 CRONOGRAMA .....</b>	<b>16</b>
<b>2 REVISÃO BIBLIOGRÁFICA.....</b>	<b>18</b>
<b>2.1 Roteadores .....</b>	<b>18</b>
<b>2.1.1 Roteamento Estático .....</b>	<b>18</b>
<b>2.1.2 Roteamento Dinâmico.....</b>	<b>18</b>
<b>2.2. SWITCHES.....</b>	<b>20</b>
<b>2.3. Alta disponibilidade em redes de computadores .....</b>	<b>21</b>

2.4 VRRP .....	23
2.5 Sistema operacional Linux.....	24
2.6 GNS3 .....	25
2.7 Oracle VirtualBox.....	26
2.8 Quagga Linux.....	28
<b>3 DESENVOLVIMENTO .....</b>	<b>29</b>
3.1 Configuração do Oracle VirtualBox.....	29
3.2 Configuração da topologia no GNS3.....	32
3.2.1 Clonedeubuntu .....	34
3.2.2 Ubuntu .....	35
3.2.3 R2 (Roteador Cisco C7200).....	36
3.2.4 Ubuntuserver: .....	38
3.2.5 Operadora (Cisco C7200).....	43
<b>4 ENSAIOS E RESULTADOS .....</b>	<b>44</b>
<b>5 CONCLUSÕES.....</b>	<b>49</b>
6.1 TRABALHOS FUTUROS .....	49
<b>6 REFERÊNCIAS.....</b>	<b>51</b>

## 1 INTRODUÇÃO

Na atualidade, os sistemas de informática estão presentes na maioria das empresas, independentemente do porte ou tipo de negócio envolvido. As redes de computadores fornecem um meio para troca de informações entre os setores da empresa (internos ou externos) e até mesmo com fornecedores e clientes, de maneira rápida e eficaz.

Muitos tipos de negócio devem funcionar de maneira ininterrupta, como por exemplo, serviços bancários, sites de vendas, aeroportos, entre outros. Interrupções no fornecimento do serviço *on-line* em casos como estes podem significar prejuízos financeiros ou morais enormes para as empresas (RUSSO, 2006). Algumas empresas, no intuito de aumentar a disponibilidade dos serviços em caso de falhas no *link* de *internet* (ou até mesmo do roteador), adotam métodos de redundância que utilizam dois ou mais roteadores, com *links* de *internet* de diferentes operadoras e por meios físicos separados, para que um *link* secundário opere de maneira provisória até que o principal seja restabelecido. Empresas de grande porte geralmente aplicam esta solução utilizando dois ou mais roteadores, que são equipamentos robustos e que cumprem especificamente a tarefa para qual foram construídos. No entanto, estes roteadores possuem um custo considerável para empresas de pequeno porte, onde a aquisição de dois ou mais equipamentos, muitas vezes se torna inviável.

Este projeto tem como finalidade apresentar uma possibilidade de solução de redundância de *links* para empresas de pequeno porte, utilizando um roteador configurado dentro de um computador comum com sistema operacional Linux, capaz de carregar os protocolos de roteamento necessários para o funcionamento da rede de maneira ininterrupta, após a queda do *link* principal, o qual estará configurado em um roteador Cisco. O cenário desta solução será simulado sobre a plataforma GNS3, que consiste em uma ferramenta capaz de carregar sistemas operacionais reais em equipamentos virtuais, tanto roteadores quanto computadores. O *software* GNS3 também permite a conexão em rede dos equipamentos virtuais, criando um cenário quase idêntico ao real. Esta funcionalidade será de extrema importância na comprovação da eficiência

da solução proposta, uma vez que facilmente podemos provar o correto funcionamento do sistema de *backup* ao desligar o *link* principal (ou até mesmo o roteador) para comprovar que o roteador Linux entrará em operação automaticamente.

Existem alguns protocolos que são implementados nos roteadores ou *switches* de camada 3, que servem para garantir a alta disponibilidade dos *links*, tais como: HSRP, GLBP e VRRP (FILIPPETTI, 2008). Neste trabalho será abordado o protocolo VRRP (*virtual Router Redundancy Protocol*), pois é um protocolo aberto (FILIPPETTI, 2008) e também suportado por um pacote de aplicação para o sistema operacional Linux, além de também estar presente em grande parte dos sistemas operacionais dos roteadores Cisco. Sendo assim, este protocolo é ideal para a implementação do cenário proposto.

## 1.1 PROBLEMA

A conectividade com a *internet* hoje é algo fundamental no processo da maioria das empresas. Perder conexão com a rede externa pode significar em prejuízos incalculáveis, dependendo da linha de negócio (RUSSO, 2006). Para empresas de pequeno porte, perder a conexão pode significar perder uma oportunidade, muitas vezes única, para um concorrente.

A redundância com dois ou mais roteadores é uma solução simples e traz uma enorme segurança para a conectividade das empresas. No entanto, apesar de ser uma solução aparentemente simples, há de se considerar que o custo de cada roteador pode chegar a alguns milhares de Reais, tornando-se inviável para empresas de pequeno porte, em fase de expansão ou com problemas financeiros, onde os recursos monetários aplicados em um roteador fariam muita diferença se aplicados em outra linha de investimento.

Ao apresentar um roteador construído dentro de um computador comum com sistema operacional Linux como opção para o equipamento secundário dentro de um sistema de redundância de alta disponibilidade, será necessário

realizar um estudo sobre a instalação, funcionamento e configuração dos pacotes e protocolos para a distribuição Linux envolvidos no processo. Estes protocolos e pacotes de aplicações devem ser compreendidos para suas corretas configurações, a fim de obter o que se espera, que é o correto funcionamento da redundância de *gateways* com alta disponibilidade, ou seja, o roteador Linux deverá entrar em operação automaticamente após ocorrer uma falha no roteador principal (Cisco).

Esta implementação deverá ser capaz de propor uma solução eficaz e com considerável redução de custos, tornando-se viável para empresas que não podem arcar com os custos de dois ou mais roteadores para manter sua conectividade com a *internet*. Serão levantados os valores finais, em comparação com a implementação convencional utilizando dois roteadores, para que o leitor possa ter uma noção dos inúmeros benefícios que se pode agregar com futuras implementações e atualizações do processo utilizando esta proposta.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Implementar um cenário virtual dentro da plataforma GNS3 para simulação de um sistema de redundância de *gateways* para alta disponibilidade, utilizando um roteador Cisco (principal) e um roteador construído dentro de um computador com sistema operacional Linux Ubuntu 17.04. A finalidade deste conjunto será manter o *link* de *internet* da rede *lan* (*Local Area Network*) através do roteador Linux, após a queda do meio ou do próprio roteador principal.

### 1.2.1 Objetivos Específicos

Para alcançar o objetivo proposto, será necessário cumprir as seguintes etapas:

1. Instalação da plataforma GNS3: Esta aplicação será instalada dentro de um computador com sistema operacional *Windows 7*, com o objetivo de simular equipamentos virtuais carregados com sistemas operacionais reais.

2. Instalação das máquinas virtuais: para as máquinas com sistema operacional Linux Ubuntu 17.04, será utilizado o *software Virtual Box*, cuja funcionalidade consiste em executar computadores virtuais com sistemas operacionais reais. Já os roteadores virtuais terão seus sistemas operacionais carregados dentro do próprio GNS3.

3. Pesquisa para configuração dos protocolos de roteamento e VRRP e no roteador Cisco e no roteador Linux.

4. Configuração e simulação do sistema redundância de *gateways* com o protocolo VRRP. Uma vez configurado o cenário, será possível realizar testes desligando a porta do roteador Cisco configurada com o protocolo VRRP, observando se a placa de rede do roteador Linux assume o endereço IPv4 (*internet protocol version 4*) do *gateway* virtual. Desta forma, um *host* na dentro da *lan* deve continuar tendo conectividade com o segmento de rede após os roteadores, que simula a *internet*.

A Figura 1 mostra o exemplo deste cenário.

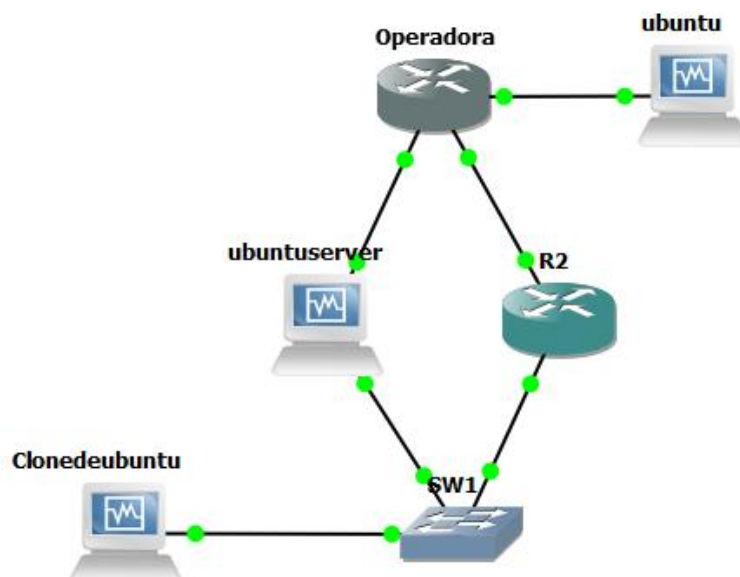


Figura 1 - Cenário de simulação dentro da plataforma GNS3

Fonte: Autoria própria

### 1.3 JUSTIFICATIVA

Os sistemas de redundância de *gateways* com alta disponibilidade utilizam, de maneira convencional, dois ou mais roteadores configurados com os protocolos adequados (GLBP, HSRP ou VRRP). Considerando que grande parte do mercado de equipamentos de redes de computadores utiliza equipamentos da marca Cisco, que dependendo do modelo pode chegar a dezenas de milhares de Reais, esta solução muitas vezes se torna inviável para empresas de pequeno porte.

Obviamente, não se pode negar que a robustez e a confiabilidade dos equipamentos da Cisco justificam o investimento. Entretanto, para empresas que não possuem condições financeiras de implementar dois roteadores cisco, onde um destes equipamentos permaneceria em operação somente para manter a redundância para alta disponibilidade, é também óbvio que haverá uma menor probabilidade de se perder toda a conectividade com a *internet* ao se utilizar um sistema de redundância no qual o roteador secundário seja construído sobre um *hardware* de computador comum, em comparação a um sistema sem nenhuma redundância.

Sendo assim, o que se pretende com este trabalho não justificar a substituição de nenhum equipamento Cisco dentro da rede, mas sim apresentar uma opção funcional para empresas que não poderiam arcar com os custos de um segundo roteador para manter a alta disponibilidade do *link* de *internet*.

### 1.4 ESTRUTURA DO TRABALHO

Este trabalho terá a estrutura abaixo apresentada.

**Capítulo 1 - Introdução:** serão apresentados o tema, o problema, os objetivos da pesquisa, a justificativa e a estrutura geral do trabalho.

**Capítulo 2 – Cronograma:** será apresentado um cronograma para o desenvolvimento e conclusão do trabalho



**Capítulo 3 – Revisão Bibliográfica:** serão apresentadas as revisões dos temas relacionados com o trabalho e necessários para o seu desenvolvimento, tais como:

**Capítulo 4 – Desenvolvimento:** neste capítulo serão apresentadas todas as configurações, métodos e recursos utilizados para a configuração de um computador com sistema operacional Linux Ubuntu 17.04 para funcionar como roteador de *backup*, em um sistema de alta disponibilidade, em conjunto com um roteador Cisco C7200 (*master*).

**Capítulo 5 – Ensaio e Resultados:** Capítulo que apresenta a descrição detalhada dos testes e resultados obtidos ao término do desenvolvimento.

**Capítulo 6 – Conclusões:** Neste capítulo é apresentada uma breve avaliação sobre os problemas propostos e os resultados obtidos, bem como os possíveis trabalhos futuros que podem utilizar este trabalho como base.

**Capítulo 7 – Referências:** Apresenta todas as referências bibliográficas utilizadas no desenvolvimento do trabalho.

## 1.5 MÉTODO DE PESQUISA

Neste projeto pretende-se apresentar um cenário de rede simulado dentro da plataforma GNS3, contendo dois roteadores, sendo um roteador Cisco C7200 e um roteador construído dentro de um computador com sistema operacional Linux Ubuntu 17.04, ambos trabalhando como um sistema de redundância de *gateway*, utilizando o protocolo VRRP.

O cenário de rede será composto pelos seguintes elementos:

- um roteador Cisco C7200.
- um PC virtual com sistema operacional Ubuntu *server* 17.04, que será configurado para trabalhar como um roteador.
- dois *Switches*, simulados dentro do próprio GNS3, utilizados apenas para ligação dos *hosts* das *lans*.

- dois computadores virtuais com sistema operacional Linux Ubuntu 17.04. Um destes computadores será um *host* da *lan* abaixo do *gateway* virtual, ou outro computador servirá como simulação de um *link* externo.

Para realização deste trabalho será necessário o entendimento do modo de funcionamento e configuração do protocolo VRRP a ser implementado nos dois roteadores (Cisco - principal e Linux – *Backup*), para que ambos trabalhem em conjunto como *gateway* virtual, sendo que o roteador Linux somente irá transmitir os pacotes entre as redes caso ocorra algum problema com o roteador Cisco. Esta comutação deverá ocorrer de forma automática.

Para a atualização das tabelas de roteamento dos roteadores, será utilizado o protocolo OSPF (*Open Shortest Path First*). No roteador Linux, será utilizado o pacote de aplicativo *Quagga*, que possui o protocolo OSPF, além de outras funcionalidades e protocolos de roteamento. A instalação e configuração do pacote *Quagga* serão descritos no desenvolvimento do trabalho. O protocolo OSPF foi escolhido aleatoriamente apenas para efeito de testes de conectividade entre as redes, não sendo o objetivo deste trabalho confrontar as vantagens ou desvantagens deste protocolo de roteamento em relação a outros protocolos.

## 1.6 CRONOGRAMA

Pretende-se realizar este projeto em um prazo aproximado de 3 meses. As principais etapas e seus respectivos prazos serão a seguir:

Etapa 1 – pesquisa teórica: nesta etapa, pretende-se determinar, através de pesquisa, os recursos dentro do sistema operacional Linux Ubuntu 17.04 que sejam capazes de cumprir com o que se espera de um roteador que trabalhe em conjunto com um roteador Cisco C7200, ambos utilizando o protocolo VRRP para trabalhar como redundância de *gateways*. Faz-se necessário também realizar uma pesquisa sobre o pacote de roteamento *Quagga* Linux, que permitirá ao roteador construído dentro da máquina com Linux utilizar os protocolos de roteamento dinâmico em conjunto com outros roteadores. Pretende-se concluir esta etapa em aproximadamente 4 semanas.

Etapa 2 – Configuração do ambiente de testes dentro do aplicativo GNS3 e *Oracle Virtual Box*, contemplando as instalações dos sistemas operacionais nas máquinas e roteadores virtuais: conclusão em aproximadamente 1 semana.

Etapa 3 – Configuração e testes dos roteadores Cisco e máquinas virtuais com Linux, utilizando o protocolo VRRP e OSPF, para roteamento dinâmico. Conclusão em aproximadamente 3 semanas.

Etapa 4 – Construção da documentação técnica (monografia): pretende-se concluir esta etapa em aproximadamente 4 semanas.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 ROTEADORES

Atualmente, as redes de computadores estão presentes em quase todas as empresas. Para o funcionamento da computação em rede, são necessários diversos equipamentos, dentre os quais encontra-se o roteador. Este equipamento é necessário quando pretende-se interconectar diferentes redes, como por exemplo, conectar à rede de uma empresa à parceiros, filiais e clientes (XAVIER, 2010).

Os roteadores são equipamentos fundamentais para o funcionamento da *internet*. São eles os responsáveis por decidir qual rota um determinado pacote deve seguir, facilitando a comunicação entre as redes (XAVIER, 2010). Em resumo, quando um computador tenta enviar um pacote para outra máquina localizada em outra rede, o roteador é quem determina a rota a ser seguida, identificando o endereço *IP* destino e consultando uma tabela interna de rotas conhecidas, que podem ter sido aprendidas pelo roteador de maneira estática ou dinâmica, sendo que esta tabela contém as informações sobre a rede destino.

#### 2.1.1 Roteamento Estático

Rotas estáticas são caminhos configurados manualmente no roteador. É a forma mais simples de se estabelecer uma rota, indicando que todo pacote enviado à rede destino siga o caminho determinado contido na tabela de roteamento. A grande desvantagem das rotas estáticas é que, ao se alterar a topologia da rede em qualquer parte das redes (origem ou destino), o administrador da rede deve configurar a rota novamente para que a comunicação volte a funcionar. No caso de redes muito grandes, o uso de roteamento estático torna-se muitas vezes inviável.

#### 2.1.2 Roteamento Dinâmico

Em roteamento dinâmico, a tabela de roteamento será automaticamente atualizada sempre que novas informações a respeito da topologia forem

recebidas por roteadores vizinhos (LAMMLE, 2002). Estas informações são trocadas através de protocolos configurados nos roteadores. Desta maneira, para que as rotas sejam aprendidas de maneira dinâmica, os roteadores vizinhos devem estar configurados com os mesmos protocolos, para que esta troca de informações seja possível. Já o método que cada protocolo de roteamento utiliza para realizar esta atualização depende dos algoritmos e das características operacionais do protocolo (FOSTER, 2011).

Os protocolos são classificados por finalidade como sendo IGPs (protocolos de *gateways* internos), utilizados em redes internas autônomas (AS); e EGPs (protocolos de *gateways* externos), utilizado para conectar as AS. O BGP (*border gateway protocol*) é o único EGP viável atualmente, sendo utilizado como protocolo oficial da *internet* (FOSTER, 2011)

Dentre os IGPs, podem-se destacar os seguintes:

**RIP (Routing information protocol):** É um protocolo que usa como base a contagem de saltos até o destino para estabelecer o que considera a melhor rota, com uma limitação de 15 saltos (LAMMLE, 2002).

A versão mais atual deste protocolo é a versão 2 (RIPv2). Mesmo a versão mais atual do RIP ainda não é escalável para implementações de grandes redes (FOSTER, 2011).

**EIGRP (Enhanced Interior Gateway Routing Protocol):** É um protocolo proprietário da Cisco, que substituiu o IGRP (*Interior Gateway Routing Protocol*), também proprietário da Cisco. É um protocolo também baseado em vetor distância, que avalia largura de banda, atraso, carga e confiabilidade para criar uma métrica (FOSTER, 2011).

O EIGRP envia propaga atualizações apenas quando ocorrem alterações na tabela de roteamento, o que reduz o volume da carga que o protocolo de roteamento coloca sobre a rede. Utiliza também um mecanismo *Hello keepalive*, que é uma pequena mensagem de aviso, trocada periodicamente entre os roteadores vizinhos com a finalidade de manter as adjacências. Isso é extremamente eficaz em relação ao uso dos recursos da rede pois durante a

operação normal, pois são enviados pequenos pacotes em vez de atualizações periódicas (LAMMLE, 2002).

O EIGRP também é capaz de operar com outros protocolos de rede além do IPV4 e IPV6, como o IPX legado e o *Appletalk* (LAMMLE, 2002).

**OSPF (Open Shortest Path First)** : É um protocolo capaz de suportar grandes redes baseado em “*link-state*”, ou seja, cada roteador guarda a topologia da rede local em sua tabela de roteamento, com informações das interfaces, número do enlace e a métrica (ou custo). Diferentemente dos protocolos baseados em vetor distância, uma rota determinada pelo OSPF pode seguir um caminho mais longo em relação ao número de saltos, desde que o custo total (ou seja, a soma das métricas) deste caminho seja menor do que o custo do caminho com menor número de saltos (LEWIS, 1999).

O envio de atualizações não é periódico, ou seja, ocorre apenas quando uma alteração na topologia é detectada. Entretanto, ao receber uma atualização (chamada LSP – *link-state packet*), um roteador inunda todas as interfaces com as LSPs, menos para a interface pela qual foi recebido o LSP original.

## 2.2. SWITCHES

Os *switches* são dispositivos usados para conectar vários *hosts* na mesma rede, direcionando o fluxo de informações a nível de camada de acesso (FOSTER, 2011). Basicamente, um *switch* serve para direcionar os quadros enviados de um dispositivo de origem para outro de destino, tomando em conta a porta pela qual foram recebidos os quadros e o endereço ao qual o quadro se destina. Os *switches* possuem uma tabela pela qual relaciona uma determinada porta aos endereços MAC diretamente conectados a ela. Sendo assim, um quadro destinado a um dispositivo sempre será encaminhado pela mesma porta

de saída, independentemente da porta por onde o quadro foi enviado (FOSTER, 2011).

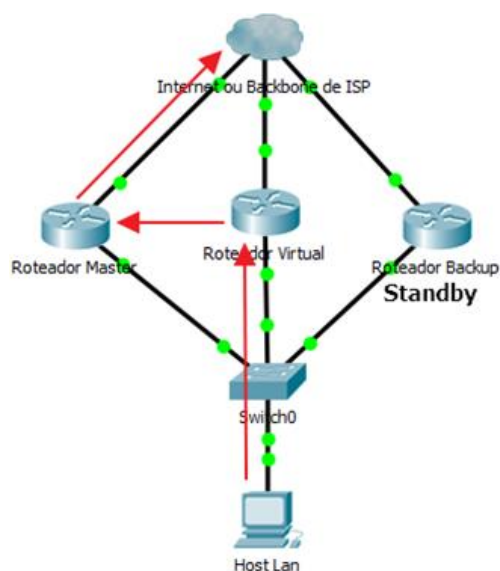
### 2.3. ALTA DISPONIBILIDADE EM REDES DE COMPUTADORES

Um sistema de alta disponibilidade é aquele capaz de manter um serviço ativo aos usuários mesmo quando houver falha de um ou mais dispositivos dentro do ambiente. Em uma rede de computadores, é possível manter a alta disponibilidade utilizando *links* e equipamentos redundantes, como *switches* e roteadores. Desta forma, a rede continuará operando mesmo quando falhar um *link*, roteador ou *switch*, independente se a falha ocorrer em uma única porta ou no equipamento inteiro (LEWIS, 1999).

Roteadores ou *switches* multicamada, operando em um sistema com redundância, oferecem a um cliente a capacidade de serem configurados como um *gateway* padrão alternativo, caso ocorra uma falha no *gateway* padrão principal. Sendo assim, para um determinado *host*, é possível existir vários caminhos para mais de um *gateway* padrão possível (LAMMLE, 2002).

O *gateway* padrão é determinado aos clientes da rede através do uso dos Protocolos de Redundância de Primeiro Salto (*First Hop Redundancy Protocols*), que servem para gerenciar o modo de operação hierárquico dos *gateways*, conforme o método utilizado para dar prioridade a cada equipamento.

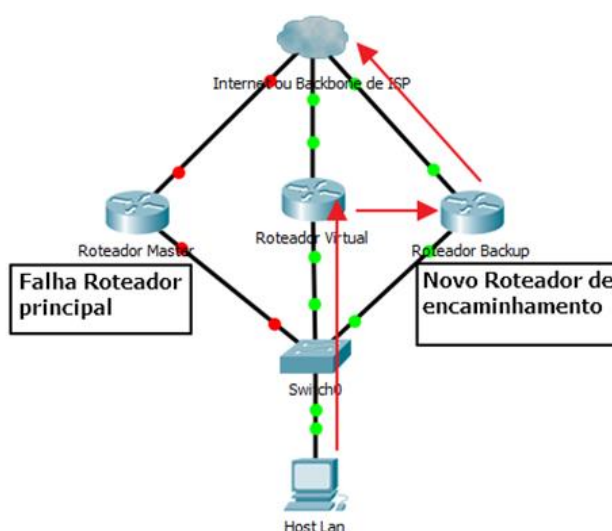
A figura 2 traz um exemplo de uma rede com dois roteadores que podem operar como *gateway* padrão, conforme a hierarquia determinada pelo protocolo. É possível observar a presença de um roteador de encaminhamento (principal), um roteador em *standby* (redundância) e um roteador virtual, que nada mais é do que a ilustração do *gateway* padrão criado virtualmente pelo conjunto configurado com um protocolo do tipo FHRP.



**Figura 2: Redundância de roteadores para alta disponibilidade**

**Fonte: Autoria própria**

Já na figura 3 podemos observar o comportamento do *cluster* quando o roteador principal falha. O protocolo de redundância utilizado é encarregado de fazer a transição do roteador em *standby* para ser o novo roteador de encaminhamento, que passa a assumir o endereço *IP* e *MAC* do roteador virtual. Isto faz com que nenhuma interrupção venha a ser constatada pelo usuário da rede.



**Figura 3: Troca de roteador de encaminhamento após falha**

**Fonte: Autoria própria**



Neste projeto, iremos abordar o protocolo VRRP, que é um exemplo de protocolo do tipo FHRP aberto, característica esta necessária para operar no sistema operacional Linux.

## 2.4 VRRP

O VRRP (*Virtual Router Redundancy protocol*) é um protocolo aberto de eleição que atribui dinamicamente a responsabilidade por um roteador virtual a um dos roteadores VRRP em uma LAN. O roteador VRRP com maior prioridade controla o endereço IP associado a um roteador virtual, sendo classificado como mestre, enquanto os roteadores de *backup* (com menor prioridade) permanecem em *standby*. O processo de eleição proporciona a adaptação dinâmica na responsabilidade de encaminhamento ao roteador de *backup* com segunda maior prioridade, em caso de falha do roteador mestre. Isso permite que qualquer um dos roteadores que fazem parte do roteador virtual na LAN seja eleito como roteador de primeiro salto padrão para o *host* final. Com isto, mantém-se a alta disponibilidade da rede sem que o *host* perceba a falha do roteador mestre (RFC2338).

O critério para a definição da hierarquia dos roteadores do grupo consiste primeiramente na verificação do parâmetro prioridade configurado no roteador. Havendo mais de um roteador com prioridade semelhante, utiliza-se o maior endereço IP dentre eles como critério de desempate para definir a maior prioridade.

Os pacotes trocados entre os roteadores do grupo para anunciar a operação são chamados “*advertisement*” e são enviados via *multicast* para o endereço 224.0.0.18 e porta 112. O roteador *master* responde pelas solicitações de ARP com o endereço Mac 0000.005e.01XX onde xx é o número do grupo. O VRRP conta com o recurso da preempção, ou seja, quando o roteador com maior prioridade voltar ao estado ativo, ele assumirá novamente o estado de *master* automaticamente (DIAS, 2015)

## 2.5 SISTEMA OPERACIONAL LINUX

O Linux é um sistema operacional criado pelo finlandês Linus Torvalds no ano de 1991. A principal característica do Linux é o fato de ser sistema operacional de código aberto, ou seja, sua instalação e distribuição são gratuitas, não havendo a necessidade de obter licenças para seu funcionamento. Seu código fonte é liberado como “*free Software*” (*software livre*), sob licença GPL (*General Public License*) (SILVA, 2010). O código fonte aberto permite que qualquer pessoa possa veja como o sistema funciona, corrija problemas e implemente melhorias. Existem diversos grupos de pessoas espalhadas pelo mundo, trabalhando voluntariamente para o desenvolvimento e melhoria do Linux.

Sozinho, o *Kernel GNU/Linux* não é o suficiente para se ter um sistema operacional funcional. Existem diversas distribuições do Linux que acompanham determinados pacotes de programas, como editores de texto, planilhas, navegadores, ferramentas de programação, dentre outras. Cada distribuição é implementada conforme seu objetivo, havendo distribuições com interfaces gráficas amigáveis (voltadas para usuários), versões com serviços básicos (que ocupam menos recursos da CPU, ideal para serviços específicos) e diversas outras, cada uma com suas características específicas (SILVA, 2010). As distribuições são executadas sobre o Kernel GNU/Linux, que é basicamente o núcleo do sistema operacional.

Dentre as distribuições mais conhecidas atualmente, estão: Ubuntu, Debian, Slackware, CentOS, Red Hat, Gentoo e Suse. Todas estas distribuições utilizam o Linux como kernel principal. A distribuição Debian é independente de kernel, ou seja, pode ser executada sob outros kernels, como o GNU hurd ou o kernel BSD.

Este trabalho utiliza a distribuição Ubuntu como base para a construção do roteador de *backup*. A distribuição Ubuntu foi desenvolvida pela empresa Canonical, sediada na ilha de Man, na África do Sul. Baseada na distribuição

Debian, o Ubuntu foi desenvolvido para ser totalmente livre e compartilhado entre as pessoas (SILVA, 2010). Possui versões para usuários e servidores, com e sem interface gráfica.

## 2.6 GNS3

O *software* GNS3 (*graphical network simulator*) é um *software* livre (*open source*) que serve para criar diagramas de redes e ao mesmo tempo simular o funcionamento destas redes. Com o GNS3, é possível utilizar imagens reais dos IOS (*internetwork operating system*) da Cisco, que são emuladas através de um programa chamado *Dynamips* (TAVARES, 2011). Sendo assim, pode-se afirmar que o GNS3 é uma interface gráfica para o *Dynamips*, que é o programa responsável por emular os equipamentos da Cisco utilizando os IOS reais (TAVARES, 2011). Além de emular os equipamentos da cisco, é possível integrar máquinas virtuais de softwares como *VirtualBox* e *VMWare*, por exemplo, e agregá-las ao diagrama de redes, o que permite simular cenários muito próximos aos reais. Outra característica interessante do GNS3 é a possibilidade de implementar uma ponte entre um equipamento emulado e a placa de rede física do computador onde está instalado. Desta forma, pode-se mesclar as redes simuladas dentro do GNS3 com redes reais, interligadas pela máquina onde o GNS3 encontra-se instalado.

O site oficial do GNS3 traz os requisitos do computador onde o *software* será instalado, descritos na tabela 1.

ITEM	REQUERIMENTO
Sistema operacional	Windows 7 (64 bits) ou superior
Processador	4 ou mais núcleos lógicos- AMD-V / RVI Series ou Intel VT-X / EPT
Virtualização	Extensões de virtualização necessárias. Pode ser necessário habilitar isto na Bios do computador
Memoria	8 GB RAM ( <i>Random Access Memory</i> )
Armazenamento	Solid-state Drive (SDD) 35 GB de espaço disponível

**Tabela 1: Recomendações de configuração do computador para instalação do GNS3**

## 2.7 ORACLE VIRTUALBOX

*VirtualBox* é uma aplicação desenvolvida pela empresa *Oracle*, que serve para realizar a virtualização de sistemas operacionais. É um *software* multiplataforma, ou seja, pode ser instalado em computadores com sistemas operacionais Windows, Mac, Linux ou Solaris, em arquiteturas intel ou AMD (ORACLE, 2017). Com o *VirtualBox*, é possível executar vários sistemas operacionais (dentro de várias máquinas virtuais) ao mesmo tempo. Por exemplo, é possível instalar e executar o Windows e o Linux em um computador com sistema operacional Mac, executar o Windows Server 2008 em um servidor Linux, executar Linux em um PC com Windows, e assim por diante. Também é possível instalar e executar quantas máquinas virtuais forem necessárias, sendo o espaço em disco e a memória disponível as únicas limitações para isto (ORACLE, 2017).

O *VirtualBox* pode ser executado em diversos tipos de computadores, desde pequenos sistemas embarcado, computadores pessoais, servidores e até mesmo em ambientes em nuvem (ORACLE, 2017).

A figura 4 mostra um ambiente com sistema operacional Mac, executando uma máquina virtual com sistema operacional Windows 8.

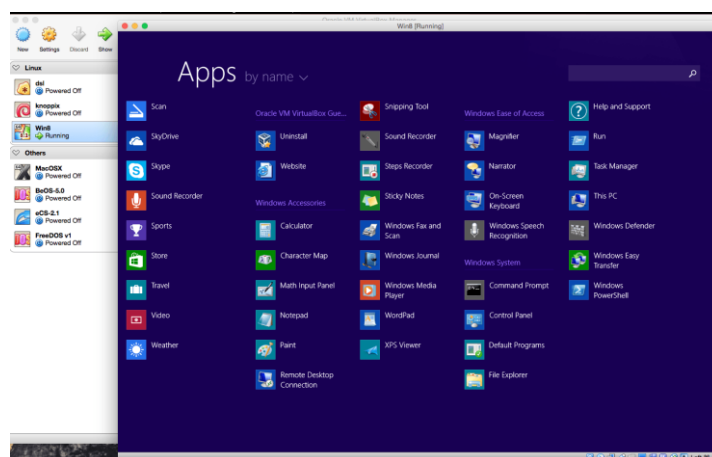


Figura 4: Sistema operacional Windows 8 virtualizado em um computador com MacOS

Fonte: Oracle VirtualBox User Manual, 2017, capítulo 1

OS recursos que o *VirtualBox* fornece podem ser úteis para as seguintes situações:

1 – Executar vários sistemas operacionais simultaneamente, sendo possível utilizar *softwares* desenvolvidos para uma determinada plataforma, sem precisar reiniciar o computador pelo sistema operacional específico (ORACLE, 2017).

2 – *Backup* para recuperação de desastres: Com o *VirtualBox*, é possível instalar sistemas de *backup* para armazenamento de dados de servidores, havendo a possibilidade de se utilizar este *backup* como armazenamento e até mesmo como um sistema espelho, que assume a execução dos serviços quando o servidor principal falhar (ORACLE, 2017).

3 – Ambiente de testes: Ao se implementar melhorias em um determinado serviço, o desenvolvedor necessita realizar testes antes de colocá-las em prática no ambiente corporativo. Desta forma, utilizando um ambiente espelho dentro do *VirtualBox*, caso ocorra alguma falha dentro da implementação, o impacto será resumido ao ambiente de testes e não causará impacto no ambiente corporativo (ORACLE, 2017).

4 – Otimização de recursos: Muitas vezes, os recursos de memória, armazenamento e processamento de alguns servidores são subutilizados, ou seja, sobram recursos que poderiam estar executando outras tarefas instaladas em outros servidores. Desta forma, é perfeitamente possível instalar vários servidores virtuais dentro de um único *hardware*, otimizando o uso dos recursos da máquina e gerando economias como por exemplo, na energia elétrica que seria necessária para alimentar os outros servidores reais (ORACLE, 2017).

Neste trabalho, o Oracle *VirtualBox* será utilizado para executar o sistema operacional Ubuntu 17.04, nas versões *server* (que será configurado como o roteador de *backup*) e *desktop* (que servirá para simular os usuários finais dentro da rede).

## 2.8 QUAGGA LINUX

O *software Quagga* para plataformas Linux é uma ramificação do GNU Zebra, que é um software livre (GPL) que gerencia protocolos de roteamento dinâmico baseados em TCP/IP. O Zebra possui serviços, denominados “*daemons*” com suporte aos protocolos de OSPFv2, OSPFv3, RIP v1, RIP v2, RIPng e BGP-4, que podem ser habilitados individualmente e comunicam as atualizações de roteamento para o núcleo Zebra (ISHIGURO, 2005).

Um computador com o *Quagga* instalado é capaz de atuar como um roteador dedicado, trocando informações de roteamento com outros roteadores através dos protocolos de roteamento. O *Quagga* usa essas informações para atualizar a tabela de roteamento do kernel para que os pacotes sejam roteados para o destino correto (ISHIGURO, 2005).

### 3 DESENVOLVIMENTO

Neste capítulo, serão apresentadas as configurações utilizadas no ambiente do *software* GNS3 para simular o ambiente de alta disponibilidade, utilizando um roteador Cisco C7200 como roteador principal e um roteador construído dentro da máquina virtual com sistema operacional Ubuntu *server* 17.04.

O computador utilizado para a implementação possui as seguintes características:

- Sistema operacional Windows 7 Professional 64 bits
- Memória RAM de 8GB, DDR3
- Processador Intel(R) Core(TM) i7-3770 CPU @3.4GHz
- Disco Rígido de 1Tera *Byte* de espaço.

#### 3.1 CONFIGURAÇÃO DO ORACLE VIRTUALBOX

O instalador do Oracle *VirtualBox* foi adquirido desde o site <http://www.oracle.com/technetwork/pt/server-storage/VirtualBox/downloads>.

Este trabalho não irá demonstrar todo o processo de instalação do Oracle *VirtualBox*, bem como a instalação de máquinas virtuais, pois estes detalhes podem ser obtidos diretamente no manual do próprio *software*.

Após instalado o *software* Oracle *Virtualbox*, foram instalados os sistemas operacionais virtuais Ubuntu *server* 17.04 e Ubuntu *desktop* 17.04 através das imagens obtidas no site <https://www.ubuntu.com/download>. Para simular os *hosts* da rede, foram criados “clones *linkados*”, ou seja, os sistemas operacionais dos *hosts* executam recursos desde o sistema operacional da máquina virtual de origem, utilizando menos recursos do *hardware* físico.

A figura 5 mostra a tela da configuração do Oracle *VirtualBox* após a instalação das máquinas virtuais que serão utilizadas no projeto, onde:

- **Clone de ubuntu:** Máquina a ser utilizada como *host* de rede, clonada desde a máquina ubuntu.

- **Clone de ubuntuserver:** Sem uso, criada apenas para ser utilizada como *host* de rede, caso necessário, pois utiliza menos recursos de *hardware* do que as máquinas com sistema operacional Ubuntu desktop. Clonada desde a máquina ubuntuserver

- **ubuntu:** Máquina com sistema operacional Ubuntu Desktop 17.04, utilizada como *host* de rede.

- **ubuntuserver:** Máquina que será utilizada para configuração do roteador de *backup*.

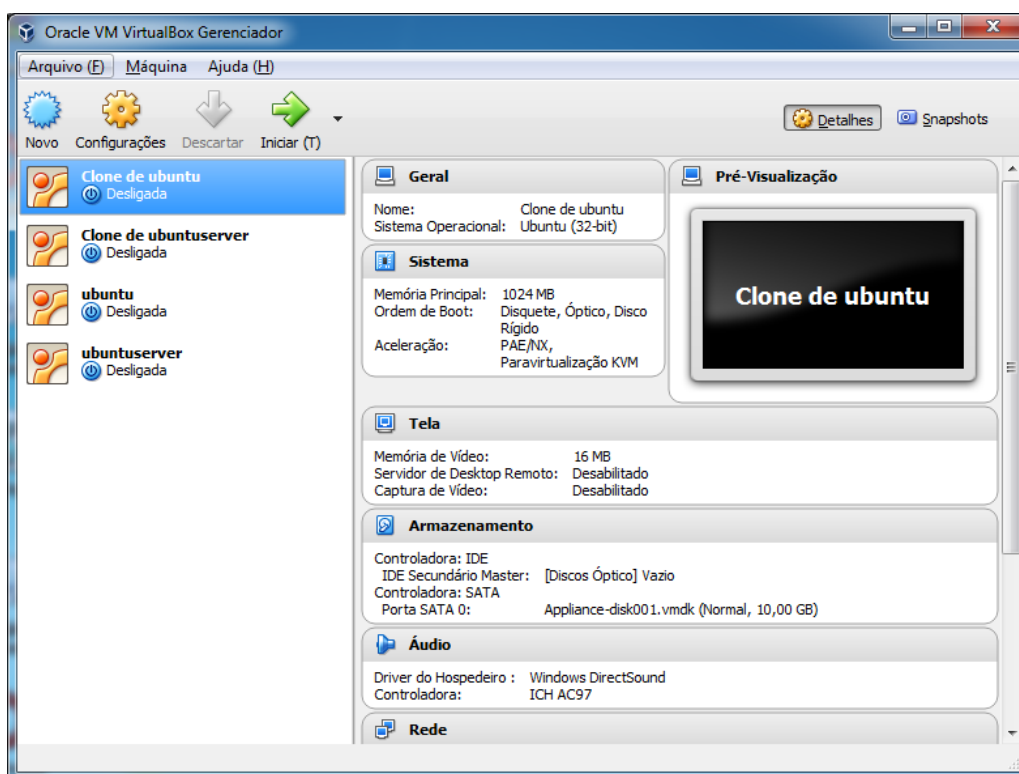
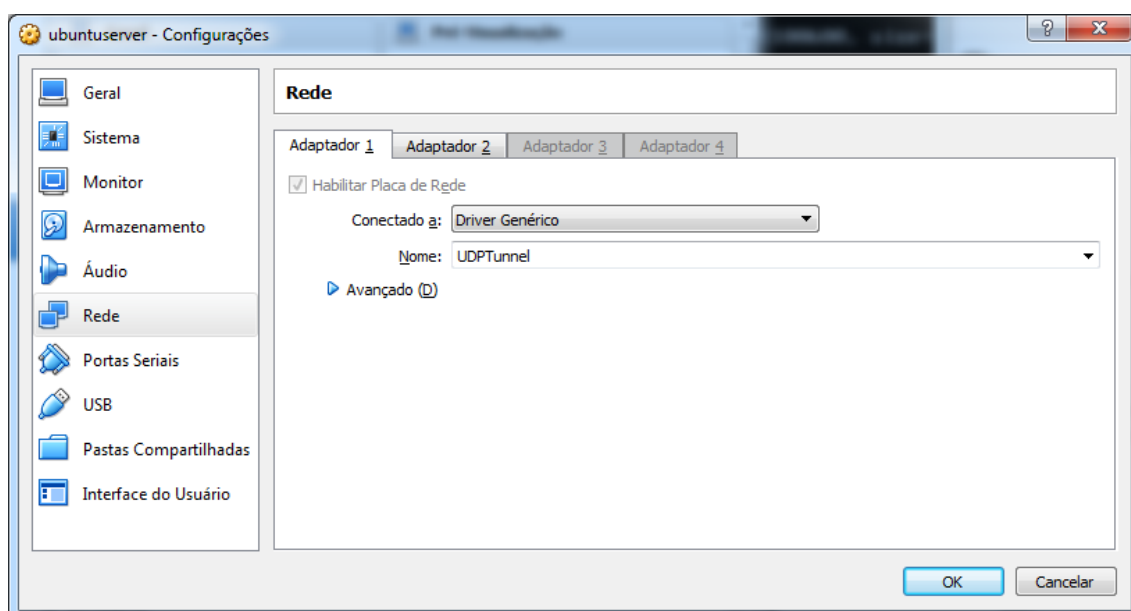


Figura 5: Oracle VirtualBox com máquinas virtuais instaladas

Fonte: Autoria própria



Na máquina `ubuntuserver`, foram adicionadas duas placas de rede, dentro de suas configurações, pois esta máquina deverá operar como um roteador, recebendo e enviando pacotes ethernet da rede local para a rede remota. Inicialmente, uma destas placas de rede foi configurada como “conectado a: NAT”, pois isto permite que a máquina virtual se conecte à mesma rede que o computador real, obtendo um endereço *IP* e permitindo conexão com a *internet*. Isto é necessário para obter os pacotes de aplicativos que serão utilizados no projeto, tais como: *Quagga* (pacote com protocolos de roteamento) e *VRRP* (protocolo para alta disponibilidade). A instalação e configuração destes pacotes de aplicativos serão detalhadas no capítulo 4.2. A figura 6 mostra detalhes da configuração das placas de rede da máquina `ubuntuserver`.



**Figura 6: Detalhes das configurações das placas de rede da máquina virtual `ubuntuserver`**

**Fonte: Autoria própria**

Na figura 6 é possível observar que existem duas guias habilitadas (Adaptador 1 e Adaptador 2), pois foram habilitadas apenas duas placas de rede nesta máquina virtual. Os campos para habilitar estão inacessíveis, pois a figura 6 foi obtida em um momento onde a máquina virtual estava sendo executada.

As configurações de rede das máquinas virtuais, bem como do roteador Cisco C7200 no ambiente do GNS3 serão detalhados no capítulo 4.2, pois as máquinas virtuais serão iniciadas pelos recursos do GNS3.

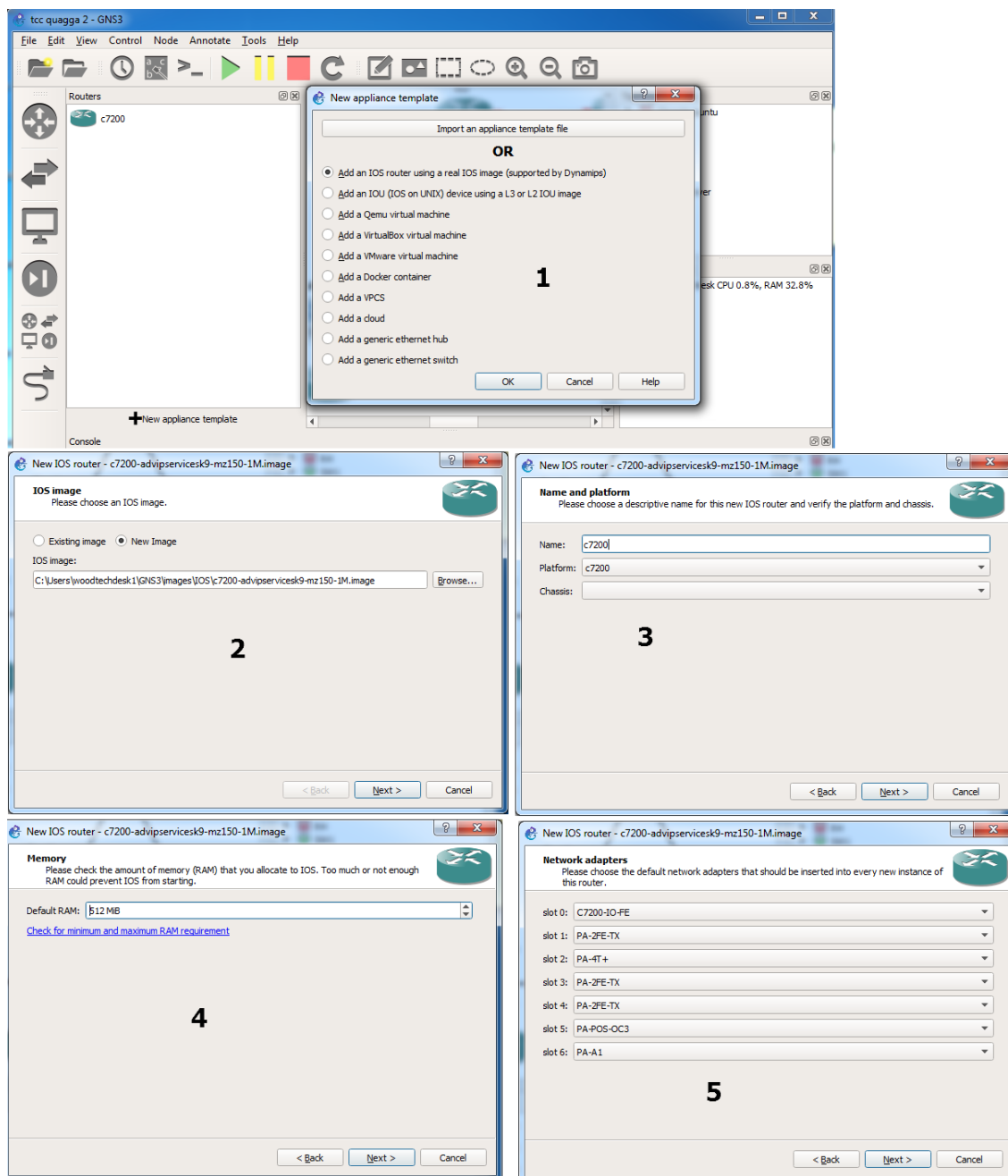
Para cada máquina virtual, foram destinados 1024MB de memória RAM e 10GB de espaço em disco. Durante os testes, isto mostrou-se suficiente para não comprometer o desempenho do computador onde as aplicações foram instaladas.

### 3.2 CONFIGURAÇÃO DA TOPOLOGIA NO GNS3

O instalador do GNS3 foi adquirido desde o site <https://www.gns3.com/software/download>.

Após instalado o GNS3, foi necessário adicionar o roteador C7200 através de uma imagem obtida desde um roteador real. Através da opção “*New Appliance template*”, seleciona-se “*Add an IOS router using a real IOS image (supported by Dynamips)*”. Após clicar em “OK”, seleciona-se o campo “*New Image*” e navega-se até a pasta do computador que contém a imagem do roteador através do botão “Browse”. Uma vez selecionada a imagem, deve-se clicar em “*Next*”, sendo possível renomear o roteador e selecionar a plataforma (modelo) a qual o roteador pertence. Clicando em “*Next*”, é possível selecionar a quantidade de memória *RAM* para o roteador e na próxima janela é possível adicionar adaptadores de rede. Clicando em “*Finish*” na próxima janela, finaliza-se a adição do roteador ao GNS3.

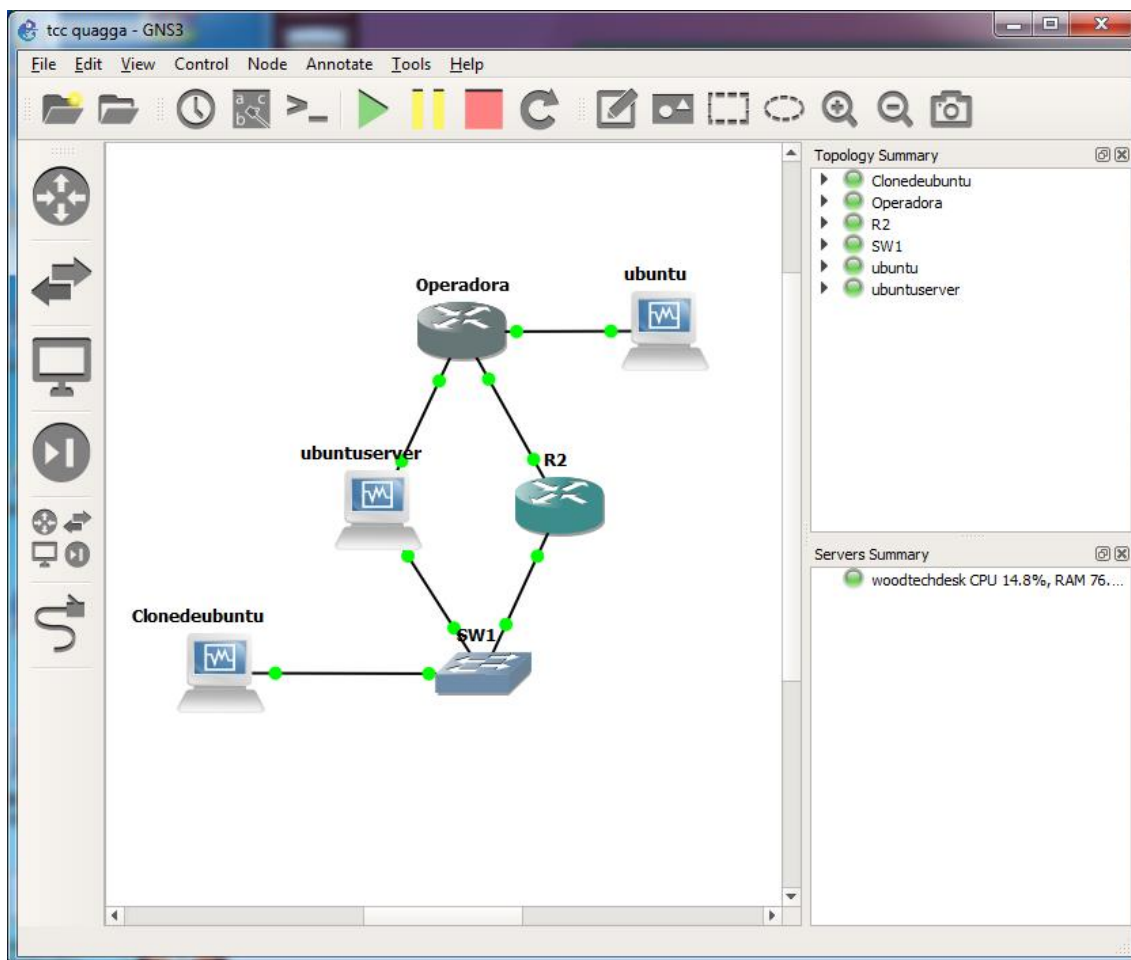
A figura 7 mostra as telas do GNS3 seguindo estas etapas:



**Figura 7: Adição de um roteador através de uma imagem do Cisco IOS no GNS3**

**Fonte: Autoria própria**

Uma vez adicionado o roteador, foi possível montar a topologia de testes para simular um ambiente com alta disponibilidade. O próprio GNS3 possui um *switch* genérico, não configurável, que serviu perfeitamente para implementar as redes locais. A figura 8 mostra o GNS3 com a topologia já montada.



**Figura 8: GNS3 com topologia de teste do ambiente de alta disponibilidade com roteador Cisco e Linux**

**Fonte: A autoria própria**

Os equipamentos foram configurados da seguinte maneira:

### 3.2.1 Clondeubuntu

Máquina que representa um *host* da rede local, que deverá acessar o *host* ubuntu pertencente a rede remota. Este equipamento foi configurado dentro da rede 192.168.1.0/24, com endereço IPv4 192;168.1.5/24, onde o *default gateway* corresponde ao endereço do *Gateway* virtual configurado dentro do grupo VRRP 1. Para efeitos de testes, esta foi a única configuração realizada nesta máquina virtual. A figura 9 mostra a configuração da placa de rede da máquina virtual **Clondeubuntu**.

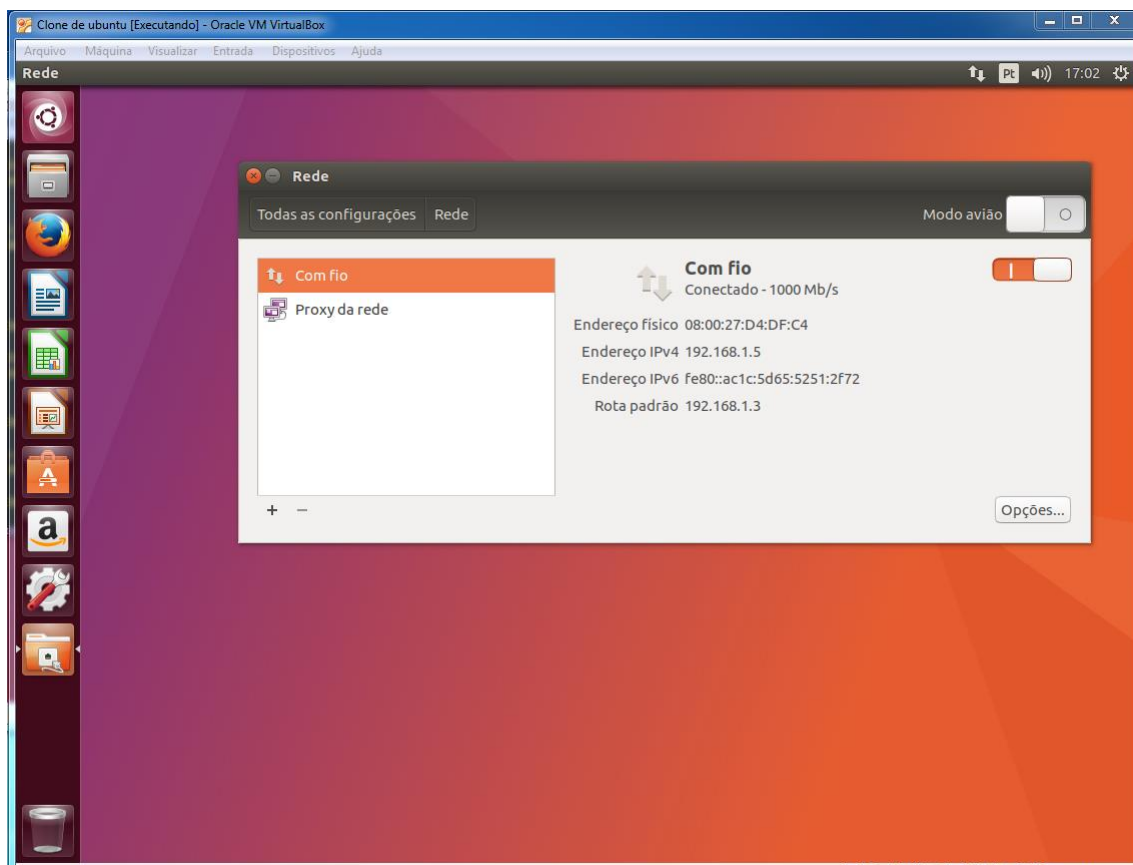


Figura 9: Detalhes da configuração da placa de rede da máquina virtual Clonedeubuntu

Fonte: Autoria própria

Para facilitar a visualização da rota percorrida pelos pacotes ICMP (ping) durante os testes, foi instalada nesta máquina a aplicação “*Traceroute*”, através do comando “*apt-get install traceroute*”. Após instalar esta aplicação, facilmente pode-se visualizar a rota percorrida pelo pacote, sendo possível diferenciar quando o roteador de *backup* assume o estado de *master*.

### 3.2.2 Ubuntu

Máquina virtual que representará um *host* remoto. Seu único objetivo é ser alcançado pela rede local da máquina **Clonedeubuntu**, tanto pelo roteador R2 (Cisco C7200) quanto pelo roteador configurado na máquina **Ubuntuserver**, quando este assumir o estado de *Master*. Assim como na máquina virtual **Clonedeubuntu**, a única configuração específica feita para esta máquina é a configuração de rede, sendo configurada com o endereço IPv4 10.10.10.2/24,

com *default gateway* 10.10.10.1 (Roteador **Operadora**). A figura 10 mostra a configuração de rede da máquina **ubuntu**.

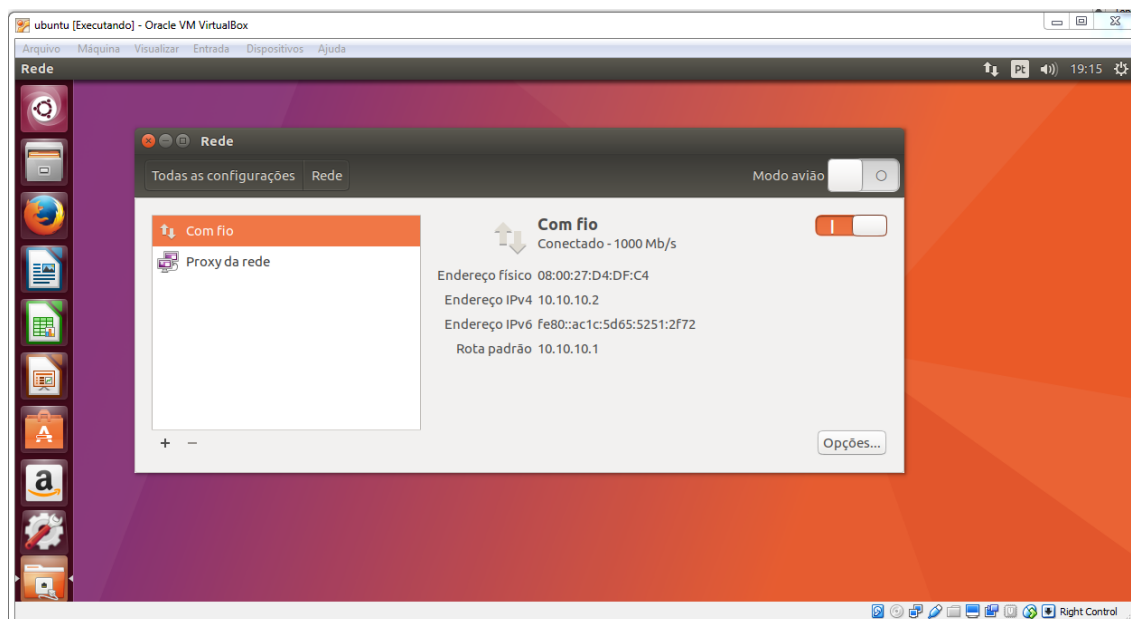


Figura 10: detalhes da configuração da placa de rede da máquina virtual ubuntu

Fonte: Autoria própria

### 3.2.3 R2 (Roteador Cisco C7200)

Este roteador terá a função de roteador principal dentro da topologia, pois em um ambiente real idêntico ao da topologia proposta, um roteador dedicado possui maior robustez do que um roteador construído dentro de um computador (como é o caso do roteador de *backup ubuntu server*).

O roteador R2 possui 2 portas *Fast Ethernet* conectadas dentro da topologia, onde F0/0 pertence a rede 192.168.1.0/24 (que corresponde à rede local) e F1/0 pertence a rede 192.168.0.0/24 (que está ligado diretamente ao roteador que representa a operadora de *internet*).

Neste roteador, foi configurado o protocolo VRRP na interface F0/0, para manter a alta disponibilidade na rede local 192.168.1.0 em caso de falha do roteador R2. A sintaxe da configuração deste protocolo ficou da seguinte forma:

**R2(config)#int f0/0**

**R2(config-if)# IP add 192.168.1.1 255.255.255.0**

**R2(config-if)#vrrp 1 description wan\_vrrp\_1** (descrição do processo VRRP)

**R2(config-if)#vrrp 1 IP 192.168.1.3** (*IP do gateway virtual*)

**R2(config-if)# vrrp 1 timers advertise 2** (tempo de envio de mensagens de advertência ao grupo vrrp 1)

**R2(config-if)# vrrp 1 priority 120** (determina a prioridade 120 para o roteador)

**R2(config-if)# vrrp 1 preempt** (comando que indica que o roteador deve assumir sua posição como *master* automaticamente após a falha ser restaurada).

Para a atualização automática das rotas entre as redes, foram configuradas duas áreas com o protocolo OSPF, sendo:

- Área 1: Rede local

- Área 0: Área de *Backbone*

A configuração do OSPF ficou da seguinte forma:

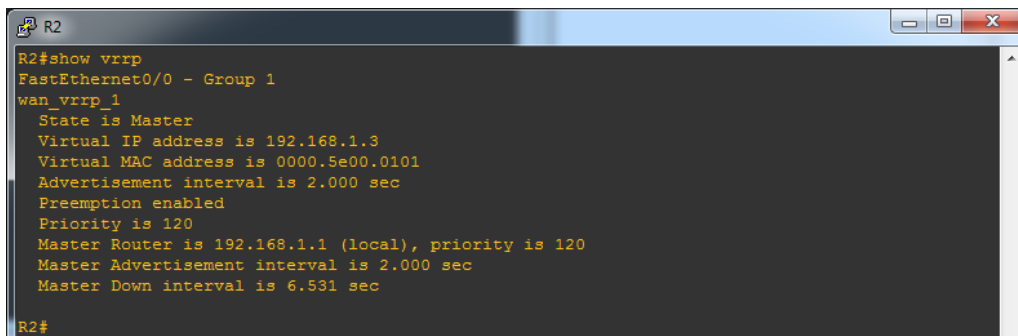
**R2(config)#router ospf1**

**R2(config-router)#router-id 1.1.1.1** (identificador do processo OSPF)

**R2(config-router)#network 192.168.1.0 0.0.0.255 area 1**

**R2(config-router)#network 192.168.0.0 0.0.0.255 area 0**

Supondo que o restante da rede esteja devidamente configurado, o resultado do comando **show vrrp** no roteador R2 mostra a seguinte saída, indicada na figura 11:

A terminal window titled 'R2' showing the output of the 'show vrrp' command. The output indicates that the router is in the 'Master' state for VRRP Group 1 on interface FastEthernet0/0. The virtual IP address is 192.168.1.3, and the virtual MAC address is 0000.5e00.0101. The advertisement interval is 2.000 seconds, and the master router is 192.168.1.1 (local) with a priority of 120. The master advertisement interval is 2.000 seconds, and the master down interval is 6.531 seconds.

```
R2#show vrrp
FastEthernet0/0 - Group 1
wan vrrp_1
  State is Master
  Virtual IP address is 192.168.1.3
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 2.000 sec
  Preemption enabled
  Priority is 120
  Master Router is 192.168.1.1 (local), priority is 120
  Master Advertisement interval is 2.000 sec
  Master Down interval is 6.531 sec
R2#
```

Figura 11: Saída do comando *show vrrp*, que mostra o roteador R2 como *master*

Fonte: Autoria própria

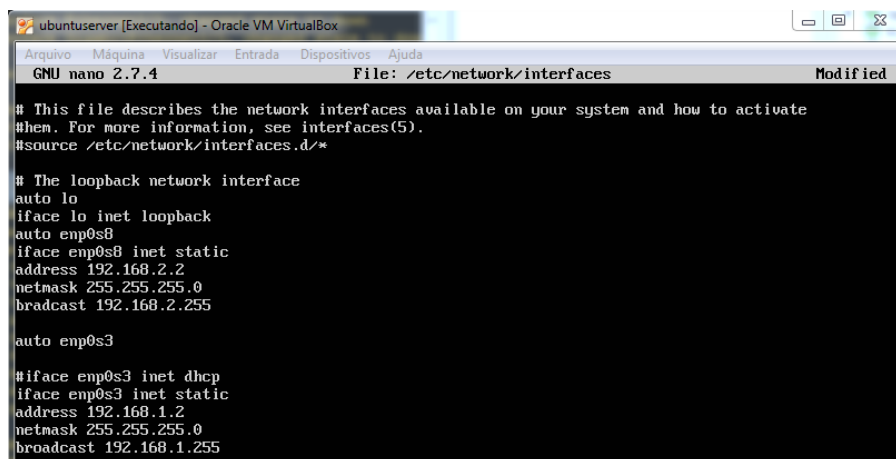
### 3.2.4 Ubuntu server:

A máquina virtual ubuntu server tem como objetivo ser um roteador de *backup* dentro do *cluster* de alta disponibilidade utilizando o protocolo VRRP. Foram criadas duas interfaces de redes virtuais, para conectividade com a rede local e remota.

Após a instalação do sistema operacional Ubuntu server 17.04, foi necessário realizar as seguintes configurações:

**a) Configurar as interfaces de rede enp0s3 e enp0s8:** A interface enp0s3 pertence ao domínio do protocolo VRRP e nela foi configurado o *IP* 192.168.1.2/24. A interface de enp0s8 representa o *link* de *backup* com a operadora e nela foi configurado o *IP* 192.168.2.2/24. Para configuração destas interfaces, foi utilizado o comando **nano /etc/network/interfaces**, que permite editar o arquivo que contém as configurações das interfaces de rede. A figura 12 mostra a tela com a configuração completa das duas interfaces de rede:





```

ubuntuserver [Executando] - Oracle VM VirtualBox
GNU nano 2.7.4 File: /etc/network/interfaces Modified
# This file describes the network interfaces available on your system and how to activate
# them. For more information, see interfaces(5).
#source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
auto enp0s8
iface enp0s8 inet static
address 192.168.2.2
netmask 255.255.255.0
broadcast 192.168.2.255

auto enp0s3

#iface enp0s3 inet dhcp
iface enp0s3 inet static
address 192.168.1.2
netmask 255.255.255.0
broadcast 192.168.1.255

```

Figura 12: Configuração das interfaces de rede na máquina ubuntuserver

Fonte: Autoria própria

Após a configuração da rede, utilizou-se o comando `/etc/init.d/networking restart` para validar as alterações.

**b) Instalar e configurar o protocolo VRRP:** O protocolo VRRP pertence a um pacote de aplicativo que deve ser instalado no Linux. Este pacote foi obtido na *internet* através do endereço <http://www.off.net/~jme/vrrpd/vrrpd-current.tgz>, sendo utilizados os seguintes comandos para instalação deste pacote.

```
root@ubuntu:/ #cd /tmp
```

```
root@ubuntu:/ #wget http://www.off.net/~jme/vrrpd/vrrpd-current.tgz
```

```
root@ubuntu:/ # tar xvfz vrrpd-current.tgz
```

```
root@ubuntu:/ # cd vrrpd
```

```
root@ubuntu:/ # make # cp vrrpd /usr/bin/
```

```
root@ubuntu:/ # cp vrrpd.8 /usr/share/man/man.8/ (RUSSO, 2006)
```

Feita a instalação do VRRP, foi realizada sua configuração com a sintaxe `vrrpd <opções> <IP_virtual> &` (RUSSO, 2006), sendo que o comando para a

máquina ubuntuserver ficou da seguinte forma: **vrrpd -i enp0s3 -v 1 -p 110 -d 2 192.168.1.3 -n &**

**Onde:**

-i é a identificação da interface de rede (no caso enp0s3).

-v é identificação do grupo VRRP (deve ser igual ao configurado no Roteador R2).

-p é a prioridade no (onde um número maior possui maior prioridade – R2 foi configurado com 120, que é maior que 110 configurado em ubuntuserver).

-d é o intervalo de verificação, em segundos, sendo utilizado o valor “2” igual ao configurado em R2.

**c) Configurar o encaminhamento de pacotes entre as interfaces de rede:** Isto é essencial para transformar uma máquina com Linux em um roteador. Para habilitar o roteamento de pacotes, foi utilizado o comando **nano /etc/sysctl.conf** para editar o arquivo sysctl.conf, sendo necessário deixar o **net.ipv4.IP\_forward** com valor **=1** (BRITO, 2017).

**d) Instalação e configuração do pacote Quagga:** A aplicação *Quagga* para Linux possui alguns pacotes de protocolos de roteamento, sendo possível configurar, além de outros protocolos de roteamento, o protocolo OSPF (utilizado na topologia) para enviar atualizações de rotas para os roteadores vizinhos.

A instalação do pacote *Quagga* foi realizada através do comando **apt-get install quagga**.

Após instalar o *Quagga*, foram copiados os arquivos de exemplo do diretório **/usr/share/doc/quagga-core/examples/** para a pasta **/etc/quagga/** (BRITO, 2017). Estes arquivos de exemplo são criados automaticamente junto com a instalação do *Quagga*.

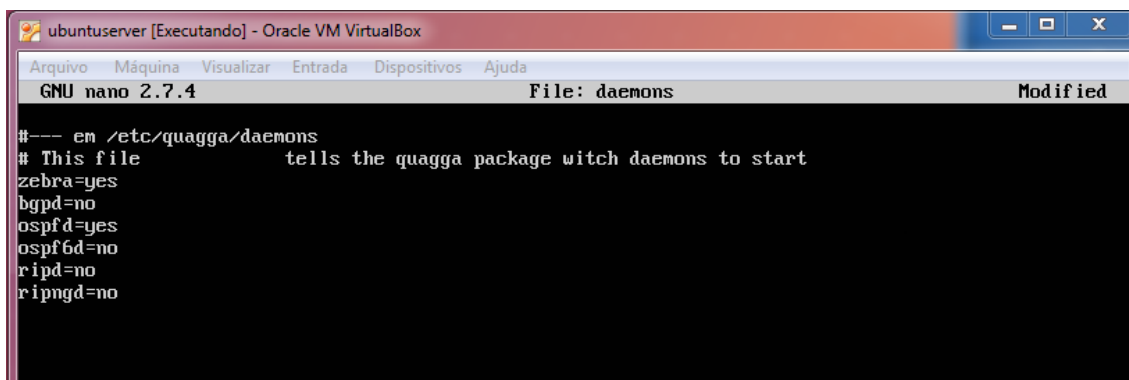
Dentro do diretório **/etc/quagga**, foram renomeados os arquivos de exemplo para cada protocolo para o nome padrão a ser utilizado pelo *Quagga*.

Foram utilizados os comandos para utilizar o protocolo OSPF e o gerenciador “zebra”:

```
mv zebra.conf.sample zebra.conf
```

```
mv ospfd.conf.sample ospfd.conf
```

No arquivo “*daemons*” contido dentro do diretório `/etc/quagga`, é necessário habilitar as aplicações que serão utilizadas, alterando o estado de “*no*” para “*yes*” (BRITO, 2017). A figura 13 mostra como ficou este arquivo com o protocolo OSPF e o gerenciador “zebra” habilitados.



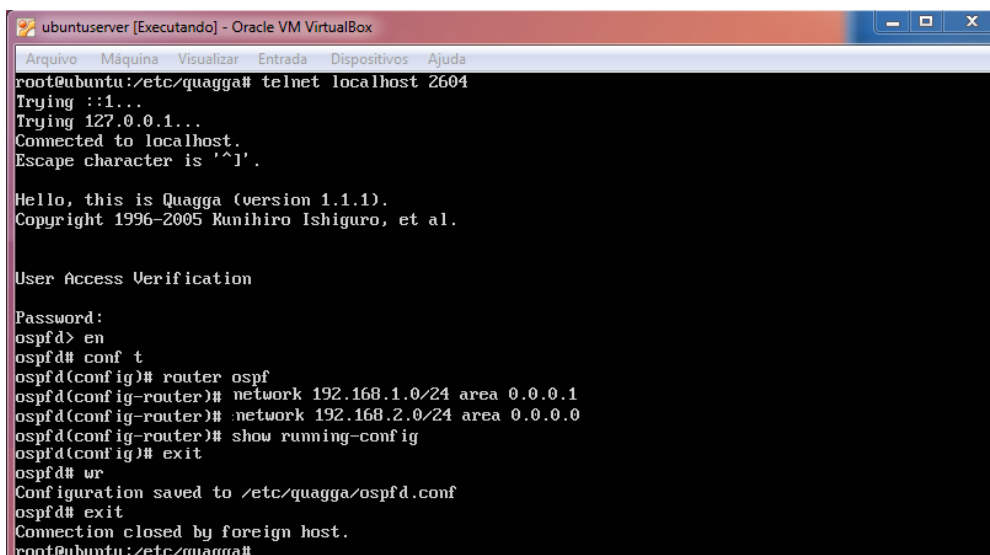
```
ubuntuserver [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
GNU nano 2.7.4      File: daemons      Modified
#--- em /etc/quagga/daemons
# This file      tells the quagga package witch daemons to start
zebra=yes
bgpd=no
ospfd=yes
ospfd=no
ripd=no
ripngd=no
```

Figura 13: Configuração dos *daemons* do Quagga

Fonte: Autoria própria

Após habilitar os *daemons* do *Quagga*, foi possível configurar o protocolo OSPF para anunciar as redes conectadas ao roteador `utuntuserver` e atualizar a tabela de roteamento de maneira dinâmica. Para ingressar na configuração do OSPF, utilizou-se o comando **telnet localhost 2604**. Desta forma, é possível ingressar por telnet na porta 2604 da máquina local, sendo esta porta utilizada pelo *daemon* OSPF. Ao ingressar este comando, é solicitada a senha, cujo padrão da instalação do pacote *Quagga* é “zebra”.

A figura 14 mostra a tela da configuração do OSPF no *Quagga* par a topologia proposta, onde serão anunciadas as redes 192.168.1.0 para a área 1 e 192.168.2.0 para a área 0.



```

ubuntuserver [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
root@ubuntu:/etc/quagga# telnet localhost 2604
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.

Hello, this is Quagga (version 1.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

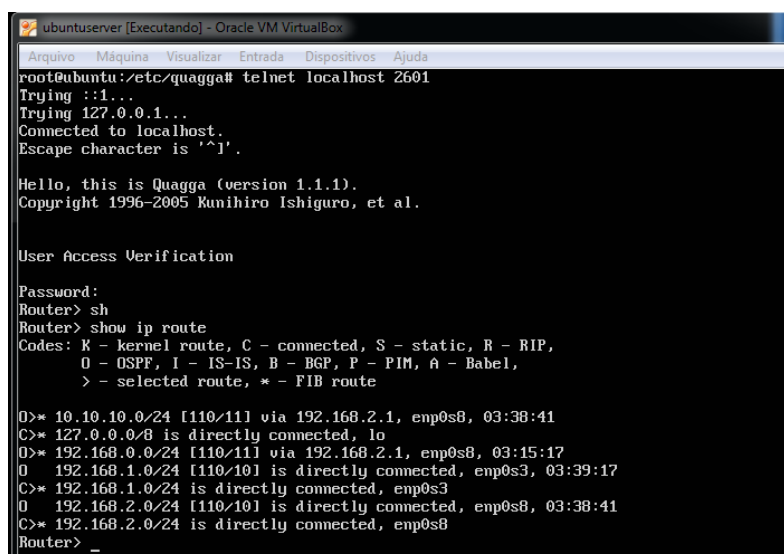
Password:
ospfd> en
ospfd# conf t
ospfd(config)# router ospf
ospfd(config-router)# network 192.168.1.0/24 area 0.0.0.1
ospfd(config-router)# network 192.168.2.0/24 area 0.0.0.0
ospfd(config-router)# show running-config
ospfd(config)# exit
ospfd# wr
Configuration saved to /etc/quagga/ospfd.conf
ospfd# exit
Connection closed by foreign host.
root@ubuntu:/etc/quagga#

```

Figura 14: Configuração do OSPF dentro do *Quagga*

Fonte: Autoria própria

Supondo que todos os outros roteadores da topologia proposta estejam devidamente configurados com o OSPF, pode-se ingressar no *daemon* zebra para verificar a tabela de roteamento. Para isto, utiliza-se o comando **telnet localhost 2601** e ingressando com a senha “zebra”. A figura 15 mostra os comandos utilizados após ingressar no *daemon* zebra e para verificar a tabela de roteamento.



```

ubuntuserver [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
root@ubuntu:/etc/quagga# telnet localhost 2601
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.

Hello, this is Quagga (version 1.1.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

User Access Verification

Password:
Router> sh
Router> show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, P - PIM, A - Babel,
       > - selected route, * - FIB route

O>* 10.10.10.0/24 [110/11] via 192.168.2.1, enp0s8, 03:38:41
C>* 127.0.0.0/8 is directly connected, lo
O>* 192.168.0.0/24 [110/11] via 192.168.2.1, enp0s8, 03:15:17
O  192.168.1.0/24 [110/101] is directly connected, enp0s3, 03:39:17
C>* 192.168.1.0/24 is directly connected, enp0s3
O  192.168.2.0/24 [110/101] is directly connected, enp0s8, 03:38:41
C>* 192.168.2.0/24 is directly connected, enp0s8
Router> _

```

Figura 15: Tabela de roteamento exibida pelo Quagga no roteador *ubuntuserver*

Fonte: Autoria própria

Os *daemons* do *Quagga* são executados automaticamente ao iniciar a máquina.

### 3.2.5 Operadora (Cisco C7200)

Este roteador terá a função de simular o *link* de *internet* para os roteadores R2 e ubuntu server.

O roteador Operadora possui 3 portas Fast Ethernet conectadas dentro da topologia, onde F0/0 pertence a rede 192.168.2.0/24 (que corresponde ao *link* para o roteador ubuntu server), F1/0 pertence a rede 192.168.0.0/24 (corresponde ao *link* para o roteador R2) e F1/1 que está configurada na rede 10.10.10.0 que representa a rede remota da topologia, cujo alvo é a máquina ubuntu.

O protocolo OSPF foi configurado neste roteador para atualizar a tabela de roteamento de maneira automática. A configuração do OSPF ficou da seguinte forma:

```
R2(config)#router ospf1
```

```
R2(config-router)#router-id 2.2.2.2 (identificador do processo OSPF)
```

```
R2(config-router)#network 10.10.10.0 0.0.0.3 area 0
```

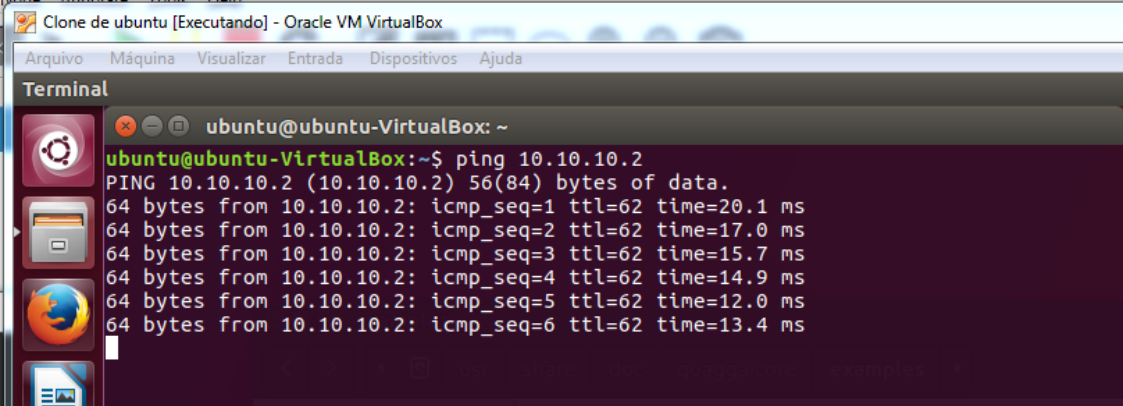
```
R2(config-router)#network 192.168.0.0 0.0.0.255 area 0
```

```
R2(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

## 4 ENSAIOS E RESULTADOS

Após concluída toda a configuração da topologia, foram realizados os testes para comprovar a funcionalidade da alta disponibilidade para a rede local, à qual pertence o *host* Clonedeubuntu.

Primeiramente, foi realizado um teste de ping para comprovar se o computador remoto ubuntu poderia ser acessado pelo computador Clonedeubuntu. A figura 16 mostra que existe conectividade entre os *hosts* das duas redes.



```
Clone de ubuntu [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Terminal
ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox:~$ ping 10.10.10.2
PING 10.10.10.2 (10.10.10.2) 56(84) bytes of data:
64 bytes from 10.10.10.2: icmp_seq=1 ttl=62 time=20.1 ms
64 bytes from 10.10.10.2: icmp_seq=2 ttl=62 time=17.0 ms
64 bytes from 10.10.10.2: icmp_seq=3 ttl=62 time=15.7 ms
64 bytes from 10.10.10.2: icmp_seq=4 ttl=62 time=14.9 ms
64 bytes from 10.10.10.2: icmp_seq=5 ttl=62 time=12.0 ms
64 bytes from 10.10.10.2: icmp_seq=6 ttl=62 time=13.4 ms
```

Figura 16: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota)

Fonte: Aatoria própria

O segundo teste a ser realizado foi desativar a interface F0/0 do roteador R2. Esta interface, em funcionamento normal, assume o *IP* virtual 192.168.1.3, que é o *gateway* do grupo VRRP1 da rede 192.168.1.0, como já mostrado na figura 11. Ao desativar esta interface, simula-se uma falha no roteador R2, sendo que após o tempo de advertência de 2 segundos, o roteador Linux *ubuntuserver*, através da interface *enp0s3*, deve assumir o papel do *gateway* virtual. A figura 17 mostra a tela do roteador R2 com os comandos utilizados para desabilitar a interface F0/0, enquanto o *host* *clonedeubuntu* permanece com conectividade ao *host* remoto *ubuntu*.

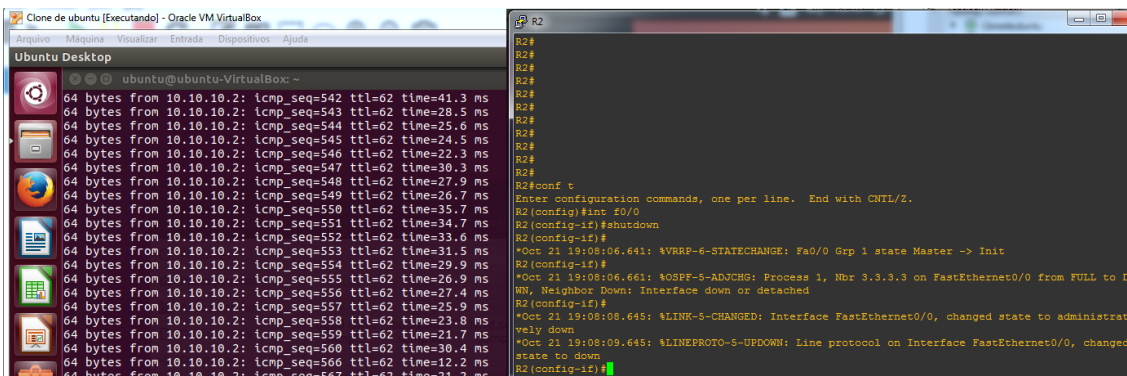


Figura 17: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota) após desabilitar F0/0 de R2

Fonte: Autoria própria

Já a figura 18 mostra o resultado do comando `IP -4 addr ls` no roteador Linux ubuntuuser, em dois momentos distintos (antes e depois de desabilitar a interface F0/0 em R2).

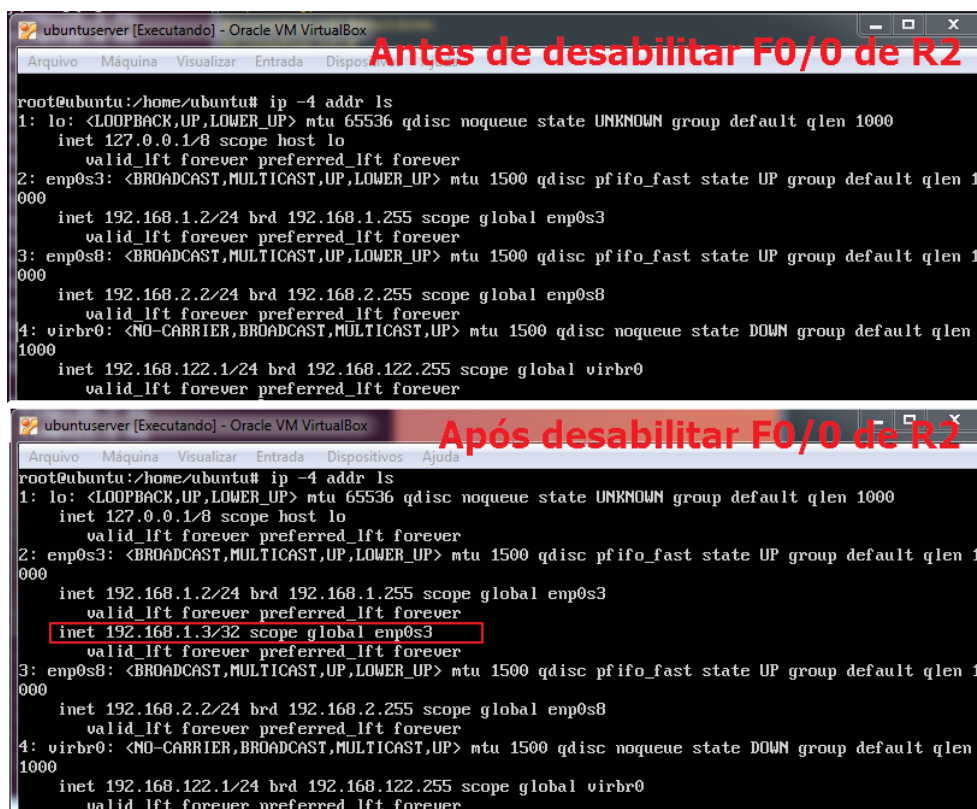
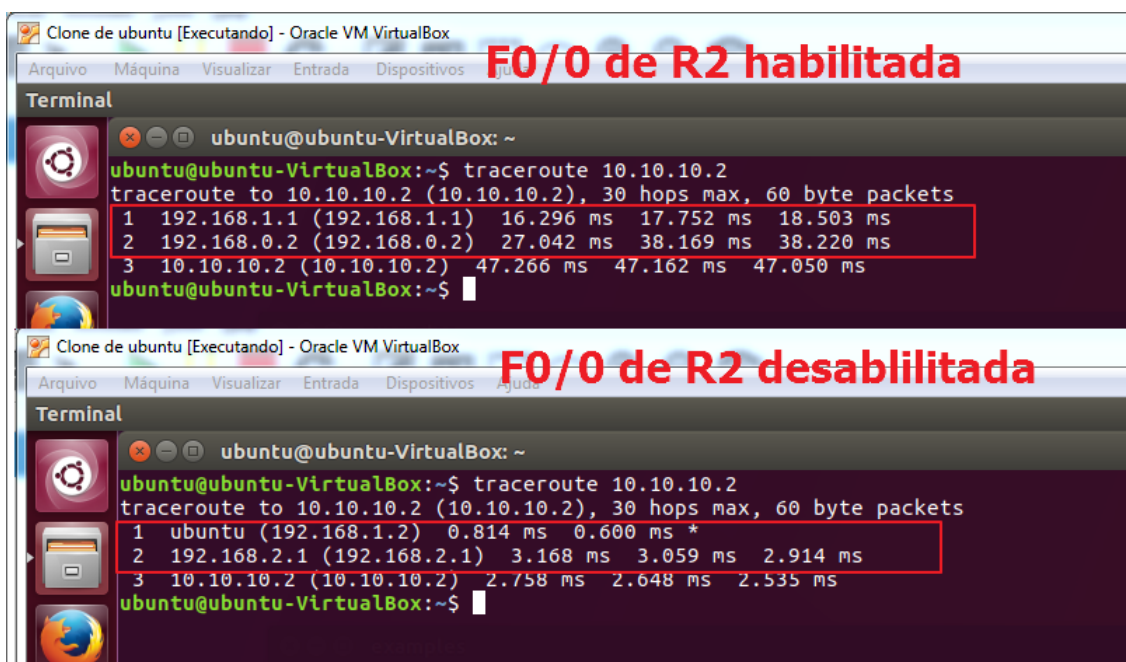


Figura 18: Teste de ping entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota), antes e após desabilitar F0/0 de R2

Fonte: Autoria própria

A figura 18 também mostra em destaque que a interface `enp0s3` passou a assumir também o *IP* do *gateway* virtual após a falha na interface `F0/0` de `R2`.

O teste de *traceroute* executado no *host* `clonedeubuntu` também é capaz de mostrar que o caminho percorrido pelos pacotes se altera após o roteador *master* apresentar falha. A figura 19 mostra o resultado do comando *traceroute* `10.10.10.2` executado em dois momentos distintos na máquina `clonedeubuntu`, sendo antes e depois de desabilitar a interface `F0/0` em `R2`. Antes de desabilitar `F0/0` de `R2`, o primeiro salto do pacote correspondia ao *IP* `192.168.1.1`, pertencente à interface `F0/0`. Após desabilitar `F0/0` de `R2`, o primeiro salto se altera para o *IP* `192.168.1.2`, pertencente à interface `enp0s3` do roteador Linux `ubuntuserver`.



**Figura 19: Traceroute entre as máquinas Clonedeubuntu (rede local) e ubuntu (rede remota), antes e após desabilitar `F0/0` de `R2`**

**Fonte: A autoria própria**

Outra possibilidade de falha dentro da rede local, que não corresponde ao domínio VRRP, é a queda da interface `F1/0` do roteador `R2`. Neste caso, o roteador `R2` não deixa de ser o *master* dentro do domínio VRRP, mas identifica que não possui mais uma rota diretamente conectada à rede `192.168.0.0` (necessária para alcançar o roteador operadora) e automaticamente direciona o



tráfego para a porta `enp0s3` do roteador `ubntuserver`. O tempo para este redirecionamento corresponde ao tempo de *dead-interval* do protocolo OSPF (por padrão corresponde a 40 segundos) e é superior ao tempo de advertência do VRRP (2 segundos), mas ainda é uma opção melhor do que perder toda a conectividade com a rede remota.

A figura 20 mostra que o tráfego é redirecionado para o *IP* 192.168.1.2, que pertence à interface `enp0s3`.

```

R2
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)#int f1/0
R2(config-if)#shut
R2(config-if)#
*Oct 21 19:58:37.525: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet1/0 from FULL to DOWN, Neighbor Down: Interface down or detached
R2(config-if)#
*Oct 21 19:58:39.505: %LINK-5-CHANGED: Interface FastEthernet1/0, changed state to administratively down
*Oct 21 19:58:40.505: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to down
R2(config-if)#
R2(config-if)#
R2(confi

```

```

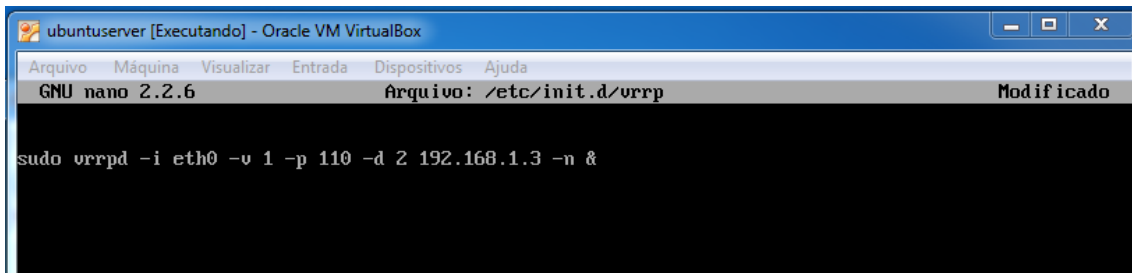
ubuntu@ubuntu-VirtualBox: ~
64 bytes from 10.10.10.2: icmp_seq=349 ttl=62 time=13.7 ms
64 bytes from 10.10.10.2: icmp_seq=350 ttl=62 time=22.2 ms
64 bytes from 10.10.10.2: icmp_seq=351 ttl=62 time=18.9 ms
64 bytes from 10.10.10.2: icmp_seq=352 ttl=62 time=18.2 ms
64 bytes from 10.10.10.2: icmp_seq=353 ttl=62 time=15.2 ms
64 bytes from 10.10.10.2: icmp_seq=354 ttl=62 time=15.6 ms
64 bytes from 10.10.10.2: icmp_seq=355 ttl=62 time=15.3 ms
64 bytes from 10.10.10.2: icmp_seq=356 ttl=62 time=17.3 ms
From 192.168.1.3: icmp_seq=357 Redirect Host(New nexthop: 192.168.1.2)
64 bytes from 10.10.10.2: icmp_seq=357 ttl=62 time=22.8 ms
64 bytes from 10.10.10.2: icmp_seq=358 ttl=62 time=21.3 ms
64 bytes from 10.10.10.2: icmp_seq=359 ttl=62 time=19.6 ms

```

**Figura 20: Tráfego redirecionado para interface `enp0s3` de `ubntuserver` após simulação de falha em `F1/0` de `R2`**

**Fonte: Autoria própria**

Por último, a máquina `ubntuserver` foi preparada para iniciar o serviço VRRP automaticamente. Para isto, foi adicionado um script chamado **VRRP** contendo o comando `sudo vrrpd -i enp0s3 -v 1 -p 110 -d 2 192.168.1.3 -n &` dentro do diretório `init.d` do Ubuntu, conforme mostra a figura 21.



```
ubuntuserver [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
GNU nano 2.2.6      Arquivo: /etc/init.d/vrrp      Modificado
sudo vrrpd -i eth0 -v 1 -p 110 -d 2 192.168.1.3 -n &
```

Figura 21: Script de inicialização do VRRP, dentro do diretório `init.d`

Fonte: A autoria própria

Para garantir que a máquina inicie o serviço juntamente com o sistema operacional, foi também necessário dar permissão para a execução do script **VRRP** com o comando **chmod 755 /etc/init.d/vrrp** e também atualizar a lista de inicialização, com o comando **update-rc.d vrrp defaults**.

## 5 CONCLUSÕES

Após finalizadas as configurações e testes, foi possível concluir que a implementação de um roteador dentro de uma máquina com sistema operacional Linux, operando em conjunto com um roteador Cisco e utilizando o protocolo VRRP para alta disponibilidade é perfeitamente possível e funcional. Os testes realizados demonstraram que a rede local 192.168.1.0 obteve conectividade com a rede remota mesmo após falhas no roteador principal R2 (Cisco C7200), através do redirecionamento automático do tráfego pelo protocolo VRRP.

A topologia simulada atende perfeitamente à proposta inicial do projeto, que é oferecer uma solução de baixo custo para redundância na conectividade com a *internet* para pequenas empresas que não dispõe de recursos para aquisição de dois roteadores Cisco.

Por não ser o objetivo do projeto, o seu desenvolvimento não aborda temas como por exemplo, o comparativo de desempenho entre os dois equipamentos. Obviamente, o roteador Cisco possui maior robustez para executar as atividades para as quais foi construído, sendo que o computador com sistema operacional Linux, além dos serviços de roteamento configurados neste projeto, possui inúmeros outros recursos que apesar de não serem relevantes para esta aplicação, permanecem ativos e consumindo processamento e memória do computador. No entanto, esta análise pode ser objeto de estudo para o próprio autor ou até mesmo de leitores interessados no assunto.

### 6.1 TRABALHOS FUTUROS

Este trabalho teve sua essência voltada para o funcionamento da alta disponibilidade do conjunto de um roteador construído dentro de um computador com sistema operacional Linux e um roteador Cisco. O resultado esperado foi atingido, mas a topologia montada ainda permite vários incrementos futuros, como por exemplo:

- Implementar balanceamento de cargas entre os dois *links* de *internet*, dividindo a banda ao invés de deixar o *link* de *backup* ocioso.

- Implementar regras e listas de acesso, tanto no roteador principal Cisco quanto no roteador Linux, através de serviços específicos, de forma que ambos sejam capazes de operar de maneira semelhante.

- Utilizar outros protocolos de roteamento dinâmico suportados pela aplicação *Quagga*.

## 6 REFERÊNCIAS

BRITO, Samuel Henrique Bucke. **Serviços de Redes em Servidores Linux**. Ed. Novatek, 2017.

DIAS, Diego. **VRRP**: Track baseado no estado de uma interface física. 10 jul. 2015. Disponível em <<http://www.comutadores.com.br/vrrp-track-baseado-no-estado-de-uma-interface-fisica/>>. Acesso em 21 set. 2017.

FILIPPETTI, Marco. **VRRP x HSRP x GLBP**. 16 dez. 2008. Disponível em <<http://blog.ccna.com.br/2008/12/16/pr-vrrp-x-hsrp-x-glbp/>>. Acesso em: 26 jul. 2017.

FOSTER, Michael. **Cisco CCNA Training: Learn to Manage Networks**. *E-Book*. 2011.

GNS3: Disponível em: < <https://www.gns3.com/>>. Acesso em: 22 de jul. 2017.

ISHIGURO, Kunihiro. **Quagga Routing Suite Manual**: Versão 1.2.0. Disponível em: <<http://www.nongnu.org/quagga/docs/docs-info.html#About-Quagga>>. Acesso em 28 set. 2017.

LAMMLE, Todd. **CCNA: Cisco Certified Cisco Certified Network Associate Study Guide**. 3. ed. Ed. Sybex, 2002.

LEWIS, Chris. **Cisco TCP/IP Routing Professional Reference**. Ed. Computing McGraw-Hill, 1999.

RFC2338. **Virtual Router Redundancy Protocol**. Disponível em <<http://www.ietf.org/rfc/rfc2338.txt>>. Acesso em: 20 set. de 2017.

RUSSO, Gilberto. **Firewalls redundantes utilizando VRRP**. 20 jun. 2006. Disponível em <<https://www.vivaolinux.com.br/artigo/Firewalls-redundantes-utilizando-VRRP?pagina=2>>. Acesso em: 26 jul. 2017.

TAVARES, Alexei C. **Simulador de Redes GNS3 para Certificações Cisco**. 22 mai. 2011. Disponível em: <<http://www.dltec.com.br/blog/cisco/simulador-de-redes-gns3/>>. Acesso em 3 out. 2017.

XAVIER, Fábio Correa. **Roteadores Cisco – Guia Básico de Configuração e Operação**. 2. ed. Ed. Novatek, 2010.