UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC

CURSO DE CIÊNCIA DA COMPUTAÇÃO

WILLIAN JEFFERSON MEURER

ESTUDO SOBRE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA UTILIZANDO O FREEBSD PFSENSE

CRICIÚMA

2016

WILLIAN JEFFERSON MEURER

ESTUDO SOBRE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA UTILIZANDO O FREEBSD PFSENSE

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel em Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof.Esp.Valter Blauth Junior

CRICIÚMA

2016

WILLIAN JEFFERSON MEURER

ESTUDO SOBRE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA UTILIZANDO O FREEBSD PFSENSE

Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Redes de Comunicações.

Criciúma, 29 de Novembro de 2016.

BANCA EXAMINADORA

Prof.Esp.ValterBlauth Junior - (UNESC) - Orientador

Prof. MSc. Paulo João Martins - (UNESC)

Prof. MSc. Rogério Antônio Casagrande- (UNESC)

AGRADECIMENTOS

A Deus por ter me dado saúde e força para superar as dificuldades.

A UNESC, seu corpo docente, direção e administração que oportunizaram uma educação de qualidade, sempre prezando pela ética e o respeito pelos seus alunos.

Ao meu orientador Prof.Esp.Valter Blauth Junior, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos.

A minha Esposa Carol e meus filhos Willian Jr. e Amabile, pelo amor, incentivo e apoio incondicional.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“A ciência de hoje é a tecnologia de amanhã.”

Edward Teller

RESUMO

No presente trabalho apresentam-se conceitos, ferramentas, configurações e técnicas para prover alta disponibilidade e balanceamento de carga da Internet através de um firewall. Destacam-se, também a importância da informação estar disponível para as organizações quando requisitadas. A Internet é uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo, serviços de armazenamento, manipulação de dados e aplicações vem sendo amplamente utilizados através da Internet. Este trabalho foi desenvolvido a partir de pesquisas bibliográficas, para que assim fosse possível a implementação prática, bem como os testes realizados. Foi utilizado na implementação, o sistema operacional PFsense, que é uma distribuição FreeBSD, que por sua vez é um sistema open source, também foi utilizado o CARP (Protocolo de redundância) e o PFsync (Protocolo de sincronismo) para implementação do cluster de alta disponibilidade, estas ferramentas são nativas do sistema operacional. Esta pesquisa mostra que é possível encontrar no software livre soluções em nível de software para criação de um ambiente de alta disponibilidade e balanceamento de carga de forma eficiente, sem a necessidade de aquisição de licenças de softwares.

Palavras-chave: Alta disponibilidade. *Clusters*. Carp. Balanceamento de carga

.PFsense

ABSTRACT

In this article introduced concepts, tools, configurations and techniques to provide high availability and load balancing of the Internet through a firewall. Also, the importance of information begins available to organizations when requested. The Internet is a computer network that interconnects thousands of computing devices around the world, storage services, data manipulation and applications has been widely used over the Internet. This work was developed from bibliographical research, so that it was possible the practical implementation, as well as the tests performed. It was used in the implementation, the PFsense operating system, which is a FreeBSD distribution, which in turn is an open source system, also the CARP (Common Address Redundancy Protocol) and PFsync (Synchronization Protocol) were used to implement the high cluster. These tools are native to the operating system. This research shows that it is possible to find in software free software-level solutions to create an environment of high availability and load balancing efficiently without the need to acquire software licenses.

Keywords: High Availability. Cluster. Carp. Load balancing .PFsense.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ADSL	*Asymmetric Digital Subscriber Line*
ARPA	*Advanced Research Projects Agency*
BSD	*Berkeley Software Distribution*
CARP	*Common Address Redundancy Protocol*
CPU	*Central Processing Unit*
DNA	*Desoxyribunucleic Acid*
GB	*Giga Bytes*
HD	*Hard Disk*
IBM	*International Business Machines*
ICMP	*Internet Control Message Protocol*
IEC	*International Electrotechnical Commission*
ISO	*International Organization of Standardization*
LAN	*Local Area Network*
MAC	*Media Access Control*
MAN	*Metropolitan Area Network*
NAT	*Network Address Translation*
OSI	*Open Systems Interconnection*
RAM	*Random Acess Memory*
SPOF	*Single Point of Failures*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
WAN	*Wide Area Network*
WEB	*World Wide* Web

SUMÁRIO

1 INTRODUÇÃO

A Internet é uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo. Há pouco tempo esses dispositivos eram basicamente *desktops* e servidores que armazenavam e transmitiam informações. A Internet de hoje é provavelmente o maior sistema de engenharia já criado pela humanidade, milhares de usuários conectam-se por meio de telefones celulares, e dispositivos como *webcams*, sensores, TVs e até mesmo máquinas de lavar vem sendo conectadas a essa rede (KUROSE, 2010). Serviços como armazenamento, processamento e recuperação de informações têm sido amplamente utilizados usando esse recurso, principalmente nas organizações que utilizam rede de computadores. A computação em nuvem tem mudado o paradigma de armazenamento e manipulação de dados e aplicações por todos os tipos de consumidores: o que antes era feito no *desktop* do cliente, atualmente tende a ser executado em uma nuvem (HAN; XU; YAO, 2009).

Para garantir que o acesso a esses serviços seja de forma segura e que não ocorram interrupções, visto que a indisponibilidade da Internet pode gerar prejuízos tanto financeiro quanto de imagem as organizações, uma solução possível é a utilização de um *firewall* baseado em *software* livre que tenha suporte a alta disponibilidade.

A alta disponibilidade caracteriza-se por um sistema projetado para impedir perda de um serviço por ele disponibilizado, reduzindo ou gerenciando falhas, diminuindo o tempo de desligamento planejado para manutenção (BRASIL, 2006).

Conforme estudo periódico que avalia a capacidade de acesso à Internet de mais de 200 países, realizado pela Akamai, com dados coletados no primeiro trimestre de 2015, o Brasil ocupa 89º lugar no *ranking* mundial e 8º na América Latina. Esta classificação se dá baseada na velocidade da Internet (AKAMAI, 2015).

Diante desse cenário faz-se necessário as organizações brasileiras buscarem tecnologias para gerenciar e otimizar esses recursos. A utilização do *software* livre é uma delas pelo fato de ter uma flexibilidade que permite a otimização de recursos, de acordo com cada necessidade. Também possui uma respeitável

estrutura para implementação de segurança, balanceamento de carga e tolerância a falhas.

Como a Internet pública e as *intranets* privadas cresceram e se transformaram de pequenas redes em grandes infraestruturas globais, a necessidade de gerenciar mais sistematicamente componentes de *hardware* e *software* dentro dessas redes tornou-se mais importante (KUROSE, 2010).

Sendo assim surge a oportunidade de realizar uma pesquisa com orientação acadêmica sobre alta disponibilidade e balanceamento de carga utilizando a distribuição FreeBSD PFsense para implementar esses serviços a fim de efetuar testes e analises, com intuito de verificar se a ferramenta será satisfatória para resolver estes problemas sem a necessidade de compra de licenças de *softwares*, ou equipamentos, como firewall ou roteadores. Também disponibilizar essa pesquisa para administradores de redes ou profissionais da área que buscam por soluções desse gênero.

1.1 OBJETIVO GERAL

Estudar o comportamento da alta disponibilidade e balanceamento de carga de um *firewall* em uma rede local.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

a) conceituar rede de computadores, *cluster* de alta disponibilidade, balanceamento de carga e *firewall*;
b) estudar *softwares* e protocolos usados para implantação de alta disponibilidade e balanceamento de carga;
c) implementar um ambiente de rede de computadores utilizando *firewall* de alta disponibilidade e balanceamento de carga;
d) testar a alta disponibilidade e balanceamento de carga no sistema implantado de *firewall.*

1.3 JUSTIFICATIVA

Com a crescente dependência da utilização da Internet nas empresas ou instituições, surgem vários problemas como de segurança e indisponibilidade das informações, através de computadores e dispositivos interconectados em rede. A indisponibilidade ou a falta de segurança pode acarretar sérios prejuízos às organizações. Sendo essa a razão da importância dessa pesquisa visto que segundo a norma ABNT NBR ISO/IEC 17799:2005A informação é um ativo importante para os negócios de uma organização e consequentemente necessita ser protegida, especialmente nos ambientes dos negócios, cada vez mais interconectados (ABNT, 2005).

A EMC Corporation realizou um estudo global no segundo semestre de 2014, que detalha as perdas financeiras sofridas em empresas, causadas por perda de dados e falta de *backups*. O estudo foi intitulado *Global Data Protection Index* (Índice Global de Proteção de Dados), o levantamento foi realizado em parceria com a empresa de pesquisa de mercado Vanson Bourne onde questionou 3,3 mil responsáveis por decisões de TI (Tecnologia da Informação), em médias e grandes empresas dos setores público e privado de 24 países.

No Brasil foram consultadas 125 organizações onde foi constatado que as empresas tiveram prejuízos superiores a US$ 26 bilhões (cerca de R$ 66 bi) devido à perda de dados e a tempo de indisponibilidade de seus ativos. Mundialmente, o montante foi de US$ 1,7 trilhão (R$ 4,3 trilhões).

Além das perdas financeiras, as interrupções causaram perda de produtividade dos funcionários e atraso nos desenvolvimentos de produtos e serviços. De acordo com a EMC, as interrupções podem ser causadas por uma série de problemas. Entre os principais estão a falha de hardware (61%), perda de energia (45% e um dos motivos mais apontados entre as empresas brasileiras) e falha de software (26%) (ROMER, 2014).

Existem vários *firewalls* ou *routers* (roteadores) proprietários disponíveis no mercado que oferecem suporte a alta disponibilidade, redundância de *link* e balanceamento de carga, porém com custos elevados.

A ferramenta escolhida para a implementação dessa pesquisa é o PFsense, que por sua vez é uma distribuição de código aberto de *firewall* baseado

em FreeBSD que fornece uma plataforma para roteamento e *firewall* flexível e poderosa (WILLIAMSON, 2011). É um sistema operacional customizado apenas com pacotes essenciais para serviços de *firewall e* roteamento, que possui características de menor consumo de recursos da CPU, possibilitando o reaproveitamento de máquinas já descartadas por serem obsoletas para outras funções.

Os serviços e protocolos que foram estudados nesse trabalho são nativos do sistema como o *Common Address Redundancy Protocol* (CARP) que é o protocolo responsável pelo sincronismo dos servidores conectados em alta disponibilidade, também o *Load Balance*, responsável por fazer o balanceamento de carga e *Failover* que da suporte a implementação do *cluster* de alta disponibilidade (FREEBSD, 2014).

O PFsense é completamente gerenciável pelos principais browsers e qualquer sistema operacional, também carrega em seu “DNA” a criação do protocolo mais utilizado no mundo da Internet o TCP/IP. Segundo Relatório Revisional Analítico FreeBSD Brasil Ltda. o sistema BSD tem, desde seus primórdios, a reconhecida fama de ter uma das pilhas TCP/IP mais robustas e com melhor *performance*, além de ter sido o sistema onde o TCP/IP foi projetado, criado e popularizado (FREEBSD, 2007), possibilitando assim obter-se melhor *performance* na filtragem de pacotes realizada por TCP/IP. Entretanto, mesmo que a ferramenta escolhida para estudo e implementação possua várias vantagens e recursos, ela por si só é ineficaz, e mal configurada pode trazer sérios prejuízos ocasionados por parada de serviços e falta de segurança. Por isso deve ser estudada, analisada a fim de obter um melhor aproveitamento dos recursos oferecidos sem comprometer a segurança.

1.4 ESTRUTURA DO TRABALHO

Neste trabalho foram, primeiramente, explanados assuntos pertinentes à sistemas distribuídos, redes de computadores, *cluster* de alta disponibilidade, disponibilidade, balanceamento de carga e *firewall*, da mesma forma, a estrutura e funcionamento dos padrões aos quais esta pesquisa se atém, a fim de obter-se uma noção para a implementação de um *firewall* de alta disponibilidade com suporte a

balanceamento de carga utilizando o sistema operacional PFsense (Capítulo 2 e 3). Logo em seguida, tomou-se lugar uma listagem de trabalhos correlatos (Capítulo 4). Já no capítulo 5 encontram-se os objetivos que o presente trabalho alcançou, bem como a metodologia utilizada. No capitulo 6, encontram-se os resultados e testes da implementação desta pesquisa. O capítulo 7 trata-se da conclusão deste trabalho.

2 REDE DE COMPUTADORES

As primeiras redes de computadores foram desenvolvidas a fim de realizar o compartilhamento de recursos como impressoras e servidores de arquivos, de modo que os computadores que estivessem presentes em uma mesma rede pudessem ter acesso a um mesmo dispositivo ou recurso que estivesse sendo compartilhado (COMER, 2007). De acordo com Coulouris (2007), as redes de computadores mudaram a forma com que as pessoas, empresas e instituições estão se comunicando. A responsável por essa mudança é a rede mundial de computadores a Internet.

Conforme o entendimento de Kurose (2010) e Coulouris (2007) pode-se definir a Internet como uma grande rede, formada pela interconexão de diversas outras redes, distribuídas mundialmente, podendo estas ser redes serem:

a) Local Area Networks (LANs), são redes privadas que atuam em um espaço geográfico limitado. Estas redes são responsáveis por permitirem o acesso dos usuários, bem como a realização do compartilhamento de recursos e a troca de informações (TANENBAUM, 2003);

b) *Wide Area Network* (WANs), são redes geograficamente distribuídas que abrangem um amplo espaço geográfico, são definidas como redes de longa distância. Tais redes possuem protocolos específicos e que os dados trafegam a uma velocidade de transmissão menor do que nas LANs em nível de usuário (MORAES, 2010);

c) *Metropolitan Area Network* (MANs), segundo Stallings (2005), são redes metropolitanas que servem de intermediário entre as redes LAN e WAN. As MANs são utilizadas em áreas metropolitanas por clientes que necessitam de maiores capacidades de transmissão, garantindo um menor custo e melhor eficiência.

2.1 MODELOS DE REFERÊNCIA

Conforme Kurose (2010), a Internet é um sistema extremamente complicado e que possui muitos componentes: inúmeras aplicações e protocolos, conexões entre roteadores além de vários tipos de meios físicos e de enlace, deve-se seguir os padrões e modelos de referência para envio e recebimento de dados.

O modelo de referência padrão para interconexão e endereçamento de rede é o *Transmission Control Protocol/Internet Protocol* (TCP/IP), a arquitetura TCP/IP é resultado de um projeto criado pela *Advanced Research Projects Agency* (ARPA), no ano de 1960, que tinha por finalidade desenvolver uma arquitetura de dados aberta, que fosse capaz de estabelecer a interligação das redes com dispositivos, mesmo que estes possuíssem uma arquitetura de *hardware* ou *software* diferentes (SOUSA, 2009).

De acordo com Sousa (2009), TCP/IP é dividido em quatro camadas que são elas:

a) camada física e enlace: Esta camada é responsável por enviar o datagrama recebido pela camada de Internet ou rede em forma de um quadro através da rede e transmite pelo meio físico, é constituída por *hardware*, possui as características físicas dos equipamentos;
b) camada de rede ou Internet: nesta camada ocorre a especificação do melhor caminho para os pacotes e os endereços lógicos tanto de origem como de destino;
c) camada de transporte: tem por objetivo realizar a conexão confiável, eficiente, e econômica dos dados entre a máquina de origem e a máquina de destino, independente do tipo, topologia ou configuração das redes físicas existentes entre elas, garantindo ainda que os dados cheguem sem erros e na sequência correta;
d) camada de aplicação: é responsável pela comunicação de aplicativos, é nesta camada onde estão os protocolos no qual são responsáveis pela comunicação destas aplicações.

A figura1 ilustra como fica a divisão das quatro camadas que compõem o modelo de referência TCP/IP.

Figura 1 – Modelo de referência TCP/IP

Camada 4	Aplicação	Comunicação de aplicativos
Camada 3	Transporte	Entrega
Camada 2	Internet	Endereço lógico e melhor caminho
Camada 1	Física	Transmissão binária e acesso aos meios

Fonte: Do Autor.

Como o TCP/IP possui um padrão para o envio e recebimento de dados, segundo Torres (2001), o *Open Systems Interconnection* (OSI) é um modelo de referência que foi desenvolvido pela *International Standards Organization* (ISO) com o objetivo de facilitar a interconexão das redes computadores, dado ao fato que as tecnologias desenvolvidas na época suportavam apenas as tecnologias do mesmo fabricante.

Conforme Tanenbaum (2003), o modelo de referencia OSI é composto de sete camadas, que são as seguintes:

a) camada física: trata-se da transmissão de *bits* brutos por um canal de comunicação;
b) camada de enlace de dados: é responsável por realizar a transformação de um canal de transmissão bruta em quadros, no qual serão adicionadas as informações necessárias para que possam continuar seu tráfego na rede;
c) camada de rede: é responsável por controlar a operação de redes, suas principais funções é o roteamento dos pacotes entre a fonte e o destino, é onde também ocorre a mudança do endereçamento dos pacotes, passando de lógico para físico;
d) camada de transporte: esta camada deve garantir a entrega de pacotes sem erros, no transmissor a camada de transporte é responsável de após o recebimento dos dados, separá-los em diferentes pacotes para o seu envio na rede, já no receptor esta camada é responsável pela montagem dos pacotes, para que retornem a sua forma original e repassá-los para a camada seguinte;
e) camada de sessão: a camada de sessão permite que os usuários de diferentes máquinas estabeleçam sessões entre eles. Uma sessão

oferece diversos serviços, inclusive o controle de diálogo (mantendo o controle de quem deve transmitir em cada momento);

f) camada de apresentação: a camada de apresentação esta relacionada a sintaxe e a semântica das informações transmitidas, permite comunicação entre computadores com diferentes representações de dados, esta camada gerencia estruturas de dados abstratas e permite a definição e o intercâmbio de estruturas de dados de nível mais alto;

g) camada de aplicação: esta camada encontra-se no topo da estrutura OSI, é responsável por realizar interconexão entre um determinado protocolo de comunicação e o aplicativo que irá receber tais dados.

A figura2 ilustra como fica a divisão das sete camadas que compõem o modelo de referência OSI.

Figura 2 – Camadas do modelo OSI

Camada 7	Aplicação	Processos de rede para aplicações
Camada 6	Apresentação	Representação de dados
Camada 5	Sessão	Comunicação entre hosts
Camada 4	Transporte	Conexões ponto a ponto
Camada 3	Rede	Endereço lógico e melhor caminho
Camada 2	Enlace	Acesso aos meios
Camada 1	Física	Transmissão binária

Fonte: Do autor.

A figura3 ilustra um comparativo entre os modelos de referências OSI e TCP/IP.

Figura3 – Modelo OSI x TCP/IP

	Modelo OSI	Modelo TCP/IP	
7	Aplicação	Aplicação	4
6	Apresentação		
5	Sessão		
4	Transporte	Transporte	3
3	Rede	Internet	2
2	Enlace	Física	1
1	Física		

Fonte: Do autor.

Uma reflexão a ser feita antes de começar a pensar em redes: um dos grandes erros em projetos de *clusters*, para qualquer que seja a sua finalidade, é achar que o processamento, ou alta capacidade de processamento, é baseado apenas em computadores, ou em processadores. Um *cluster* precisa ser visto como um organismo completo, cada peça tem sua importância e pode ser o responsável por um melhor desempenho do sistema globalmente. E é exatamente nesse aspecto que a rede de comunicações tem sua importância na hora de projetar o *cluster*. A rede pode se tornar um gargalo de desempenho para o *cluster*. A criação de camadas separadas para dados e para gerenciamento dos sistemas, a possível utilização de várias interfaces de rede, ou outras tecnologias de comunicação, precisam ser avaliadas a fim de obter-se o resultado desejado (BRASIL, 2006).

2.2 *CLUSTER*

O termo *cluster* começou a ser difundido na década de 60, quando a IBM interligou dois de seus *mainframes* para realizarem tarefas coordenadamente (PITANGA, 2008). Segundo Dantas (2005), as configurações de *cluster*, em termos computacionais podem ser entendidas como uma agregação de computadores de forma dedicada ou não. A partir desse princípio tem-se a ideia de Sistemas Distribuídos que para Tanenbaum (2003), consiste em computadores autônomos que trabalham juntos para dar a aparência de um único sistema coerente. Com a ideia inicial de se obter maior poder de processamento, um *cluster* pode ser formado

por milhares de unidades de processamento, onde se pode distribuir ou replicar a execução de um programa através dessas unidades de processamento (COULOURIS, 2007).

Yeo et al (2006) e Pitanga (2008), afirmam haver duas divisões distintas de *clusters,* os de alto desempenho, implementados para prover um alto poder de processamento; e os de alta disponibilidade, focados em suprir, por meio de redundância de equipamentos, eventuais falhas físicas em componentes do *cluster*, visando manter recursos e serviços disponíveis sempre.

Os objetivos principais para a formação de *clusters* são alta disponibilidade ou alta *performance.* Ainda é possível a situação onde o *cluster* deve atingir os dois objetivos juntos.

De acordo com o Guia de Estruturação e Administração do Ambiente de *cluster* e *grid* (Brasil, 2006), as vantagens técnicas de utilização de *cluster* e *grid* consistem em:

a) utilização de *hardwares comuns* no mercado em sistemas críticos através da transferência do gerenciamento das funções de alta disponibilidade, tolerância à falhas e balanceamento de carga do *hardware* para o software. Minimizando a necessidade de *hardware* especializado, aumentando a concorrência entre as empresas fornecedoras e propiciando a independência tecnológica de fornecedores de *hardware*;
b) em geral, as tecnologias de *cluster* e *grid* possuem padrões abertos e interoperáveis, facilitando a integração de sistemas baseados nestas tecnologias, já em sistemas em computação de grande porte que utilizam, em sua grande parte, tecnologias proprietárias e padrões fechados;
c) disponibilidade de soluções baseadas em *software* livre que permitem a implementação sem a necessidade de licenças de *software* permite também a melhoria, alteração, distribuição e compartilhamento de soluções, segurança, transparência e possibilidade de auditoria plena do sistema;
d) possibilidade de realizar o aproveitamento de “ciclos ociosos” de computadores já existentes na infraestrutura de TI atual.

2.2.1 *Cluster* de Alta Disponibilidade

Conforme Lopes Filho (2008), em sistemas de informações, pode-se dizer que o serviço proposto ficará sempre disponível independentemente da hora, do dia ou local, a alta disponibilidade tem por seu maior objetivo a continuidade do serviço sem interrupções por longos períodos de tempo.

Para Santos (2004), existem diferentes métodos e esquemas para prover a capacidade dos sistemas continuarem funcionando mesmo na ocorrência de falhas. O comportamento de *clusters* de alta disponibilidade deve contar basicamente com:

a) relógios sincronizados*:* é um requisito para algumas aplicações, pois a diferença entre os relógios dos nós pode levar a problemas de coordenação do *cluster*;
b) repositório estável*:* um sistema de armazenamento comum, com sua integridade do conteúdo preservada mesmo em caso de falhas, estável para as aplicações, sendo capaz de tolerar os tipos de falhas mais prováveis e comuns;
c) detecção de defeitos e diagnóstico de falhas: aplicações e métodos, em sua maioria, assumem que na ocorrência de falha em um nó, outros nós perceberão esta falha dentro de um período de tempo finito, e então executarão as medidas de reparo necessárias;
d) entrega confiável de mensagens: perdas de mensagens e erros em meios de comunicação podem ocorrer. Portando são necessários mecanismos para garantir que uma mensagem enviada por um nó chegue incorrupta ao seu destino e que a ordem das mensagens seja preservada entre dois nós.

Com a grande dependência da Internet nas organizações, a pressão sobre gestores de tecnologia da informação vem aumentando, pois se o sistema ou o serviço não estiver disponível, o negócio pode ser ameaçado e sofrer com perdas de produtividade, receita e bom relacionamento com o cliente. A alta disponibilidade é uma solução que pode combinar baixo investimento, tanto financeiro como de manutenção para empresas que necessitam implantar redes, sistemas e segurança. Ela auxilia na redução da perda de informações e da negação do serviço, tendo

como suas principais características a disponibilidade, o sistema tolerante a falhas e a redundância (SANTOS, 2004).

2.2.2 Disponibilidade

Segundo Brandão (2013), em informática a disponibilidade é entendida como o espaço de tempo em que estes serviços estão disponíveis, por exemplo, 12 horas por dia, 5 dias por semana. A disponibilidade pode ser calculada através de uma fórmula simples:

$$\mathbf{D} = \frac{\mathbf{TUP} - \mathbf{DT}}{\mathbf{TUP}}$$

Onde,

D é a Disponibilidade;**TUP** é o Total de Unidade de Tempo; **DT** é o *DownTime* (tempo parado).

Para exemplificar, considere-se o cálculo da disponibilidade de um equipamento (servidor) em um período de 1 ano, levando em consideração 3 horas de DT *(DowTime*) para manutenção.

TUP = 1 ano ou 8760 horas ***DowTime* = 2 horas**

Aplicando a fórmula, é obtido o seguinte resultado:

$$\mathbf{D} = \frac{\mathbf{8760} - \mathbf{2}}{\mathbf{8760}} = \mathbf{99,9771}\%$$

neste caso, tem-se uma grande disponibilidade do serviço, sendo que chegar aos 100 % é praticamente impossível por inúmeros fatores. Este tempo indisponível do equipamento é medido em porcentagem, então quanto mais noves (9), da direita para a esquerda no resultado da fórmula da disponibilidade, aparecer no cálculo, menor será o *DowTime*. Para Brandão (2013), isso representa “O significado dos noves” que estão indicados na tabela 1.

Tabela 1 – Significado dos Noves

%	Downtime (por ano)
100	Nenhum downtime
99.999 (5 noves)	Menos de 5.26 min.
99.99 (4 noves)	De 5.26 a 52 min.
99.9 (3 noves)	De 52 min. a 8 horas e 45 min.
99 (2 noves)	De 8 horas e 45 min. a 87 horas e 56 min.
90.0-98.9 (1 nove)	87 horas e 56 min. a 875 horas e 54 min.

Fonte: BRANDÃO (2013).

2.2.3 Redundância

A redundância de recursos é uma necessidade para se ter alta disponibilidade, do ponto de vista tecnológico, existe basicamente três alternativas para conseguir-se um maior nível de disponibilidade (SANTOS, 2004):

a) implementação de mecanismos de redundância e tolerância à falhas no *hardware*. Exemplos: Fontes Redundantes, Espelhamento de discos, Utilização de Tecnologias *HotSwap* para troca de componentes do servidor, entre outras;
b) implementação de mecanismos de tolerância à falhas via software. Esta opção consiste na utilização de um sistema de alta disponibilidade em software, capaz de gerenciar um conjunto de servidores, onde que se algum dos servidores vier a ter alguma falha a execução da aplicação que é controlada pelo sistema de alta disponibilidade não seja afetada. Exemplos destes sistemas são: HeartBeat, Mon,Carp, entre outros;
c) redundância via tempo: Uma técnica comum que consiste em executar certa instrução várias vezes, necessita de tempo extra para executar tarefas para tolerar falhas.

Os pontos únicos de falha ou *Single Point of Failures* (SPOF), são recursos que não podem falhar, pois todo o sistema depende deles. A redundância de recursos é um pré-requisito para se eliminar os SPOFs.

Os SPOFs não estão somente em servidores, em uma rede de computadores podem ser encontrados em: *switches, routers, hubs*, cabeamento,

energia elétrica e até mesmo no espaço físico que este por sua vez pode sofrer algum desastre, vindo a comprometer todo o sistema (SANTOS, 2004).

2.2.4 Balanceamento de Carga

Quando se projeta um sistema computacional, sabe-se a carga média e máxima que este irá suportar. Quando a carga de utilização do sistema começa a se tornar excessiva, é preciso buscar uma solução para o aumento de capacidade do sistema, como por exemplo:

a) aquisição de uma máquina ou serviço de maior capacidade computacional;
b) melhoria de *performance* do sistema;
c) utilização de sistemas de balanceamento de carga.

Os sistemas de balanceamento de carga são geralmente a repartição da execução do serviço por várias máquinas. Estas soluções podem se especializar em pequenos grupos sobre os quais se faz um balanceamento de carga: utilização da CPU, de armazenamento, ou de rede. Qualquer uma delas faz-se o uso do conceito de *clustering*, já que o balanceamento será, provavelmente, feito para vários servidores (BRASIL, 2006).

Um *cluster* de aplicação é um conjunto de servidores que respondem coletivamente por um serviço. Esse conjunto de servidores pode dividir entre si a carga do sistema, ou simplesmente aguardar um dos servidores falhar para assumir o serviço. Esta tecnologia tem sido muito utilizada na Internet, como em sites de portais, *e-commerce*, *e-business*, abrangendo desde situações onde é necessário tratar milhares de requisições simultâneas, até a demanda do serviço que exige 99.99% do tempo *on-line, por exemplo,* (BRASIL, 2006).

A alta disponibilidade estudada neste trabalho visou, através de redundância de recursos de *hardware* e *software*, tornar a Internet disponível o maior tempo possível e automatizando de forma transparente a recuperação de eventuais falhas.

O *firewall* de alta disponibilidade foi implementado por um *cluster* composto por dois computadores com os recursos de *hardware* necessários e os *softwares* devidamente instalados e configurados e dois *links* de Internet de

operadoras diferentes a fim de obter-se a alta disponibilidade não apenas nos servidores como também nos *links,* utilizando-se técnicas de balanceamento de carga nos mesmos com o objetivo de obter-se melhor desempenho quanto ao uso da Internet na rede interna.

2.3 *FIREWALL*

Para Tanenbaum (2003), os *firewalls* são apenas uma adaptação atual de uma antiga forma de segurança medieval: um fosso profundo em torno do castelo. Onde obrigava todos que quisessem entrar ou sair do castelo a passar por uma única ponte levadiça, onde eram revistados por guardas, e completa afirmando que nas redes, é possível usar o mesmo princípio: uma empresa pode ter muitas LANs conectadas, mas todo o tráfego de saída ou de entrada da empresa e feito através de uma ponte levadiça eletrônica (*firewall*).

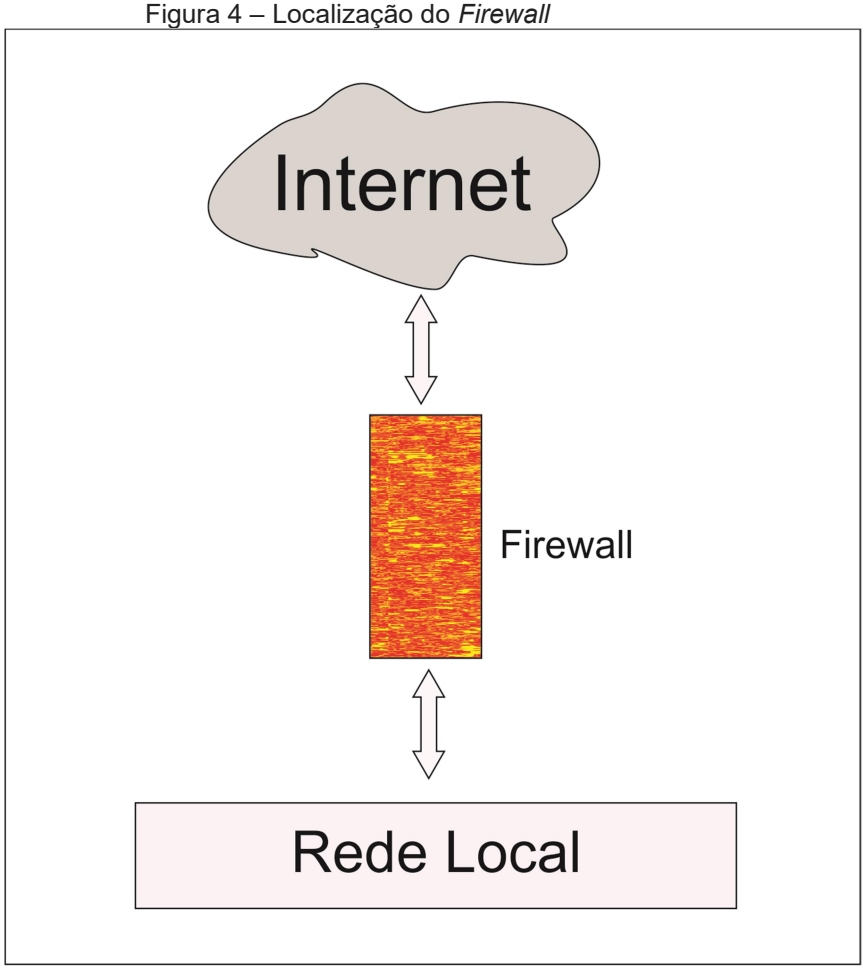
Separando-se a palavra nos termos *fire* e *wall*, tem-se como tradução mais próxima parede de fogo, conforme Cheswick (2005), este termo é definido como sendo um filtro de tráfego de rede, podendo ser um dispositivo, arranjo ou *software* que impõe limite para acessar a rede. Um *firewall* pode ser uma forma de defesa, mas devem ser adotados outros mecanismo e procedimentos para aumentar a segurança da rede a ser protegida. O *firewall* protege uma rede interna de ataques com origem de fora da rede, completa dizendo que este instrumento é importantíssimo para implementação da política de segurança, pois bem configurado torna mais custoso acessos não autorizados. Segundo Stallings (2008,), na maioria dos casos, o *firewall* está localizado entre a rede local e a rede externa, mas pode ser necessário o uso de *firewalls* internos, onde este tem o objetivo de isolar e proteger sub-redes umas das outras, para evitar ou conter a propagação de ataques.

Para a Cisco IOS Security (2006), a função mais básica de um *firewall* é monitorar e filtrar o tráfego.

Firewalls podem ser simples ou complexo, dependendo das necessidades da rede. *Firewalls* simples são geralmente mais fácil de configurar e gerenciar. No entanto, pode-se ser necessário maior flexibilidade, regras especificas e otimizaçao sendo necessário um *firewall* melhor elaborado.

A figura 4 ilustra a localização do *firewall* entre a Internet e a rede local.

Figura 4 – Localização do *Firewall*



Fonte: Do autor.

3 PFSENSE

PFsense é uma distribuição de código aberto de *firewall*, concebido em meados de setembro de 2004, por Chris Buechler e Scott Ullrich, com base no extinto m0n0wall (projeto que visava criar pacotes de *software firewall*, que quando utilizado juntamente com um PC, fornecia todas as características de um *firewall* embutido comercial), fundamentado no sistema operacional FreeBSD.

Fornece uma plataforma para roteamento e *firewall* flexível e poderosa. É um sistema operacional customizado apenas com pacotes essenciais para serviços de *firewall ou router*. Possui características de menor consumo de recursos da CPU, possibilitando o reaproveitamento de máquinas já descartadas por serem obsoletas para outras funções (WILLIAMSON, 2011).

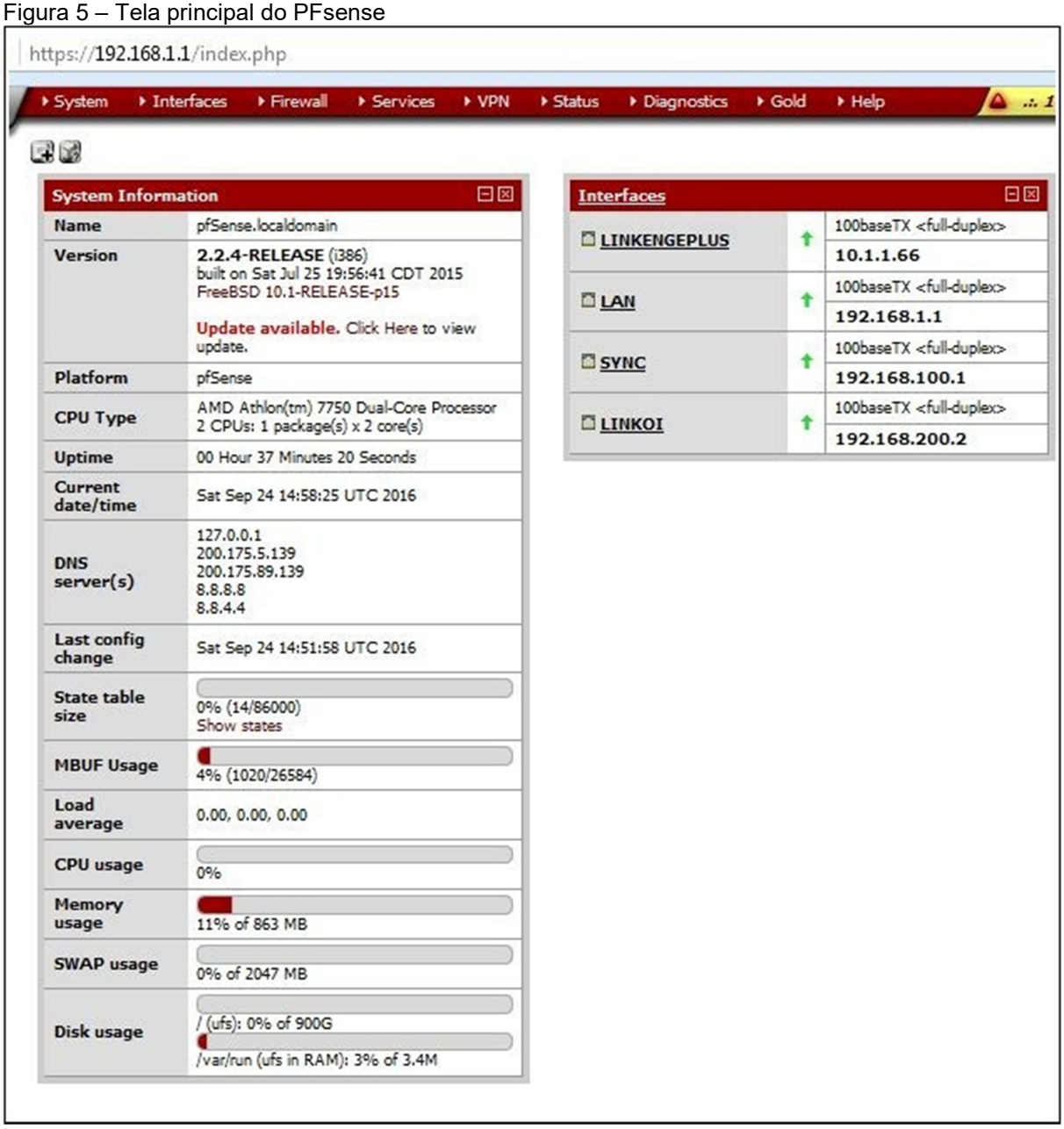
A utilização do sistema operacional *PFSense* tem como foco a segurança e administração da rede. É completamente gerenciável pelos principais *browsers* e qualquer sistema operacional (WILLIAMSON, 2011).

Segundo Relatório Revisional Analítico FreeBSD Brasil Ltda. o sistema BSD tem, desde seus primórdios a reconhecida fama de ter uma das pilhas TCP/IP mais robustas e com melhor *performance*, além de ter sido o sistema onde o TCP/IP foi projetado, criado e popularizado (FREEBSD, 2007), possibilitando assim obter-se melhor *performance* na filtragem de pacotes realizada por TCP/IP. Entretanto a ferramenta por si só ou mal configurada é ineficaz e pode comprometer a segurança, por isso a importância desse estudo.

Para utilizar a ferramenta de forma correta é indispensável obter conhecimentos sobre redes de computadores bem como protocolos e portas dos serviços de redes, é necessário também um estudo aprofundado sobre *firewall,* para assim obter excelência nos serviços implantados.

A figura 5 apresenta a tela principal do sistema *PFSense,* onde aparecem as informações do sistema, o tempo que o servidor está ligado, *status* das interfaces, *status* dos serviços sendo executados, consumo de CPU, consumo de memória, espaço em disco entre outros.

Figura 5 – Tela principal do PFsense



Fonte: FreeBsd

3.1 CARP

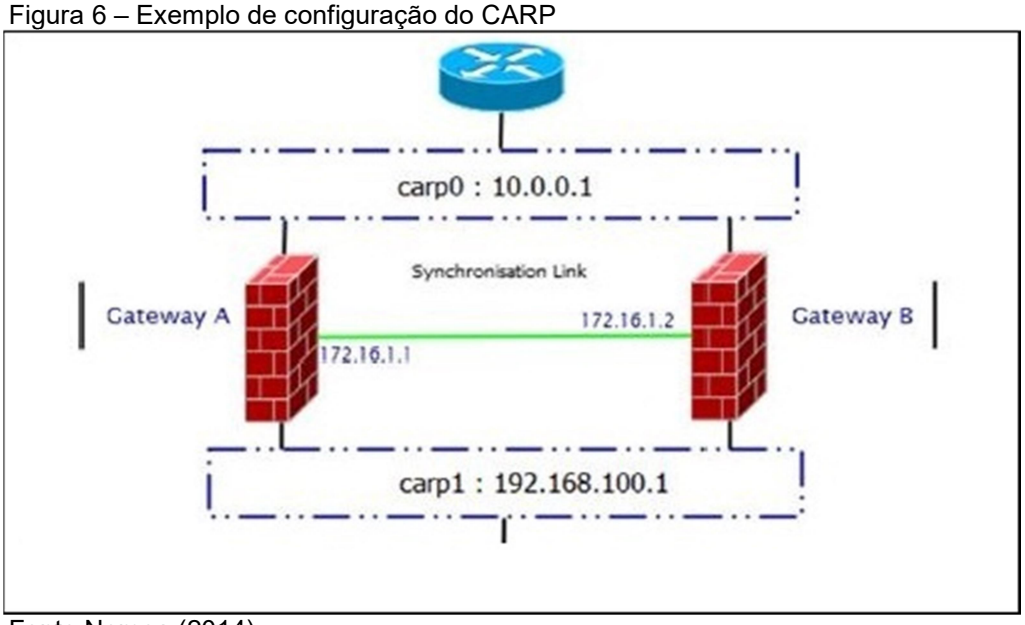
O Protocolo de Redundância de Endereço Comum ou *Common Address Redundancy Protocol* (CARP), com suporte para IPV6 ou IPV4, conforme o Manual *do Freebsd é* o responsável pelo sincronismo entre os servidores conectados em alta disponibilidade. Esta ferramenta tem por objetivo prover a redundância do sistema com vários computadores compartilhando em uma única interface virtual de

rede entre eles, garantindo o sincronismo de estado, a fim de obter-se a alta disponibilidade entre os servidores (FREEBSD, 2013).

A tecnologia tem como base um endereço IP flutuante entre os computadores que fazem parte do grupo. Um integrante do grupo é escolhido como mestre e fica respondendo todos os pacotes destinados ao grupo, os outros ficam aguardando para serem substitutos automáticos do mestre, não importando o endereço IP ou endereço MAC da interface física. O mestre utiliza o IP para comunicar a sua situação ao grupo e a sua frequência é zero, sendo esse o valor de maior prioridade. Caso estiver indisponível por um determinado período de tempo, todos do grupo são comunicados e a máquina que está programada com um valor de frequência menor e acima de zero torna-se o novo mestre. Quando o mestre volta a ser ativo na rede, tornar-se novamente assumir como mestre, retomando todas as funcionalidades, desta forma, os clientes da rede não são afetados e os serviços continuam disponíveis (FREEBSD, 2013).

A figura 6 exemplifica uma configuração, onde um *gateway* com redundância faz a separação da rede externa da rede interna. Uma placa de rede dos *gateways* está com o IP 10.0.0.1 e com o nome de “carp0” e faz a ligação dos *gateways* com a rede externa, a segunda placa de rede dos *gateways* com o IP 192.168.100.1 e de nome “carp1”, sendo responsável pela comunicação com a rede interna. A terceira placa de rede dos *gateways* com o IP 172.16.1.1/172.16.1.2 faz as consultas e a sincronização entre os *gateways*.

Figura 6 – Exemplo de configuração do CARP



Fonte:Nomoa (2014).

4 TRABALHOS CORRELATOS

Os trabalhos mostrados a seguir estão relacionados com o estudo e implementação de alta disponibilidade, podendo esta ser utilizada praticamente em todas as áreas da computação.

4.1 ALAVANCANDO OS CONTAINERS DO LINUX DE ALTA DISPONIBILIDADE PARA SERVIÇOS EM NUVEM

Abdelouahed Gherbi, Ali Kanso e Wubin Li (2015), abordam na Conferência Internacional de Engenharia de Nuvem realizada na cidade de Tempe no estado do Arizona EUA, a utilizaçao do Linux para alcançar alta disponibilidade para aplicações em nuvem, com a utilizaçao de um middleware que é constituído por um conjunto de agentes de alta disponibilidadeonde este, faz a mediaçao entre outros softwares na realização da alta disponibilidade.

A abordagem começa com a modelagemdo nível da aplicação, considerando as dependências entre os componentes do aplicativo. Após isso gerou-se o esquema de escalonamento adequado e, em seguida, implantou-se o aplicativo com repositório na nuvem.

Para cada repositório que contém componente crítico, seja monitorado continuamente o seu estado, e em seguida, reagir a uma fallha, reiniciando localmente ou via failover para outro host a fim de retomar o serviço mais rapido possivel e de um estado mais recente. Ao usar esta estratégia, todos os componentes hospedados em um repositorio são preservados sem haja modificação no lado da aplicação (GHERBI; LI; KANSON, 2015).

4.2 OTIMIZAÇÃO DE *PERFORMANCE* DE SISTEMAS *WEB* ATRAVÉS DE TÉCNICAS DE CLUSTERIZAÇÃO

Rafael Boufleuer (2013) abordou em seu trabalho de conclusão de curso na Universidade Federal de Santa Maria em Santa Maria RS, a aplicação de técnicas de clusterização em sistemas *Web* desenvolvidos na linguagem de programação Java.

Os objetivos buscados pelo trabalho foram: melhorar a *performance* destes sistemas, proporcionar uma estrutura de alta disponibilidade e também otimizar os recursos computacionais existentes. Todos os estudos, implementações e testes foram efetuados utilizando o Sistema Integrado de Gestão Acadêmica da Educação.

A estrutura do *cluster* implementada neste trabalho proporcionou a alta disponibilidade através do balanceamento de carga entre os nós. Além disso, com a falha de algum nó escravo, o usuário conectado é redirecionado automaticamente para outro nó do *cluster* sem perder a conectividade e sem a necessidade de fazer o *login* novamente. Com isso, consegue-se um sistema mais robusto, confiável e sempre disponível (BOUFLEUER, 2013).

4.3 IMPLEMENTAÇÃO DE *FIREWALL* DE ALTA DISPONIBILIDADE UTILIZANDO O *SOFTWARE* HEARTBEAT

Os acadêmicos Jorge Luiz Borges Correa Junior e Shirley Mirian Silva da Silva (2013),apresentaram no trabalho de conclusão de curso no Instituto Federal de Santa Catarina, na cidade de Sombrio SC, um estudo e implementação de um *firewall* de alta disponibilidade fazendo o uso do software Hearbeat.

Para alcançar os objetivos, o trabalho foi elaborado e desenvolvido com base na pesquisa bibliográfica, aplicada e experimental, e através do uso destes métodos de pesquisa foi possível realizar o estudo necessário e montar o respectivo cenário.

Para a implementação do ambiente de alta disponibilidade, foram utilizados dois computadores, na qual um responde por *firewall* ativo (*o* primário*)* e o outro é determinado como *firewall* passivo (o *backup),* e em ambos foram instalados o *software* Heartbeat. A implementação foi realizada em computadores reais em laboratório cedido pelo IFSC-Sombrio. Após a realização de testes, foi possível validar a implementação do *firewall* de alta disponibilidade, o qual correspondeu satisfatoriamente, cumprindo requisitos necessários para tal denominação (CORRÊA JUNIOR; SILVA, 2013).

4.4 IMPLEMENTAÇÃO DE UM AMBIENTE DE ALTA DISPONIBILIDADE PARA SERVIÇOS DE VOZ SOBRE IP BASEADOS EM ASTERISK

Em 2008 o acadêmico Fernando Fraga Rodrigues em seu trabalho de conclusão de curso, realizado na Universidade do Extremo Sul Catarinense (UNESC), na cidade de Criciúma SC, realizou um estudo e implementação de um ambiente de alta disponibilidade para serviços de voz sobre IP baseados em Asterisk.

O trabalho apresentou os testes realizados em dois ambientes, sendo o primeiro desprovido de qualquer dispositivo ou arquitetura que vise a alta disponibilidade e o segundo com o Asterisk sendo executado em um failover *cluster*.

Os ambientes foram testados por meio de simulações de falhas reais que exigissem a completa restauração do hardware do servidor bem como a comparação dos registros dos tempos necessários a sua completa restauração em ambos os cenários apontando a viabilidade do uso de *clusters* em ambientes de voz sobre IP baseados em Asterisk (RODRIGUES, 2008).

5 TRABALHO DESENVOLVIDO

A alta disponibilidade estudada neste trabalho visou, por meio de redundância de recursos de *hardware* e *software*, tornar um *firewall* e a Internet disponíveis o maior tempo possível em uma rede local, recuperando de forma transparente eventuais falhas. Também teve como objetivo o uso de técnicas de balanceamento de carga para o consumo de Internet na rede interna.

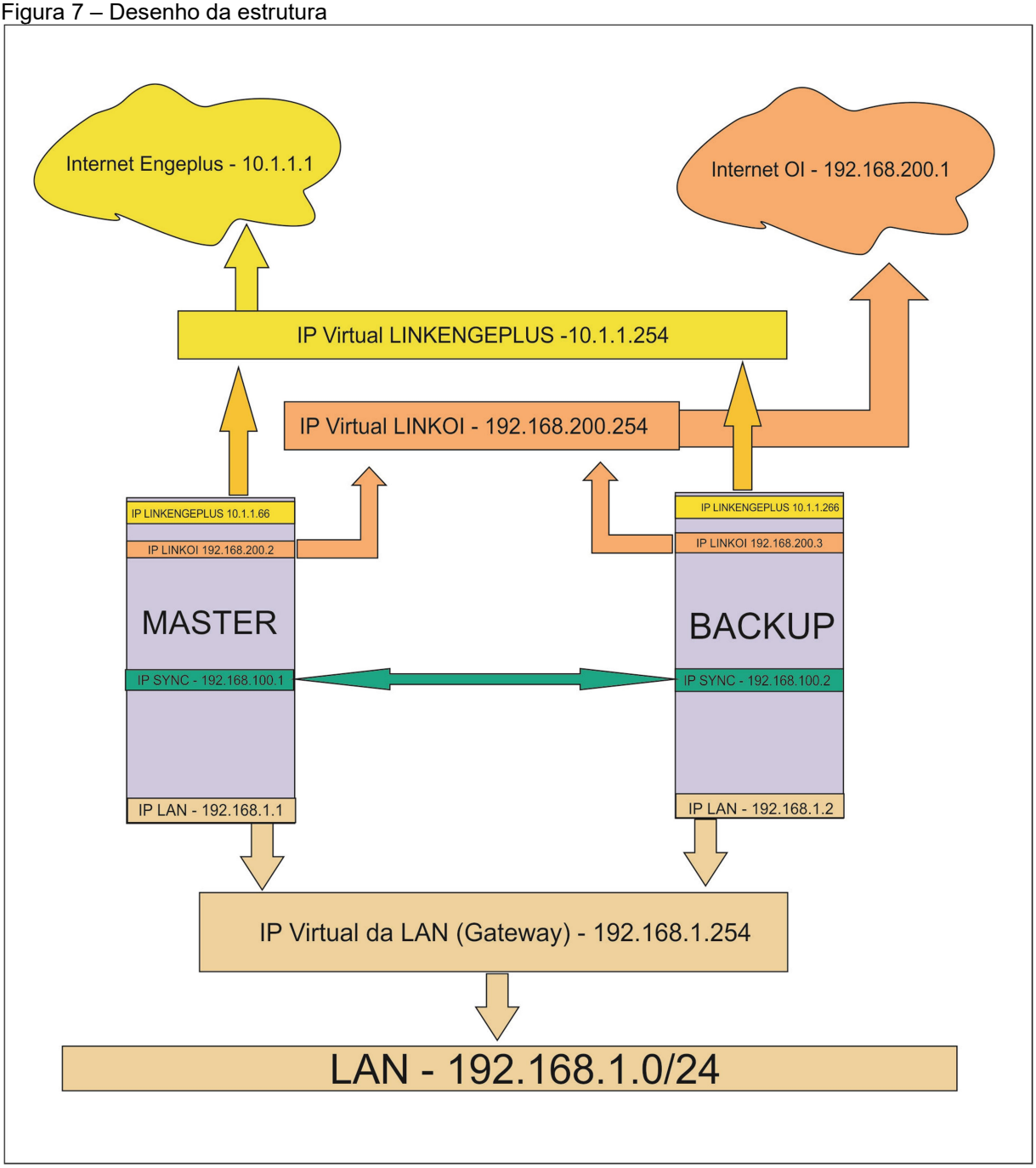
O *firewall* de alta disponibilidade foi implementado por um *cluster* composto por dois computadores onde cada computador possuía quatro placas de redes instaladas, dois *links* de Internet de operadoras diferentes a fim de obter-se a alta disponibilidade não apenas nos servidores como também nos *links.*

Para que fosse possível a implementação deste trabalho foi necessário a instalação do sistema FreeBSD PFsense versão 2.2.4 nas duas máquinas.

Após a instalação do sistema operacional FreeBSD Pfsense nos dois computadores, foi configurado e ajustado os mesmos para que trabalhassem de forma redundantes, também foi preciso realizar configurações e ajustes a fim de obter redundância e balanceamento de carga entre os dois *links* de Internet.

A figura 7 exemplifica como foi montado os dois computadores para implementação do *cluster* de alta disponibilidade, bem como os IPs reais e virtuais de cada interface, sendo o da esquerda o servidor principal chamado de *Master e* o da direita como servidor secundário chamado de *Backup*.

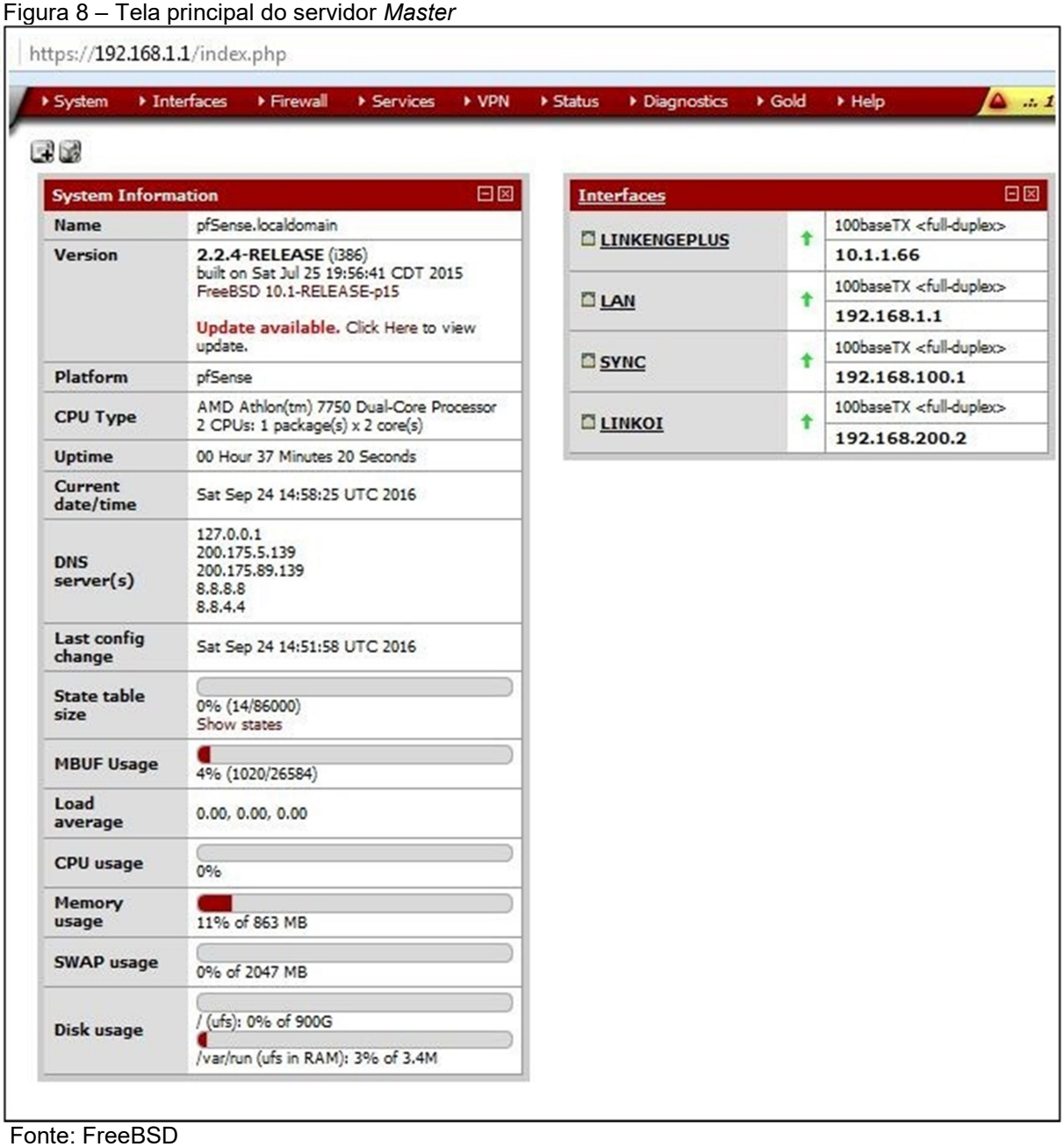
Figura 7 – Desenho da estrutura



Fonte: Do autor.

Na figura 8 pode ser observado o servidor principal acessado pelo seu endereço IP 192.168.1.1 utilizando o navegador Firefox, na qual traz as informações de sua configuração como: endereço das interfaces de rede, a versão do sistema, tipo de processador, o tempo que o servidor está ligado, consumo da memória e do disco.

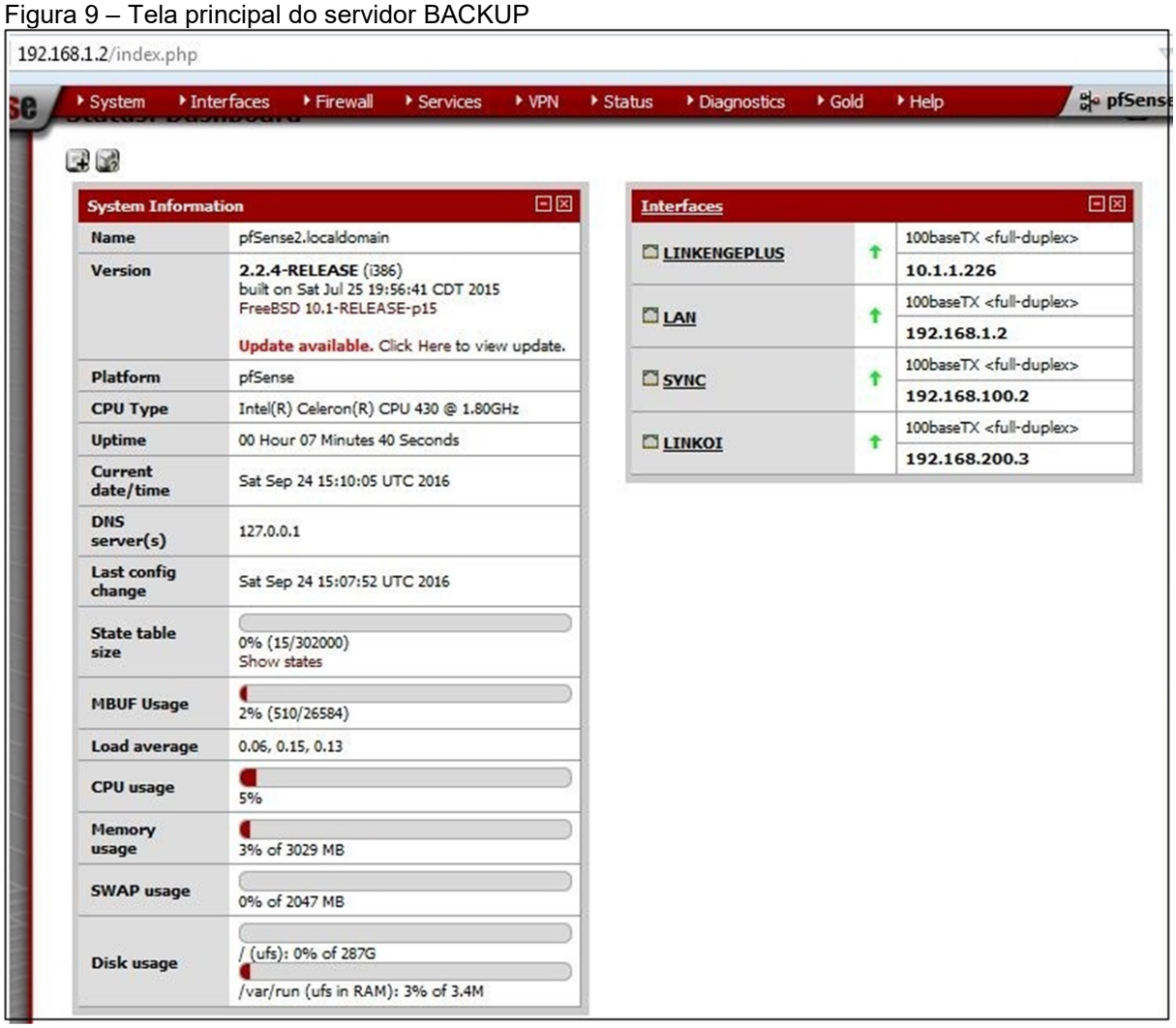
Figura 8 – Tela principal do servidor *Master*



Fonte: FreeBSD

A figura 9 ilustra o servidor secundário acessado pelo seu endereço IP 192.168.1.2 utilizando o navegador Firefox, onde é possível ver as informações de sua configuração como: endereço das interfaces de rede, a versão do sistema, tipo de processador, o tempo que o servidor está ligado, consumo da memória e do disco.

Figura 9 – Tela principal do servidor BACKUP



Fonte: FreeBSD

A Tabela 2 exemplifica os endereços IPs reais e virtuais atribuídos em cada interface de rede dos servidores *Master e Backup*.

Tabela 2 – Endereços IPs

Servidores	MASTER	BACKUP
IP da interface LAN	192.168.1.1	192.168.1.2
IP virtual da interface LAN	192.168.1.254	192.168.1.254
IP da interface Sync	192.168.100.1	192.168.100.2
IP da interface Oi	192.168.200.2	192.168.200.3
IP virtual da interface Oi	192.168.200.254	192.168.200.254
IP da interface Engeplus	10.1.1.66	10.1.1.226
IP virtual da interface Engeplus	10.1.1.254	10.1.1.254

Fonte: Do autor.

5.1 CONFIGURAÇÕES DA ALTA DISPONIBILIDADE

As configurações foram realizadas nas duas máquinas, tanto no servidor principal (*Master*) quanto no secundário (*Backup).*

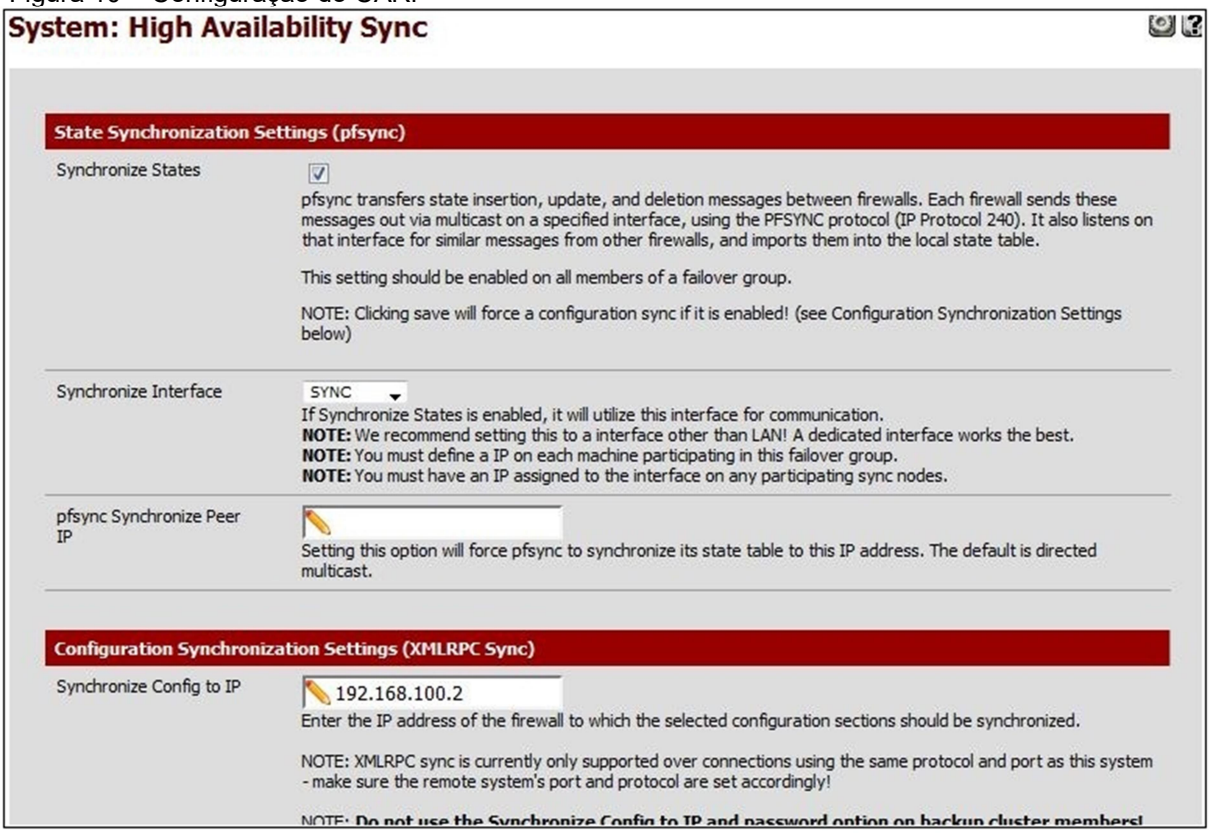
O protocolo redundante CARP e a interface de sincronismo *PFSync*, por padrão, vêm com o sistema *PFSense*, dispensando a instalação adicional de pacotes.

PFSync é a interface de rede utilizada em máquinas para compartilhar e sincronizar o estado, ou seja, faz a certificação de que uma máquina redundante tenha as mesmas configurações da máquina principal, sincronizando as duas máquinas que utilizam esta interface, deste modo fica apto a aceitar conexões sem perda, mesmo com a máquina principal desligada. Deve ser executado em uma rede confiável, pois a mesma não faz autenticação do protocolo, e de preferência por meio de um cabo *crossover* (também conhecido como cabo cruzado, é um cabo de rede par trançado que permite a ligação de dois computadores pelas respectivas placas de redes sem a necessidade de um concentrador, onde se tem: em uma ponta o padrão T568A, e, em outra, o padrão T568B) entre as duas máquinas (FREEBSD, 2013).

O Endereço Comum de Protocolo de Redundância (CARP) permite que múltiplos *hosts* compartilhem o mesmo endereço IP (Virtual) e ID do *Host* Virtual (VHID), a fim de fornecer a alta disponibilidade para um ou mais serviços. Isto significa que um ou mais *hosts* pode falhar, e os outros *hosts* transparentemente assumir o serviço de forma que os usuários não percebam uma falha no serviço (FREEBSD, 2013).

Para ajustar o sincronismo foi necessário acessar as configurações do CARP, conforme ilustra a figura 10, e ativar a regra de sincronismo de estados. Com essa opção ativada, tudo o que for alterado no servidor principal *(Master)*, é replicado no servidor secundário (*Backup*). Também é definido qual interface, IP e usuário e senha para o sincronismo; no caso em estudo, a interface definida é a SYNC e o IP é 192.168.100.2 sendo este o IP da interface SYNC do servidor secundário (*Backup*).

Figura 10 – Configuração do CARP



Fonte: FreeBSD

Nesta mesma tela também é configurado tudo o que se quer e pode ser sincronizado no *PFSense,* conforme mostra a figura 11, de modo que se houver alterações ou criação regras ou aliases no servidor *Master* as mesmas serão replicadas no servidor *Backup*. No presente trabalho foi selecionado o sincronismo das regras de *firewall*, *aliases* e *Network Address Translation* (NAT), que é uma técnica que consiste em reescrever, utilizando-se de uma tabela *hash*, os endereços IPs de origem de um pacote que passam por um *router* ou *firewall* de maneira que

um computador de uma rede interna tenha acesso ao exterior ou rede mundial de computadores (rede pública).

Figura 11 - Sincronismo

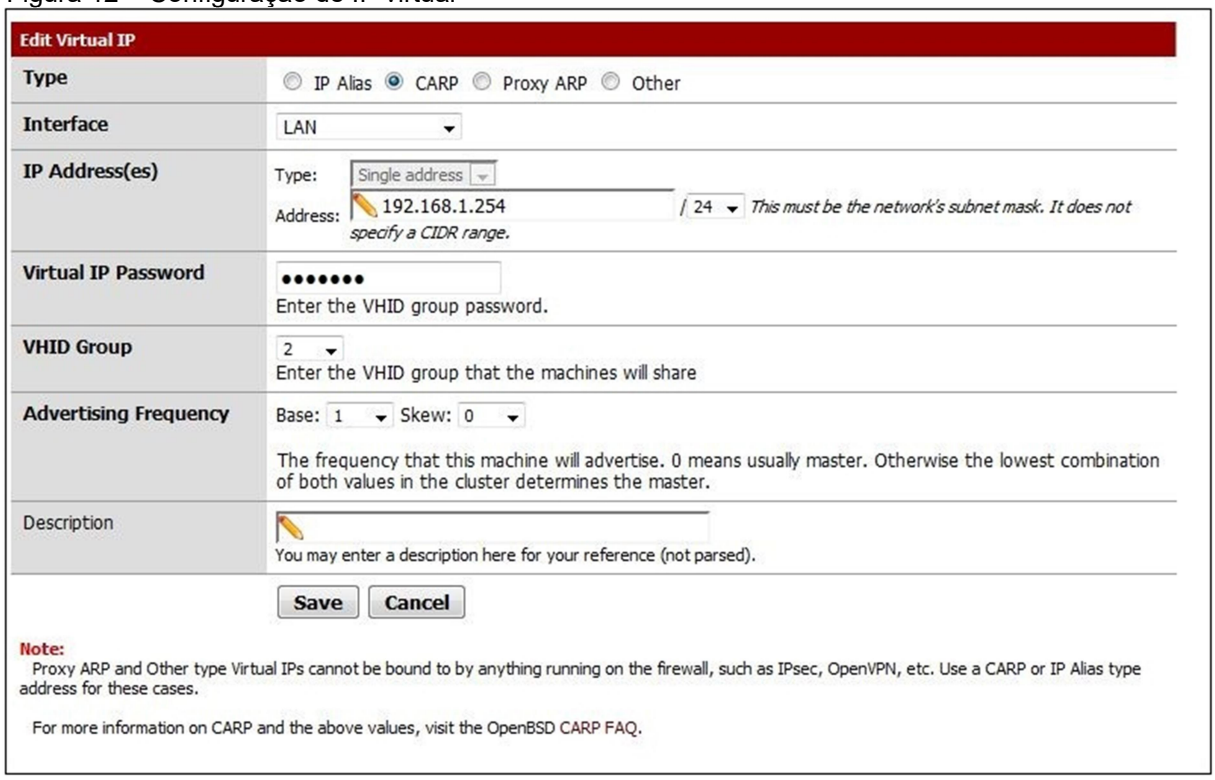
Fonte: FreeBSD

A figura 12 mostra a configuração do IP virtual, do tipo CARP, realizado na interface LAN, onde se atribuiu o IP 192.168.1.254, essas configurações foram efetuadas nas duas máquinas de forma que o IP 192.168.1.254 ficasse flutuante.

O campo “*Advertising Frequency*” é quem define a prioridade de ser o mestre ou escravo, quanto menor o valor da frequência, maior na hierarquia para assumir o lugar do mestre, o mestre sempre assume o valor “0”, definido na configuração “*Skew*”.

Para os IPs virtuais das interfaces WANs, seguiu-se os mesmos princípios.

Figura 12 – Configuração de IP virtual



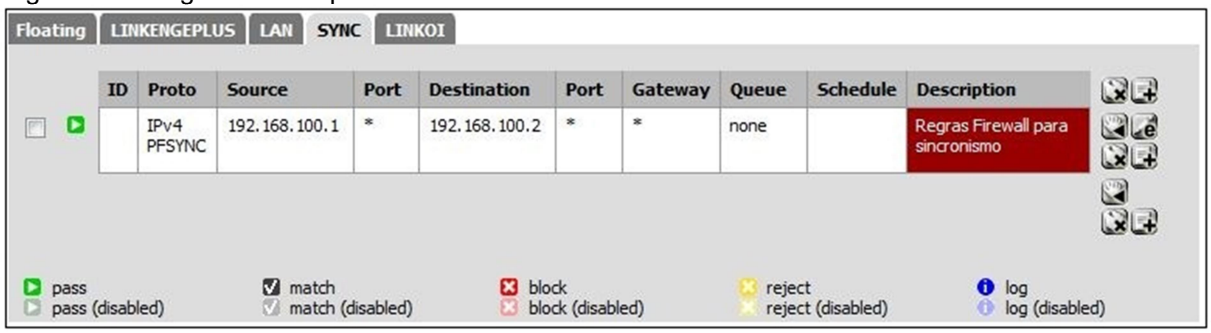
Fonte: FreeBSD

A fim de obter-se o sincronismo entre os servidores, foi necessário criar uma regra no *firewall* para que as interfaces de sincronismo “SYNCs” pudessem se comunicar.

Na figura 13, é possível perceber que esta regra permite acesso somente entre as interfaces SYNCs, com o protocolo ipv4 PFSYNC, o endereço de origem 192.168.100.1 e o endereço de destino 192.68.100.2. Podendo essa regra ser mais especifica, como por exemplo, informando o endereço MAC e até mesmo o Sistema Operacional que pode ter acesso a essa conexão, quanto mais específica for as regras de *firewall*, maior é a segurança, dificultando assim acessos indesejados.

Mesmo sendo a alta disponibilidade o foco desta pesquisa, a atenção com a segurança é fundamental, por se tratar da utilização de *firewalls.* Foi necessário criar essa regra nos dois servidores (*Master* e *Backup*).

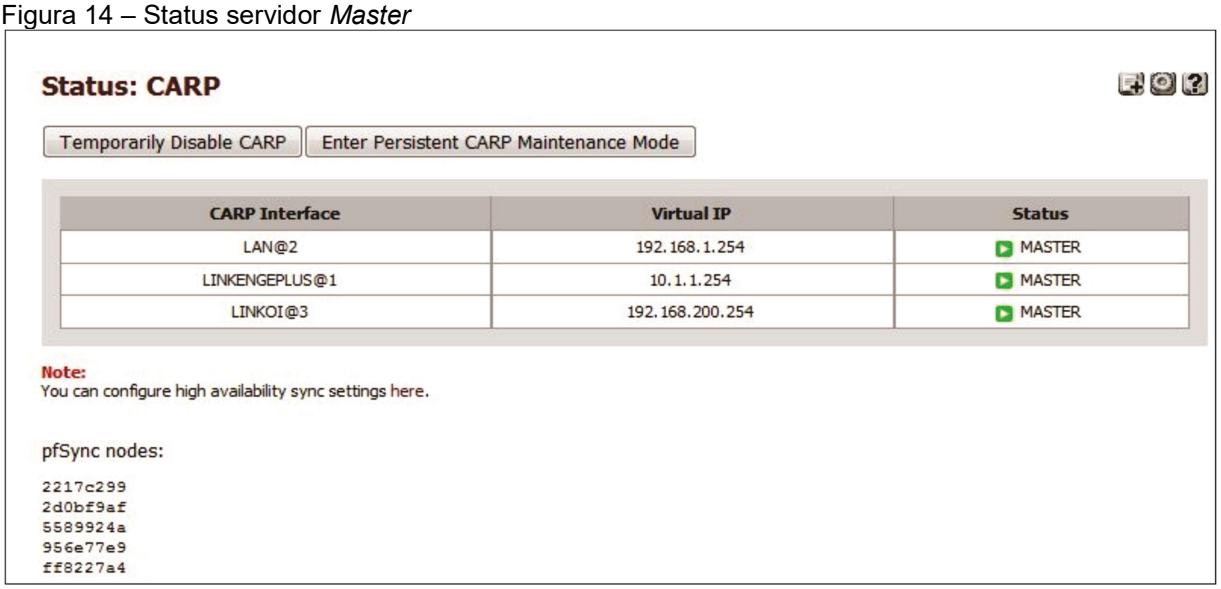
Figura 13 – Regras firewall para sincronismo



Fonte: FreeBSD

A figura 14 traz informações da interface, IP virtual e o status do sistema de alta disponibilidade implantado no servidor principal (*Master*), essa tela pode ser acessada na opção *Status* - CARP *(failover)*.

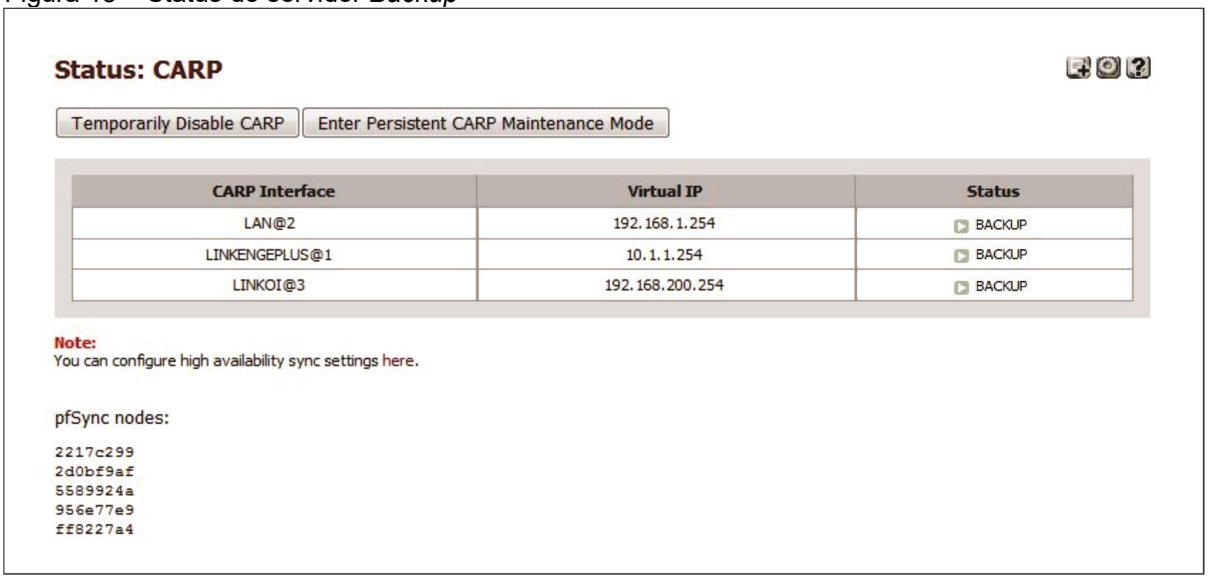
Figura 14 – Status servidor *Master*



Fonte: FreeBSD

Já na figura 15 traz informações da interface, IP virtual e o status do sistema de alta disponibilidade implantado no servidor secundário (*Backup*), essa tela pode ser acessada clicando no menu *Status* - CARP *(failover)*.

Figura 15 – Status do servidor *Backup*

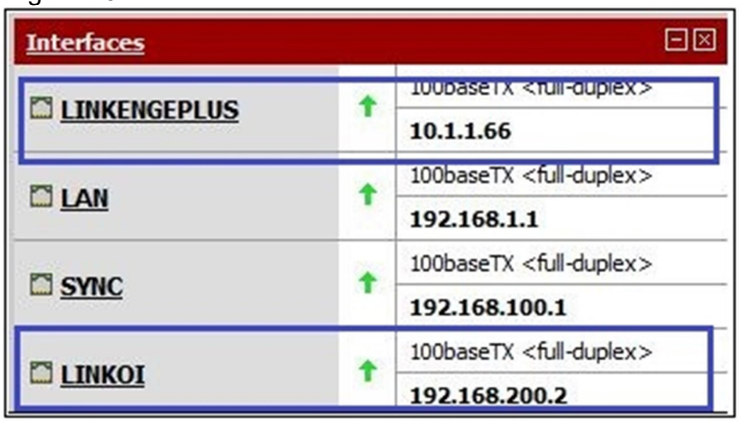


Fonte: FreeBSD

5.2 CONFIGURAÇÕES DO *FAILOVER* E *LOAD BALANCE* DOS *LINKS*

Para que fosse possível a implementação da alta disponibilidade dos *links* e do balanceamento de carga, foi necessária a configuração de duas placas de redes como entrada de Internet uma placa chamada LINKENGEPLUS e a outra LINKOI conforme ilustra a figura 16 onde as mesmas estão destacadas em azul.

Figura 16 – Interfaces WANs

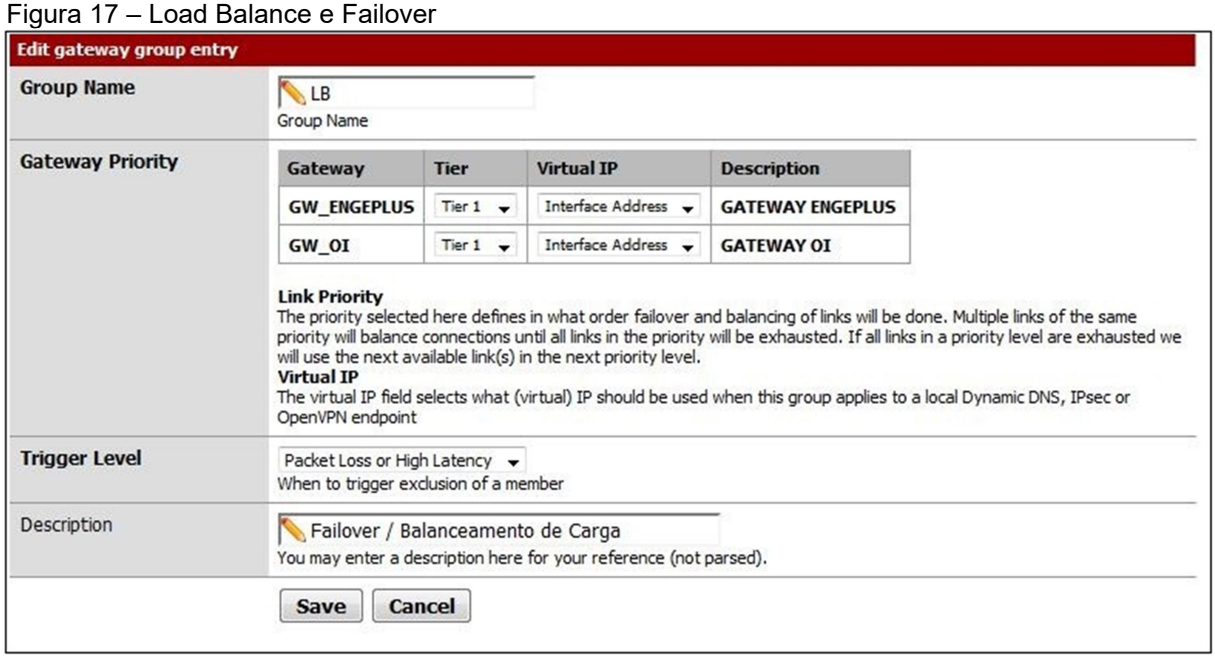


Fonte:FreeBSD

A configuração do *Failover e Load Balance* foi realizada adicionado um novo grupo de *gateway* contendo os seguintes parâmetros, conforme ilustra a figura 17: Group Name refere-se ao nome da regra, que no presente trabalho foi chamada de “LB”, a opção *Gateway Priority* é onde se define as prioridades dos *links* bem como seu peso, neste trabalho optou-se em definir como *Tier* 1 para as duas

interfaces, isto quer dizer um balanceamento de 50% (cinquenta por certo) por exemplo: os usuários ao se conectarem na Internet, se foi realizada pelo “LINKENGEPLUS” a próxima conexão será pelo “LINKOI”, porém pode ser ajustado pesos diferentes caso um *link* seja melhor do que o outro, exemplo caso o “LINKENGEPLUS” seja ajustado para *Tier* 2 e o “LINKOI” Tier 1, o acesso a Internet será de duas conexões pelo “LINKENGEPLUS” para uma conexão pelo “LINKOI” . A opção *Tigger Level* é onde se configura o *Failover*, no presente trabalho optou-se em *Package loss or Higth Latency* ou seja caso haja perda de pacote ou alta latência em algum *gateway* o outro deve assumir. A opção *Description* é apenas para descrever com mais detalhes a regra em questão, este campo aceita qualquer caractere.

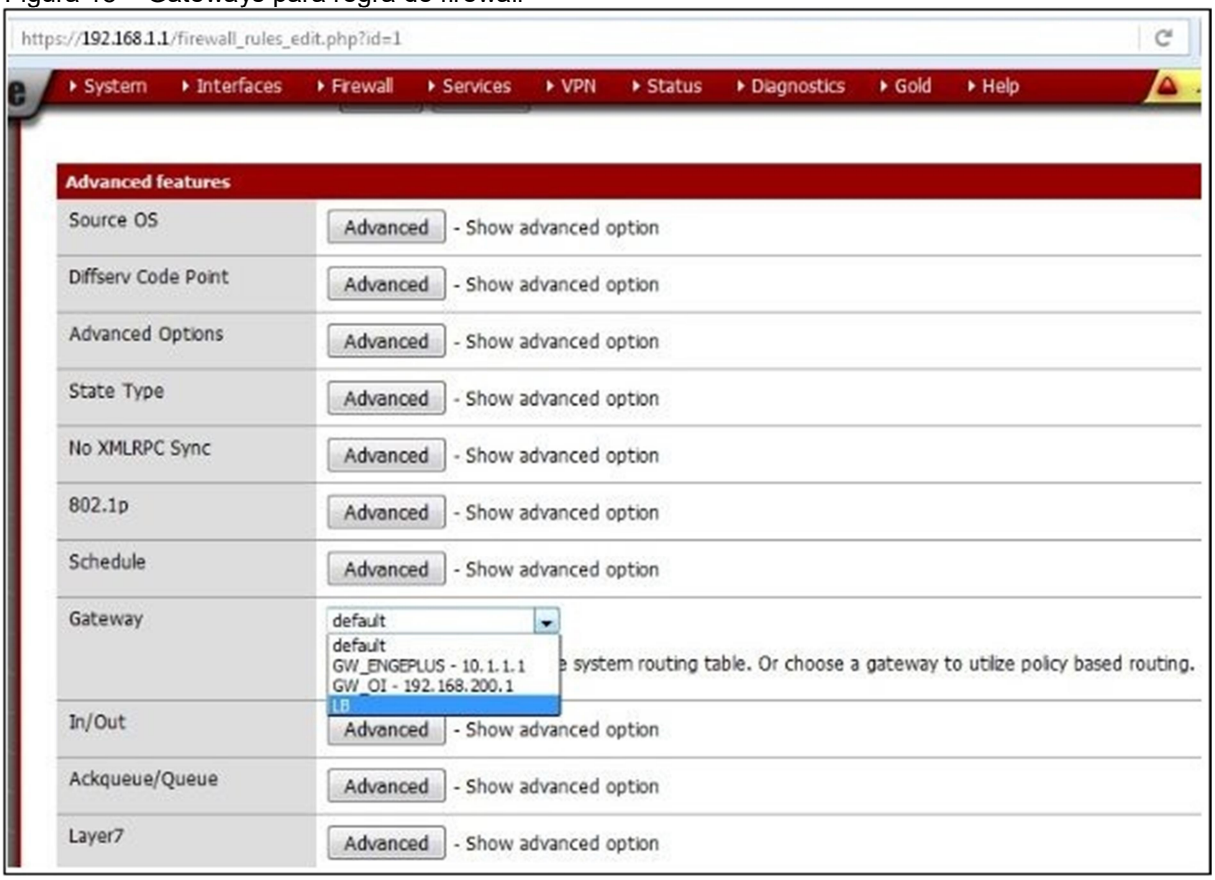
Figura 17 – Load Balance e Failover



Fonte:FreeBSD

Para que os *hosts* conectassem a Internet de forma balanceada e com redundância dos *links* foi necessário o ajuste nas regras do *firewall* onde é possível definir para cada *host,* qual *Gateway* ou Grupo de *Gateway* ele usará. Como o foco dessa pesquisa é a alta disponibilidade e balanceamento de carga, criou-se apenas uma regra para rede local, onde foi ajustado que todas as conexões realizadas tenham o “LB” como *Gateway*. A figura 18 exemplifica este cenário.

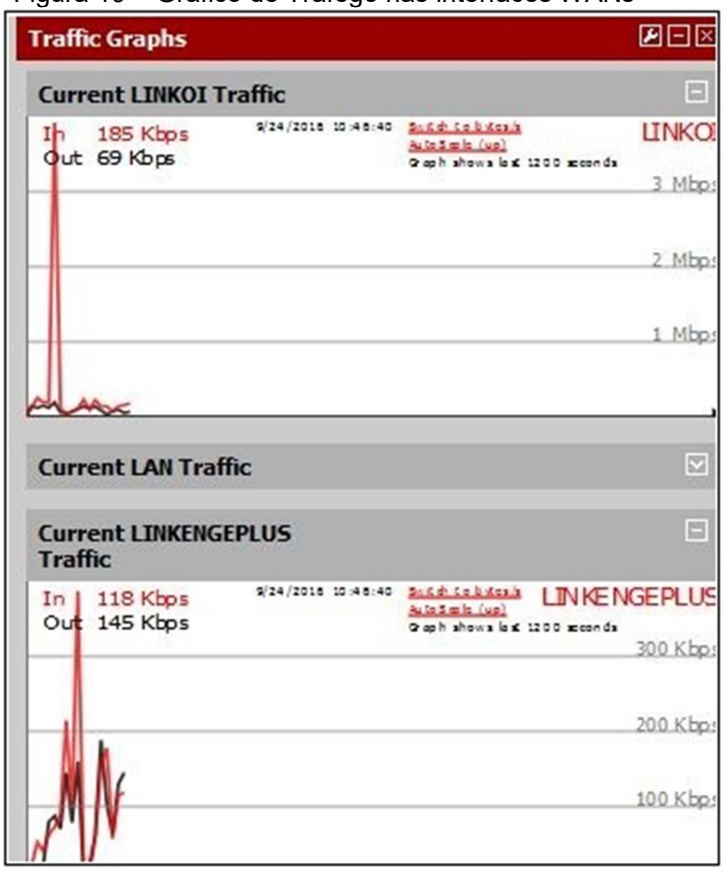
Figura 18 – Gateways para regra de firewall



Fonte:FreeBSD

Após as configurações e ajustes de alta disponibilidade e balanceamento de carga, observou-se através da ferramenta *Traffic Graphs,* disponível no próprio PFsense, se estava havendo tráfego nas duas interfaces WANs em tempo real que no presente trabalho são: “LINKOI” e “LINKENGEPLUS”, exemplificado na figura 19, as linhas que estão em vermelho representam a entrada de dados já as linhas que estão em preto representam a saída de dados. Pode-se observar no gráfico, o tráfego tanto de entrada como saída dos dados, em ambas as interfaces.

Figura 19 – Gráfico de Tráfego nas interfaces WANs



Fonte:FreeBSD

6 RESULTADOS

6.1 TESTES DE REDUNDÂNCIA DOS LINKS

Esse teste foi realizado com os dois servidores *Master e Backup* e os dois *links* “LINKENGEPLUS” e “LINKOI” ligados, e a rede local utilizando a Internet*,* em um dado momento foi desligado um *link*, o “LINKOI”, simulando uma falha, após ser desligado verificou-se se houve interrupção da Internet, mais especificamente na navegação, *downloads* que estavam ativos e *downloads* pos-desligamento do *link*. O mesmo teste foi repetido, porém com desligamento do *link* “LINKENGEPLUS”.

A figura 21 traz a informação de um teste de *Ping* que foi disparado no endereço google.com onde é possível observar a perda de pacotes durante a troca dos *links.*

O *Ping* verifica a conectividade de nível IP com outro computador TCP/IP enviando mensagens de solicitação de eco ICMP (*Internet Control Message Protocol)*. O recebimento de mensagens de resposta de eco correspondentes é exibido juntamente com ida. O *Ping* é o principal comando TCP/IP usado para solucionar problemas de conectividade, alcance e resolução de nomes. Neste teste foi utilizado o comando *Ping* com a opção –t, que significa que o *Ping* continua enviando mensagens de solicitação de eco ao destino até que seja interrompido.

Figura 21 – Teste de conexão utilizando o *Ping*

Fonte: Microsoft

Mesmo com a perda desses pacotes a redundância dos *links* funcionou perfeitamente de forma que os usuários não perceberam o problema no *link*. Já nos *downloads* que estavam em execução, que haviam sidos iniciados pelo o *link* que foi derrubado, houve interrupção com a troca do *link*, exigindo assim o usuário reiniciar

os *downloads* manualmente, a mesma regra segue para os *downloads* pós desligamento do *link*, ou seja, se houver troca de *link* com *downloads* em andamento, o mesmo deve ser reiniciado manualmente pelo o usuário.

A tabela 4 explica que no momento em que ocorreu a interrupção do *link* “LINKOI”, houve uma falha no *download* que havia sido inicializado por esta interface, já os *downloads* ativos que foram iniciados pelo *link* “LINKENGEPLUS” não apresentaram nenhuma falha. Ocorreu um erro no balanceamento de carga também, pois para haver o balanceamento os dois *links* devem estar ativos.

Tabela 4 – Resultados com desligamento do LINKOI

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Sim	-	Ok	Ok
BACKUP	Sim	-	Ok	Ok
LINKOI	Não	Falha	Ok	Falha
LINKENGEPLUS	Sim	Falha	Ok	Ok

Fonte: Do autor

Já na tabela 5, observa-se que no momento em que ocorreu a interrupção do link “LINKENGEPLUS”, houve uma falha no *download* que havia sido inicializado por esta interface já os *downloads* ativos que foram iniciados pelo *link* “LINKOI” não apresentaram nenhuma falha. Houve falha no balanceamento de carga, pois para que haja o balanceamento são necessários os dois *links* funcionando, entretanto a redundância dos *links* funcionou como planejada.

Tabela 5 – Resultados com desligamento do LINKENGEPLUS

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Sim	-	Ok	ok
BACKUP	Sim	-	Ok	ok
LINKOI	Sim	Falha	Ok	ok
LINKENGEPLUS	Não	Falha	Ok	Falha

Fonte: Do autor

O tempo de recuperação das falhas dos *links*, tanto o “LINKENGEPLUS” quanto o “LINKOI” foi de 29 segundos, conforme apresenta a tabela 6.

Tabela 6 – Tempo de recuperação de falhas nos *links*

SERVIDORES/ LINKS	TEMPO (EM SEGUNDOS)
MASTER	-
BACKUP	-
LINKOI	29s
LINKENGEPLUS	29s

Fonte: Do autor

Aplicando a formula da disponibilidade, considerando que o tempo de teste foi de oito dias (691,200 segundos) e que houve apenas uma queda em um dos *links* obteve-se o seguinte resultado:

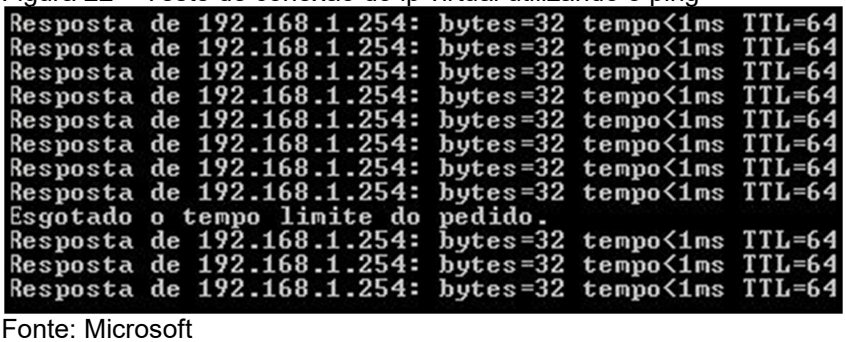
$$\mathbf{D} = \frac{\mathbf{691200 - 29}}{\mathbf{691200}}$$

D = 99,9958043981%, Neste caso foi obtido uma grande disponibilidade, chegando próximo a 100%**.**

6.2 TESTES DE ALTA DISPONIBILIDADE DOS SERVIDORES FIREWALL

Esse teste foi realizado com os dois servidores (*Master e Backup)* ligados e com a rede local (LAN) utilizando a Internet, em um dado momento foi desligado o servidor *Master* a fim de simular uma falha. Antes do desligamento foi disparado o comando *Ping* com a opção –t, no IP virtual flutuante da LAN 192.168.1.254, sendo este o IP do *Gateway* utilizado nos *hosts* na LAN. Observou-se que na transição dos servidores houve uma perda de pacote conforme ilustrado na figura 22.

Figura 22 – Teste de conexão do ip virtual utilizando o ping

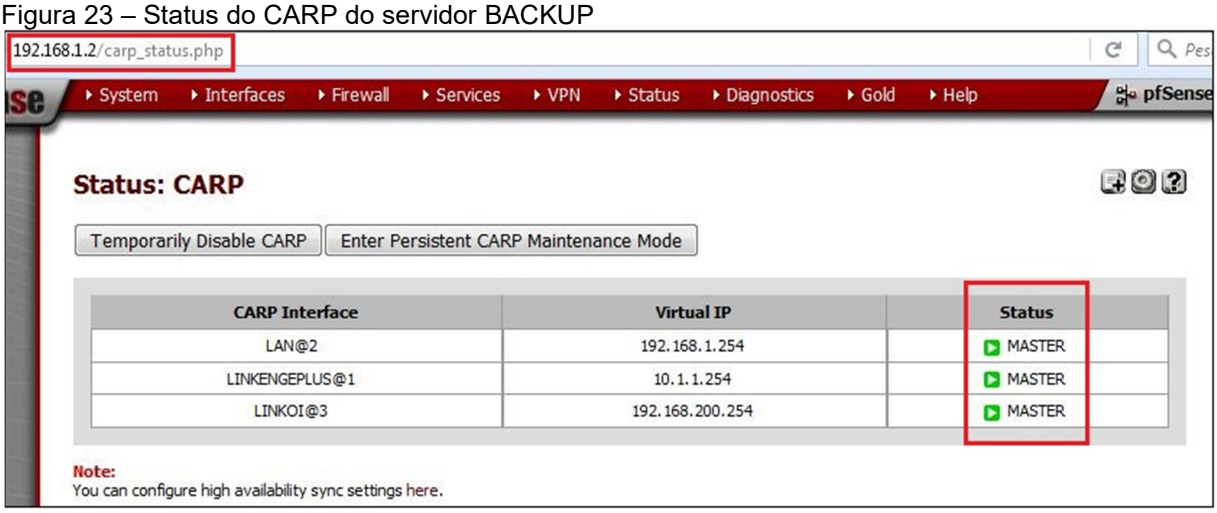


Fonte: Microsoft

Com o desligamento do servidor *Master* o servidor *Backup* passou a assumir a posição de principal de forma transparente, executando todas as regras e configurações que haviam sido definidas para o sincronismo entre os servidores, deste modo não houve interrupção nos serviços de Internet na rede local, com exceção os *downloads* que estavam em execução que foi necessário reiniciá-los manualmente.

Observou-se que acessando o IP 192.168.1.2, que é o endereço do servidor "*BACKUP*", no *Status* do CARP, onde antes do desligamento do servidor *Master*, o *status* do servidor trazia a informação "*BACKUP*", passou a assumir o status de "*MASTER*", conforme ilustra a figura 23.

Figura 23 – Status do CARP do servidor BACKUP



Fonte: Do autor.

A tabela 7 exemplifica o momento que foi desligado o servidor *Master* onde houve falha nos *downloads* em andamento, porem a redundância dos servidores funcionou perfeitamente. Quanto ao balanceamento de carga entre os *links* não houve erro, pois os mesmos continuaram ativos.

Tabela 7 – Resultados com desligamento do servidor *Master*

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Não	-	Ok	Falha
BACKUP	Sim	-	Ok	Ok
LINKOI	Sim	OK	Ok	Ok
LINKENGEPLUS	Sim	OK	Ok	Ok

Fonte: Do autor.

O próximo item que foi analisado foi o religamento do servidor *Master*, quando o mesmo foi ligado, voltou a assumir como servidor principal ocupando o status de "*MASTER*", já o servidor *Backup* que estava realizando o "papel" de principal, voltou a ser o "*BACKUP*".

A tabela 8 exemplifica o momento que o servidor *Master* foi religado, onde mostra uma falha nos *downloads* ativos no momento da transição dos servidores, sendo necessário reiniciá-los de forma manual.

Tabela 8 – Resultados com religamento do servidor *Master*

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Sim	-	Ok	-
BACKUP	Sim	-	Ok	Falha
LINKOI	Sim	OK	Ok	Ok
LINKENGEPLUS	Sim	OK	Ok	Ok

Fonte: Do autor.

O tempo de recuperação de falhas nos servidores, tanto no *Master* quanto no *Backup*, foi de 2 segundos, conforme apresenta a tabela 9.

Tabela 9 – Tempo de recuperação de falhas nos servidores

SERVIDORES/ LINKS	TEMPO (EM SEGUNDOS)
MASTER	**2s**
BACKUP	**2s**
LINKOI	**-**
LINKENGEPLUS	**-**

Fonte: Do autor.

Aplicando a formula da disponibilidade, considerando que o tempo de teste foi de oito dias (691,200 segundos) e que houve apenas uma falha e em um dos servidores, obteve-se o seguinte resultado:

$$\mathbf{D} = \frac{\mathbf{691200 - 2}}{\mathbf{691200}}$$

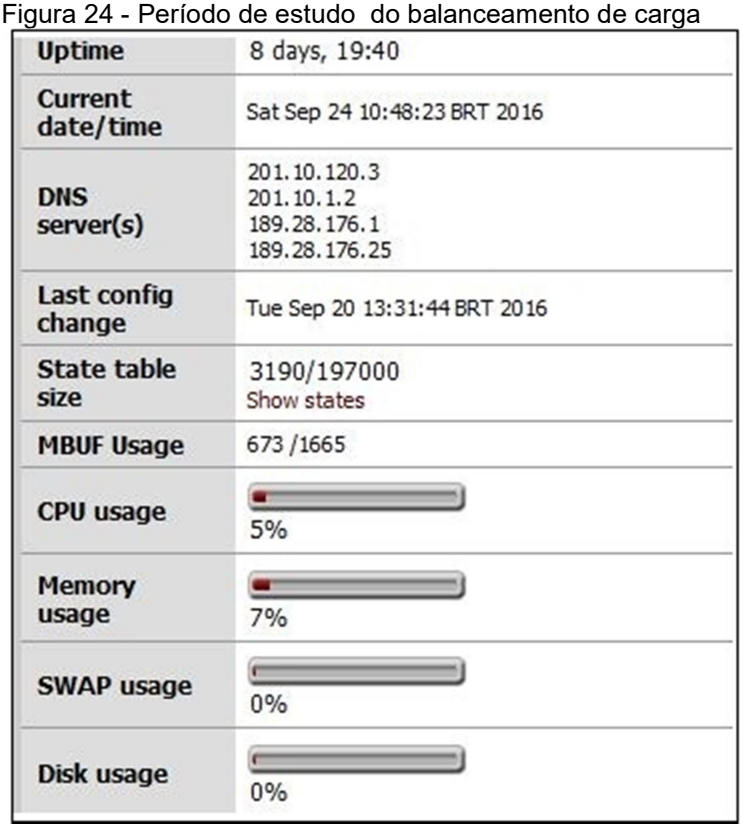
D = 99,9997106481%, Neste caso foi obtido uma grande disponibilidade, chegando próximo a 100%**.**

6.3 TESTES DE BALANCEAMENTO DE CARGA

Os testes do Balanceamento de Carga entre os *links* de Internet foram realizados num período de oito dias com todos os computadores da rede interna consumindo a Internet para navegação, envio e recebimento de *emails* e *downloads*.

A figura 24 ilustra o período que em que o servidor ficou ligado para realização do estudo.

Figura 24 - Período de estudo do balanceamento de carga



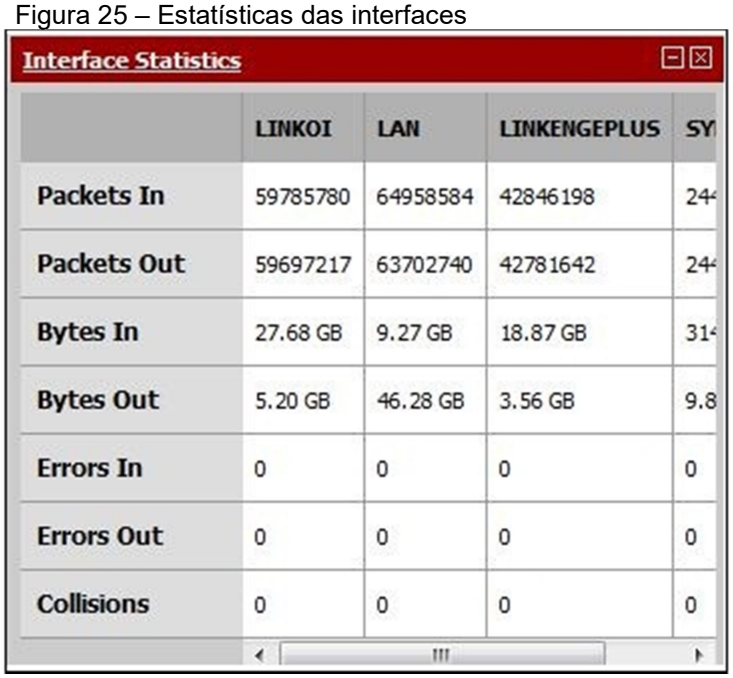
Fonte: FreeBSD

Durante esses oito dias perceberam-se um melhor desempenho quanto a velocidade para a navegação, envio e recebimento de *emails* e *downloads*, pois com o uso da técnica de balanceamento de carga, toda a carga de consumo da Internet foi distribuída entre os dois *links* e com isso foi possível minimizar o tempo de resposta e evitou sobrecarga.

Utilizando múltiplos componentes com o balanceamento de carga ao invés de um único componente além de otimizar os recursos, pode aumentar a confiabilidade através da redundância.

A figura 25 traz a informação através de estatísticas de consumo de carga de cada interface, disponibilizada pelo próprio PFsense, que houve um consumo de forma balanceada entre as interfaces “LINKOI” e “LINKENGEPLUS” no período de oito dias, onde a interface “LINKOI” teve como entrada 27,68 GB e 5,20 GB como saída e a interface “LINKENGEPLUS” teve 18,87 GB entrada e 3,56 GB de saída.

Figura 25 – Estatísticas das interfaces



Fonte: FreeBSD

Com o uso do balanceamento de carga entre os *links* foi constatado que alguns sites, principalmente sites de bancos que exigem que quando é realizada a conexão pelo um determinado IP, este não seja alterado durante a operação.

Para resolver esse problema, foi necessário acessar o menu *System – Advanced – Miscellaneous* e marcar a opção “*use sticky conections*”, conforme a figura 26. Ao marcar essa opção conexões sucessivas são redirecionadas para os servidores de um modo *round-robin*, com conexões a partir da mesma fonte a serem enviados para o mesmo servidor web.

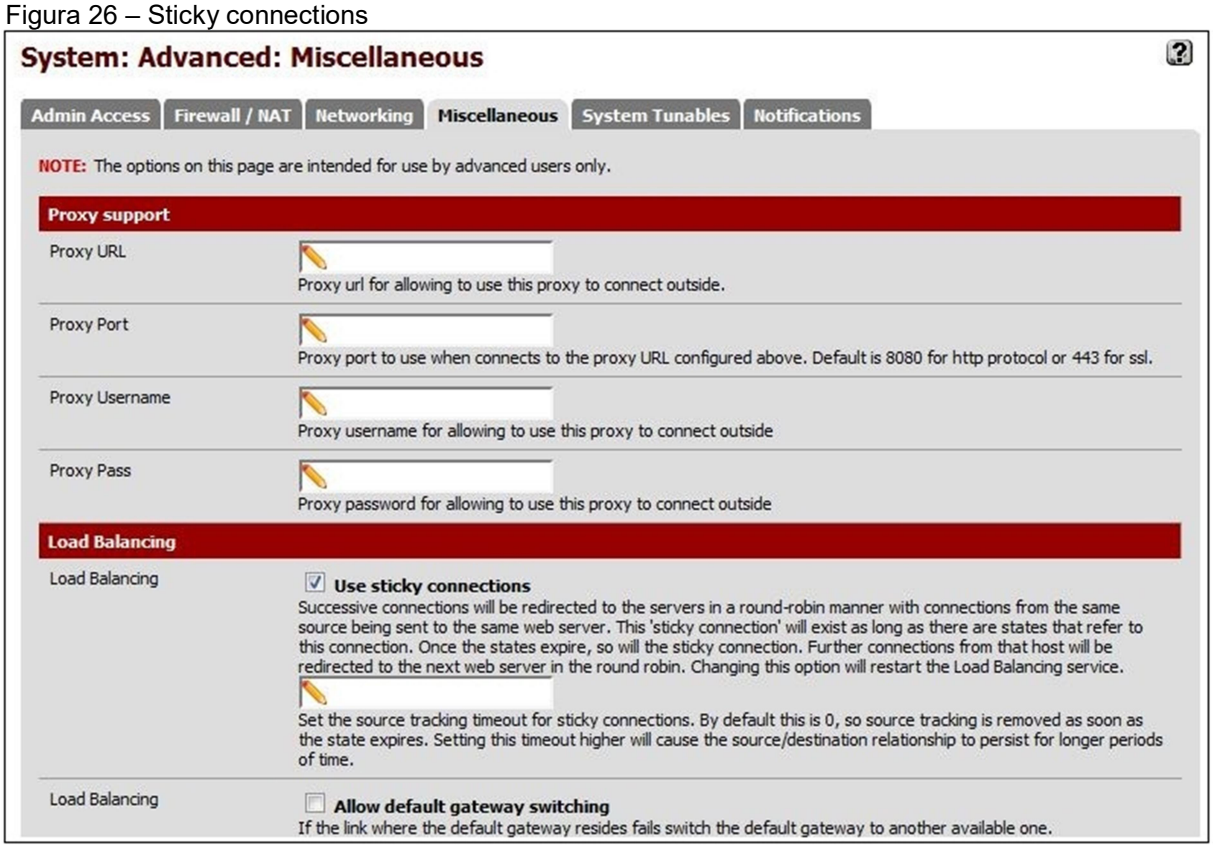
Round-robin é um algorítimo de agendamento de processos de um sistema operacional que atribui frações de tempos iguais para cada processo de

forma circular, sem prioridades. Este escalonamento também pode ser aplicado em outros problemas de agendamento, como agendamento de transmissão de pacotes de dados em redes de computadores.

Essa "conexão fixa" vai existir enquanto houver estados que se referem a esta conexão. Uma vez que o estado expirar, outras conexões a partir do *host* serão redirecionadas para o próximo servidor *web* de modo *round robin*.

É possivel definir o tempo limite rastreamento de origem para conexões *stickys* (conexões que se pega), conforme mostra a figura 26, que neste caso não foi definido valor logo o valor atribuido é o padrão que é 0, portanto, o rastreamento de origem é removido assim que o estado expirar. Ajustando o tempo limite com um valor superior irá fazer com que a relação de origem e destino persista por períodos mais longos de tempo.

Figura 26 – Sticky connections



Fonte: FreeBSD

7 CONCLUSÃO

Neste trabalho foram estudados os principais conceitos de rede de computadores, *firewall* e *cluster* de alta disponibilidade e balanceamento de carga. Foram também apresentadas ferramentas que implementam *cluster* de alta disponibilidade e balanceamento de carga em um *firewall* baseando em FreeBsd, mais especificamente o PFsense.

Foi constatado nesse estudo que um dos grandes erros em projetos de *clusters*, para qualquer que seja a sua finalidade, é achar que o processamento, ou alta capacidade de processamento, é baseado apenas em computadores, ou em processadores. Um *cluster* precisa ser visto como um organismo completo, cada peça tem sua importância e pode ser o responsável por um melhor desempenho do sistema globalmente. E é exatamente nesse aspecto que a rede de comunicações teve sua importância para projetar o *cluster* de alta disponibilidade. A rede pode se tornar um gargalo de desempenho para o *cluster*, sendo esta a razão do estudo sobre conceitos de redes de computadores bem como os modelos de referências TCP/IP e OSI e suas respectivas camadas.

A partir de conceitos de *cluster*, foi implementado um *sistema* de alta disponibilidade de *firewall por* meio de redundância de máquinas, fazendo uso do Protocolo de Redundância de Endereço Comum (CARP). Foi utilizado também o *Failover* e *Load Balance* para a redundância e balanceamento de carga dos *links,* deste modo aparentou para a rede local, um único sistema coerente de Internet, foi possível observar um melhor desempenho nas conexões.

Uma das vantagens do uso de balanceamento de carga é que além de dividir a carga, que no presente trabalho foram as conexões com a Internet em uma rede local, é possível obter a alta disponibilidade dos *links,* através da redundância.

Conclui-se nesse estudo que a alta disponibilidade não é apenas um produto ou um *software* específico, mas a um conjunto de mecanismos e técnicas que tem por objetivo detectar, contornar de forma transparente falhas que venham a ocorrer ocasionando perda de acessibilidade.

A distribuição *PFSense,* mostrou-se uma ótima ferramenta *open-source* para implementação de *firewall* com ou sem alta disponibilidade e balanceamento de carga. Através de sua interface *web*, fica mais fácil e prático o gerenciamento da

rede, porém não dispensou um estudo aprofundado sobre rede de computadores e como funciona um *firewall*, pois um *firewall* mal configurado compromete toda a segurança de uma rede.

A implementação prática deste trabalho foi de suma importância, pois somente vivenciando a experiência prática, surgiram dificuldades que precisaram ser contornadas que foram além da conceituação teórica. Podem ser citadas algumas dificuldades durante a implementação, como: a incompatibilidade do sistema PFsense com o *hardware* escolhido inicialmente, sendo necessário providenciar um *hardware* compatível; algumas placas de redes não possuíam *driver* nativo no PFsense, exigindo a instalação manual, para que assim fosse possível a continuação do trabalho.

Após os estudos realizados neste trabalho, conclui-se que os objetivos de alta disponibilidade e balanceamento de carga dos *links* de Internet, utilizando software livre para um *firewall,* foram atingidos, sendo esta uma solução possível para empresas ou organizações que buscam por soluções de alta disponibilidade e balanceamento de carga de Internet em uma rede local.

Com grande dependência da Internet nas organizações, a pressão sobre gestores de tecnologia da informação vem aumentando, pois, se o sistema ou o serviço não estiver disponível, o negócio pode ser ameaçado e sofrer com perdas de produtividade, receita e bom relacionamento com o cliente.

Este estudo mostrou que é possível criar um ambiente de alta disponibilidade sem a necessidade da aquisição de licenças de *softwares*, preocupando-se apenas nos *hardwares* envolvidos como computadores, *switches e no-breaks*, sendo que, alguns *hardwares*, como computadores podem ser reaproveitados, uma vez que foram descartados para outras funções, a o invés de serem jogados no lixo, muitas vezes em lugares impróprios, podem ser utilizados para algumas tarefas específicas como, por exemplo, na utilização de projetos de *cluster* de alta disponibilidade ou alta *performance.*

Um fator importante para o êxito da alta disponibilidade é a energia elétrica, que não foi abordado nesse estudo, contudo é relevante um estudo para trabalhos futuros.

REFERÊNCIAS

ABNT, Associação Brasileira De Normas Técnicas. Nbr Iso/Iec 17799. **Information technology -- Security techniques -- IT network security**, 2005.

Akamai's [**State of the Internet**] Q1 2015 report. Disponível em:<https://www.akamai.com/us/en/multimedia/documents/content/akamai-state-of-the-internet-report-q1-2015.pd f />Acesso em 12 de Outubro 2015.

BOUFLEUER, Rafael. **Otimização de Performance de Sistemas Web Através de Técnicas de Clusterização.** 2013. 50 f. TCC (Graduação) - Curso de Curso de Tecnologia em Redes de Computadores, Universidade Federal de Santa Maria, Santa Maria, 2013.

BRANDÃO, Robson. **Conceitos sobre Disponibilidade - Parte I**. Disponível em: <http://technet.microsoft.com/pt-br/library/cc668492.aspx>. Acesso em: 09 Abr. 2016.

CHESWICK, William R. **Firewalls e Segurança na Internet**. Repelindo o Hacker Ardiloso.2.ed. Porto Alegre: Bookman, 2005.

COMER, Douglas E. **Redes de Computadores e a Internet**. 4ed.,Porto Alegre: Bookman, 2007.

CORRÊA JUNIOR, Jorge Luiz Borges; SILVA, Shirley Mirian Silva da. **Implementação de Firewall de Alta Disponibilidade Utilizando o Software Heartbeat.** 2013. 53 f. TCC (Graduação) - Curso de Tecnologia em Redes de Computadores, Instituto Federal de Educação, Ciência e Tecnologia Catarinense - Campus Sombrio, Sombrio, 2013.

COULOURIS, George et al. **Sistemas Distribuídos: conceitos e projeto**. 5. ed., Rio de Janeiro: Bookman, 2013.

Guia de **Estruturação e Administração do Ambiente de Cluster e Grid** do governo federal. Brasília, 2006. 454 p. : il.

HAN, L.; XU, Z.; YAO, Q. **An approach to web service composition based on service-ontology.In: Fuzzy Systems and Knowledge Discovery**, 2009. FSKD 09. Sixth International Conferenceon. [S.l.: s.n.], 2009. v. 2, p. 173 –177.

KUROSE, James F.; ROSS, Keith W**. Redes de Computadores e a Internet: uma abordagem top-down**. 5. ed. São Paulo: Addison Wesley, 2010. 618 f.

LOPES FILHO, Edmo.**Arquitetura de Alta Disponibilidade para Firewall e IPS baseada** em SCTP. Dissertação (Mestrado)-Universidade Federal de Uberlândia, Minas Gerais, 2008.

MANUAL do FREEBSD: **O Projeto de Documentação do FreeBSD**. Revisão: 42953. Disponível em: <http://www.freebsd.org/doc/en_US.ISO8859-/books/handbook/carp.html>. Acesso em: 12 de outubro 2015.

MORAES, Alexandre Fernandes de. **Redes de Computadores: fundamentos.** 7. ed. São Paulo: Érica, 2010.

NOMOA. **Hot Failover with CARP** - Parte I. Disponível em:http://nomoa.com/bsd/gateway/highavailability/carp.html Acesso em: 13 jun. 2016.

Relatório Revisional Analítico Sobre a Pilha TCP/IP. Disponível em: <www.freebsdbrasil.com.br/fbsdbr_files/File/public_docs/melhorias-stack-rede.pdf> Acesso em: 12 de Outubro 2015.

RODRIGUES, Fernando Fraga. **Implementação de um Ambiente de Alta Disponibilidade para Serviços de Voz Sobre IP Baseados em Asterisk.** 2007. 114 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2008.

ROMER, Rafael**.Empresas brasileiras tem prejuízo de US$ 26 bi com perda de dados, aponta estudo.** Disponível em**:**https://corporate.canaltech.com.br/noticia/protecao-de-dados/Empresas-brasileiras-tem-prejuizo-de-US-26-bi-com-perda-de-dados-aponta-estudo/. Acesso em 18 de Outubro 2016

SANTOS, Daniel Campos dos**. Alta Disponibilidade de Serviços de Rede**. 2004:48 f. TCC (Graduação) Curso de Ciência da Computação,Universidade Federal de Santa Catarina, Florianópolis, 2004.

SOUSA, Lindeberg Barrros de**. Projetos de implementação de redes: fundamentos, soluções, arquiteturas e planejamento.** 2 ed**.,** São Paulo: Érica, 2009.

STALLINGS, William. **Criptografia e segurança de redes**. Princípios e práticas. 4.ed. São Paulo: Pearson Prentice Hall, 2008.

STALLINGS, William. **Redes e sistemas de comunicação de dados:** teoria e aplicações corporativas. Rio de Janeiro: Elsevier, 2005.

TANENBAUM, Andrew S.; **Redes de Computadores**. 4. ed., Rio de Janeiro: Elsevier,2003.

THOMAS Tom. **Segurança de Redes: Primeiros passos.** Ciência moderna.Rio de Janeiro.Brasil. 2007. 398 f.

TORRES, Gabriel**. Redes de Computadores: curso completo.** Rio de Janeiro: Axcel Books, 2001.

W. Li,A. Kanso and A. Gherbi*,* **Leveraging Linux Containersto Achieve High Availability for Cloud Services***,* Cloud Engineering (IC2E), 2015 IEEE International Conference on*,* Tempe, AZ, 2015, pp. 76-83.

WILLIAMSON, Matt. **PFsense 2 CookBook.** Birmingham: Packt Publishing Ltd., 2011. 240 f.

APÊNDICE(S)

APÊNDICE A – Artigo científico

ESTUDO SOBRE ALTA DISPONIBILIDADE COM BALANCEAMENTO DE CARGA UTILIZANDO O FREEBSD PFSENSE

Valter Blauth Junior1, Willian Jefferson Meurer2

[1]Professor do Curso Ciências da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC-Brasil

[2]Acadêmico do Curso Ciências da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC-Brasil

valterblauth@unesc.net[1], willian@rednetinfo.com.br[2]

Abstract. *In this article introduced concepts, tools, configurations and techniques to provide high availability and load balancing of the Internet through a firewall. It was used in the implementation, the PFsense operating system, which is a FreeBSD distribution, which in turn is an open source system, also the CARP (Common Address Redundancy Protocol) and PFsync (Synchronization Protocol) were used to implement the high cluster. This research shows that it is possible to find in software free software-level solutions to create an environment of high availability and load balancing efficiently without the need to acquire software licenses.*

Resumo*. No presente trabalho apresentam-se conceitos, ferramentas, configurações e técnicas para prover alta disponi¬bilidade e balanceamento de carga da Internet através de um firewall. Foi utilizado na implementação, o sistema operacional PFsense, que é uma distribuição FreeBSD, que por sua vez é um sistema open source, também foi utilizado o CARP (Protocolo de redundância) e o PFsync (Protocolo de sincronismo) para implementação do cluster de alta disponibilidade. Esta pesquisa mostra que é possível encontrar no software livre soluções em nível de software para criação de um ambiente de alta disponibilidade e balanceamento de carga de forma eficiente, sem a necessidade de aquisição de licenças de softwares.*

1. Introdução

A Internet é uma rede de computadores que interconecta milhares de dispositivos computacionais ao redor do mundo. Há pouco tempo esses dispositivos eram basicamente desktops e servidores que armazenavam e transmitiam informações. A Internet de hoje é provavelmente o maior sistema de engenharia já criado pela humanidade, milhares de usuários conectam-se por meio de telefones celulares, e dispositivos como webcams, sensores, TVs e até mesmo máquinas de lavar vem sendo conectadas a essa rede (KUROSE, 2010). Serviços como armazenamento, processamento e recuperação de informações têm sido amplamente utilizados usando esse recurso, principalmente nas organizações que utilizam rede de computadores. A computação em nuvem tem mudado o paradigma de armazenamento e manipulação de dados e aplicações por todos os tipos de consumidores: o que antes era feito

no desktop do cliente, atualmente tende a ser executado em uma nuvem (HAN; XU; YAO, 2009).

Para garantir que o acesso a esses serviços seja de forma segura e que não ocorram interrupções, visto que a indisponibilidade da Internet pode gerar prejuízos tanto financeiro quanto de imagem as organizações, uma solução possível é a utilização de um firewall baseado em software livre que tenha suporte a alta disponibilidade.

A alta disponibilidade caracteriza-se por um sistema projetado para impedir perda de um serviço por ele disponibilizado, reduzindo ou gerenciando falhas, diminuindo o tempo de desligamento planejado para manutenção (BRASIL, 2006).

Sendo assim surge a oportunidade de realizar uma pesquisa com orientação acadêmica sobre alta disponibilidade e balanceamento de carga utilizando a distribuição FreeBSD PFsense para implementar esses serviços a fim de efetuar um estudo e testes, com intuito de verificar se a ferramenta é satisfatória para resolver estes problemas sem a necessidade de aquisição de licenças de softwares, ou equipamentos, como firewall ou roteadores. Também disponibilizar essa pesquisa para administradores de redes ou profissionais da área que buscam por soluções desse gênero. Este trabalho tem como objetivo principal estudar o comportamento da alta disponibilidade e balanceamento de carga de um firewall em uma rede local.

2. Cluster

O termo cluster começou a ser difundido na década de 60, quando a IBM interligou dois de seus mainframes para realizarem tarefas coordenadamente (PITANGA, 2008). Segundo Dantas (2005), as configurações de cluster, em termos computacionais podem ser entendidas como uma agregação de computadores de forma dedicada ou não. A partir desse princípio tem-se a ideia de Sistemas Distribuídos que para Tanenbaum (2003), consiste em computadores autônomos que trabalham juntos para dar a aparência de um único sistema coerente. Com a ideia inicial de se obter maior poder de processamento, um cluster pode ser formado por milhares de unidades de processamento, onde se pode distribuir ou replicar a execução de um programa através dessas unidades de processamento (COULOURIS, 2007).

Yeo et al (2006) e Pitanga (2008), afirmam haver duas divisões distintas de clusters, os de alto desempenho, implementados para prover um alto poder de processamento; e os de alta disponibilidade, focados em suprir, por meio de redundância de equipamentos, eventuais falhas físicas em componentes do cluster, visando manter recursos e serviços disponíveis sempre.

Os objetivos principais para a formação de clusters são alta disponibilidade ou alta performance. Ainda é possível a situação onde o cluster deve atingir os dois objetivos juntos.

2.1. Cluster de Alta Disponibilidade

Conforme Lopes Filho (2008), em sistemas de informações, pode-se dizer que o serviço proposto ficará sempre disponível independentemente da hora, do dia ou local, a alta disponibilidade tem por seu maior objetivo a continuidade do serviço sem interrupções por longos períodos de tempo.

Para Santos (2004), existem diferentes métodos e esquemas para prover a capacidade dos sistemas continuarem funcionando mesmo na ocorrência de falhas. O comportamento de clusters de alta disponibilidade deve contar basicamente com:

> a) relógios sincronizados: é um requisito para algumas aplicações, pois a diferença entre os relógios dos nós pode levar a problemas de coordenação do cluster;

b) repositório estável: um sistema de armazenamento comum, com sua integridade do conteúdo preservada mesmo em caso de falhas, estável para as aplicações, sendo capaz de tolerar os tipos de falhas mais prováveis e comuns;

c) detecção de defeitos e diagnóstico de falhas: aplicações e métodos, em sua maioria, assumem que na ocorrência de falha em um nó, outros nós perceberão esta falha dentro de um período de tempo finito, e então executarão as medidas de reparo necessárias;

d) entrega confiável de mensagens: perdas de mensagens e erros em meios de comunicação podem ocorrer. Portando são necessários mecanismos para garantir que uma mensagem enviada por um nó chegue incorrupta ao seu destino e que a ordem das mensagens seja preservada entre dois nós.

2.2 Disponibilidade

Segundo Brandão (2013), em informática a disponibilidade é entendida como o espaço de tempo em que estes serviços estão disponíveis, por exemplo, 12 horas por dia, 5 dias por semana. A disponibilidade pode ser calculada através de uma fórmula simples:

$$\boldsymbol{D = \frac{TUP - DT}{TUP}}$$

Onde,

D é a Disponibilidade; **TUP** é o Total de Unidade de Tempo; **DT** é o DownTime (tempo parado).

Para exemplificar, considere-se o cálculo da disponibilidade de um equipamento (servidor) em um período de 1 ano, levando em consideração 3 horas de DT (DowTime) para manutenção.

TUP = 1 ano ou 8760 horas **DowTime = 2 horas**

Aplicando a fórmula, é obtido o seguinte resultado:

$$\mathbf{D = \frac{8760 - 2}{8760} = 99,9771\%}$$

neste caso, tem-se uma grande disponibilidade do serviço, sendo que chegar aos 100 % é praticamente impossível por inúmeros fatores. Este tempo indisponível do equipamento é medido em porcentagem, então quanto mais noves (9), da direita para a esquerda no resultado da fórmula da disponibilidade, aparecer no cálculo, menor será o DowTime. Para Brandão (2013), isso representa "O significado dos noves" que estão indicados na tabela 1.

Tabela 1 – Significado dos Noves

%	Downtime (por ano)
100	Nenhum downtime
99.999 (5 noves)	Menos de 5.26 min.
99.99 (4 noves)	De 5.26 a 52 min.
99.9 (3 noves)	De 52 min. a 8 horas e 45 min.
99 (2 noves)	De 8 horas e 45 min. a 87 horas e 56 min.
90.0-98.9 (1 nove)	87 horas e 56 min. a 875 horas e 54 min.

Fonte: BRANDÃO (2013).

2.3. Balanceamento de Carga

Os sistemas de balanceamento de carga são geralmente a repartição da execução do serviço por várias máquinas. Estas soluções podem se especializar em pequenos grupos sobre os quais se faz um balanceamento de carga: utilização da CPU, de armazenamento, ou de rede. Qualquer uma delas faz-se o uso do conceito de clustering, já que o balanceamento será, provavelmente, feito para vários servidores (BRASIL, 2006).

Um cluster de aplicação é um conjunto de servidores que respondem coletivamente por um serviço. Esse conjunto de servidores pode dividir entre si a carga do sistema, ou simplesmente aguardar um dos servidores falhar para assumir o serviço. Esta tecnologia tem sido muito utilizada na Internet, como em sites de portais, e-commerce, e-business, abrangendo desde situações onde é necessário tratar milhares de requisições simultâneas, até a demanda do serviço que exige 99.99% do tempo on-line, por exemplo, (BRASIL, 2006).

A alta disponibilidade estudada neste trabalho visou, através de redundância de recursos de hardware e software, tornar a Internet disponível o maior tempo possível e automatizando de forma transparente a recuperação de eventuais falhas.

O firewall de alta disponibilidade foi implementado por um cluster composto por dois computadores com os recursos de hardware necessários e os softwares devidamente instalados e configurados e dois links de Internet de operadoras diferentes a fim de obter-se a alta disponibilidade não apenas nos servidores como também nos links, utilizando-se técnicas de balanceamento de carga nos mesmos com o objetivo de obter-se melhor desempenho quanto ao uso da Internet na rede interna.

3. Carp

O Protocolo de Redundância de Endereço Comum ou Common Address Redundancy Protocol (CARP), com suporte para IPV6 ou IPV4, conforme o Manual do Freebsd é o responsável pelo sincronismo entre os servidores conectados em alta disponibilidade. Esta ferramenta tem por objetivo prover a redundância do sistema com vários computadores compartilhando em uma única interface virtual de rede entre eles, garantindo o sincronismo de estado, a fim de obter-se a alta disponibilidade entre os servidores (FREEBSD, 2013).

A tecnologia tem como base um endereço IP flutuante entre os computadores que fazem parte do grupo. Um integrante do grupo é escolhido como mestre e fica respondendo todos os pacotes destinados ao grupo, os outros ficam aguardando para serem substitutos automáticos do mestre, não importando o endereço IP ou endereço MAC da interface física. O mestre utiliza o IP para comunicar a sua situação ao grupo e a sua frequência é zero, sendo esse o valor de maior prioridade. Caso estiver indisponível por um determinado período de tempo, todos do grupo são comunicados e a máquina que está programada com um valor de frequência menor e acima de zero torna-se o novo mestre. Quando o mestre volta a ser ativo na rede, tornar-se novamente assumir como mestre, retomando todas as funcionalidades, desta forma, os clientes da rede não são afetados e os serviços continuam disponíveis (FREEBSD, 2013).

A figura 1 exemplifica uma configuração, onde um gateway com redundância faz a separação da rede externa da rede interna. Uma placa de rede dos gateways está com o IP 10.0.0.1 e com o nome de “carp0” e faz a ligação dos gateways com a rede externa, a segunda placa de rede dos gateways com o IP 192.168.100.1 e de nome “carp1”, sendo responsável

pela comunicação com a rede interna. A terceira placa de rede dos gateways com o IP 172.16.1.1/172.16.1.2 faz as consultas e a sincronização entre os gateways.

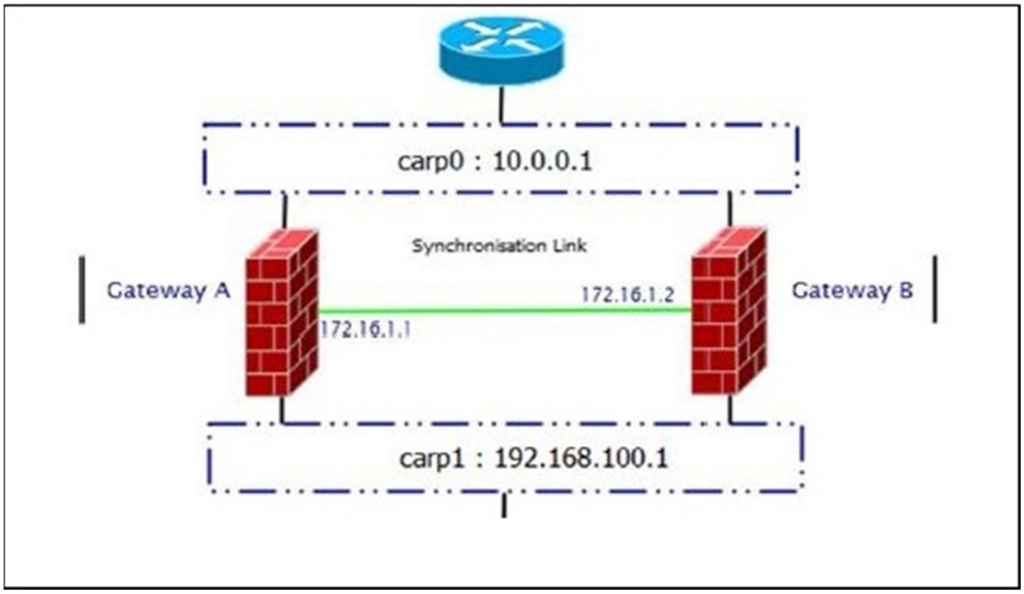


Figura 1. Exemplo de configuração do Carp
Fonte: Nomoa, 2014

4. Resultados

A alta disponibilidade estudada neste trabalho visou, por meio de redundância de recursos de hardware e software, tornar um firewall e a Internet disponíveis o maior tempo possível em uma rede local, recuperando de forma transparente eventuais falhas. Também teve como objetivo o uso de técnicas de balanceamento de carga para o consumo de Internet na rede interna. O firewall de alta disponibilidade foi implementado por um cluster composto por dois computadores onde cada computador possuía quatro placas de redes instaladas, dois links de Internet de operadoras diferentes a fim de obter-se a alta disponibilidade não apenas nos servidores como também nos links. Para que fosse possível a implementação deste trabalho foi necessário a instalação do sistema FreeBSD PFsense versão 2.2.4 nas duas máquinas.

A figura 2 exemplifica como foi montado os dois computadores para implementação do cluster de alta disponibilidade, bem como os IPs reais e virtuais de cada interface, sendo o da esquerda o servidor principal chamado de Master e o da direita como servidor secundário chamado de Backup.

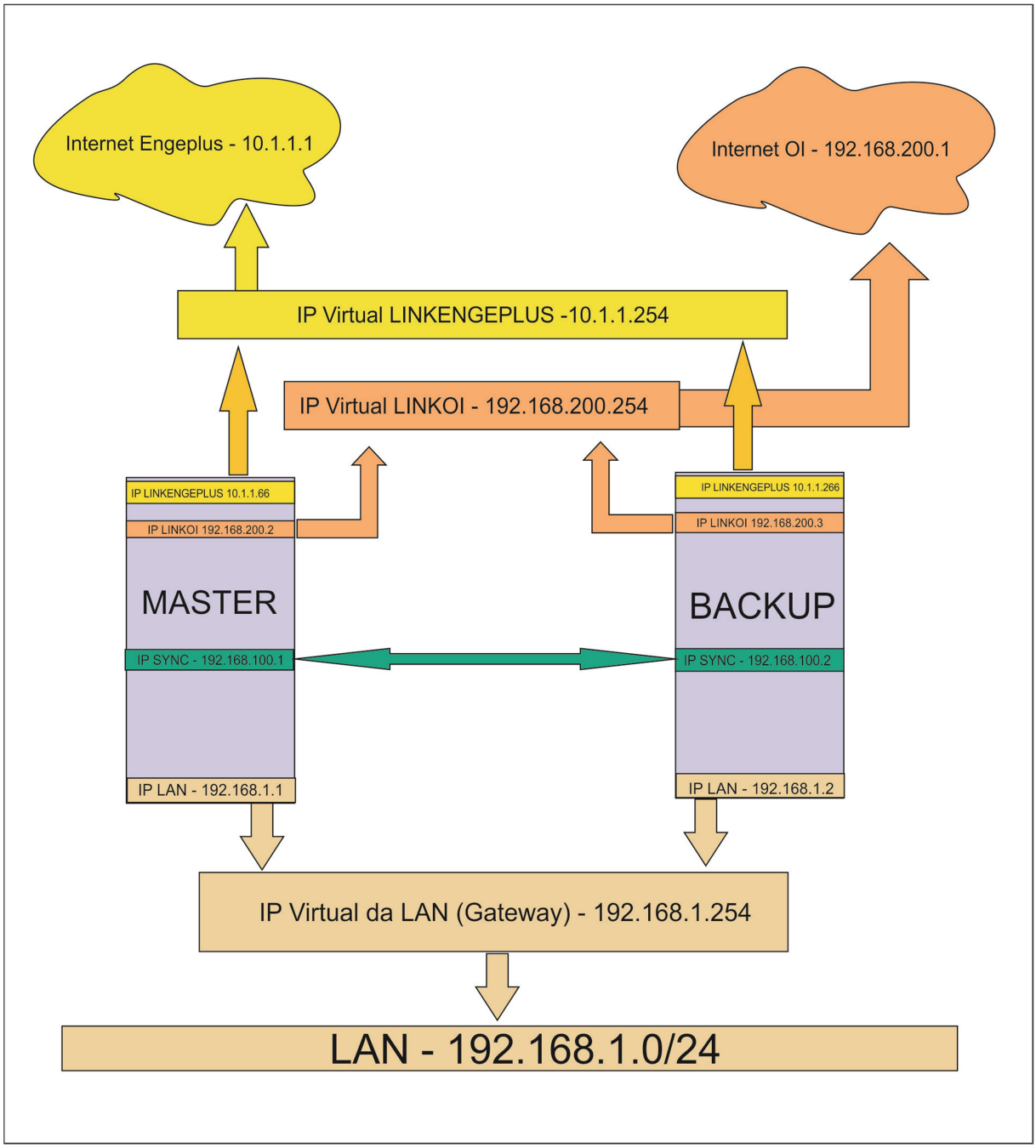


Figura 2. Desenho da estrutura
Fonte: Do autor

A figura 3 mostra a configuração do IP virtual, do tipo CARP, realizado na interface LAN, onde se atribuiu o IP 192.168.1.254, essas configurações foram efetuadas nas duas máquinas de forma que o IP 192.168.1.254 ficasse flutuante.

O campo “Advertising Frequency” é quem define a prioridade de ser o mestre ou escravo, quanto menor o valor da frequência, maior na hierarquia para assumir o lugar do mestre, o mestre sempre assume o valor “0”, definido na configuração “Skew”.

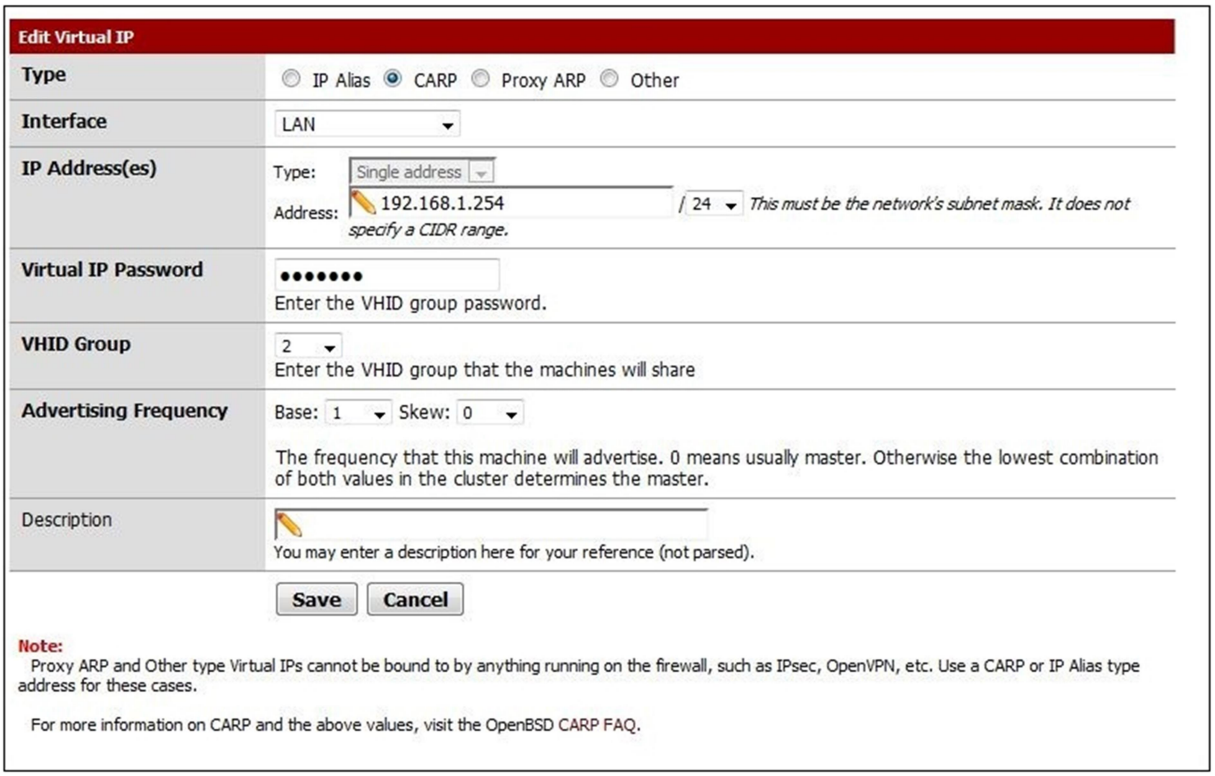


Figura 3. Exemplo de configuração do IP Virtual
Fonte: FreeBSD

4.1 Configurações do failover e balanceamento de carga dos links

A configuração do Failover e Load Balance foi realizada adicionado um novo grupo de gateway contendo os seguintes parâmetros, conforme ilustra a figura 4: Group Name refere-se ao nome da regra, que no presente trabalho foi chamada de “LB”, a opção Gateway Priority é onde se define as prioridades dos links bem como seu peso, neste trabalho optou-se em definir como Tier 1 para as duas interfaces, isto quer dizer um balanceamento de 50% (cinquenta por certo) por exemplo: os usuários ao se conectarem na Internet, se foi realizada pelo “LINKENGEPLUS” a próxima conexão será pelo “LINKOI”, porém pode ser ajustado pesos diferentes caso um link seja melhor do que o outro, exemplo caso o “LINKENGEPLUS” seja ajustado para Tier 2 e o “LINKOI” Tier 1, o acesso a Internet será de duas conexões pelo “LINKENGEPLUS” para uma conexão pelo “LINKOI” . A opção Tigger Level é onde se configura o Failover, no presente trabalho optou-se em Package loss or Higth Latency ou seja caso haja perda de pacote ou alta latência em algum gateway o outro deve assumir. A opção Description é apenas para descrever com mais detalhes a regra em questão, este campo aceita qualquer caractere.

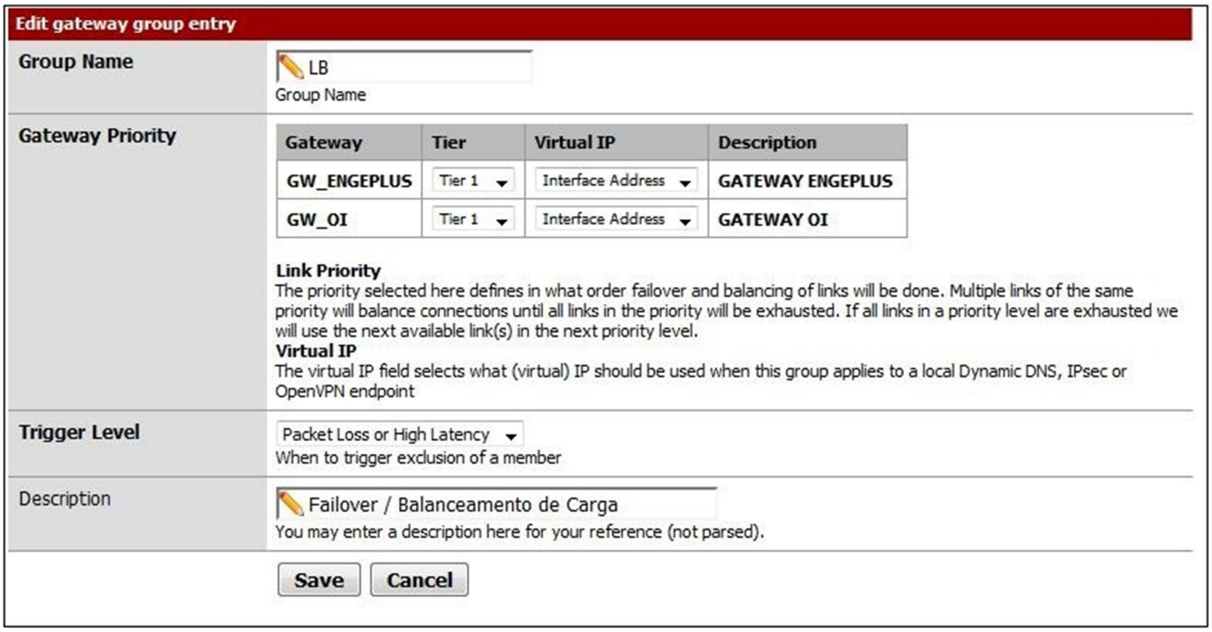


Figura 4. Exemplo de configuração do Balanceamento de carga entre os links
Fonte: FreeBSD

Para que os hosts conectassem a Internet de forma balanceada e com redundância dos links foi necessário o ajuste nas regras do firewall onde é possível definir para cada host, qual Gateway ou Grupo de Gateway ele usará. Como o foco dessa pesquisa é a alta disponibilidade e balanceamento de carga, criou-se apenas uma regra para rede local, onde foi ajustado que todas as conexões realizadas tenham o "LB" como Gateway.

4.2 Testes de redundância dos links

Esse teste foi realizado com os dois servidores Master e Backup e os dois links "LINKENGEPLUS" e "LINKOI" ligados, e a rede local utilizando a Internet, em um dado momento foi desligado um link, o "LINKOI", simulando uma falha, após ser desligado verificou-se se houve interrupção da Internet, mais especificamente na navegação, downloads que estavam ativos e downloads pos-desligamento do link. O mesmo teste foi repetido, porém com desligamento do link "LINKENGEPLUS".

A figura 5 traz a informação de um teste de Ping que foi disparado no endereço google.com onde é possível observar a perda de pacotes durante a troca dos links.

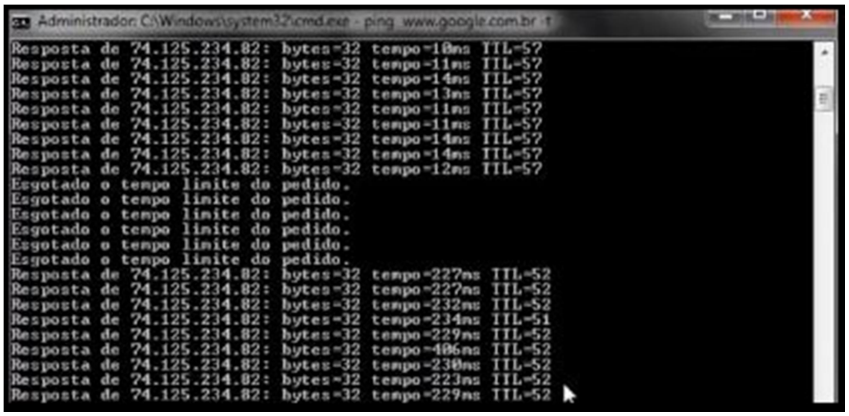


Figura 5. Teste de conexão utilizando o Ping
Fonte: Microsoft

Mesmo com a perda desses pacotes a redundância dos links funcionou perfeitamente de forma que os usuários não perceberam o problema no link. Já nos downloads que estavam em execução, que haviam sidos iniciados pelo o link que foi derrubado, houve interrupção com a troca do link, exigindo assim o usuário reiniciar os downloads manualmente, a mesma regra segue para os downloads pós desligamento do link, ou seja, se houver troca de link com downloads em andamento, o mesmo deve ser reiniciado manualmente pelo o usuário.

A tabela 2 explica que no momento em que ocorreu a interrupção do link "LINKOI", houve uma falha no download que havia sido inicializado por esta interface, já os downloads ativos que foram iniciados pelo link "LINKENGEPLUS" não apresentaram nenhuma falha. Ocorreu um erro no balanceamento de carga também, pois para haver o balanceamento os dois links devem estar ativos.

Tabela 2 – Resultados com desligamento do LINKOI

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Sim	-	Ok	Ok
BACKUP	Sim	-	Ok	Ok
LINKOI	Não	Falha	Ok	Falha
LINKENGEPLUS	Sim	Falha	Ok	Ok

Fonte: Do autor

Já na tabela 3, observa-se que no momento em que ocorreu a interrupção do link "LINKENGEPLUS", houve uma falha no download que havia sido inicializado por esta interface já os downloads ativos que foram iniciados pelo link "LINKOI" não apresentaram nenhuma falha. Houve falha no balanceamento de carga, pois para que haja o balanceamento são necessários os dois links funcionando, entretanto a redundância dos links funcionou como planejada.

Tabela 3 – Resultados com desligamento do LINKENGEPLUS

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Sim	-	Ok	ok
BACKUP	Sim	-	Ok	ok
LINKOI	Sim	Falha	Ok	ok
LINKENGEPLUS	Não	Falha	Ok	Falha

Fonte: Do autor

O tempo de recuperação das falhas dos links, tanto o "LINKENGEPLUS" quanto o "LINKOI" foi de 29 segundos, conforme apresenta a tabela 4.

Tabela 4 – Tempo de recuperação de falhas nos *links*

SERVIDORES/ LINKS	TEMPO (EM SEGUNDOS)
MASTER	-
BACKUP	-
LINKOI	**29s**
LINKENGEPLUS	**29s**

Fonte: Do autor

Aplicando a formula da disponibilidade, considerando que o tempo de teste foi de oito dias (691,200 segundos) e que houve apenas uma ocorrência de queda de link obteve-se o seguinte resultado:

$$\mathbf{D = \frac{691200 - 29}{691200}}$$

D = 99,9958043981%, Neste caso foi obtido uma grande disponibilidade, chegando próximo a 100%.

4.3. Testes de alta disponibilidade dos servidores firewalls

Esse teste foi realizado com os dois servidores (Master e Backup) ligados e com a rede local (LAN) utilizando a Internet, em um dado momento foi desligado o servidor Master a fim de simular uma falha. Antes do desligamento foi disparado o comando Ping com a opção –t, no IP virtual flutuante da LAN 192.168.1.254, sendo este o IP do Gateway utilizado nos hosts na LAN. Observou-se que na transição dos servidores houve uma perda de pacote conforme ilustrado na figura 6.

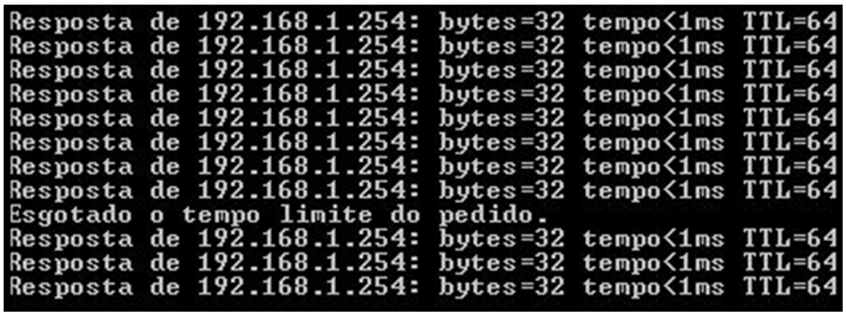


Figura 6. Teste de conexão com o Ping no IP virtual
Fonte: Microsoft

Com o desligamento do servidor Master o servidor Backup passou a assumir a posição de principal de forma transparente, executando todas as regras e configurações que haviam sido definidas para o sincronismo entre os servidores, deste modo não houve interrupção nos serviços de Internet na rede local, com exceção os downloads que estavam em execução que foi necessário reiniciá-los manualmente.

Observou-se que acessando o IP 192.168.1.2, que é o endereço do servidor "BACKUP", no Status do CARP, onde antes do desligamento do servidor Master, o status do servidor trazia a informação "BACKUP", passou a assumir o status de "MASTER".

A tabela 5 exemplifica o momento que foi desligado o servidor Master onde houve falha nos downloads em andamento, porem a redundância dos servidores funcionou perfeitamente. Quanto ao balanceamento de carga entre os links não houve erro, pois os mesmos continuaram ativos.

Tabela 5 – Resultados com desligamento do servidor *Master*

SERVIDORES/ LINKS	LIGADO	BALANCEAMENTO DE CARGA	NAVEGAÇÃO	DOWLOADS ATIVOS
MASTER	Não	-	Ok	Falha
BACKUP	Sim	-	Ok	Ok
LINKOI	Sim	OK	Ok	Ok
LINKENGEPLUS	Sim	OK	Ok	Ok

Fonte: Do autor.

O tempo de recuperação de falhas nos servidores, tanto no Master quanto no Backup, foi de 2 segundos, conforme apresenta a tabela 6.

Tabela 6 – Tempo de recuperação de falhas nos servidores

SERVIDORES/ LINKS	TEMPO (EM SEGUNDOS)
MASTER	2s
BACKUP	2s
LINKOI	-
LINKENGEPLUS	-

Fonte: Do autor.

Aplicando a formula da disponibilidade, considerando que o tempo de teste foi de oito dias (691,200 segundos) e que houve somente uma ocorrência de falha no servidor, obteve-se o seguinte resultado:

$$\mathbf{D = \frac{691200 - 2}{691200}}$$

D = 99,9997106481%, Neste caso foi obtido uma grande disponibilidade, chegando próximo a 100%.

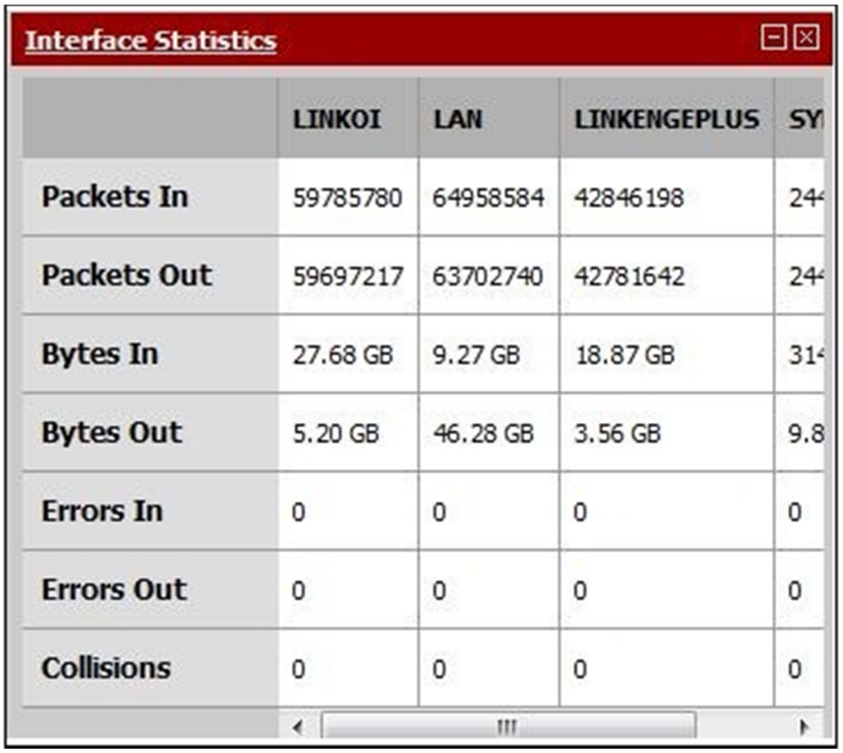
4.4. Testes de balanceamento de carga

Os testes do Balanceamento de Carga entre os links de Internet foram realizados num período de oito dias com todos os computadores da rede interna consumindo a Internet para navegação, envio e recebimento de emails e downloads.

Durante esses oito dias perceberam-se um melhor desempenho quanto a velocidade para a navegação, envio e recebimento de emails e downloads, pois com o uso da técnica de balanceamento de carga, toda a carga de consumo da Internet foi distribuída entre os dois links e com isso foi possível minimizar o tempo de resposta e evitou sobrecarga.

Utilizando múltiplos componentes com o balanceamento de carga ao invés de um único componente além de otimizar os recursos, pode aumentar a confiabilidade através da redundância.

A figura 25 traz a informação através de estatísticas de consumo de carga de cada interface, disponibilizada pelo próprio PFsense, que houve um consumo de forma balanceada entre as interfaces “LINKOI” e “LINKENGEPLUS” no período de oito dias, onde a interface “LINKOI” teve como entrada 27,68 GB e 5,20 GB como saída e a interface “LINKENGEPLUS” teve 18,87 GB entrada e 3,56 GB de saída.



	LINKOI	LAN	LINKENGEPLUS	SY
Packets In	59785780	64958584	42846198	24
Packets Out	59697217	63702740	42781642	24
Bytes In	27.68 GB	9.27 GB	18.87 GB	31
Bytes Out	5.20 GB	46.28 GB	3.56 GB	9.8
Errors In	0	0	0	0
Errors Out	0	0	0	0
Collisions	0	0	0	0

Figura 7. Estastísticas de entrada e saída de pacotes nas interfaces
Fonte: FreeBSD

Com o uso do balanceamento de carga entre os links foi constatado que alguns sites, principalmente sites de bancos que exigem que quando é realizada a conexão pelo um determinado IP, este não seja alterado durante a operação.

Para resolver esse problema, foi necessário acessar o menu System – Advanced – Miscellaneous e marcar a opção “use sticky conections”. Ao marcar essa opção conexões sucessivas são redirecionadas para os servidores de um modo round-robin, com conexões a partir da mesma fonte a serem enviados para o mesmo servidor web. Essa "conexão fixa" vai existir enquanto houver estados que se referem a esta conexão. Uma vez que o estado expirar, outras conexões a partir do host serão redirecionadas para o próximo servidor web de modo round robin.

5.Conclusão

Neste trabalho foram estudados os principais conceitos de rede de computadores, firewall e cluster de alta disponibilidade e balanceamento de carga. Foram também apresentadas ferramentas que implementam cluster de alta disponibilidade e balanceamento de carga em um firewall baseando em FreeBsd, mais especificamente o PFsense. A partir de conceitos de cluster, foi implementado um sistema de alta disponibilidade de firewall por meio de redundância de máquinas, fazendo uso do Protocolo de Redundância de Endereço Comum (CARP). Foi utilizado também o Failover e Load Balance para a redundância e balanceamento de carga dos links, deste modo aparentou para a rede local, um único sistema coerente de Internet, foi possível observar um melhor desempenho nas conexões.

Uma das vantagens do uso de balanceamento de carga é que além de dividir a carga, que no presente trabalho foram as conexões com a Internet em uma rede local, é possível obter a alta disponibilidade dos links, através da redundância.

Conclui-se nesse estudo que a alta disponibilidade não é apenas um produto ou um software específico, mas a um conjunto de mecanismos e técnicas que tem por objetivo detectar, contornar de forma transparente falhas que venham a ocorrer ocasionando perda de acessibilidade.

A distribuição PFSense, mostrou-se uma ótima ferramenta open-source para implementação de firewall com ou sem alta disponibilidade e balanceamento de carga. Através de sua interface web, fica mais fácil e prático o gerenciamento da rede, porém não dispensou um estudo aprofundado sobre rede de computadores e como funciona um firewall, pois um firewall mal configurado compromete toda a segurança de uma rede.

Após os estudos realizados neste trabalho, conclui-se que os objetivos de alta disponibilidade e balanceamento de carga dos links de Internet, utilizando software livre para um firewall, foram atingidos, sendo esta uma solução possível para empresas ou organizações que buscam por soluções de alta disponibilidade e balanceamento de carga de Internet em uma rede local.

Este estudo mostrou que é possível criar um ambiente de alta disponibilidade sem a necessidade da aquisição de licenças de softwares, preocupando-se apenas nos hardwares envolvidos como computadores, switches e no-breaks, sendo que, alguns hardwares, como computadores podem ser reaproveitados, uma vez que foram descartados para outras funções, a o invés de serem jogados no lixo, muitas vezes em lugares impróprios, podem ser utilizados para algumas tarefas específicas como, por exemplo, na utilização de projetos de cluster de alta disponibilidade ou alta performance.

Um fator importante para o êxito da alta disponibilidade é a energia elétrica, que não foi abordado nesse estudo, contudo é relevante um estudo para trabalhos futuros.

6. Referências

BRANDÃO, Robson. Conceitos sobre Disponibilidade - Parte I. Disponível em: <http://technet.microsoft.com/pt-br/library/cc668492.aspx>. Acesso em: 09 Abr. 2016.

COULOURIS, George et al. Sistemas Distribuídos: conceitos e projeto. 5. ed., Rio de Janeiro: Bookman, 2013.

Guia de Estruturação e Administração do Ambiente de Cluster e Grid. Brasília, 2006. 454 p. : il.

HAN, L.; XU, Z.; YAO, Q. An approach to web service composition based on service-ontology.In: Fuzzy Systems and Knowledge Discovery, 2009. FSKD 09. Sixth International Conferenceon. [S.l.: s.n.], 2009. v. 2, p. 173 –177.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet: uma abordagem top-down. 5. ed. São Paulo: Addison Wesley, 2010. 618 f.

LOPES FILHO, Edmo.Arquitetura de Alta Disponibilidade para Firewall e IPS baseada em SCTP. Dissertação (Mestrado)-Universidade Federal de Uberlândia, Minas Gerais, 2008.

MANUAL do FREEBSD: O Projeto de Documentação do FreeBSD. Revisão: 42953. Disponível em: <http://www.freebsd.org/doc/en_US.ISO8859-/books/handbook/carp.html>. Acesso em: 12 de outubro 2015.

NOMOA. Hot Failover with CARP - Parte I. Disponível

em:http://nomoa.com/bsd/gateway/highavailability/carp.html Acesso em: 13 jun. 2016.

SANTOS, Daniel Campos dos. Alta Disponibilidade de Serviços de Rede. 2004:48 f. TCC (Graduação) Curso de Ciência da Computação,Universidade Federal de Santa Catarina, Florianópolis, 2004.

TANENBAUM, Andrew S.; Redes de Computadores. 4. ed., Rio de Janeiro: Elsevier,2003.