JETMA Jornal de Engenharia, Tecnologia e Meio Ambiente

IMPLANTAÇÃO DE UM BALANCEAMENTO DE CARGA DINÂMICO

IMPLEMENTATION OF A DYNAMIC LOAD BALANCE

VANDERLEY COSTA DE LIMA JÚNIOR[1]
EDUARDO FERNANDES SAAD[2]
DIOVANI DOS SANTOS MILHORIM[3]

FACTHUS
FACULDADE DE TALENTOS HUMANOS.
UBERABA (MG)

[1] e-mail: junior_delima@outlook.com
[2] e-mail: edusaad@hotmail.com
[3] e-mail: diovani.milhorim@facthus.edu.br

AUTOR CORRESPONDENTE
CORRESPONDING AUTHOR

DIOVANI DOS SANTOS MILHORIM
FACTHUS - Faculdade de Talentos Humanos
Rua Manoel Gonçalves de Rezende, 230 - Bairro Vila São Cristóvão - Uberaba/MG
e-mail: diovani.milhorim@facthus.edu.br

RESUMO:
Atualmente, os recursos tecnológicos nos meios corporativos estão em grande crescimento, deste modo, o uso da internet torna-se essencial para a funcionalidade dos mesmos. No entanto, as empresas que utilizam aplicações podem não suportar a alta demanda de informações em sua conexão devido a elevada quantidade das mesmas. O presente projeto tem como objetivo demonstrar uma solução utilizando um algoritmo com intuito de efetuar um balanceamento de carga dinâmico entre dois ou mais links de internet, garantindo uma alta disponibilidade e qualidade. Para a implantação, utilizou-se um computador de baixo custo com duas placas de redes, dois links de internet de qualidades diferentes e recursos open source. Para a análise do seu funcionamento foram realizados testes que necessitavam de um alto consumo de recursos de internet. Utilizou-se um cálculo de balanceamento de carga que para tratar as conexões e assim determinar os pesos automaticamente. Portanto, o projeto demonstrou grande eficiência no processamento das requisições, obtendo sucesso no balanceamento e aliviando a carga da rede.

PALAVRAS-CHAVES:
ptables; iproute; debian; shell script; Balanceamento de carga.

ABSTRACT:
Nowadays, the technological resources in the corporate means are in great growth, in this way, the use of the internet becomes essential for their functionality. However, companies that use applications may not support the high demand for information on their connection, due to their high number of applications. The present project aims to demonstrate a solution using an algorithm to dynamically load balance between two or more internet links, ensuring high availability and quality. For the implementation, a low-cost computer with two network cards, two internet links of different qualities and open source resources were used. For the analysis of its operation were carried out tests that needed a high consumption of internet resources. A load balancing calculation was used to treat the connections and thus determine the weights of the connections automatically. Therefore, the project demonstrated great efficiency in the processing of the requisitions, obtaining success in the balancing and relieving the load of the network.

KEYWORDS:
iptables; iproute; debian; shell script; Load balance;

INTRODUÇÃO

Com a utilização da internet, cada vez mais as pessoas se conectam e por consequência a quantidade de dados cresce vertiginosamente. Segundo o relatório da Conferência das Nações Unidas sobre Comércio e Desenvolvimento (UNCTAD, 2017), o Brasil ocupa o quarto lugar com o maior número absoluto de usuários de Internet, ficando atrás de Estados Unidos, Índia e China. No mesmo sentido, é observado um significativo crescimento da utilização da internet, de acordo com a pesquisa da União Internacional de Telecomunicações, demonstra que no ano de 2000 os internautas eram 6,5% da população mundial. Já em 2015 esse índice subiu para 43% (UIT, 2015).

As empresas têm muita dificuldade de controlar o tráfego que sai para a rede mundial pois a quantidade de computadores só cresce e também o acesso. Uma das soluções encontradas é a utilização de 2 ou mais links como possível solução para melhorar a navegação por parte dos usuários. Esta solução cria um novo problema de utilização destas conexões devido ao congestionamento de uma delas, pois as velocidades dos links são diferentes, porém fixas. Outro aspecto existente é que alguns provedores de internet fornecerem um link de diferentes qualidades (ANDRIOLI e DA ROSA RIGHI, 2017).

Este estudo pretende a criação de um algoritmo com objetivo de executar um balanceamento de carga entre dois links de internet, para um garantir alta disponibilidade e qualidade. O resultado esperado é fornecer um balanceamento eficiente e instantâneo, evitando problemas com congestionamento da rede, demonstrado através de gráficos e tabelas.

REFERENCIAL TEÓRICO

BALANCEAMENTO DE CARGA

A internet transformou-se em algo bastante complexo, em que a rede deve ter capacidade de suportar milhares de requisições ao mesmo tempo. Assim, tornou-se imprescindível o uso do balanceamento de carga como mecanismo para rede de computadores. É indispensável que os recursos de uma rede sejam bem aplicados, para que a internet possa sustentar os grandes tráfegos e manter um tempo de resposta consideravelmente rápida para os usuários. O ideal é que cada equipamento ou servidor processe uma quantidade de informações correspondente a sua capacidade. Desta maneira, potencializando o rendimento, o balanceamento de carga criou uma forma de processar as requisições de modo que todos os recursos possam ser bem aproveitados, aumentando sua capacidade. Assim, o balanceamento em redes de computadores é uma técnica utilizada para compartilhar a tarefa em diversos links (UPPAL e BRANDON, 2010).

MÉTRICA *MINIMUM LOSS WITH ADDITIVE COST*

A métrica *Minimum Loss* se constitui em probabilidades de sucesso na transmissão de pacotes no nível de enlace. A ideia é obter rotas que minimizem a probabilidade de perda dos pacotes. A melhor rota entre dois nós será aquela com a maior probabilidade de sucesso na transmissão do pacote, uma vez que os pesos sejam atribuídos a todos os enlaces da rede (PASSOS, 2007). Para que exista probabilidade de sucesso na transmissão do pacote, pode-se utilizar uma expressão em que nela são adicionados parâmetros ajustáveis representando um custo. Estes parâmetros foram definidos por λ conforme equação.

$$MLAC_n = \left[\prod_{i=0}^{n-1}\left(\frac{1}{P_{a_i a_{i+1}}} + \lambda\right)\right]^{-1} \quad (1)$$

O significado de λ nessa expressão pode ser definido como o conjunto de fatores que influenciam no aumento da probabilidade de perda de pacotes na rede. Estes fatores provocam auto interferência dos nós em uma rota ou o consumo de recursos na rede.

DEBIAN

Iniciado em 1993 por Ian Murdock, o Debian foi criado como uma nova distribuição aberta no espírito do Linux e do projeto GNU. Um grupo pequeno de hackers de software livre cresceu e fortaleceu para se tornarem uma comunidade grande e organizada de desenvolvedores e usuários.

O Debian é um sistema operacional livre para seu computador. Sistema operacional é um conjunto de programas e utilitários que fazem seu computador funcionar (DEBIAN, 2017). No núcleo do sistema operacional está o *kernel*. *Kernel* é o programa mais fundamental no computador e faz todas as operações mais básicas, permitindo que você execute outros programas. Os sistemas Debian atualmente usam o *kernel* Linux ou o *kernel* FreeBSD. O Linux é uma peça de software criada inicialmente por Linus Torvalds com a ajuda de milhares de programadores espalhados pelo mundo. (DEBIAN, 2017)

IPTABLES

De acordo com a descrição do criador desse pacote (*NetFilter*), O Iptables é uma ferramenta para criar e administrar regras e assim filtrar pacotes de redes. O iptables pode funcionar baseado no endereço, porta de origem, destino do pacote, prioridade. Ele tem como função a comparação de regras para saber se um pacote tem ou não permissão para passar. Em firewalls mais restritivos, o pacote é bloqueado e registrado para que o administrador do sistema tenha discernimento sobre o que está acontecendo em seu sistema (MUNIZ, 2014).

Com o uso desta ferramenta determinamos regras especiais para entrada, saída e passagem de pacotes entre interfaces, podendo operar antes ou depois das ações referentes ao roteamento.

IPROUTE2

A base do pacote iproute2 é a ferramenta ip. Ela traz toda a funcionalidade existente nos comandos arp, ifconfig e route (BIANCHETTI, 2003).

SHELL SCRIPT

Shell Script é uma vasta ferramenta de automação de instruções. Com um arquivo de texto executável o usuário ou sistema é capaz de executar uma sequência de operações, instruções e testes.

Qualquer sequência de instruções, utilizada com regularidade e que seja capaz de ser automatizada pode ser implementada em *Shell Script* (VIRGÍLIO, 2007).

SSH – *SECURE SHELL*

O protocolo ssh (*secure shell*), foi desenvolvido basicamente para comunicação, administração de máquinas linux remotamente e transferir arquivos de diversas formas. As implementações de SSH usam como padrão a porta 22 do TCP/IP.

O SSH é dividido em dois padrões. O SSHD é o padrão servidor, um serviço que fica hospedado na máquina que será acessada, enquanto o SSH é o padrão do cliente, uma ferramenta que você utiliza para acessar a máquina.

Alguns equipamentos como cisco, utilizam o protocolo para acessar e modificar as configurações dos roteadores (REDERIO, 2001).

PUTTY

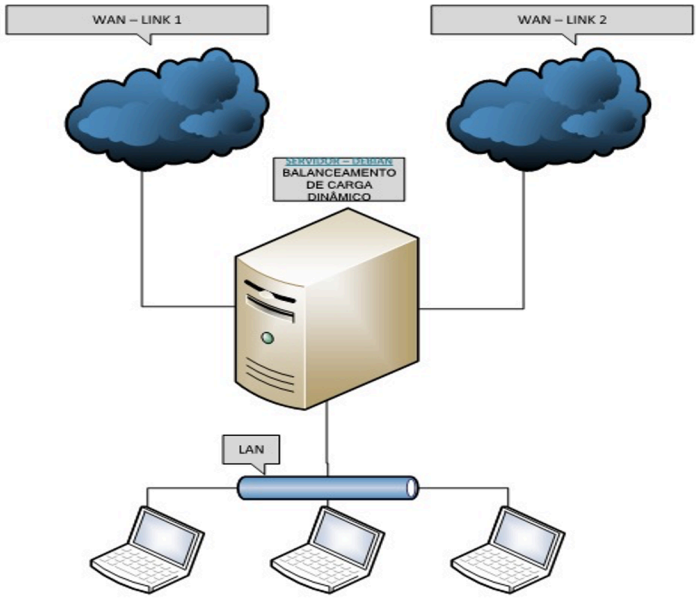
Putty é um software de simulação *open source*, onde foi desenvolvido para atuar como um cliente de conexões remotas. Ele é utilizado para fazer diversas conexões como, raw, telnet, rlogin, SSH e porta serial. Ideal para estabelecer conexões de acesso remoto a servidores via SSH, sendo assim necessário para efetuar comandos em terminal Linux (TECHTUDO, 2013).

MATERIAIS E MÉTODOS

O presente projeto foi desenvolvido utilizando um computador com processador AMD Athlon II x3 425 de 2.7Ghz, 2Gb de memória RAM, disco rígido de 250Gb, 3 placas de rede, dois links de internet (um link de 100 Mbps e um link de 8 Mbps). O Sistema Operacional escolhido foi o Debian 9.3 devido à várias facilidades de implantação da programação.

A Figura 1 apresenta a topologia de rede aplicada para criação do projeto.

Figura 1: Topologia de rede.



Fonte: O autor

Na primeira inicialização do sistema operacional, executa-se os comandos apt—get update para atualização dos pacotes do sistema operacional, apt—get install iptables para instalação da ferramenta *iptables* e em seguida, o comando apt—get install iproute para instalação da ferramenta *iproute*.

O sistema operacional define automaticamente na instalação, os endereços da interface de rede em modo automático dependendo de um serviço DHCP (*Dynamic Host Configuration Protocol*) para lhe fornecer o endereço IP. Entretanto, para este projeto existe a necessidade de as placas serem configuradas manualmente com endereços estáticos, definidos desta forma:

```
#Rede Lan
iface enp0s7 inet static
address 192.168.200.20
netmask 255.255.255.0

#Rede wan1

iface enp1s8 inet static
address 192.168.0.60
netmask 255.255.255.0

#Rede wan2

iface ensp1s10 inet static
```

```
address 192.168.20.2
netmask 255.255.255.0
```

Conforme configuração vista, existem três placas instaladas no servidor com tarefas específicas. Uma é responsável pela rede local (LAN) e as outras duas placas são responsáveis pela internet (WAN), sendo que uma recebe um link de 8Mbps e a outra um link de 100Mbps.

O servidor está localizado em outro ambiente, sendo necessário a instalação e configuração do serviço SSH para comunicação remota. Em seguida, ocorreu a definição da porta e IP de comunicação do servidor dentro do arquivo /etc/ssh/sshd_config. Conforme figura 2.

Figura 2: Configuração do SSH.

```
#       $OpenBSD: sshd_config,v 1.100 2016/08/15 12:32

# This is the sshd server system-wide configuration fi
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/

# The strategy used for options in the default sshd_co
# OpenSSH is to specify options with their default val
# possible, but leave them commented.  Uncommented opt
# default value.

Port 2222
#AddressFamily any
ListenAddress 192.168.0.60
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO
```

Fonte: O autor

o

Posteriormente, configura-se o balanceamento de carga permitindo que a rede trabalhe com dois links ao mesmo tempo. Cria-se as tabelas de roteamento no arquivo rt_tables, localizado em /etc/iproute2/rt_tables. Essas tabelas devem conter identificação e nome:

```
200 nome_tabela_1
201 nome_tabela_2
```

Para o funcionamento adequado do balanceamento é necessário que os *gateways* não estejam definidos no momento da configuração das placas de redes, assim o balanceamento será responsável por estabelecer o *gateway* padrão apropriado.

Foram criados dois *scripts* referente ao balanceamento e cálculo dos links que ficarão armazenados em uma nova pasta localizada na raiz do sistema operacional com o nome de "internet", gerada pelo administrador do servidor. Nesse momento foi definido no *script* balanceamento.sh pesos fixos no tráfego da rede. A figura 3 mostra o arquivo do balanceamento.

Figura 3: Primeira parte do arquivo de configuração balanceamento.sh.

```
# Dados da interfaces de rede conectadas com a internet - Link 1
placa_rede_1=enp1s8
tabela_1=link01
gateway_1=192.168.0.1
peso_balanceamento_link_1=1

# Dados da interfaces de rede conectadas com a internet - Link 2
placa_rede_2=enp1s10
tabela_2=link02
gateway_2=192.168.20.1
peso_balanceamento_link_2=5

# Dados da Rede Interna
placa_rede_interna=enp0s7
rede_interna=192.168.200.0/24

### Fim das Variáveis

#Carregar módulos
modprobe iptable_nat
modprobe ip_nat_ftp
modprobe ipt_MARK
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/ip_dynaddr

# Zero (0) para desativado e (1) para ativado
echo "1" > /proc/sys/net/ipv4/conf/default/rp_filter
echo "1" > /proc/sys/net/ipv4/conf/all/rp_filter

# Deletar a rota default
ip route del default

# Limpa as regras das tabelas
ip route flush table $tabela_1
ip route flush table $tabela_2
```

Fonte: O autor

No momento da criação do *script*, como demonstrado na figura 3, foi inserido variáveis com informações referentes às placas de redes de internet, tabelas de roteamento, *gateway* de cada rede e o peso dos balanceamentos. Em relação à rede local (LAN) foram determinadas variáveis que receberam informações da placa de rede e qual a informação da rede local.

Deste modo, são acrescentados no *script* diversos comandos de segurança da rede que executam os módulos do *iptables*; comando para habilitar o bloqueio de *syn cookies*; comando para habilitar o repasse de pacotes de *kernel* (roteamento) e comando para habilitar os repasses de pacotes de IPs dinâmicos.

No mesmo *script* adiciona-se o *rp_filter* fazendo com que o *firewall*

sempre responda aos pacotes na mesma interface de origem, prevenindo ataques diversos, onde podem tirar proveito da regra que permite conexões na interface *loopback*.

Para que o *script* deletasse a rota padrão, foi utilizado o comando ip route del default, assim o balanceamento é responsável por encaminhar os pacotes para outras rotas. Em seguida, adiciona-se o comando ip route flush table para limpar as regras das tabelas de roteamento.

Foi imprescindível adicionar no *script* comandos encarregados em definir as rotas para as tabelas, comandos para definir as regras para o balanceamento dos links e por último e essencial comando para compartilhar a conexão entre os links.

Após feito todo o procedimento no *script* responsável pelo balanceamento de carga, ajusta-se o arquivo balance.sh, sendo este o *script* referente ao cálculo dos links.

Em primeiro momento adota-se o site www.registro.br como referência, devido a grande disponibilidade e confiabilidade da conexão. Iniciou-se o *script* utilizando o comando ping, pois a partir dele que se colheu as informações necessárias para efetuar o cálculo. Ao analisar a rede foram identificados que a perda de pacotes e o RTT (*Round-Trip Time*) tem grande importância para a performance dos links. A figura 4 demonstra a forma de coleta das informações utilizando o cálculo de uma das interfaces.

Na outra placa com o nome de enp1s10, utiliza-se o mesmo cálculo.

Para o presente trabalho adotaremos como probabilidade de sucesso de transmissão em cada enlace a diferença entre um e a probabilidade de insucesso representado neste estudo pela perda média de pacotes no enlace. Conforme se observa na equação (2).

$$P_{sucesso} = 1 - \frac{\$perda}{100} \tag{2}$$

Adotaremos como fator aditivo (lambda) a relação baseada no tempo médio de retorno, como se vê abaixo na equação (3).

$$\lambda = \frac{\$rttmedio}{100} \tag{3}$$

Finalmente, conforme equação representada a seguir, para leitura de perda e rttmedio da velocidade do nó a ser calculado, carrega-se o valor obtido através do *ping* na variável $perda e $rttmedio. Obtemos a equação (4) utilizada para cálculo do peso de balanceamento de cada enlace.

$$peso = \left[\frac{1}{\left(1 - \frac{\$perda}{100}\right)} + \left(\frac{\$rttmedio}{100}\right) \right]^{-1} \tag{4}$$

No cálculo anteriormente representado, a variável $rttmedio foi definido como um fator que influencia no aumento da probabilidade de perda dos pacotes das redes. Portanto, é alterada a qualidade do link de conexão, modificando o resultado do cálculo, assim aplicando um novo peso no balanceamento dos links das redes.

O *script* é executado em um tempo pré-determinado para que analise o estado das conexões e refaça o cálculo, consequentemente ele verifica a qualidade dos links e se necessário estipula novas rotas. Para que esse processo aconteça foi editado o arquivo crontab, e nele configurado parâmetros que executa o *script* balance.sh de dois em dois minutos, armazenando os resultados em um arquivo de log do sistema na pasta /var/log/roteamento.log.

RESULTADOS E DISCUSSÃO

Após os procedimentos descritos anteriormente, foram realizados testes nos quais executa-se os *scripts* e analisa-se o arquivo *log* para verificação do funcionamento do cálculo, logo em seguida foi efetuado teste de desempenho de conexão da rede local. Na figura (5) demonstra o arquivo *log* e comprova o resultado dos cálculos executados pelo *script* balance.sh.

Com um procedimento de *streaming* de vídeo executando em um computador conectado na rede local, observou-se a efetividade do balanceamento de carga, onde nota-se que em determinados momentos o serviço mudava a rota sem afetar o desempenho do *streaming*. Demonstra-se na figura (6) a seguir nos horários 23:12 e 23:20, onde o balanceamento de carga era realizado com o processo de *streaming* em execução, a variação de velocidade de download dos links nestes tempos provocada pela alteração de pesos de cada link no balanceamento de carga.

Foram utilizados alguns parâmetros para tratar a qualidade das conexões, tais como perda de pacotes e *rttmedio*. Observa-se que a perda de pacote apresenta influencia no cálculo do peso do enlace. Nas figuras (7) e (8) no horário das 19:46 observarmos que apesar do tempo baixo de retorno, o peso do enlace 2 se mantém baixo (figura (9)) compensado pela alta perda de pacotes neste mesmo momento.

Já o *rttmedio*, teve grande importância no tratamento das conexões, com a variação do mesmo os pesos eram modificados frequentemente e com tais modificações a qualidade das conexões foram melhores ajustadas.

Observa-se que a cada momento de execução do *script* de cálculo, o

Figura 4: Cálculo da interface 1.

```
#################################################################################################################
#
#  calculo do peso para interface 1
#
#################################################################################################################
data=`date +%H:%M-%d/%m/%Y`

IF1="enp1s8"
IF2="enp1s10"

placa_rede_1=enp1s8
placa_rede_2=enp1s10

#################################################################################################################
#
#  calculo do peso para interface 1
#
#################################################################################################################

result1=`ping www.registro.br -c 10 -I $IF1`

echo "ping =" $result1

perdal=`echo $result1 | awk -F\, '{print $3 ; }' |awk -F' ' '{print $1 ; } ' | awk -F'%' '{print $1 ;}'`
rttmedio1=`echo $result1 | awk -F\, '{print $4 ; }' |awk -F' ' '{print $6 ; } ' | awk -F\/ '{print $2 ; } '`
pesoA1=`echo  "scale=4; 10*(1 /((1 / (1 - $perdal / 100)) + (($rttmedio1)/100) ))" | bc `
peso1=`echo "scale=0; $pesoA1/1 " | bc`

echo "######################### calculo de pesos link 1 ###################################################"

echo "perda 1=" $perdal
echo "rttmedio 1=" $rttmedio1

echo "Peso sem arredondar 1=" $pesoA1
echo "Peso arredondado 1=" $peso1
```

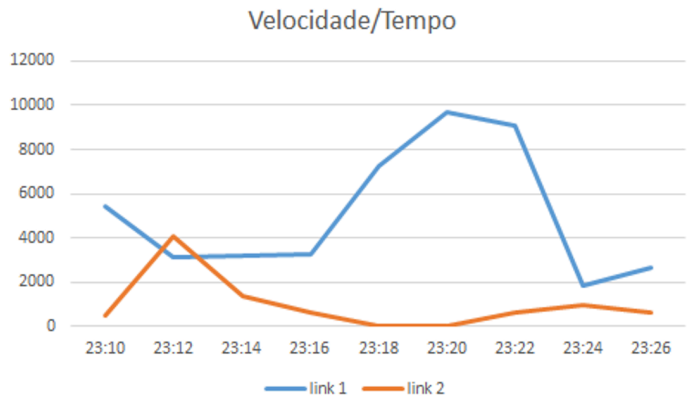
Fonte: O autor

Figura 5: Arquivo *Log*.

```
17:00-25/05/2018  0 16.475 8.5850 8 0 515.959 1.6230 1
17:02-25/05/2018  10 32.537 6.9610 6 0 384.067 2.0650 2
17:04-25/05/2018  0 19.490 8.3680 8 0 406.758 1.9730 1
17:06-25/05/2018  0 24.536 8.0300 8 0 59.953 6.2510 6
17:08-25/05/2018  0 24.040 8.0610 8 0 63.798 6.1050 6
17:10-25/05/2018  0 31.360 7.6120 7 0 83.700 5.4430 5
17:12-25/05/2018  0 37.805 7.2560 7 0 65.639 6.0370 6
17:14-25/05/2018  0 22.765 8.1450 8 0 57.189 6.3620 6
17:16-25/05/2018  0 18.122 8.4650 8 0 559.799 1.5150 1
17:18-25/05/2018  0 11.424 8.9750 8 0 453.434 1.8060 1
17:20-25/05/2018  0 10.519 9.0480 9 0 426.924 1.8970 1
17:22-25/05/2018  0 11.244 8.9890 8 0 269.545 2.7060 2
17:24-25/05/2018  0 10.400 9.0570 9 0 356.803 2.1890 2
17:26-25/05/2018  0 10.713 9.0320 9 0 407.308 1.9710 1
17:28-25/05/2018  0 10.593 9.0420 9 0 191.831 3.4260 3
17:30-25/05/2018  0 11.101 9.0000 9 0 463.743 1.7730 1
```

Fonte: O autor

Figura 6: Streaming de vídeo.

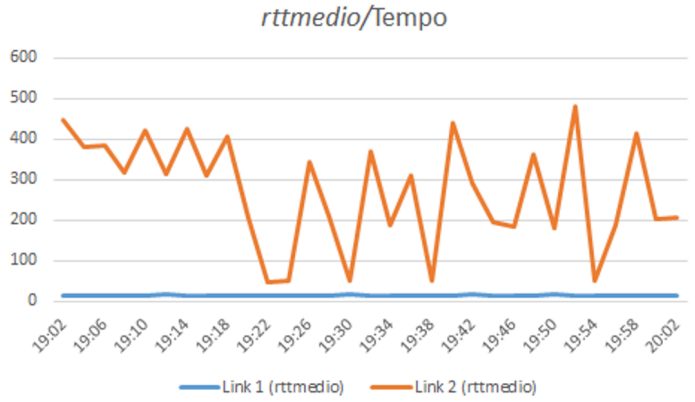


Fonte: O autor

link 2 sofre maior variação do *rttmedio*, diferentemente da perda de pacote.

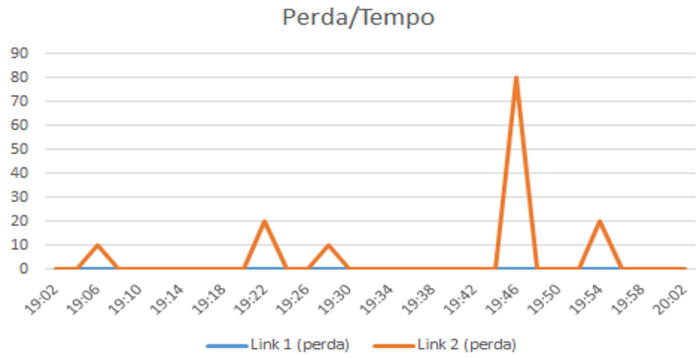
A figura (9) demonstra os pesos determinados após a execução do *script* de cálculo. Os mesmos são inseridos automaticamente pelo *script*.

Figura 7: *rttmedio*/tempo.



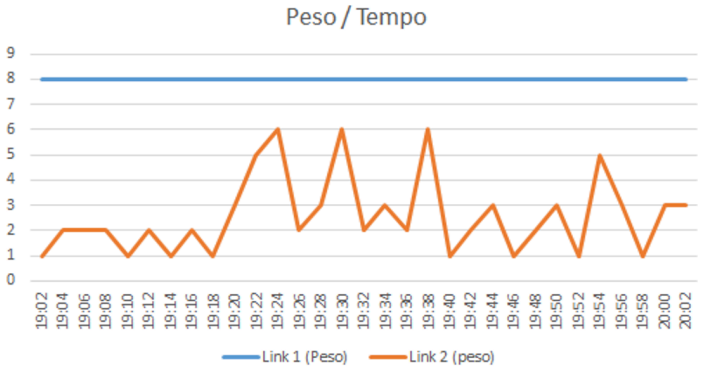
Fonte: O autor

Figura 8: Perda/Tempo



Fonte: O autor

Figura 9: Pesos do balanceamento.



Fonte: O autor

CONCLUSÃO

Após a finalização dos testes, constatou-se que o projeto teve um resultado bastante satisfatório. Concluiu-se que através de um computador com um mínimo de recurso necessário e um algoritmo bem implementado é possível fazer um grande tratamento das conexões, e assim, realizando um balanceamento de carga dinâmico entre dois ou mais links de forma eficiente e sobretudo sem a interferência humana. Esse balanceamento foi determinado pelo nosso *script* que tem uma função de qualificação, em seguida, retornando um valor de controle da nossa rede, que no caso em questão é o peso das nossas conexões. Os pesos eram definidos automaticamente pelo nosso cálculo e inserido no *script* responsável pelo balanceamento.

Desta maneira, o projeto atende com eficiência qualquer instalação cuja demanda de recursos de internet é demasiadamente elevada, garantindo o equilíbrio de todo tráfego da rede e mantendo o acesso com alta disponibilidade e qualidade na conexão.

Analisando e finalizando o projeto, diagnosticou-se que o parâmetro perda de pacotes teve maior influência na equação do balanceamento de carga. Assim, para um próximo trabalho seria desejável, analisar com maior rigor o parâmetro perda de pacote, pois para este estudo, não houve grande relevância deste cálculo.

REFERÊNCIAS

ANDRIOLI, Leandro; DA ROSA RIGHI, Rodrigo. **AGANT: Proposta de um modelo ciente do tráfego da rede para ambientes inteligentes.** In: Anais da XVII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul. SBC, 2017.

BIANCHETTI, D. **Manual de Configuração Advance Routing Linux.** Disponível em: http://www.dicas-l.com.br/arquivo/manual_de_configuracao_advance_routing_linux_para_bisonhos.php#.WhYzMEqnHIV . Acesso em: 08 de outubro de 2017.

DEBIAN. **Debian**. Disponível em: https://www.debian.org/. Acesso em: 26 de novembro de 2017.

EDGARD, A. **Roteamento avançado com Linux.** Disponível em: https://memoria.rnp.br/newsgen/0201/roteamento_linux.html#ng-1/. Acesso em: 30 de outubro de 2017.

HARDPLUS. **O que é load balance.** Disponível em: http://hardplus.com.br/blog/o-que-e-load-balance/. Acesso em: 18 de março de 2018.

MUNIZ. V. **O que é IPTables, para que serve, como usar?** Disponível em: http://viniciusmuniz.com/pt/o-que-e-iptables-para-que-server-como-usar/ . Acesso em: 08 de outubro de 2017.

ONU. **Brasil é o quarto país com mais usuários de Internet do mundo, diz relatório da ONU.** Disponível em: https://nacoesunidas.org/brasil-e-o-quarto-pais-com-mais-usuarios-de-internet-do-mundo-diz-relatorio-da-onu/. Acesso em: 04 de outubro de 2017.

PASSOS, Diego. **Métricas de Roteamento para Redes em Malha Sem**

Fio. Monografia apresentada ao departamento de Ciência da Computação. 2007

REDE RIO. **SSH**. Disponível em: http://www.rederio.br/downloads/pdf/nt00301.pdf. Acesso em: 16 de abril de 2018.

TECHTUDO. **Putty**. Disponível em: http://www.techtudo.com.br/tudo-sobre/putty.html. Acesso em: 16 de abril de 2018.

UPPAL, H.; BRANDON, D. **Openflow based load balancing.** In: Proceedings of CSE561: Networking Project Report. University of Washington, Spring, 2010.

VIRGÍLIO, J. **O que é Shell Script.** Disponível em: https://www.vivaolinux.com.br/artigo/O-que-e-Shell-Script. Acesso em: 26 de novembro de 2017.

VIVA O LINUX. **Balanceamento de links - Load balance + Failover + Failback.** Disponível em: https://www.vivaolinux.com.br/artigo/Balanceamento-de-links-Load-balance-Failover-Failback. Acesso em: 29 de abril de 2018.