

Uma Análise comparativa dos Efeitos no Tempo de Convergência do Protocolo OSPF com a Arquitetura RouteFlow

Sandro T. Pinto¹, Fernando Barreto², Alexandre L'Erário¹

¹ Programa de Pós-Graduação em Informática – PPGI -Universidade Tecnológica Federal do Paraná (UTFPR) - Cornélio Procópio – PR – Brasil

² Programa de Pós-Graduação em Informática – PPGI -Universidade Tecnológica Federal do Paraná (UTFPR) - Apucarana – PR – Brasil

sandropinto21@gmail.com, fbarreto@utfpr.edu.br, alerario@utfpr.edu.br

Abstract. *The convergence time of a network is the total time spent by the routing protocol to update the routes of routers. In case of OSPF, it is possible to reduce that time by adjusting the values of the parameters. This paper presents an analysis of the effects of the Hello Interval, and Dead-Interval parameters in the convergence time in a RouteFlow architecture. It made a practice analysis using standard virtualized network infrastructure and other RouteFlow of Abilene backbone. After analysis, it was revealed that the time can be improved by adjusting the Hello Interval, and Dead-Interval for the values of 1 and 4 seconds respectively, and it was found that the stage size has influence on set values.*

Resumo. *O tempo de convergência de uma rede é o tempo total gasto pelo protocolo de roteamento para atualizar as rotas dos roteadores. No caso do OSPF, é possível reduzir esse tempo através de ajustes nos valores dos parâmetros. Este trabalho apresenta uma análise dos efeitos dos parâmetros Hello-Interval e Dead-Interval no tempo de convergência em uma arquitetura RouteFlow. É feita uma análise prática utilizando a infraestrutura de rede virtualizada padrão do RouteFlow e outra de um backbone Abilene. Após a análise, revelou-se que o tempo pode ser melhorado com o ajuste dos Hello-Interval e Dead-Interval para os valores de 1 e 4 segundos respectivamente, e identificou-se que o tamanho do cenário tem influencia nos valores configurados.*

1 Introdução

Em um Sistema Autônomo (SA), utiliza-se geralmente protocolos de roteamento do estado do enlace para organizar as rotas internas. O protocolo de roteamento *Internet Protocol* (IP) mais utilizado é o *Open Shortest Path First* (OSPF) (MOY, 1998). A detecção de falha no OSPF é reativa, e tem início quando as mensagens *HELLO* trocadas periodicamente entre roteadores OSPF adjacentes ficam um período sem resposta. Nesse caso, o OSPF identifica a existência de uma falha adjacente. Os roteadores adjacentes à falha atualizam, então, suas *Routing Information Bases* (RIBs) e anunciam mensagens *Link State Advertisement* (LSA) para sinalizar aos demais roteadores do SA sobre a alteração da topologia e um recálculo das novas rotas é realizado.

O tempo despendido nos processos de detecção da falha, modificação e notificação dos novos estados do enlace com LSA, atualização da RIB, recálculo do SPF e o preenchimento da *Forwarding Information Base* (FIB) (VASSEUR; PICKAVET; DEMEESTER, 2004) compõem o tempo de convergência. Durante esse período, as rotas geralmente ficam instáveis, acarretando em loops de roteamento e descarte de pacotes. Dependendo da topologia, o tempo de convergência pode variar de vários segundos até minutos, o que influencia diretamente na confiabilidade de uma infraestrutura de rede.

Como a maioria dos roteadores implementam uma arquitetura composta de camada de software fechada e hardware proprietário (NASCIMENTO et al., 2011), torna-se inviável realizar modificações no protocolo de roteamento ou a adoção de novas abordagens, porque tais alterações mudam a estrutura padrão do protocolo OSPF definido pela norma (MOY, 1998).

A arquitetura RouteFlow é uma arquitetura de roteamento IP que procura combinar o desempenho dos equipamentos de fabricantes conhecidos com a flexibilidade de softwares executados remotamente através de computadores, que desacopla os planos de encaminhamento e controle (NASCIMENTO et al., 2011).

Baseado em uma análise sistemática (PETERSEN et al., 2008), a proposta deste trabalho é apresentar um estudo do tempo de convergência do protocolo OSPF em infraestruturas de rede comuns com arquitetura RouteFlow na alteração dos parâmetros *Hello-Interval* e *Dead-Interval* do protocolo OSPFv2.

Este artigo está organizado da seguinte forma: na seção 2 é apresentado os parâmetros de configuração *Hello-Interval* e *Dead-Interval*, na seção 3, os métodos e procedimentos, na seção 4 o cenários de teste, na seção 5 os resultados e na seção 6 a conclusão.

2 Parâmetros *Hello-Interval* e *Dead-Interval* do protocolo OSPF

Os parâmetros, definidos na norma (MOY, 1998), permitem ao administrador de rede modificar o comportamento do protocolo de roteamento, podendo influenciar diretamente no tempo de convergência.

Os parâmetros de configuração do OSPF *Hello-Interval* (P1) e *Dead-Interval* (P2) permitem ao administrador de rede modificar o comportamento do protocolo de roteamento. Estes parâmetros estão contidos na mensagem *HELLO*, onde o P1 tem por finalidade determinar o tempo entre o envio de pacotes *HELLO*, já o P2 é o tempo para detectar se houve uma mudança na infraestrutura, ambos são responsáveis pela detecção das falhas no enlace ou de algum roteador.

O protocolo OSPF envia a mensagem *HELLO* com o tempo configurado no parâmetro P1 a cada 10 segundos em enlaces broadcast e ponto a ponto, ou a cada 30 segundos nos demais tipos de enlace. O objetivo é verificar a conectividade entre os roteadores adjacentes. O parâmetro P2, segundo a norma (MOY, 1998), apresenta um tempo 4 vezes o valor do tempo do P1, responsável por trazer a informação da mudança ocorrida, caso aconteça uma queda de enlace ou de um roteador.

3 Métodos e procedimentos

Essa pesquisa realizou um levantamento de trabalhos científicos que modificaram algum dos parâmetros de configuração do OSPF relatados na Tabela 1 e de configurações sugeridas por fabricantes como Cisco®, HP®, Avaya®, Huawei® e Juniper® levantadas em manuais e em seus sites (CISCO, 2014)(NETENGINE; ROUTER, 2011)(NETWORKS, 2015).

Para a condução desta pesquisa, foi utilizado um mapeamento que compõe uma análise sistemática descrito por(PETERSEN et al., 2008).

Na execução do mapeamento sistemático, a busca inicial é feita com elementos definidos com palavras chaves necessárias. As palavras chaves identificadas foram: RouteFlow; ((OSPF) AND (time convergence)); ((RouteFlow) AND (OSPF)); ((configuring) AND (OSPF) AND (time convergence)); ((reducing) AND (OSPF) AND (time convergence)).

Cada conjunto de palavras chaves foram conduzidas nos sites ieexplore, acm, webofknowledge, scopus e sciencedirect resultando em um total de 1457 trabalhos. Com o uso da ferramenta Jabref (biblioteca *open source* para gerenciar referências) as referências duplicadas foram removidas resultando no total de 653 trabalhos.

Para seleção dos trabalhos relevantes foi utilizado como critério de exclusão três fatores: o primeiro são trabalhos relacionados com alteração do protocolo, como algoritmos para melhorar o tempo; em segundo, trabalhos que utilizaram especificamente redes sem fio e em terceiro, trabalhos que utilizaram parâmetros proprietários não contidos na norma do OSPF(MOY, 1998).

A quantidade de trabalhos obtidos após a execução do critério de corte entre estudos primários e secundários foram 17. Nestes 17, o abstract foi interpretado, com o intuito de verificar se o mesmo enquadrava no escopo deste trabalho. O único trabalho que apresentou estudos secundários foi (LUCENA, 2011).

Com análise do abstract, os trabalhos apresentados por (LUCENA, 2011)(REIS; COLETTI; SENNE, 2013)(BINDHU; RAMESH, 2014)(STRINGER; FU; LORIER, 2013)(SHARMA; STAESSENS; COLLE, 2013)(JAKMA; LAMPARTER, 2014)(NUNES; MENDONCA, 2013)(LARA; KOLASANI; RAMAMURTHY, 2014)(GUEDES et al., 2012)(LIN et al., 2014)(FARIAS et al., 2012) abordam RouteFlow, contudo são trabalhos que não abordam o tratamento do tempo de convergência do protocolo OSPF utilizando a arquitetura RouteFlow, segue na sequência uma síntese de cada trabalho.

O trabalho de (LUCENA, 2011), apresenta sobre o gerenciamento de redes virtualizadas utilizando a arquitetura RouteFlow, com um comparativo entre o protocolo BGP tradicional e atômico, (REIS; COLETTI; SENNE, 2013) apresenta a implantação da arquitetura RouteFlow em uma rede de produção, onde implementa como uma alternativa viável no roteamento virtualizado para controle centralizado de pacotes no intercambio com a rede física, (BINDHU; RAMESH, 2014) faz uma abordagem sobre a história de redes definas por software, onde apresenta o RouteFlow como uma plataforma de controle

de encaminhamento utilizando o protocolo OpenFlow, (STRINGER; FU; LORIER, 2013) utiliza o RouteFlow por ser um projeto de código aberto, juntamente no projeto Cardigan, pois permite a agregação de vários caminhos de dados, para ser utilizado como roteadores distribuídos, (SHARMA; STAESSENS; COLLE, 2013) apresenta uma forma de configuração automática de plataformas de controle de roteamento em redes OpenFlow e para os testes utiliza a arquitetura RouteFlow, (JAKMA; LAMPARTER, 2014) apresenta uma introdução sobre roteadores Quagga, e diz que o projeto RouteFlow tem capacidade disponível como um plano de encaminhamento para utilização do Quagga como uma interface baseada em padrões de redes, (NUNES; MENDONCA, 2013) apresenta uma pesquisa sobre a história de redes definidas por software, e com isso, demonstra o RouteFlow como sendo um controlador compatível com o protocolo OpenFlow que fornece roteamento IP virtualizado, (LARA; KOLASANI; RAMAMURTHY, 2014) apresenta uma pesquisa sobre inovação usando redes OpenFlow, e o RouteFlow como uma arquitetura que fornece serviço com utilização do Quagga e propõe uma abordagem baseada em OpenFlow que permite a introdução de sistemas avançados de roteamento, (GUEDES et al., 2012), apresenta uma abordagem sistêmica no desenvolvimento de pesquisas em redes de computadores em SDN, e o RouteFlow como sendo uma solução distribuída que possa ser aplicado como protocolo de roteamento em SDN, (LIN et al., 2014) apresenta uma estrutura de roteamento em um ambiente de SDN, e o RouteFlow como uma única solução criada para roteamento IP com protocolo OpenFlow, (FARIAS et al., 2012) apresenta um estudo comparativo de roteamento no uso de SDN, e cita o RouteFlow apenas como um gerenciador de switches de roteamento e virtualização, onde, switches OpenFlow são conectados a roteadores virtuais e entre eles são estabelecidas as conectividades virtuais conforme a topologia física.

Estes trabalhos estão relacionados a esta pesquisa entretanto, tais trabalhos não fizeram abordagem que tratam o tempo de convergência no protocolo OSPF. Desta maneira existem 5 trabalhos sobre o assunto sem o RouteFlow que foram identificados (PITKANEN; LUOMA, 2005) (GOYAL; RAMAKRISHNAN; FENG, 2003)(SINGH, 2013)(JIN, 2015)(BASU; RIECKE, 2001) e mais um que se refere ao próprio autor do RouteFlow (NASCIMENTO et al., 2011) e 3 fabricantes de equipamentos(CISCO, 2014)(NETENGINE; ROUTER, 2011)(NETWORKS, 2015)que são apresentados na Tabela 1.

Tabela 1- Valores dos Parâmetros utilizados no Protocolo OSPF

TRABALHOS	HELLO	DEAD
	Tempo em Segundos	
(PITKANEN; LUOMA, 2005)	1	3
(GOYAL; RAMAKRISHNAN; FENG, 2003)	0.25	1
(SINGH, 2013)	0.25	1
(JIN, 2015)	0.25	1
(BASU; RIECKE, 2001)	0.2	1
(NASCIMENTO et al., 2011)	1	4
(CISCO, 2014)	0.25	1

(NETENGINE; ROUTER, 2011)	10	30
(NETWORKS, 2015)	2	8

Em seguida, realizou-se testes obedecendo às alterações de configuração citadas por cada trabalho individualmente em dois cenários de testes comuns com o objetivo de identificar os melhores resultados

4 Cenários de Teste

Para realizar testes nas topologias com RouteFlow foram utilizadas apenas duas máquinas virtuais. A primeira máquina virtual tem por finalidade acomodar a arquitetura RouteFlow, que compõe alguns softwares para o seu funcionamento, como: engine Quagga, controlador POX, o banco de dados MongoDB, LXC containers e o Open VSwitch, descritos na Tabela 2 e uma segunda máquina com o programa Mininet para execução dos testes.

Tabela 2 - Softwares necessários para o funcionamento do RouteFlow

RouteFlow	
Softwares	Finalidade
LXC Containers	Construir as topologias em containers com Sistema Operacional Linux e a Engine de Roteamento Quagga
OpenVswitch	Interligar os containers
DBMongo	Armazenar todo o estado de configuração do mapeamento entre topologias física e virtual
NOX/POX	Controlar a comunicação entre o cliente e o servidor da arquitetura
Quagga	Roteador Virtual
Mininet	Simular a rede física para testes

Os cenários utilizados foram dois: primeiro o denominado de RouteFlow 4 Rts fornecido pelo projeto original do RouteFlow com 4 roteadores, 4 estações e 9 enlaces conforme Figura 1(a), e outro RouteFlow Abilene com 11 roteadores, 2 estações e 16 enlaces conforme mostra a Figura 1(b).

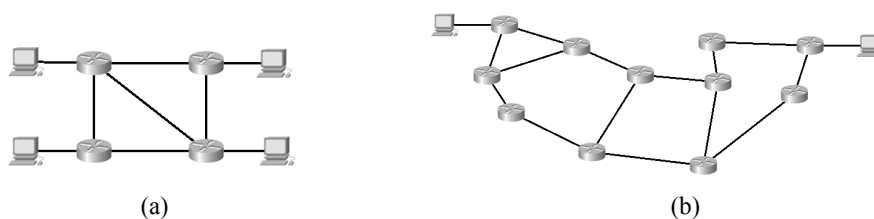


Figura 1 - Cenário de Teste RouteFlow 4 Rts

Com a montagem dos cenários foi aplicado as configurações dos tempos de *Hello-Interval* e *Dead-Interval* descritas conforme na Tabela 1.

5 Resultados

Com as informações obtidas através dos testes pode-se observar que o tempo de convergência aumenta proporcionalmente com o aumento dos parâmetros *Hello-Interval* e *Dead-Interval*.

No cenário original 4 Rts foi criado seguindo as instruções utilizado pela arquitetura original RouteFlow. Mesmo com uma arquitetura robusta de roteamento IP, composta de outros programas para o seu funcionamento, apresentou uma estabilidade no tempo de convergência, com baixa oscilação nos valores como pode ser visto nas Figuras 2(a), já o cenário Abilene, os testes também apresentaram uma estabilidade no tempo de convergência, com baixa oscilação nos valores como pode ser visto na Figura 2(b), contudo para os tempos menores que 1 segundo com o parâmetro *Hello-Interval*, havia perda de conexão durante todo o tempo e muitas mensagens de destino não alcançado, impossibilitando um levantamento médio.

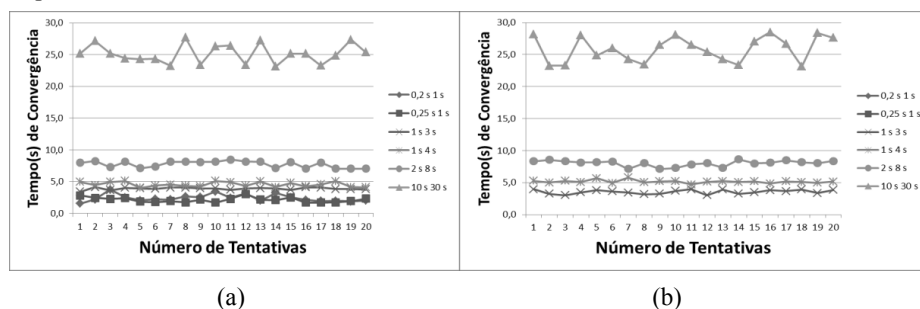


Figura 2 - Resultado do tempo de convergência da configuração dos parâmetros *Hello-Interval* e *Dead-Interval*

Contudo para tempos menores que 1 segundo do *Hello-Interval*, no cenário original no caso do 0.25 segundos (JIN, 2015), (BASU; RIECKE, 2001), (GOYAL; RAMAKRISHNAN; FENG, 2003), (CISCO, 2014), possibilitou uma média de 2,110 segundos no tempo de convergência, e em algumas vezes apresentava uma mensagem com “destino não alcançado”. Isso ocorreu devido ao aumento de mensagens *Hello*. No caso do 0.20 segundos utilizado por (SINGH, 2013), apresentou uma inconsistência durante os testes, aumentando a frequência de mensagens destino não alcançado, mas mesmo assim foi possível identificar que a média do tempo de convergência ficou em 2,402 segundos, ou seja, um aumento de 13,8% em relação ao tempo do *Hello-Interval* de 0.25 segundos, mostrando assim que realmente o número de mensagens gera um congestionamento de pacotes na rede.

6 Conclusão

Este artigo selecionou e analisou alguns trabalhos, além de configurações de fabricantes, que sugerem a modificação dos parâmetros definidos na especificação do protocolo OSPF para obter uma redução no tempo de convergência, juntamente com a arquitetura RouteFlow.

Os parâmetros de tempo convergência, *Hello-Interval* *Dead-Interval* se destacaram por apresentarem uma melhora acima dos 90% no tempo de convergência quando utilizados os tempos para P1 entre 0.20 e 1 segundo. Porém, toda alteração deve ser ajustada de acordo com a quantidade de equipamentos utilizados no cenário.

Os valores de P1 abaixo de 1 segundo deve ser usado com cautela, isso porque, os testes mostraram que no cenário Abilene, com um maior número de equipamentos não funcionou adequadamente, gerando assim uma instabilidade na rede.

Os testes demonstraram também que na configuração do parâmetro *Hello-Interval*, abaixo de 0.25 segundos, o tempo de convergência piorou ao invés de melhorar, isso porque, aumentou da quantidade de pacotes *HELLO* transmitidas em um segundo, possibilitando uma sobrecarga de pacotes na rede. Os dois cenários tiveram o mesmo problema com esta configuração, dessa forma, esse caso deve ser evitado.

Para trabalhos futuros, serão abordados cenários com um maior volume de roteadores, e testes com outros protocolos, juntamente com redes sem fio.

Referências

- BASU, A.; RIECKE, J. Stability issues in OSPF routing. **ACM SIGCOMM Computer Communication Review**, v. 31, p. 225–236, 2001.
- BINDHU, M.; RAMESH, G. . The Journey to SDN: A Peek into the History of Programmable Networks. **International Journal of Computer Science and Engineering Communications**, v. 2, n. 5, p. 500–506, 2014.
- CISCO. **ASA Release 9.2.1 OSPF Enhancements Configuration Example**. Disponível em: <<http://www.cisco.com/c/en/us/support/docs/security/asa-5500-x-series-next-generation-firewalls/118098-config-ospf-00.html>>. Acesso em: 16 out. 2015.
- FARIAS, L. DE et al. **Um Estudo Comparativo de Softwares de Roteamento para Uso em Redes Definidas por Software**. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wperformance/2013/004.pdf>>. Acesso em: 11 nov. 2014.
- GOYAL, M.; RAMAKRISHNAN, K. K.; FENG, W. F. W. Achieving faster failure detection in OSPF networks. **IEEE International Conference on Communications, 2003. ICC '03.**, v. 1, p. 296–300, 2003.
- GUEDES, D. et al. Redes Definidas por Software: uma abordagem sistêmica para o desenvolvimento de pesquisas em Redes de Computadores. **Sbrc**, n. February, p. 160–210, 2012.
- JAKMA, P.; LAMPARTER, D. Introduction to the quagga routing suite. **Network, IEEE**, n. April, p. 42–48, 2014.
- JIN, J. Precisely Control Routing Convergence by Adjusting Parameters Based on Network State. **2015 Fifth International Conference on Communication Systems and Network Technologies**, p. 278–282, 2015.

LARA, A.; KOLASANI, A.; RAMAMURTHY, B. Network Innovation using OpenFlow: A Survey. **IEEE Communications Surveys & Tutorials**, v. 16, n. 1, p. 493–512, 2014.

LIN, Z. et al. A Routing Framework in Software Defined Network Environment. **2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing**, p. 907–911, 2014.

LUCENA, S. C. DE. **Projeto de Pesquisa Gerenciamento de Redes Virtualizadas Baseadas na Arquitetura RouteFlow**. Disponível em: <<http://www2.uniriotec.br/ppgi/informacoes-sobre-o-programa/projetos-de-pesquisa/gerenciamento-de-arquiteturas-de-rede-baseadas-em-openflow-com-foco-em-roteamento-bgp>>. Acesso em: 11 nov. 2015.

MOY, J. OSPF Version 2. **Ascend Communications, Inc.**, p. Request for Comments: 2328, 1998.

NASCIMENTO, M. et al. RouteFlow: Roteamento Commodity Sobre Redes Programáveis. **XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, 2011.

NETENGINE, H.; ROUTER, C. Configuration Guide - IP Routing Huawei. 2011.

NETWORKS, J. **Example: Configuring OSPF Timers**. Disponível em: <http://www.juniper.net/documentation/en_US/junos15.1/topics/topic-map/ospf-timers.html>. Acesso em: 16 out. 2015.

NUNES, B.; MENDONCA, M. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. ... **Surveys and ...**, p. 1–17, 2013.

PETERSEN, K. et al. Systematic mapping studies in software engineering. **EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering**, p. 68–77, 2008.

PITKANEN, M.; LUOMA, M. OSPF flooding process optimization. **HPSR. 2005 Workshop on High Performance Switching and Routing**, 2005.

REIS, V. DOS; COLETTI, C.; SENNE, E. Implementando RouteFlow em uma Rede em Produção. **sbrc2013.unb.br**, v. 3, p. 3–8, 2013.

SHARMA, S.; STAESSENS, D.; COLLE, D. Automatic configuration of routing control platforms in OpenFlow networks. **Proceedings of the ...**, n. 1, p. 491–492, 2013.

SINGH, H. **Effects OSPF Timers Configurations on Network Convergence in New Generation Routers**Nacional Conference on Emerging in Electrical, Instrumentation & Communication Engineering. **Anais...Varanasi**: 2013

STRINGER, J.; FU, Q.; LORIER, C. Cardigan: Deploying a distributed routing fabric. **Proceedings of the ...**, p. 169–170, 2013.

VASSEUR, J. P.; PICKAVET, M.; DEMEESTER, P. **Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS**. São Francisco : Elsevier, 2004.