

Tolerância a falhas em multi-controladores SDN: eficiência de controladores *RYU* usando Mininet

Eonay Gurjão^{1,2}, Klenilmar Dias², Marcos Amaris¹

¹Universidade Federal do Pará (UFPA) - Tucuruí - PA - Brasil

²Instituto Federal do Amapá (IFAP) - Macapá - AP - Brasil

{eonay.web, klenilmar, marcos.amaris}@gmail.com

Abstract. *The SDN Paradigm has emerged as a way to realize an interaction between networks and applications. Its centralized architecture makes a controller have a central view of the network and its scheduling feature allows applications to schedule the network. In this direction, Mininet is a widely used tool for emulating SDN network scenarios. However, when emulating a scenario with multi-controllers, Mininet's native module generates an overhead in the utilization rate of Memory and CPU resources. In this sense, this paper presents a module, with functional characteristics of a multi-controller, generating a better compromise (trade-off) in the utilization rate of Memory and CPU, in case of unavailability of a controller in a SDN scenario with hierarchical network topology.*

Resumo. *O Paradigma SDN surgiu como um caminho para a realização de uma interação entre redes e aplicações. Sua arquitetura centralizada faz com que um controlador tenha uma visão central da rede e sua característica de programação permite que os aplicativos programem a rede. Nessa direção, o Mininet é uma ferramenta amplamente utilizada para emular cenários de redes SDN. Contudo, ao emular um cenário com multi-controladores, o módulo nativo do Mininet gera uma sobrecarga na taxa de utilização dos recursos de Memória e CPU. Nesse sentido, este artigo apresenta um módulo, com características funcionais de um multi-controlador, gerando um melhor compromisso (trade-off) na taxa de utilização de Memória e CPU, em caso de indisponibilidade de um controlador em um cenário SDN com topologia de rede hierárquica.*

1. Introdução

Redes Definidas por Software (*Software Defined Networks* ou *SDN*) é um paradigma de arquitetura de redes de computadores emergentes, onde o controle da rede está dissociado do encaminhamento de pacotes e pode ser diretamente programável [Foundation 2021]. Ou seja, a SDN apresenta como principal característica a separação do plano de controle do plano de encaminhamento, também conhecido como plano de dados, permitindo que o plano de controle possa ser movido do hardware de comutação (switch ou roteador) para um servidor dedicado.

Mininet é uma ferramenta de emulação, capaz de criar topologias de rede completas com hosts, enlaces e switches em uma única máquina, de maneira customizada utilizando a linguagem Python. Através da interface CLI (*Command Line Interface*) do

Mininet é possível interagir facilmente com a rede SDN, personalizá-la, compartilhá-la, ou simplesmente implantá-la em um hardware real.

Porém, ao emular uma topologia SDN com multi-controladores, o módulo nativo do Mininet com suporte a esta funcionalidade, gera uma sobrecarga excessiva na taxa de utilização dos recursos de Memória e CPU no cenário de rede emulado. Como proposta de um melhor compromisso na relação da taxa de utilização Memória x CPU, este artigo apresenta um módulo com arquitetura multi-controlador integrando o Mininet ao controlador RYU, com características de sistema distribuído.

A estrutura do documento está organizada da seguinte forma, Seção 2 apresenta os conceitos e trabalhos relacionados, na Seção 3 mostra os materiais e métodos, na Seção 4 a os resultados e discussões e, por fim, na Seção 5 conclui-se este trabalho e sugere-se trabalhos futuros.

2. Conceitos e Trabalhos Relacionados

Segundo a [ONF 2015], Redes Definidas por Software (também conhecido como SDN, de *Software Defined Network*) é uma arquitetura de redes emergente em que o plano de controle da rede é dissociado do plano de encaminhamento sendo possível programá-lo diretamente.

ONIX [Koponen et al. 2010] foi a proposta pioneira no controle distribuído em redes SDN, particionando o escopo dos controladores, agregando as informações, e compartilhando o estado, via APIs, para diferentes tipos de *data stores* em função dos requisitos de consistência.

Em [Fonseca et al. 2013] os autores apresentam um estudo em SDN em que as estratégias de replicação ativa e passiva são usadas para implementar controladores distribuídos. Além disso, indicam cenários em que utilização de cada técnica pode ser mais efetiva.

Dentre as ferramentas de desenvolvimento e testes, o emulador Mininet é o mais difundido, foi aplicado nos trabalhos de [Romanov et al. 2021] e [Hasan et al. 2020] para criar cenários para os estudos de SDN, tratando-se de multi-controladores significativamente são necessários maiores *pools* de recursos para garantir o funcionamento de ambos os controladores.

O protocolo OpenFlow foi o primeiro padrão de interface de comunicação entre plano de controle e o plano de dados de uma arquitetura SDN [ONF 2015]. O OpenFlow [ONF 2015] é uma proposta que permite a implementação das SDN, uma vez que define como protocolo de comunicação entre o plano de encaminhamento e o plano de controle de forma simples e flexível para que seja possível a instalação de regras de encaminhamento baseadas em diversos parâmetros de protocolos em camadas distintas.

O Ryu é um controlador desenvolvido na linguagem Python e que possibilita o desenvolvimento de aplicações de gerenciamento e controle de redes SDN através da disponibilização de diversas APIs. Além disso, suporta protocolos de comunicação como o NetConf e o Openflow [Sánchez Calderón et al. 2021]. Seu código fonte é aberto e disponibilizado aos desenvolvedores de aplicações e gerentes de rede. Desta forma, o Ryu possui uma comunidade ativa e uma vasta documentação. Isto faz com que o mesmo seja uma boa escolha para a implementação de uma rede SDN [ONF 2015].

3. Materiais e Métodos

O cenário para os experimentos de desenvolvimento e testes do módulo foi totalmente em ambiente virtual orquestrado via um *hypervisor*. Para equidade dos testes foi alocado um único núcleo (*core*) para cada máquina virtualizada e 1GB de Memória RAM, onde duas VM (*Virtual Machine*) são responsáveis por atuar como controladores RYU, logo faz referência à ideia de multi-controladores. A terceira VM representa a estrutura de rede local (LAN de *Local Área Network*) que tem a função de interligar os ativos de rede na topologia hierárquica em camadas de um data center.

O módulo se encarrega de iniciar todos os controladores além de criar instâncias dos hosts, OVS, links e demais requisitos do cenário de rede proposto, seguidamente realiza a integração dos controladores ao módulo em camadas de forma a subsidiar o funcionamento da rede e demais serviços na camada de aplicação.

4. Resultados e Discussões

O resultado dos testes automatizados em ambos controladores RYU (c1 e c2) conforme Figura 1 retratam a redução da carga do controlador c2 mantendo a alternância em casos de indisponibilidade, mantendo assim a tolerância a falhas com menor custo computacional. A prioridade do controle dos fluxos fica vinculada ao carregamento do módulo de encaminhamento de pacotes em ordem de inicialização, sendo as regras aplicadas aos fluxos (flow) de rede são contingenciados assim reduzindo o uso de recursos físicos e consequentemente energéticos.

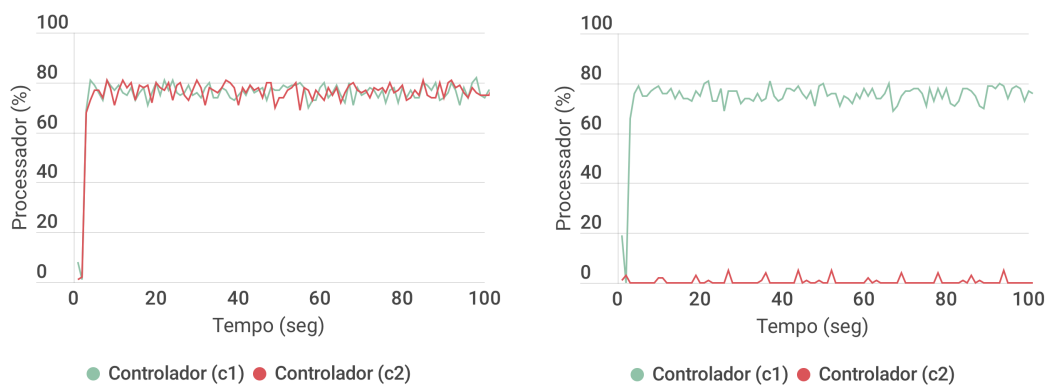


Figura 1. Carga de CPU (Antes [Esquerda], Depois [Direita])

Na prática a implementação do módulo possibilita a aplicação em equipamentos com baixos recursos, implicando em menores custos de desenvolvimento e testes. A redução no processamento do controlador c2, ocasionou uma pequena elevação no uso de Memória RAM (Figura 2), onde os controladores mantiveram os princípios de disponibilidade em caso de falha além da redução substancial de recursos de máquina.

5. Conclusão e Trabalhos Futuros

Contudo, a análise e avaliação de desempenho através da aplicação de multi-controladores RYU, considerando a proposta do módulo para Mininet de múltiplas instâncias mantendo a característica de alta disponibilidade em modelos de rede SDN hierárquica, mostrou-se eficaz na redução do uso de recursos de hardware. Sobretudo, obteve-se resultados

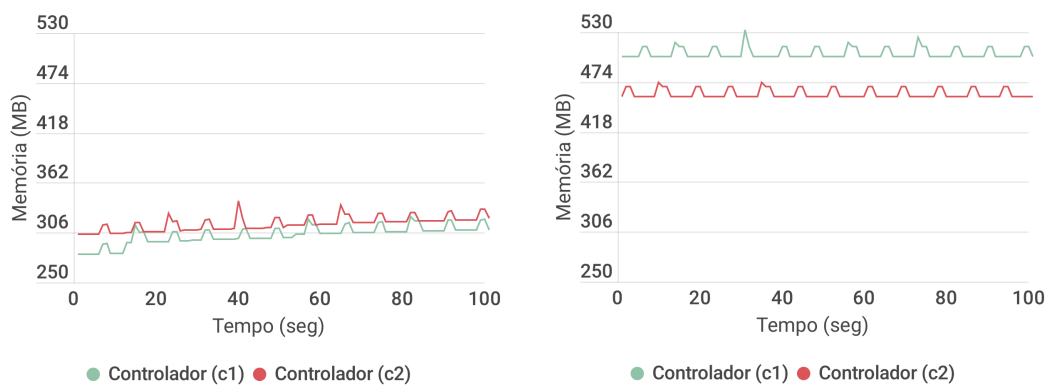


Figura 2. Uso de Memória RAM (Antes [Esquerda], Depois [Direita])

preliminares nos experimentos e simulações, que avaliou carga de CPU e Memória como parâmetros iniciais de análise.

No futuro, pode-se avaliar o desempenho dos controladores RYU, considerando diferentes topologias no emulador Mininet e avaliar outros parâmetros de desempenho, também integrar o módulo a outros controladores OpenFlow SDN em decorrência da aplicação eficiente dos planos de controle. Outros parâmetros podem ser avaliados e ajustados na versão atual, o módulo sendo codificado na linguagem Python abre um leque de possibilidades na área acadêmica de pesquisa, apresentando incentivando novos estudos para característica de redução de custos e recursos em redes programáveis.

Referências

- Fonseca, P., Bennesby, R., Mota, E., and Passito, A. (2013). Resilience of sdn based on active and passive replication mechanisms. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 2188–2193. IEEE.
- Foundation, O. N. (2021). Software-defined networking: The new norm for networks. Executive Summary.
- Hasan, M., Dahshan, H., Abdelwanees, E., and Elmoghazy, A. (2020). Sdn mininet emulator benchmarking and result analysis. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pages 355–360.
- Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., et al. (2010). Onix: A distributed control platform for large-scale production networks. In *OSDI*, volume 10, pages 1–6.
- ONF (2015). Openflow switch specification. Technical report, Open Networking Foundation. Technical Manual.
- Romanov, O., Saychenko, I., Marinov, A., and Skolets, S. (2021). Research of sdn network performance parameters using mininet network emulator. *Information and Telecommunication Sciences*, (1):24–32.
- Sánchez Calderón, J. D., Guzmán Pineda, J. J., et al. (2021). Balanceador de carga utilizando algoritmos de calidad de servicio (qos) en redes definidas por software (sdn).