

REDES DEFINIDAS POR SOFTWARE NO CONTEXTO DE NUVEM PRIVADA

PINTO, R. P.

Fatec Americana - Segurança da Informação
rossano.pinto@fatec.sp.gov.br

Software defined networking in the context of private clouds

Eixo Tecnológico: Informação e Comunicação

Resumo

A nuvem privada da FATEC Americana é composta de nós computacionais e de switches de software que permitem criar topologias diversas (este último item sendo objeto de pesquisa do projeto corrente). A fim de controlar a nuvem privada de maneira centralizada e eficiente, uma plataforma de controlador SDN, extensível e passível de criar aplicações de rede, pode trazer grande benefício para gerência do parque computacional da Fatec Americana (nuvem privada). No projeto anterior, o controlador SDN apareceu apenas como um coadjuvante. Nesse projeto o controlador SDN possui um papel de destaque e pretende-se explorar a grande variedade de aplicações existentes no controlador. Um dos objetivos principais é o estudo de desenvolvimento de aplicações de rede (de um sistema operacional de rede) que tenha como plataforma o controlador SDN escolhido. Como resultado espera-se os seguintes entregáveis: estudo sobre o estado da arte em controladores SDN (descrevendo um ou mais em profundidade); implantação do controlador SDN escolhido; estudo de algumas aplicações de rede nativas do controlador SDN escolhido; criação de uma ou mais aplicações de rede que possam ser instaladas no controlador SDN escolhido.

Palavras-chave: Controlador SDN, Aplicações de rede, Nuvem privada.

Abstract

FATEC Americana's private cloud is composed of computational nodes and software switches that allow the creation of different topologies (this last item being the object of research in the current project). In order to control the private cloud in a centralized and efficient way, an extensible SDN controller platform, capable of creating network applications, can bring great benefit to the management of Fatec Americana's computing park (private cloud). In the previous project, the SDN controller only appeared with a supporting role. In this project the SDN controller has a prominent role and it is intended to explore the wide variety of applications existing in the controller. One of the main objectives is to study the development of network applications (of a network operating system) that have the chosen SDN controller as the platform. As a result, the following deliverables are expected: study on the state of the art in SDN controllers (description of one or more controllers in depth); deployment of the chosen SDN controller; study of some native network applications of the chosen SDN controller; creation of one or more network applications that can be installed on the chosen SDN controller.

Key-words: SDN controller, Network applications, Private cloud.

1. Introdução

Uma nuvem é composta de nós computacionais para o oferecimento de serviços como processamento paralelo e armazenamento. Estes nós podem ser físicos ou virtuais. A interconexão dos nós, por sua vez, também pode se dar por equipamentos físicos ou virtuais. Em datacenters é comum que equipamentos de interconexão virtuais (switches oferecidos em software) [1](Seção 8.2, Figura 42; Seção 8.3.2, Figura 48*) sejam utilizados, uma vez que

Anais da VII Mostra de Docentes em RJJ

vários nós virtuais existem em cima de um nó físico. Com isso é possível utilizar o poder computacional de nós individuais para criar redes complexas, isoladas e que atendem a diversos requisitos de isolamento, segurança e desempenho. O ambiente de nuvem desenvolvido nos projetos anteriores oferece redes flat por meio do mecanismo de Linux Bridge (built-in L2 switch)[2] e Linux namespaces [3,4], especialmente o "Network namespaces". ; além da possibilidade da criação de topologias de redes diversas por meio do uso de switches de software de alto desempenho.

Com o aumento da complexidade da rede, mecanismos de gerência tradicionais não são ideais e acabam oferecendo opções limitadas relacionadas a customização. No projeto vigente, a inserção de controladores SDN [5] possibilita que um elemento central (que pode ser replicado para oferecer alta disponibilidade e balanceamento de carga) seja utilizado e que aplicações diversas sejam executadas sobre este controlador. Apenas para fins de exemplo e ilustração, um controlador que está em contante desenvolvimento e oferece uma plataforma ideal para inovação e desenvolvimento de novas aplicações de rede é o ONOS "Open Network Operating System" [6]. Um outro controlador bastante popular é o Open Daylight [7].

Todo um novo ecossistema voltado para SDN tem sido criado e várias empresas relevantes como CISCO, Juniper, Broadcom, NVIDIA e Intel tem investido em peso em tecnologias SDN e de programabilidade da rede. Como a tecnologia e conceito SDN é relativamente novo, vários padrões e arquiteturas ainda estão em definição. Projetos de pesquisa como este podem ajudar a desenhar tais padrões. Sendo assim, um projeto como esse justifica-se.

A nuvem privada da FATEC Americana, composta de máquinas dos laboratórios de informática, vem sendo criada ao longo de vários projetos de RJJ. Cada projeto oferece a implementação de funcionalidades adicionais. O objetivo geral desse projeto é dotar a nuvem privada da FATEC Americana de aplicações de rede que sejam executadas em uma plataforma de controlador SDN. Os objetivos específicos são: 1 - Escolher uma ou mais plataformas de controlador SDN para ser utilizada na nuvem privada da Fatec Americana; 2 - Explorar as funcionalidades e aplicações existentes em uma plataforma de controlador SDN escolhida; 3 - Verificar a possibilidade de estender / evoluir alguma aplicação existente no controlador SDN escolhido; 4 - Explorar uma ou mais maneiras de construir aplicações de rede no controlador escolhido, com compilação in-tree e out-tree; 5 - Desenvolver aplicações de rede que sejam executadas em uma plataforma SDN (que rodem no controlador SDN ONOS [8,9.10], se este for o escolhido, por exemplo).

2. Materiais e métodos

2.1. Materiais

O projeto utiliza os computadores dos laboratórios de informática em dias e horários em que estejam disponíveis, sem que haja interferência no cotidiano de uso normal pela Fatec Americana, e sem prejudicar as atividades acadêmicas regulares. Além disso, alguns computadores de uso pessoal do pesquisador são utilizados para fazer provas de conceito (PoC). Apenas softwares livres são utilizados na execução dos experimentos e criação das PoCs.

2.2. Metodologia

Anais da VII Mostra de Docentes em RJJ

Como esta pesquisa é classificada como tecnológica e está associada à área de ciência da computação, o método empregado é o experimental, já que computadores e softwares necessários para o projeto estão amplamente disponíveis para experimentação e uso. Nesse contexto, a fim de conduzir os experimentos necessários, esse projeto de pesquisa contempla as atividades de levantamento bibliográfico e exploração de plataformas alvo, tais como escolha de produtos (ou a programação/desenvolvimento de software - controlador SDN e suas aplicações/programas de rede) para a gerência SDN de uma nuvem privada. Nesse contexto, o método empregado é o experimental. Os experimentos necessários contemplam diversos aspectos, separados em 3 grandes grupos. Cada grupo contempla atividades a serem desenvolvidas. Os grupos e atividades, de maneira resumida, são: 1 - Escolher e explorar a instalação e operação de um controlador SDN com as aplicações nativas já existentes; 2 - Explorar as APIs existentes no controlador para que novas aplicações de rede possam ser desenvolvidas ou estendidas; 3 - Programar OU estender uma ou mais aplicações de rede no controlador SDN escolhido.

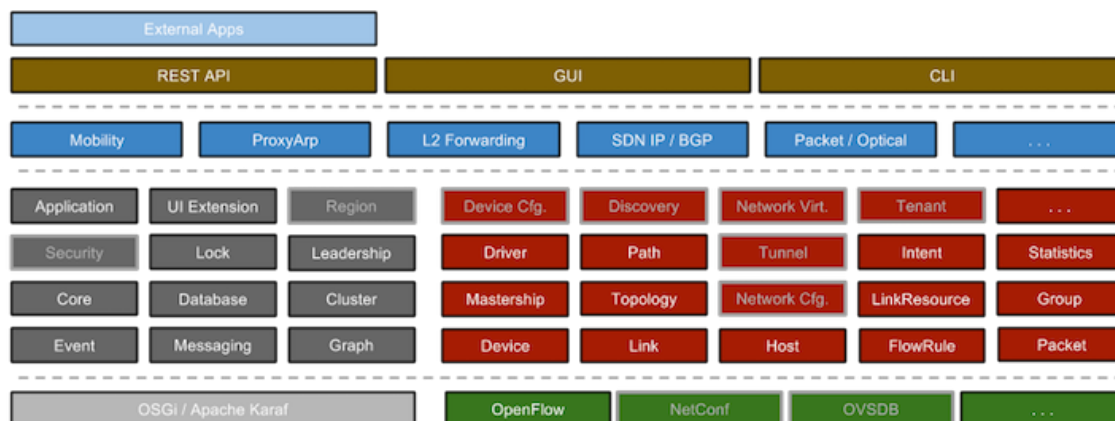
3. Resultados e Discussão

O controlador SDN ONOS continua sendo a escolha dos controladores disponíveis como software livre, pois é o que a comunidade adotou de fato.

Em ambientes de testes que utilizam switches programáveis, principalmente aqueles que suportam a linguagem P4, vê-se um misto de uso do ONOS e controladores bem pequenos escritos de maneira dedicada aos experimentos. Para o propósito do uso do ONOS para controlar uma nuvem privada, este mostrou-se suficiente, versátil e com uma gama bastante grande de aplicações e protocolos já existentes, como: SNMP, NETCONF, P4Runtime, gNMI, OSPF, BGP, DHCP, ODTN, ROUTING, VROUTER, dentre várias outras. Sendo assim, e em testes experimentais, optou-se em escolher o ONOS como controlador SDN para este projeto.

A Fig. 1 exibe os diversos subsistemas da arquitetura ONOS. O componente “Application” representa o subsistema que permite rodar parte das aplicações mencionadas anteriormente (omitindo-se os protocolos, que fazem parte de outro subsistema, mas que são utilizados por aplicações específicas).

Fig. 1 - Subsistemas disponíveis no ONOS.



Fonte: ONOS , 2021.

Anais da VII Mostra de Docentes em RJI

As aplicações são disponibilizadas com empacotamento baseado em OSGi, comum em sistemas que permitem descarregar aplicações sob demanda. Uma nova aplicação ONOS pode ser carregada, configurada, utilizada e, se necessário/desejável, removida. Tudo isso utilizando os protocolos REST ou SSH/cli, sem que seja necessário parar ou reiniciar o ONOS. Essa característica é bastante importante para um ambiente de nuvem, pois a criação e deploy de novas aplicações não necessita paralizar aplicações já em execução e nem tão pouco o próprio ONOS. A Fig. 2 exibe a lista completa de aplicações disponíveis no ONOS.

Fig. 2 - Lista de aplicações disponíveis no controlador SDN ONOS.

```
[rossano@asti-new VII-MOSTRA-RJI]$ ls ~/git-repos/onos-from-asti/onos/apps/
acl                flowalyzer         metrics            optical-model     scalablegateway
actn-mdsc          flowspec-api       mfwf              optical-rest      sdnip
artemis           fwd                mlb               p4-tutorial      segmentrouting
bgprouter         gangliametrics     mobility          packet-stats     simplefabric
castor            gluon              netconf           packet-throttle  t3
cfm               graphitemetrics    network-troubleshoot pathpainter      tenbi
cip               imr                newoptical        pcep-api         test
config            inbandtelemetry   node-diagnosis    pim              tetopology
configsync        influxdbmetrics    nodemetrics       portloadbalancer tunnel
configsync-netconf intentsync          odtn              powermanagement virtual
cord-support      iptopology-api    onlp-demo         proxyarp         virtualbng
cpman             k8s-networking    onos-topo         rabbitmq         vpls
dhcp              k8s-node          openroadm         reactive-routing vrouter
dhcprelay         kafka-integration openstacknetworking restconf         vtn
dpistats          l3vpn             openstacknetworkingui roadm            workflow
drivermatrix      layout            openstacknode     routeradvertisement yang
events            learning-switch   openstacktelemetry route-service    yang-gui
evpnopenflow     linkprops         openstacktroubleshoot routing          yms
evpn-route-service mappingmanagement openstackvtap     routing-api
faultmanagement  mcast
```

Fonte: Autoria própria.

Para a criação de novas aplicações pode-se partir de templates disponíveis e o sistema de build atual do ONOS é baseado em bazel[12], que é um sistema de build mantido e utilizado pelo google. Por exemplo, a aplicação DHCP possui os seguintes diretórios raiz, vistos na Fig. 3 (A).

Fig. 3 - (A) Exemplo de estrutura da aplicação DHCP. (B) Conteúdo do arquivo BUILD da aplicação DHCP.

```
[rossano@asti-new dhcp]$ ls
api  app  BUILD
[rossano@asti-new dhcp]$
```

```
[rossano@asti-new dhcp]$ cat BUILD
BUNDLES = [
  "//apps/dhcp/api:onos-apps-dhcp-api",
  "//apps/dhcp/app:onos-apps-dhcp-app",
]

onos_app(
  category = "Utility",
  included_bundles = BUNDLES,
  required_apps = ["org.onosproject.hostprovider"],
  title = "DHCP Server",
  url = "http://onosproject.org",
)
```

(A)

(B)

A Fig. 3 (B) exibe o conteúdo do arquivo BUILD do sistema de build bazel. Pode-se ver que o arquivo informa como a aplicação deve ser empacotada, as dependências

Fonte: Autoria própria.

Anais da VII Mostra de Docentes em RJJ

("org.onosproject.hostprovider"), categoria para facilitar a exibição e busca, além do título.

Vale destacar que parte da equipe que mantém o ONOS também desenvolve o μ ONOS, uma versão do ONOS baseada em microserviços. A versão μ ONOS não foi explorada nesse projeto por ser pouco madura e por optar-se por uma abordagem de controle centralizado.

Um vez instanciado, o ONOS pode ser acessado via interface WEB. A Fig. 4 (B) exibe a tela de login.

Fig. 4 - (A) Topologia emulada de uma rede óptica com controle via NETCONF. (B) Tela de login do controlador SDN ONOS.



Fonte: Autoria própria.

Nos primeiros testes, o ONOS foi utilizado para se conectar em switches virtuais de que emulam parte de uma rede óptica. A conexão entre ONOS e switches virtuais se deu via NETCONF. A Fig. 4 (A) exibe a topologia detectada pelo ONOS.

4. Considerações finais

Os objetivos propostos na linha do tempo do cronograma aprovado foram alcançados. A junção da ferramenta de gerência da nuvem, elaborada em projetos anteriores, com o controle de rede baseado em SDN (em especial o ONOS), no contexto de nuvem privada, pode trazer algumas inovações interessantes, tais como a descarga e utilização de novas aplicações sem que seja necessário parar todas as aplicações da nuvem. No modelo de gerência anterior, baseado somente na ferramenta de gerência escrita em python e com o framework Flask, a atualização de aplicações exigia a parada completa da ferramenta. Com o controlador ONOS, apenas a aplicação sendo atualizada é que precisa ser interrompida. Isso traz agilidade na manutenção e criação de novas aplicações.

Referências

[1] PETERSON, LARRY L.; CASONE, CARMELO; O'CONNOR, BRIAN; VACHUSKA, THOMAS; DAVIE, BRUCE. Software-Defined Networks: A Systems Approach. Disponível em <https://sdn.systemsapproach.org/uses.html>. 2020. Acesso em 05/09/2023.

[2] THE LINUX FOUNDATION. Bridge. Disponível em <https://wiki.linuxfoundation.org/networking/bridge>. Apr 27, 2021. Acesso em 05/09/2023

Anais da VII Mostra de Docentes em RJI

[3] OVENS, STEVE. The 7 most used Linux namespaces. Disponível em <https://www.redhat.com/sysadmin/7-linux-namespaces>. 21/01/2021. Acesso em 05/09/2023.

[4] LINUX PROGRAMMAER'S MANUAL. Namespaces(7) -- Linux manual page. Disponível em <https://man7.org/linux/man-pages/man7/namespaces.7.html>. 27/08/2021. Acesso em 20/09/2021.

[5] GÖRANSSON, PAUL; BLACK, CHUCK. Software Defined Networks: A Comprehensive Approach. 2014. Morgan Kaufmann.

[6] ONOS. Open Network Operating System. Disponível em <https://wiki.onosproject.org/display/ONOS/Developer+Guide>. Acesso em 05/09/2023.

[7] OPENDAYLIGHT. Open Daylight. Disponível em <https://www.opendaylight.org/>. Acesso em 05/09/2023.

[8] ONOS. Developer Guide. Disponível em <https://opennetworking.org/onos/>. Acesso em 27/09/2022.

[9] ONOS. ONOS Java API (2.7.0). Disponível em <https://api.onosproject.org/2.7.0/apidocs/>. Acesso em 05/09/2023.

[10] ONOS. Architecture and Internals Guide. Disponível em <https://wiki.onosproject.org/display/ONOS/Architecture+and+Internals+Guide>. Acesso em 30/09/2022.

[11] ONOS. Open Network Operating System Applications. <https://github.com/opennetworkinglab/onos/tree/master/apps>. Acesso em 05/09/2023

[12] BAZEL. Bazel - {Fast, Correct} - Choose two. Disponível em <https://bazel.build/>. Acesso em 30/09/2022.